A hybrid hierarchical task network planning and ontological approach for dynamic cross-enterprise business process formulation

Ko, Ryan Kok Leong

2011

Ko, R. K. L. (2011). A hybrid hierarchical task network planning and ontological approach for dynamic cross-enterprise business process formulation. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/43992>

<https://doi.org/10.32657/10356/43992>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**A HYBRID HIERARCHICAL TASK NETWORK
PLANNING AND ONTOLOGICAL APPROACH FOR
DYNAMIC CROSS-ENTERPRISE BUSINESS
PROCESS FORMULATION**

**A HYBRID HIERARCHICAL TASK NETWORK PLANNING AND
ONTOLOGICAL APPROACH FOR DYNAMIC CROSS-ENTERPRISE
BUSINESS PROCESS FORMULATION**

RYAN K L KO

KO KOK LEONG RYAN

School of Mechanical & Aerospace Engineering

2011

2011

**A HYBRID HIERARCHICAL TASK NETWORK
PLANNING AND ONTOLOGICAL APPROACH FOR
DYNAMIC CROSS-ENTERPRISE BUSINESS
PROCESS FORMULATION**

KO KOK LEONG RYAN

RYAN K L KO

School of Mechanical & Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2011

Soli Deo Gloria

To my parents.

Abstract

Every day, information is transacted in the Internet across different enterprises in business-to-business (B2B) collaborations. In a bid to manage the growing volume and complexity of supply chain information, many companies are turning to business process management systems (BPMS) and systems supporting B2B collaboration standards such as *Electronic Data Interchange (EDI)*, *RosettaNet* and *ebXML (Electronic Business using eXtensible Markup Language)*. According to a 2007 Gartner report, the BPMS market reached nearly US\$1.7 billion in total software revenue at the end of 2006. This is testimony to a rapid growth of interest in business process management (BPM) over the last decade.

Despite this growth, many current techniques are still *static* in nature, requiring substantial modifications/ (re)coding to re-establish the link between collaborating companies. Such modifications are expensive and impractical in a *dynamic*, global business environment characterized by frequent, rapid changes. At the same time, many prominent business process (BP) modelling techniques are rooted in computer process theories like state machines, Pi Calculus and Petri Nets, which are computationally efficient but are void of context and semantics so necessary in very complex, multi-party, high level business communications. Most importantly, these BP modelling techniques cannot dynamically synthesise business processes based on a user's high-level business goals and constraints.

The goal of this thesis is to discover, implement, test and verify a technique that enables agile B2B collaborations through the dynamic formulation of cross-enterprise business processes based on high-level user business goals and criteria. An extensive literature review into current BP research, industry best practices, BP modeling standards and B2B industry standards, revealed that the realization of this goal is contingent upon solving four sub-problems: (1) the problem of decomposing high level business goals and constraints (e.g. item price, lead time) on-the-fly into corresponding low-level tasks of adequate granularity, (2) the task of dynamically chaining these decomposed low-level tasks with the appropriate control and information flows, (3) the problem of representing the semantic characteristics and relationships between business processes components (e.g. tasks, sub-tasks, actors, methods of decomposition, control flow, etc.) and (4) the chained low-level tasks have to be implemented openly and directly on the WWW.

These four sub-problems were addressed by a hybrid methodology which entails the dynamic decomposition and sequencing of tasks facilitated by Hierarchical Task Network (HTN) planning concepts, and the adoption of ontologies. HTN Planning employs so-called *methods* to establish the various permutations by which high level business processes can be

decomposed into *primitive tasks* which are chained sequentially in a single level from left-to-right. Current business processes in industry do not use these methods. However, the primitive tasks do not reflect the true nature of real life cross-enterprise BP's, which are a network of tasks with associated workflow patterns (e.g. merge, join, split, etc.). To address this reality, traditional HTN Planning was augmented with additional constructs such as workflow patterns and business constraints (e.g. item cost, quantity ordered, etc.).

A methodology was proposed to model the hierarchical task networks to support dynamic B2B collaborations. Decomposition *methods*, compound and primitive tasks, and other components (e.g. actors, document information, etc.) needed to describe typical B2B collaborations were modeled into the proposed "Business-OWL" (BOWL) ontologies.

Unlike the traditional problem description files used in HTN planning, ontologies can richly and semantically describe complex direct and indirect relationships between *compound* and *primitive tasks*, *public-private business processes message exchange patterns*, *input-behaviour-output characteristics* of executable tasks and *actors*, thereby facilitating implementation over the WWW.

The augmented HTN Planning approach and the supporting ontologies were embodied in a prototype called *Genesis*. The business processes enabling business goals of *order management* and *procurement*, which are the main B2B collaboration processes, were successfully generated on-the-fly in *Genesis*. *Genesis* was further validated in a case study of a regional pallet-leasing company. Low-level operational tasks (e.g. issuing of hire notes, exchange notes, etc) were successfully decomposed on-the-fly from higher level business goals (e.g. buying raw materials from suppliers, selling pallets to food producers and the leasing pallets to supermarkets).

With the ontology-augmented HTN Planning methodology, business goals can now be dynamically addressed on-the-fly, without the need to overhaul existing information technology (IT) infrastructures.

Acknowledgments

It is my firm belief that a person's life is shaped by a combination of predestination, his or her personal life choices, and special individuals whose influence mould that person's mindset and actions at different stages of his or her life. This section is dedicated to these special people who have picked me up or spurred me on in my apprenticeship as an independent researcher.

I am very blessed to have Associate Professor Dr Stephen Lee Siang Guan and Dr Lee Eng Wah as my PhD supervisors. Stephen, thank you for moulding me, guiding me, and giving me the freedom to experience the record number of epiphanies in my life. I will not forget your teachings on how to be a good scientist and teacher, and how to be a useful person in God's eyes. Eng Wah, thank you for always giving me encouragement and supporting me in all aspects of my PhD. I always leave our meetings with a boosted mindset, and that has a lot to do with your industry experience and sharing of personal experiences.

I am also fortunate to have Professor Angela Goh from whom I can speak to for guidance on research methods, and valuable, objective advice on life and my career. My initial work on HTN planning and AI research received a lot of support from Assistant Professor Miao Chunyan and her Emerging Research Lab. I also wish to thank Associate Professor Stephane Bressan of National University of Singapore for his useful advice for my academic career and his input during the IMSS meetings. I am also indebted to Dr. Tony KY Chan, my former undergraduate FYP supervisor, and eventually my mentor and friend, for introducing me to this project *via* Professor Goh, and for his advice on PhD and the academic career.

I also wish to thank Mr. Andre Jusuf, Mr. Karel Mous, Dr. Le Duy Ngan, Ms. PS Tan, Ms. Rohayati Bte. Kamaruddin and many other colleagues, technical staff and laboratory staff at ADaM Lab and SIMTech. In the course of my research, I also had the chance to mentor and learn from discussions with the following final-year project undergraduate students: Mathew Wong, Koh Zhen Rong, Foo Xuexun, Louise Lai, Woo Yaoying, Lim Deqiang, Pan Kepei Justin, and Ooi Yee Loong.

I benefited tremendously from interviews and on-site visits with several industry representatives: Mr. Thomas Yeo – Manager, Kim Hiap Lee Company (Pte) Ltd Singapore, Mr. Basil Sim – Manager, Meiban Mold Manufacture Pte. Ltd., Ms. Serina Lim – Procurement Officer, Meiban Mold Manufacture Pte. Ltd., Mr. Albert Tan – Manager, Dell Singapore Pte. Ltd, Mr. TM Chua – Manager, Motorola Electronics Pte. Ltd., Singapore, and Mr. Bobby Koh – Founder, Managing Director, Endela Trading & Manufacturing Pte. Ltd.

The following researchers have also contributed significantly through their comments, discussions and advice during the course of my research: Professor Mark S. Fox (University of Toronto) – for his advice on looking at AI planning for clues, Dr. Charles Petrie (SRI International) – for letting me know during his visit to NTU that he also feels the top-down planning approach is more logical than conventional ways to achieve dynamic business process synthesis, and Dr Hamid Motahari, Dr Min Luo, Assistant Professor Yuhong Yan and Dr Liang-Jie Zhang for their input and advice during my presentation and demonstration at the IEEE Service Cup 2009 and ICWS 2009.

I also wish to thank all friends and people who have helped me in any way, especially the staff of NTU library for doing a wonderful job acquiring very updated and useful content for researchers like myself. The research was supported by Singapore's Agency for Science, Technology and Research (A*STAR)'s thematic Integrated Manufacturing Systems and Services (IMSS) project (Grant number: 0521160077).

Most importantly, I wish to thank my parents and my sister for their continued support and prayers, and Denise for being my ever encouraging cheerleader.

Last but not least, I wish to thank the Lord for His provision and guidance.

TABLE OF CONTENTS

ABSTRACT	V
ACKNOWLEDGMENTS	VII
TABLE OF CONTENTS	IX
LIST OF FIGURES.....	XVI
LIST OF TABLES	XIX
NOMENCLATURE	XX
CHAPTER 1. INTRODUCTION	1
1.1 BACKGROUND.....	1
1.2 MOTIVATION FOR THIS RESEARCH	2
1.2.1 Objectives of Research	3
1.2.2 Scope of Work.....	4
1.3 ORGANISATION OF THESIS	4
PART I: LITERATURE REVIEW.....	6
CHAPTER 2. BUSINESS PROCESSES DEFINED	7
2.1 DEFINITIONS OF BUSINESS PROCESSES	7
2.2 TYPES OF BUSINESS PROCESSES	8
2.2.1 Level Perspective.....	8
2.2.2 Core Competency Perspective.....	10
2.2.3 Perspective used in this thesis.....	10
2.3 DEMYSTIFYING BPM-RELATED TERMINOLOGIES.....	10
2.4 CHAPTER SUMMARY	11
CHAPTER 3. BUSINESS PROCESS MODELING THEORY	12
3.1 THE FLOW-BASED APPROACH.....	12
3.1.1 Graphical Flow-Based Methods	13
3.1.1.1 Petri Nets.....	13
3.1.1.2 State Machines.....	14
3.1.1.3 Flow Charts and Role-Activity Diagrams	15
3.1.2 Grammatical Flow Based Methods.....	16
3.1.2.1 Pi-Calculus	16
3.1.2.2 Process Specification Language.....	18
3.1.3 Flow-based Approaches in the Industry	19
3.2 THE COMMUNICATIVE APPROACH	20
3.2.1 Rationale for Communicative Approaches.....	20
3.2.2 Language Action Perspective (LAP)	21
3.2.2.1 Rationalistic approaches vs LAP approaches.....	21
3.2.2.2 The birth of the term “Language Action”	22
3.2.2.3 Importance of the human decision maker in the loop	22
3.2.3 Applications of the LAP in BPM.....	22
3.2.3.1 Conversation for Action (CfA) Approach.....	22
3.2.3.2 Speech Act-based information analysis Methodology with comPUter-aided tOols (SAMPO).....	24
3.2.3.3 Action Workflow Approach.....	26
3.2.3.4 Business as Action Theory (BAT) Framework	28
3.2.3.5 Dynamic Essential MOdeling (DEMO) Framework	31
3.2.4 Automating the Negotiation Phase	34
3.2.4.1 Speech-Act-Based Negotiation Protocol (SANP)	34
3.2.4.2 Formal Language for Business Communication (FLBC).....	35
3.2.5 Limitations of LAP and Related LAP Works	35

3.3	APPLICATIONS TO RESEARCH	36
3.4	CHAPTER SUMMARY	38
3.4.1	<i>Flow Based Approaches in BPM.....</i>	38
3.4.2	<i>Communicative Approaches in BPM.....</i>	38
CHAPTER 4.	INDUSTRY B2B STANDARDS	40
4.1	ROSETTANET	40
4.1.1	<i>Scope of RosettaNet</i>	41
4.1.1.1	<i>The RosettaNet Framework and its Components</i>	41
4.1.2	<i>Challenges in RosettaNet Adoption.....</i>	42
4.1.3	<i>Application to the Author's Research.....</i>	43
4.2	ELECTRONIC BUSINESS USING EXTENSIBLE MARKUP LANGUAGE (EBXML)	43
4.2.1	<i>Differences of ebXML from RosettaNet.....</i>	44
4.2.2	<i>Key Components of ebXML.....</i>	44
4.2.3	<i>How Companies Collaborate in ebXML</i>	45
4.2.4	<i>Adoption of ebXML</i>	46
4.2.5	<i>Limitations of ebXML.....</i>	47
4.2.6	<i>Application to the implementation aspects of the author's research.....</i>	47
4.3	OPEN APPLICATION GROUP INTEGRATION SPECIFICATION (OAGIS)	48
4.3.1	<i>Key Components of OAGIS.....</i>	48
4.3.1.1	<i>Business process (Scenarios)</i>	48
4.3.1.2	<i>Business Object Documents (BOD)</i>	49
4.3.2	<i>Limitations of OAGIS.....</i>	49
4.3.3	<i>OAGIS' Role in This Research</i>	50
4.4	LIMITATIONS OF CURRENT B2B STANDARDS.....	50
4.5	LEGACY IT-BASED INDUSTRY B2B PRACTICES.....	50
4.5.1	<i>Electronic Data Interchange (EDI).....</i>	50
4.5.2	<i>Vendor Managed Inventory (VMI) Systems.....</i>	51
4.5.3	<i>The ICAM Definition Languages (IDEF) Standards Family</i>	51
4.6	CHAPTER SUMMARY	53
CHAPTER 5.	CROSS-ENTERPRISE BUSINESS PROCESS (CBP) RESEARCH	54
5.1	CBP RESEARCH: WEB SERVICES AND SERVICE-ORIENTED COMPUTING	54
5.1.1	<i>Web Services.....</i>	54
5.1.2	<i>Service Oriented Architecture (SOA).....</i>	55
5.1.3	<i>Web Service Composition Challenges.....</i>	56
5.1.3.1	<i>Possible reasons for non-progress in dynamic Web Service Composition</i>	57
5.1.3.2	<i>Bottom-Up vs. Top-Down approach.....</i>	58
5.1.4	<i>Why Top-Down Approach was Adopted in This Research</i>	59
5.2	CBP RESEARCH: SEMANTIC WEB	59
5.2.1	<i>State of the Art in the Semantic Web</i>	60
5.2.1.1	<i>Syntactic Representation - Flexible Storage of Data</i>	60
5.2.1.2	<i>Semantic Representation - Description of Meaning in the Data</i>	60
5.2.2	<i>Related Research: Semantic Web Services.....</i>	62
5.3	CBP RESEARCH: PRAGMATIC WEB	63
5.3.1	<i>Limitations of the Semantic Web</i>	63
5.3.2	<i>Vision of the Pragmatic Web</i>	63
5.4	CBP RESEARCH: CONTEXT-AWARENESS	64
5.5	CBP RESEARCH: DYNAMIC CBP FORMULATION.....	65
5.5.1	<i>CrossFlow Project.....</i>	65
5.5.2	<i>Matchmaking of Ad Hoc Processes</i>	67
5.5.3	<i>Semantics Utilised for Process Management within and between Enterprises (SUPER) Integrated Project</i>	68
5.5.4	<i>SemBiz Project.....</i>	68
5.5.5	<i>A*STAR IMSS Ont-CAF Project.....</i>	69
5.6	APPLICATIONS TO RESEARCH	71

5.7	CHAPTER SUMMARY	72
CHAPTER 6.	POTENTIAL TECHNIQUES FOR DYNAMIC CBP FORMULATION RESEARCH	73
6.1	NATURE OF RESEARCH PROBLEM	73
6.2	HIERARCHICAL TASK NETWORK (HTN) PLANNING	75
6.2.1	<i>Understanding the Nature of Artificial Intelligence Planning.....</i>	75
6.2.1.1	Main components of an AI Planning methodology	75
6.2.1.2	“Sequence of Actions” vs. “Action Strategies”	75
6.2.1.3	Planning vs. Scheduling vs. Searching.....	76
6.2.2	<i>Classical Planning</i>	76
6.2.2.1	Forward State Space Search.....	77
6.2.2.2	Backward Chaining.....	77
6.2.2.3	Partial-Order Planning.....	77
6.2.3	<i>Difficulties of Classical Planning and Partial Order Planning</i>	78
6.2.4	<i>Hierarchical Task Network Planning</i>	78
6.2.4.1	Why HTN Planning for Dynamic cBP Formulation?	78
6.2.4.2	HTN Planning in a Nutshell	79
6.2.5	<i>Why HTN Planning alone is insufficient for Dynamic cBP Formulation?</i>	80
6.3	ONTOLOGIES	81
6.3.1	<i>A Brief Overview of Web Ontologies</i>	81
6.3.2	<i>Why Model HTN’s as Ontologies Described in OWL?</i>	82
6.3.3	<i>Related enterprise-related ontology research projects.....</i>	83
6.4	CONSIDERATION OF EXECUTION PHASE, LOWER-LEVEL REQUIREMENTS	84
6.4.1	<i>Embedding Control Flows with Workflow patterns.....</i>	84
6.4.2	<i>Output Format for the Synthesized Plans</i>	86
6.4.2.1	BPEL in a nutshell	87
6.5	CHAPTER SUMMARY	89
CHAPTER 7.	RESEARCH ISSUES AND DIRECTIONS.....	91
7.1	RESEARCH ROADMAP	91
7.2	BPM LANDSCAPE.....	92
7.2.1	<i>Research Contributions of the BPM Landscape</i>	92
7.2.2	<i>Organisation of the BPM Landscape</i>	92
7.2.3	<i>Topics with research potential.....</i>	93
7.3	THE BPM RESEARCH GENEALOGY	95
7.3.1	<i>The Scientific Development of BPM.....</i>	95
7.3.2	<i>Chronological Progression of BPM Research</i>	95
7.4	SUMMARY OF GAPS IN BPM RESEARCH	98
7.4.1	<i>No universal BPM terminology.....</i>	98
7.4.2	<i>Too many duplicating BPM languages and standards.....</i>	98
7.4.3	<i>Inadequate basic research in business process essentials and formulation.....</i>	98
7.4.4	<i>Lack of high level business process representations</i>	98
7.4.5	<i>Too much focus on operational processes.....</i>	99
7.4.6	<i>Gap between business analysts and IT implementers</i>	99
7.5	SUMMARY OF GAPS IN CURRENT B2B SYSTEMS AND APPROACHES.....	99
7.5.1	<i>High setup and licensing costs</i>	99
7.5.2	<i>Static B2B setups not flexible and not responsive</i>	99
7.5.3	<i>Lack of Research into Dynamic B2B Approaches.....</i>	100
7.5.3.1	Overlooking the Planning Phase of B2B Collaboration	100
7.5.3.2	Current Emphasis of the Bottom-Up Approach	100
7.5.4	<i>Current B2B standards’ emphasis on functional perspective of business processes</i>	100
7.5.5	<i>Limited semantic representations in current B2B standards and systems.....</i>	100
7.5.6	<i>Lack of open access collaboration framework.....</i>	101
7.6	RESEARCH GOAL AND SUB-OBJECTIVES.....	101
7.6.1	<i>Formulation of Research Questions</i>	101
7.6.2	<i>Scope and Domain of Research.....</i>	102

7.7	CHAPTER SUMMARY	103
PART II: METHOD, IMPLEMENTATION AND RESULTS.....		104
CHAPTER 8. PROPOSED CONCEPTUAL FOUNDATIONS AND CBP FRAMEWORK 105		
8.1	THE ESSENTIAL CBP (ECBP) FRAMEWORK.....	105
8.1.1	<i>How the ECBP Framework is Derived.....</i>	<i>105</i>
8.1.2	<i>Other Influences From LAP</i>	<i>107</i>
8.1.3	<i>Phases in the ECBP Framework</i>	<i>109</i>
8.1.3.1	Discovery Phase	109
8.1.3.2	Establish Contact Phase	109
8.1.3.3	Negotiation Phase.....	109
8.1.3.4	Contract Agreement Phase.....	110
8.1.3.5	Contract Fulfillment Phase.....	110
8.1.4	<i>Buyers and Sellers.....</i>	<i>111</i>
8.2	CBP AXIOMS DERIVED FROM ECBP FRAMEWORK	111
8.3	KEY USES OF THE ECBP FRAMEWORK	112
8.3.1	<i>Simplifying B2B Collaboration Modelling.....</i>	<i>112</i>
8.3.2	<i>B2B Collaboration Systems and Architecture Design</i>	<i>113</i>
8.3.3	<i>Refining Current ECBP research.....</i>	<i>113</i>
8.3.3.1	EDI from the Perspective of the ECBP Framework.....	114
8.3.3.2	RosettaNet from Perspective of the ECBP Framework.....	114
8.3.3.3	ebXML from the Perspective of the ECBP Framework.....	114
8.3.3.4	Related B2B research.....	114
8.4	CONSTRAINTS AND CRITERIA IN B2B COLLABORATIONS	114
8.5	CHAPTER SUMMARY	116
CHAPTER 9. AN ONTOLOGY WHICH ENABLES DYNAMIC CBP FORMULATION		118
9.1	RATIONALE FOR CREATING BOWL	118
9.1.1	<i>Lack of dynamic cBP task decomposition methodologies</i>	<i>120</i>
9.1.2	<i>Limitations of Traditional HTN Planning Problem and Domain Description Files</i>	<i>120</i>
9.2	CREATING BOWL.....	120
9.2.1	<i>Creating the Hierarchical Task Ontology</i>	<i>120</i>
9.2.1.1	Manual Synthesis of Tasks.....	120
9.2.1.2	Sorting and Categorising into 5 B2B Phases	121
9.2.2	<i>Creating and Linking the Methods Ontology.....</i>	<i>122</i>
9.3	POPULATING THE BOWL KNOWLEDGE BASE	124
9.4	FEATURES OF BOWL.....	124
9.4.1	<i>Tasks.....</i>	<i>125</i>
9.4.2	<i>Methods and their components</i>	<i>125</i>
9.4.2.1	Modelling task decomposition permutations.....	125
9.4.2.2	Components encapsulated in Methods.....	126
9.4.3	<i>Actors of Tasks</i>	<i>128</i>
9.4.4	<i>Effects</i>	<i>128</i>
9.4.5	<i>Organization-Defined Thresholds</i>	<i>128</i>
9.4.6	<i>Primitive Actions.....</i>	<i>128</i>
9.5	HOW BUSINESS GOALS ARE DECOMPOSED INTO PRIMITIVE TASKS	129
9.6	TECHNICAL MERITS OF BOWL.....	130
9.7	CHAPTER SUMMARY	131
CHAPTER 10. ENHANCED HTN PLANNING ALGORITHM FOR DYNAMIC CBP FORMULATION		133
10.1	GENESIS ALGORITHM IN A NUTSHELL.....	133
10.1.1	<i>Task Decomposition and Sub-Task Chaining</i>	<i>134</i>
10.1.2	<i>How Genesis Algorithm traverses through BOWL's HTN.....</i>	<i>136</i>

10.2	OUTPUT AND EVENTUAL WEB SERVICE MATCHING AND EXECUTION.....	137
10.3	CONTRIBUTIONS OF THE GENESIS ALGORITHM	137
10.4	CHAPTER SUMMARY	138
CHAPTER 11. IMPLEMENTATION, VERIFICATION AND VALIDATION.....		139
11.1	IMPLEMENTING THE GENESIS SYSTEM.....	139
11.1.1	<i>Architecture of the Genesis System</i>	139
11.1.2	<i>Types of Outputs of Genesis System</i>	141
11.1.2.1	Annotated Abstract BPEL for Eventual SOA Implementation	141
11.1.2.2	Displaying Nesting of Control Flows and Parent-Child Task Relations.....	142
11.1.2.3	Graphical Display For Visual Inspection of Generated Decision Routes	144
11.2	ADAPTING TO DIFFERENT SETS OF GOALS AND CRITERIA.....	148
11.3	SUCCESSFUL APPLICATION INTO A*STAR IMSS ONT-CAF PROJECT	150
11.4	PROTOTYPE VERIFICATION PROCEDURE AND RESULTS	151
11.4.1	<i>Path Testing</i>	152
11.4.1.1	Reducing Factors	152
11.4.1.2	Minimum Scenarios Needed For Complete Branch Coverage	153
11.4.1.3	Path Testing Verification Process	154
11.4.1.4	Fault Tolerance Tests and Limitations Remedied	155
11.4.1.5	HTN Planning-Related Limitation Revealed via Verification	156
11.5	VALIDATION AGAINST INDUSTRY PROCESSES.....	158
11.5.1	<i>Validation Background</i>	159
11.5.1.1	Why KHL?	159
11.5.1.2	Introduction to KHL's Pallet Leasing Business Processes	159
11.5.1.3	Hire Order	161
11.5.1.4	Issue Note.....	161
11.5.1.5	Retrieve Note	163
11.5.1.6	Exchange Note	163
11.5.1.7	Transfer Note	163
11.5.2	<i>Validation Goal</i>	164
11.5.3	<i>Validation Protocol</i>	165
11.5.3.1	Constants.....	165
11.5.3.2	Validation Variables	165
11.5.3.3	Industry Expert Validation	166
11.5.4	<i>Validation Highlights and Results</i>	167
11.5.4.1	High Similarity against Real-Life Processes	167
11.5.4.2	Future Validation Work.....	168
11.5.4.3	An example of the LeaseOut goal achieved	169
11.5.4.4	An example of the LeaseFrom Goal Achieved	171
11.6	CHAPTER SUMMARY	171
PART III: DISCUSSIONS AND CONCLUSION.....		173
CHAPTER 12. DISCUSSIONS AND KEY RESEARCH CONTRIBUTIONS.....		174
12.1	OUTLINE OF THESIS RESEARCH CONTRIBUTIONS	174
12.1.1	<i>General Overview of Contributions</i>	174
12.1.2	<i>Addressing Research Challenge 1- Encapsulation of High-level Business Goals and Constraints</i>	176
12.1.2.1	Highest level business goals in B2B collaborations	176
12.1.2.2	Finding the "top priority" criteria for each B2B collaboration	176
12.1.2.3	The Need for a Generic B2B Collaboration Framework.....	178
12.1.2.4	Seeking a more-natural way to express high-level goals and constraints	178
12.1.3	<i>Addressing Research Challenge 2 - Decomposition of High-level Goals into Granular Executable Tasks</i>	179
12.1.3.1	What methodologies decompose compound tasks into primitive tasks?.....	179
12.1.3.2	Any current modeling techniques available for describing dynamic decompositions? ..	180
12.1.4	<i>Addressing Research Challenge 3- Composition of granular tasks into appropriate control flows for appropriate roles</i>	181

12.1.4.1	Ensuring that business process' control flows are considered during composition of granular tasks.....	181
12.1.4.2	Modeling augmentations/ techniques to ensure composed primitive tasks can be easily executed by existing execution-phase technologies	182
12.2	HOW THE RESEARCH ADDRESSED BPM GAPS	182
12.2.1	<i>Absence of universally agreed BPM terminologies.....</i>	182
12.2.2	<i>Presence of many duplicating languages and standards</i>	183
12.2.3	<i>Current State of BP Modeling Theory.....</i>	183
12.2.4	<i>Overlooking the Planning Approach to BP Modeling</i>	184
12.2.5	<i>Top-down vs. Bottom-Up approaches</i>	185
12.3	HOW THE RESEARCH ADDRESSED B2B GAPS.....	185
12.3.1	<i>High setup and licensing costs</i>	185
12.3.2	<i>Static setups not flexible and responsive</i>	186
12.3.3	<i>Current B2B standards' emphasis on functional perspective of business processes</i>	186
12.3.4	<i>Limited semantic representations in current B2B standards and systems.....</i>	186
12.4	COMPARISONS WITH RELATED WORK.....	186
12.4.1	<i>Planning Processes with Web Services</i>	187
12.4.2	<i>Semantics of Business Vocabulary and Business Rules (SBVR)</i>	187
12.4.3	<i>Web Service Composition Methods using Traditional HTN-Planners.....</i>	188
12.4.4	<i>Ontological and Reference Model Approaches.....</i>	189
12.4.5	<i>Other Related Approaches</i>	190
12.5	LIMITATIONS OF STUDY.....	190
CHAPTER 13. CONCLUSIONS AND FUTURE WORK		192
13.1	HIGHLIGHTS OF THE RESEARCH	192
13.2	FUTURE WORK	194
13.2.1	<i>Modeling Transition Points between Public and Private Processes</i>	194
13.2.2	<i>Enhancing BOWL with Task Input-Behaviour-Output Description Ontologies</i>	195
13.2.3	<i>Measurement of Quality of CBPs Generated.....</i>	196
13.2.4	<i>High-Level Language to Describe Goals and Criteria</i>	196
REFERENCES.....		198
RELATED PUBLICATIONS AND AWARDS.....		211
APPENDICES.....		213
APPENDIX A – DIFFERENTIATING THE BPM TERMINOLOGIES		214
APPENDIX B – A REVIEW OF BPM STANDARDS		215
APPENDIX C – MAIN SYNTAX OF PI-CALCULUS EXPLAINED		216
APPENDIX D – SPEECH ACTS THEORY AND THEORY OF COMMUNICATIVE ACTIONS		217
APPENDIX E – SYNTHESISED CBP TASK LISTS FROM ROSETTANET, EBXML AND OAGIS REFERENCE MODELS.....		218
APPENDIX F – TQRDCEB MODEL FOR SUPPLIER EVALUATION		219
APPENDIX G – INDUSTRY VISIT QUESTIONNAIRE ENQUIRING HIGH-PRIORITY VENDOR SELECTION CRITERIA		220
APPENDIX H – BOWL - VISIO FILE FORMAT		221
APPENDIX I – BOWL – ENHANCED OWL/RDF FILES FORMAT.....		222
APPENDIX J – COMPILATION OF GENESIS SOURCE CODES.....		223
APPENDIX K – BPELTODOT.XSL FILE		224

APPENDIX L – VERIFICATION/ PATH TESTING SCENARIOS AND RESULTS	225
APPENDIX M – SAMPLE VERIFICATION SCENARIO IN VISIO FORMAT	226
APPENDIX N – COLLAGEN SYNTAX IN EBNF	227

List of Figures

FIGURE 2.1: STRATEGIC, MANAGEMENT AND OPERATIONAL BUSINESS PROCESSES	9
FIGURE 3.1: A SIMPLE PETRI NET AND ITS COMPONENTS	13
FIGURE 3.2: A PETRI NET FOR THE PROCESSING OF COMPLAINTS. SRC.: [34].....	13
FIGURE 3.3: AN EXAMPLE OF A SIMPLISTIC PURCHASING BUSINESS PROCESS IN RAD FORMAT. SRC.: HTTP://WWW.INSTREAM-DEV.CO.UK/INSTREAM/RAD.JSP.....	16
FIGURE 3.4: CONVERSATION FOR ACTION (CfA) PROPOSED BY WINOGRAD AND FLORES [104]	23
FIGURE 3.5: AN EXAMPLE OF A CONVERSATION GRAPH FOR AN INVITATION DISCOURSE. SRC: [107].....	25
FIGURE 3.6: THE ACTION WORKFLOW LOOP, THE MOST FUNDAMENTAL TRANSACTION IN ALL BUSINESS PROCESSES, ACCORDING TO [88].....	26
FIGURE 3.7: AN EXAMPLE WORKFLOW LOOPS MAKING UP THE CORE STRUCTURE OF A CANDIDATE INTERVIEW REVIEW PROCESS [88]	27
FIGURE 3.8: AN ILLUSTRATION OF A BAT GENERIC BUSINESS FRAMEWORK IN ACTION. SRC: [115].....	28
FIGURE 3.9: THE CONSTITUENTS OF THE BUSINESS TRANSACTION IN BAT. SRC: [115]	29
FIGURE 3.10: A SIMPLIFIED VERSION OF THE BUSINESS AS ACTION GAME. SRC: [115].....	30
FIGURE 3.11: 3 LEVELS OF ABSTRACTION OF AN ORGANISATION [85].....	32
FIGURE 3.12: THE O-E-R TRANSACTION, THE ESSENCE/ DEEP STRUCTURE OF BUSINESS PROCESS TRANSACTIONS	32
FIGURE 3.13: SUMMARY OF CHAPTER 2 AND CHAPTER 3	39
FIGURE 4.1: AN EXAMPLE OF A CLUSTER-SEGMENT-PIP-ACTION BREAKDOWN IN ROSETTANET FOR THE “3A1- REQUEST QUOTE” PIP.....	40
FIGURE 4.2: OVERVIEW OF THE RNIF AND ROLE OF EACH ROSETTANET COMPONENT (SRC: HTTP://WWW.IBM.COM/DEVELOPERWORKS/WEBSERVICES/LIBRARY/WS-ROSE2/INDEX.HTML)	42
FIGURE 4.3: EBXML TECHNICAL ARCHITECTURE SPECIFICATION. SRC: [145]	45
FIGURE 4.4: A SCENARIO DESCRIBING A SUPPLY CHAIN INTEGRATION BETWEEN THE MANUFACTURING, PURCHASING, ORDER MANAGEMENT, BILLING, SHIPPING, AND FINANCIALS DEPARTMENTS. SRC: HTTP://WWW.IBM.COM/DEVELOPERWORKS/XML/LIBRARY/X-OAGIS/	48
FIGURE 4.5: GENERAL STRUCTURE OF ALL BODS	49
FIGURE.4.6: AN EXAMPLE OF STRUCTURE OF THE PROCESSPURCHASEORDER BOD	49
FIGURE 4.7: AN IDEFo DECOMPOSITION DIAGRAM	52
FIGURE 5.1: SCREEN CAPTURE OF THE SAP ENTERPRISE PORTAL, SHOWING WEB SERVICE DEPLOYMENTS VIA IVEWS	55
FIGURE 5.2: THE LAYERS WITHIN SERVICE ORIENTED ARCHITECTURE [140].....	56
FIGURE 5.4: COMPARING WEB PARADIGMS	62
FIGURE 5.5: THE CROSSFLOW ARCHITECTURE.....	66
FIGURE 5.6: THE SEMBIZ ARCHITECTURE	69

FIGURE 5.7: THE DECOMPOSITION OF HIGH LEVEL BUSINESS PROCESSES INTO EXECUTABLE SERVICES VIA 3 STAGES IN THE IMSS ONT-CAF PROJECT. SRC: [163]	70
FIGURE 6.2: A PARTIALLY ORDERED PLAN	78
FIGURE 6.3: DIFFERENT 'METHODS' FOR THE TRAVEL COMPOUND TASK TO DECOMPOSE	79
FIGURE 6.4: A TRAVEL PROBLEM REPRESENTED AS A HTN. [245]	80
FIGURE 6.5: A PLAN GENERATED BY THE HTN ALGORITHM DECOMPOSING TRAVEL(X, Y) [245]	80
FIGURE 6.6: AN EXAMPLE OF AN ONTOLOGY OF AN ORGANISATION FROM THE VIEWPOINT OF FOX ET AL. [253]	83
FIGURE 7.1: SUMMARY OF AUTHOR'S APPROACH TOWARDS ADDRESSING THE RESEARCH GOAL	91
FIGURE 7.2: THE BPM LANDSCAPE	94
FIGURE 7.3: THE GENEALOGY OF BUSINESS PROCESS RESEARCH	96
FIGURE 8.1: HOW ECBP IS DERIVED, AND HOW ITS CONTRIBUTIONS SERVE AS BASIS FOR BOWL	105
FIGURE 8.2: THE ESSENTIAL cBP FRAMEWORK	108
FIGURE 9.2: SYNTHESISING B2B COLLABORATION TASK LISTS FROM B2B STANDARDS INTO FIVE HIERARCHIES OF TASKS FOR EACH COLLABORATION PHASE	122
FIGURE 9.3: OVERVIEW OF THE VISIO VERSION OF THE DISCOVERY PHASE HTN	123
FIGURE 9.4: THE CLASSES IN THE BOWL ONTOLOGY	124
FIGURE 9.5: SIMPLE TASK DESCRIPTION IN BOWL	125
FIGURE 9.6: METHODS FOR THE TASK "BUY"	126
FIGURE 9.7: LOGICAL SNAPSHOT OF TASK "EC-ESTABLISH CONTACT" WITH METHODS EC.A, EC.B, EC.C AND EC.D	128
FIGURE 9.8: SIMPLIFIED OVERVIEW OF HOW BOWL TASKS ARE DECOMPOSED TO THE PRIMITIVE TASK LEVEL	130
FIGURE 10.1: SIMPLIFIED GENESIS PROCEDURE	133
FIGURE 10.2: HOW MATCHING METHOD "EC.B" DECOMPOSES THE COMPOUND TASK "EC" INTO SUB-TASKS CHAINED BY NESTED CONTROL FLOWS.	134
FIGURE 10.3: HOW THE TASKS ARE DECOMPOSED AND CHAINED DURING THE GENESIS ALGORITHM EXECUTION	136
FIGURE 11.2: SCREENSHOT OF GENESIS PROTOTYPE INPUT GUI, AND THE COMPONENTS INCLUDED IN THE GUI	140
FIGURE 11.3: SCREEN CAPTURE OF THE ABSTRACT BPEL OUTPUT GENERATED BY GENESIS	141
FIGURE 11.4: TEXT BASED OUTPUTS OF THE GENESIS PROTOTYPE	142
FIGURE 11.5: SCREEN CAPTURE OF THE TREE STRUCTURE OUTPUT OF GENESIS	143
FIGURE 11.6: LIST OF PRIMITIVE TASKS (I.E. WEB SERVICES) REQUIRED TO FULFILL THE USER'S GOALS AND CONSTRAINTS	143
FIGURE 11.7: PROGRESS BAR TO REDUCE USER WAITING FRUSTRATION	143
FIGURE 11.9: OVERCOMING CONTROL FLOW REPRESENTATION LIMITATIONS OF GRAPHVIZ	146

FIGURE 11.10: SAMPLE ABSTRACT BPEL FILE SEGMENT WHICH CORRESPONDS TO THE .DOT AND THE GRAPH FIGURES BELOW	147
FIGURE 11.11: CORRESPONDING .DOT FILE FROM THE ABSTRACT BPEL.....	147
FIGURE 11.12: GRAPHICAL VISUAL OUTPUT GENERATED BY GENESIS USING GRAPHVIZ TOOLS, CORRESPONDING TO THE EARLIER ABSTRACT BPEL AND .DOT FILE DIAGRAMS.....	148
FIGURE 11.13: SNIPPET OF RESULTING TASKS IN SCENARIO 1'S CBP	150
FIGURE 11.14: SNIPPET OF RESULTING TASKS IN SCENARIO 2'S CBP	150
FIGURE 11.15: RELATIONSHIPS BETWEEN PRODUCT TYPE AND PROCESS TYPE	153
FIGURE 11.16: SCREEN CAPTURE OF PART OF THE <i>METHOD</i> CHECKLIST	155
FIGURE 11.17: EXAMPLES OF LOOPS WHICH CAN BE PROPERLY FORMULATED INTO THE OUTPUT CBP BUT MAY RESULT IN INFINITE LOOPS DURING RUNTIME	157
FIGURE 11.18: SUBSIDIARY COMPANIES OF LHT HOLDINGS LIMITED.....	159
FIGURE 11.19: TYPICAL FLOW OF KHL PALLET LEASING PROCESSES	160
FIGURE 11.20: PALLET LEASING LIFE CYCLE	160
FIGURE 11.21: ISSUE NOTE BUSINESS PROCESS	162
FIGURE 11.22: RETRIEVE NOTE BUSINESS PROCESS	162
FIGURE 11.23: EXCHANGE NOTE BUSINESS PROCESS.....	163
FIGURE 11.24: TRANSFER NOTE BUSINESS PROCESS.....	164
FIGURE 11.25: GENESIS SYSTEM FOR DYNAMIC CREATION OF LEASING BUSINESS PROCESSES	166
FIGURE 11.26: VALIDATION PROCESS.....	167
FIGURE 11.27: SCREEN CAPTURE OF LEASEOUT GOAL AND THE INPUT CRITERIA DURING VALIDATION	169
FIGURE 11.28: HIRE ORDER BUSINESS PROCESS TASKS DYNAMICALLY FORMULATED IN THE CORRECT CONTROL FLOW AND ORDER	170
FIGURE 11.29: ISSUE NOTES AND EXCHANGE NOTES BUSINESS PROCESS TASKS DYNAMICALLY GENERATED FOR LEASEOUT GOAL	170
FIGURE 11.30: EXCHANGE NOTES, TRANSFER NOTES AND RETRIEVE NOTES TASKS SUCCESSFULLY GENERATED FOR LEASEFROM GOALS.....	171
FIGURE 12.1: OUTLINE OF THESIS RESEARCH CONTRIBUTIONS	175
FIGURE 12.2: COMPARING THE STRUCTURE AND SYNTAX OF PDDL AND BOWL	180
FIGURE 13.1: AN EXAMPLE OF SELL COMMAND IN COLLAGEN SYNTAX	197

List of Tables

TABLE 3.1: CONTEMPORARY BPM STANDARDS AND THEIR UNDERLYING FLOW-BASED THEORIES.....	19
TABLE 3.2: APPLICATIONS OF CHAPTER 3'S THEORIES TO PHD METHODOLOGIES.....	36
TABLE 5.1: APPLICATIONS OF CHAPTER 5'S DISCUSSIONS TO PHD METHODOLOGIES.....	71
TABLE 6.1: STATUS AND DIFFERENCES BETWEEN PLANNING AND EXECUTION PHASE	74
TABLE 6.2: WORKFLOW PATTERNS.....	85
TABLE 8.1: ECBP PHASES DISTILLED FROM LAP APPROACHES	106
TABLE 8.2: SUMMARY OF CONTRIBUTIONS FROM LAP METHODS TO THE ECBP FRAMEWORK.....	107
TABLE 8.3: USING THE ECBP FRAMEWORK TO EVALUATE CURRENT B2B METHODS	113
TABLE 8.4: CONSTRAINTS CONSIDERED IN B2B COLLABORATION BASED ON 1966 DICKSON STUDY. SRC:[284].....	115
TABLE 9.1: BREAKDOWN OF B2B TASKS IN BOWL.....	122
TABLE 9.2: SUPERVISED CRITERIA AND THEIR RANGE OF VALUES.....	126
TABLE 11.1: A SUMMARY OF GRAPHVIZ NOTATIONS FOR CURRENT CONTROL FLOW TAGS IN THE GENERATED ABSTRACT BPEL.....	145
TABLE 11.2: BUSINESS GOALS AND CRITERIA OF 2 SELECTED SAMPLE SCENARIOS.....	149
TABLE 11.3: 14 SCENARIOS THAT CAN VERIFY BOWL AND GENESIS VIA PATH TESTING.....	154
TABLE 11.4: INPUT PARAMETERS AND ILLEGAL INPUT ENTERED FOR ROBUSTNESS TESTING.....	156
TABLE 11.5: VALIDATION RESULTS WERE HIGHLY SIMILAR TO REAL-LIFE PROCESSES	168

Nomenclature

B2B	<i>Business-to-Business</i>
BAM	<i>Business Activity Modelling</i>
BAT	<i>Business as Action Theory</i>
BOWL	<i>Business-OWL</i>
BPA	<i>Business Process Analysis</i>
BPDM	<i>Business Process Definition Metamodel</i>
BPEL	<i>Business Process Execution Language</i>
BPM	<i>Business Process Management</i>
BPM	<i>Business Process Modelling (Context provided when focus is on modelling)</i>
BPML	<i>Business Process Modelling Language</i>
BPMN	<i>Business Process Modelling Notation</i>
BPQL	<i>Business Process Query Language</i>
BPR	<i>Business Process Reengineering</i>
BPRI	<i>Business Process Runtime Interface</i>
BPSS	<i>Business Process Specification Schema</i>
cBP	<i>Cross-Enterprise Business Processes</i>
CfA	<i>Conversations for Action</i>
DEMO	<i>Dynamic Essential Modelling</i>
ebXML	<i>Electronic Business using eXtensible Markup Language</i>
ECBP	<i>Essential Cross-Enterprise Business Process Framework</i>
EDI	<i>Electronic Data Interface</i>
IS	<i>Information Systems</i>
LAP	<i>Language/ Action Perspective</i>
OASIS	<i>Organization for the Advancement of Structured Information Standards</i>
OMG	<i>Object Modelling Group</i>
OWL	<i>Web Ontology Language</i>
PSL	<i>Process Specification Language</i>
SOA	<i>Service Oriented Architecture</i>
UML	<i>Unified Modelling Language</i>
VMI	<i>Vendor Managed Inventory</i>
W3C	<i>World Wide Web Consortium</i>
WfM	<i>Workflow Management</i>
WfMC	<i>Workflow Management Coalition</i>
WS-CDL	<i>Web Services Choreography Description Language</i>
WSOI	<i>Web Services Choreography Interface</i>
WSCL	<i>Web Service Conversation Language</i>
WSFL	<i>Web Services Flow Language</i>
XLANG	<i>The eXtensible Language</i>
XML	<i>eXtensible Markup Language</i>
XPDL	<i>XML Process Definition Language</i>

Chapter 1. Introduction

1.1 Background

Every human endeavour, from planning a holiday to managing complex manufacturing processes, is governed by *processes*. *Processes* can be optimised either through experience (e.g. planning a holiday) or by prudent scientific investigations (e.g. manufacturing processes). Likewise, there are *business processes in business*. Business processes (e.g. purchase orders, price negotiations, shipping management, request for quotations, merger-and-acquisition procedures, etc.) are commonly found in business organizations and across organizations.

Fundamentally, business processes are either *private* or *public* business processes. Private business processes are those internal to the enterprise and can be at the *strategic, management* or *operational level*. Public business processes involve external organisations, e.g. delivery of goods, ordering of materials, etc. Public business processes are also commonly known as *cross-enterprise business processes (cBP)*. With intensified globalisation, cBP's are becoming more important because of (1) the rise in frequency of goods ordered and delivered, (2) the need for fast information transfer, (3) quick decision making, (4) the need to adapt to demand change, (5) more competition from similar businesses all over the world and (6) shorter cycle times.

In a bid to deal with these challenges, Information Technology (IT) was harnessed to manage business processes. Previously manual hand-filled forms have increasingly been replaced by their “paperless” electronic counterparts. This gave rise to Business Process Management (BPM). According to prominent BPM researcher van der Aalst [1], BPM is defined as “*supporting business processes using methods, techniques and software to design, enact, control and analyze operational processes involving humans, organizations, applications, documents and other sources of information.*” Software tools supporting the management of such operational processes became known as Business Process Management systems (BPMS).

At the end of 2006, the BPMS market reached nearly US\$1.7 billion in total software revenue [2] and began to exhibit the characteristics of an early mainstream software market, i.e. proven technology, stable vendors, vendor consolidation and rapid user adoption. The BPMS market is also the second-fastest-growing middleware (a type of integrative software) market segment; Gartner estimates that the BPMS market will have a compound annual growth rate of more than 24% from 2006 to 2011 [2].

Interest in BPM from among practitioners and researchers grew rapidly. A wide variety of paradigms and methodologies from organization management theory, computer science,

mathematics, linguistics, semiotics, and philosophy were adopted, making BPM a cross-discipline “theory in practice” subject. Perhaps because it is cross-disciplinary, BPM practice and research are fraught with duplication and misunderstandings, and are starting to show signs of not addressing the urgent needs of today’s supply chains.

Another high-level business process not yet comprehensively researched into is the burgeoning *Cross-Enterprise Business Process* (cBP) (a.k.a. B2B Collaborations, B2B Process Integration). Gartner forecasts that by 2011, midsize-to-large companies will at least double the number of multi-enterprise integration and interoperability projects they're managing and spend at least 50% more on B2B projects, compared to 2006 [3]. Global 2000 companies will also double the number of automated multi-enterprise transactions, documents and process execution events between 2006 and 2011 [3]. These forecasts affirm the emerging importance of cBP.

1.2 Motivation for this Research

In 2007, a Forrester Research publication [4] states that most business applications are too inflexible to keep pace with the businesses they support. Today's applications force people to figure out how to map isolated pockets of information and functions to their tasks and processes, and they force IT professionals to spend too much budget to keep up with evolving markets, policies, regulations, and business models. The report also states that IT's primary goal during the next five years should be to invent a new generation of enterprise software that adapts to the business and its work and evolve with it.

Forrester calls this new generation “Dynamic Business Applications”, emphasizing close alignment with business processes and work as designed for people and adaptability to business change (build for change). The report also states that the service-oriented architecture (SOA) and business process management (BPM) techniques are the tools that will enable the so-called “Dynamic Business Applications”. A recent survey by Goh *et al* [5] has also revealed the pressing need for dynamic service-oriented approaches to overcome challenges in current supply chains such as sudden changes in demand, or the delay of flight cargo due to a natural disaster.

While current technology can manage both public and private *operational* business processes, the representation and **dynamic** formulation of higher-level *strategic, management* and *cross-enterprise* business processes is still not addressed. In the area of B2B collaborations, many companies are still dependent on the largely traditional manual modes of information exchange such as emails and procurement portals.

While it works well presently, growing information overload and the complexity of an increasingly globalised and Web-dependant supply chain will undoubtedly demand a more scalable solution for companies to realise business goals *on-the-fly*.

Part of the fundamental solution is to (1) propose a methodology that bridges the gap between high-level business goals and low-level operational tasks and (2) enhance responsiveness to sudden changes in an increasingly globally inter-connected world.

Hence, the **goal of this research** is to formulate a methodology that enables agile (i.e. quick, responsive, well coordinated) B2B collaborations through the ***dynamic formulation of cross-enterprise business processes based on high level user business goals and criteria***. To achieve this goal, three fundamental objectives were identified:

1.2.1 Objectives of Research

The first objective is to review and propose how ***high-level strategic cross-enterprise business goals are decomposed into low-level operational tasks***. While it is common for many service-oriented architecture (SOA) literature to describe the modelling and execution of the flow of business processes at the *operational level (i.e. workflow)* [6-9], few actually upscale these *low-level* tasks to *high-level* strategic tasks. This research investigates the gaps of current practices in research and industry, in particular the missing link between high-level goals and low-level tasks.

The second objective is to discover a methodology to ***decompose, on-the-fly, compound business process tasks into primitive operational tasks of appropriate granularity, for direct execution by atomic, modular programs***. Current technologies are unable to dynamically decompose high level business goals into business process tasks with the appropriate granularity. Also, current decompositions of compound tasks into sub-tasks are mostly hard-coded into business process definition languages such as Business Process Modelling Notation (BPMN) [10, 11] but dynamic decomposition is still not possible.

If business process modelling approaches can be reengineered to facilitate dynamic decomposition, we will not only be able to decompose high to low level tasks, but also dynamically generate meaningful chains of primitive tasks for execution, e.g. performed *via* Web services or even agents. Furthermore, decomposition can also reduce complex problems into manageable sub-problems, thereby lightening the inferential and computational load of solving the problem.

The third objective is to enable ***dynamic chaining of the decomposed operational tasks into the appropriate control flows required to fulfil the business goal***. Once we know the tasks to

decompose into, their sequence, the actor and, the control flow of the execution of tasks (e.g. in parallel, in sequence) must be resolved.

These three objectives can render organisations highly adaptive and capable of collaborating and integrating with others within a matter of seconds or minutes. The formulation and decomposition of high level cross-enterprise business processes can be automated.

1.2.2 Scope of Work

In this thesis, the author will only focus on the domain of *cross-enterprise business processes* (*cBP's*) as they always precede the development of supporting *strategic*, *management* and *operational* business processes. Furthermore, compared to private BP's which are highly dependent on organisational structure and management style, cBP's are largely standardised across industries (e.g. issuance of invoices, sending of purchase orders, etc) regardless of the sectors.

Dynamic formulation of cBP's involves two phases: *planning* and *execution*. *Planning* means generating the steps to decompose tasks and control flows required for the business goals stated in each collaboration scenario while *execution* entails the actual implementation and execution of the tasks generated. This is likened to planning a driving journey from point A to B *versus* the execution of the driving from point A to B. This thesis will focus solely on the planning phase of dynamic formulation of cBP's as its results will form the foundation for many possibilities of execution (which can be further researched).

The research will be validated in the supply chain industry, where cBP's is prevalent. The proposed dynamic cBP modelling and formulation will be validated by an on-site dynamic creation of cross-enterprise business processes for a regional pallet leasing company.

1.3 Organisation of Thesis

Following this chapter, Chapter 2 defines business processes and the types of business processes. Chapter 3 then introduces current approaches to business process modelling, and review their strengths and limitations. From BPM, we delve deeper into the state of the art of business-to-business (B2B) collaboration [10, 11] techniques in industry (Chapter 4) and academic research (Chapter 5). After reviewing the gaps related to the author's work in BPM and B2B, Chapter 6 investigates related techniques which gave clues to solving the research objective. This leads on to Chapter 7, which summarises the literature review and the underlying research challenges and questions of the research objective. Chapter 8 introduces the theoretical underpinnings for the author's novel approach in Chapters 9 and 10. Chapter 11 then demonstrates the implementation, verification and validation of the methods proposed,

followed by discussions on research contributions and limitations in Chapter 12. Last but not least, Chapter 13 concludes this thesis with highlights of the research and future work.

Part I: Literature Review

Chapter 2. Business Processes Defined

This thesis focuses on cross-enterprise business processes (cBP's), and more specifically, their dynamic formulation. cBP's are a subset of the broader category of “business processes”, and hence, we need to first understand the fundamentals of business processes and how business processes are currently modelled in research and in practice before cBPs can be dynamically formulated. A deep appreciation of the gaps in current business process modelling techniques and the associated IT implementations is a pre-requisite.

This chapter introduces the differing perspectives and definitions of business processes, and sets the foundation for Chapter 3, a survey of BPM theoretical underpinnings and schools of thought.

2.1 Definitions of Business Processes

Perhaps due to the multi-disciplinary nature of BPM, the inherent knowledge spans a very large spectrum. The need for a taxonomy and classification of the literature was identified in [12, 13], motivating the author to devote much time in surveying the BPM landscape. Baring the traditional process views of Frederick W. Taylor in scientific management, modern and explicit definitions of the term *business process* can be traced back to the definitions by proponents of business process re-engineering (BPR) [14] in the early 1990's.

The seminal works of Hammer and Champy [15] defined a business process as “*a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer. A business process has a goal and is affected by events occurring in the external world or in other processes*”. This definition is robust and comprehensive and effectively sums up all possible permutations of business process flows in reality. Put simply, business processes are triggered by goals and some constraints happening in the external world; each of its associated activities or tasks bears one or more inputs, with a single value added output. However, instead of viewing business processes as a mere “*collection of activities*”, business processes are better thought of as a systematic ordering of specific work activities across time and space. With this understanding, those aspects of cross-enterprise business processes that need to be optimized will be evident.

According to another seminal work by Davenport [16], this structure must be implemented with information technology. Hence, in his book, Davenport defined a business process as “*a structured, measured set of activities designed to produce a specified output for a particular customer or market. It implies a strong emphasis on ‘how’ work is done within an organization, in contrast to a product focus’s emphasis on ‘what’*. A process is thus a specific

ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs: a structure for action."

While the first two definitions defined the *goals*, *location* and the *flow structure* of a business process, two other important elements are conspicuously missing in their definitions: the **actors** of the specific work activities and the **collaborative nature** of these actors. According to Ould [17], a business process:

- *embodies purposeful activity*
- *is collaboratively executed by a group (of humans and/ or machines)*
- *Often crosses functional boundaries*
- *Is invariably driven by the outside world*

This description of business process introduced the elements of (1) *actors/ roles* and (2) *collaboration between the actors/ roles involved*. Hence, it is important to note that a business process, being a structured sequence of specific activities, is not only carried out by a single individual or department, but also ***involves many people/ machines/ systems from different organizations***, working together to achieve a common business goal.

In the author's own definition, business processes are a series or network of value-added activities, performed by their relevant roles or collaborators, to purposefully achieve the common business goal. This also implies that fulfilling business goals are the ultimate ***goal*** of collaborators; a true blue BPM or B2B software needs to express business goals and how they affect the network of activities that support them.

This definition also lays the foundation for a new way to model cross-enterprise business processes in Chapters 8 to 11.

2.2 Types of Business Processes

To the author's best knowledge, there is currently no universally accepted academic or industrial classification or taxonomy of the different types of business processes. From a higher level viewpoint, there are two main perspectives of business processes [12, 13]: the ***level perspective*** and the ***core competency perspective***.

2.2.1 Level Perspective

The level perspective basically classifies business processes into levels like that of traditional organization charts. Robert N. Anthony defines three levels of management activities [18, 19]:

1. ***Operational control***, which is "the process of assuring that specific tasks are carried out effectively and efficiently"

2. **Management control**, which is "the process by which managers assure that resources are obtained and used effectively and efficiently in the accomplishment of the organization's objectives"
3. **Strategic planning**, which is "the process of deciding on the objectives of the organization, on changes in these objectives, on the resources used to obtain these objectives, and on the policies that are to govern the acquisition, use, and disposition of these resources".

Figure 2.1: Strategic, Management and Operational Business Processes

In the author's opinion, these three levels respectively form what is known today as **Operation-Level Business Processes**, **Management-Level Business Processes** and **High-Level/ Strategic Level Business Processes** as shown (in a high-level fashion) in Figure 2.1. The three business process levels in Figure 2.1 apply to internal business processes. However, in the author's opinion, the very need for the formation of these three levels is usually triggered by an external business process, namely **Cross-Enterprise (or Collaborative) Business Processes (cBP)**. It is because of cBP's that trade flourishes. cBP's define the

business collaborations across entities and enterprises. Some examples are purchasing requests, shipments, outsourcing of services, etc. This is further elaborated in Chapter 3 and Chapter 5 later.

2.2.2 Core Competency Perspective

While the level perspective is hierarchical in nature, the *Core Competency* perspective groups business processes by their relationship to the overall operation of the organization [20]. There are mainly three groups:

1. **Core Business Processes**- These are the revenue generating processes. E.g. Software Development Departments in IBM and Microsoft.
2. **Management Business Processes** – These include the processes that ensure efficiency, corporate compliance and governance. E.g. requests, notifications, etc.
3. **Support Business Processes** – These are non-revenue generating cost components which are nevertheless crucial to the fulfilment of business goals. E.g. Transportation business processes of a manufacturing firm, or an IT department in a retail outlet chain.

2.2.3 Perspective used in this thesis

The *core competency perspective* is currently adopted by many prominent B2B standards such as RosettaNet, ebXML and OAGIS [21] (See Chapter 3). They classify common B2B tasks into scenarios, functions or departments. The advantage of the core competency perspective is that IT consultants and process owners can easily and rapidly customize the BPs. However, it creates an “*over-the-wall*” silo mentality when modelling B2B collaborations. BP’s in reality require level-based and cross-functional/ cross-department processes to realize B2B goals. For example, in typical “*Procurement*” processes, the inventory department needs to provide the current inventory status and the finance department the budget available in one transaction. Hence, a *level perspective coupled with a “process-oriented” perspective* of B2B collaboration tasks is something which the author believes is more aligned to the realities of supply chain cBP’s. We will discuss this in detail in Chapter 6.

2.3 Demystifying BPM-related terminologies

While it may seem unbelievable for a discipline with a history of about three decades, there is still a lack of clarity of basic BPM terminologies like *business process management versus workflow management, workflow, business process reengineering*. The clarifications of these terms will not be reviewed in this thesis’ main text but may be found in Appendix A and [12].

2.4 Chapter Summary

This chapter introduces the many types and perspectives of business processes.

In Section 2.1, we established the fact that business processes are *series or networks of value-added activities, performed by their relevant roles or collaborators, to purposefully achieve the common business goal*. Conversely, we can infer that each business goal will require its own set of value-added activities. Hence, it is the author's opinion that modelling methodologies for business processes must encompass all levels from high-level business goals right down to the granular operation level tasks.

In Section 2.2.3, we also established the need to express BPs as a combination of the level and cross-functional perspectives.

Chapter 3. Business Process Modeling Theory

After having defined business processes, it is appropriate to introduce current BPM theoretical underpinnings, their strengths and limitations. From the late eighties to the present, the pervasiveness of business process management triggered an increased interest in a more disciplined approach to business process modelling [22]. Within this time period, there are primarily two schools of thought observed by the author: the *Flow-based Approach* and the *Communicative Approach*.

Flow-based approaches have their background in mathematics and computer science, and offer strong modelling techniques of common control flows, and easily facilitate the automation of operational business processes into software environments. On the other hand, the **communicative approaches** (Language/ Action Perspective and Linguistics) which have their background in the study of conversations, speech acts and commitments expressed in a natural way, attempt to offer an intuitive, “natural” platform to express business processes in a context-less computing environment.

Let us now explore the key perspectives that form the theoretical underpinnings of BPM today, starting with the flow-based approach.

3.1 The Flow-Based Approach

The **flow-based approach** is the most commonly found approach in BPM practice and in the author’s observation is currently more widely accepted than the **communicative approach**. The **flow based approach** had all along been adopted in computational science and operations research.

Through formalisms, strict syntax and logic rules, the flow-based approach models flows for control and optimisation and uncovers common flow problems like deadlocks and infeasible paths. These approaches have roots in Church and Kleene’s *Lambda Calculus* [23], introduced in the 1930s, C. A. Petri’s [24] *Petri Nets* (1960’s), and Milner’s [25-27] *Pi Calculus* in the 1980s. Because the **flow based approach** is succinct in expressing complicated flows and is robust in modelling operational business processes, it enjoys widespread acceptance by industry. Examples of the flow based approach range from simple *flow-charts* to the syntactically complex but powerful *Pi-Calculus*.

Despite its strengths, the flow based approach is often mistaken as simplistic mathematical flows lacking semantics and context. The inability to model high-level semantics (e.g. goals, level of commitments, constraints, etc) partly explains its current struggle to address business level constraints and a more communicative nature of business processes in reality. Many

flow-based approaches lean heavily on web services and associated IT technologies at the expense of treating high level goals of business collaborations even in today's digital world. Generally, the flow-based modelling of business processes may be expressed either **graphically** or **grammatically**.

3.1.1 Graphical Flow-Based Methods

3.1.1.1 Petri Nets

Amongst the graphical methods, Petri Nets is one of the most widely adopted theory in BPM Standards (see Section 3.1.2; further discussions of BPM standards in Appendix B). Some of these standards include the WSFL [28], BPMN [10] and BPEL [29, 30]. The classical Petri Net was developed by mathematician Carl A. Petri [24] in the 1960s.

Figure 3.1: A Simple Petri Net and its components

The main constructs of a Petri Net are *places* (stopping points in the process, but not states), *transitions* (events that drive process movement), *tokens* and *arcs* (connecting transition to place or place to transition). The movement of *tokens* traces the flow in a Petri Net. In recent literature, the classical Petri Net has been enhanced with colour (guards and conditional branching) and hierarchy (subnets) [31-33].

Figure 3.2: A Petri Net for the Processing of Complaints. Src.: [34]

One of the earliest application of Petri Nets to formalise workflow management was in 1998 when van der Aalst [34] demonstrated the flow control and mathematical robustness of

business processes. Other high-impact applications include the Workflow Patterns [35], which use Petri Nets as the basis for the flow semantics in patterns and also in the BP modelling language Yet Another Workflow Language (YAWL) [36]. An example of a “Processing of Complaint” business process was the Petri Net model created by van der Aalst in 1998 see Figure 3.2. Although the model traces the control flow of this business process, there are limitations:

1. Information (e.g. data, decisions made, etc) that is to be passed through the process are not found in the model.
2. There is no clear distinction between roles and the activities. The labels on the transitions may mean *either* the role performing the activity or the activity itself. This causes a mix-up between the actor and the activity and both cannot be simply treated the same way.
3. The business goal and related business semantics are not clear. Within the typical Petri Nets, one would find that there is no notation stating the desired goal of the process, which is an integral part of business processes (recall Section 2.1).

Hence, while Petri Nets are good at describing control flow it is lacking in high-level business semantics. Despite these limitations, many so-called BPM standards are purportedly founded on “Petri Nets”, even when their limitations are clear [37].

3.1.1.2 State Machines

A finite state machine (FSM), or simply a state machine, is a model of behaviour composed of a finite number of states, transitions between those states, and actions [38, 39]. The State Machine has been around since the advent of computer science, with key contributions from Turing, Moore, Mealy and Harel [7]. It enables computer scientists and programmers to inspect the way in which the logic runs when certain conditions are met. Like Petri Nets, state machines were also adopted to model business processes, mostly in the mid-nineties. A prominent example of such an application is in Microsoft’s Windows Workflow (WinWF) [40, 41], which is one of several workflow types supported by Microsoft .NET 3.0 Framework.

However, from the viewpoint of business process modelling, the usage of state machines as the basis for modelling BP are usually confined to artificial intelligence-related methods [42, 43], for instance, the ‘matchmaking’ of *ad hoc* business processes [44-46]. In the author’s view, state machines are ideal for inspecting logical flows in low level operational tasks of business processes, but are devoid of the contextual and semantic aspects of high level compound tasks.

Hence this thesis will not elaborate on state machines and their limited applications to workflow and business process management. However, a good online tutorial of state

machines and their application into the Windows Workflow can be found at: <http://www.odetocode.com/Articles/460.aspx>.

3.1.1.3 Flow Charts and Role-Activity Diagrams

Besides Petri Nets and State Machines, there are two traditional and commonly-adopted methods to depict business processes:

1. Flowcharts

First proposed by Gilbreth in 1921 as “Process Charts” [47, 48], flow charts are now adopted in many areas, mainly in scientific management. While many vendors and practitioners advocate adopting BPM graphical standards like BPMN [10, 11, 49] and UML AD [50, 51], in the author’s industry experience, many end-users are still most comfortable expressing their business processes in simple flowchart notations.

The advantage of using flowcharts is the ease of depicting and communicating process flow and constraints between business process owners and IT-specialists. However, while states, decision points and actions are clearly annotated, there is a lack of clarity in the expression of roles (unless explicitly labelled in the stages). Another disadvantage is that representing business processes in flowchart notation creates redundant work, because the business process owner has first to articulate the process blueprint in flowchart format to the IT-specialist, who then translates the info into standard BPM notation.

2. Role-activity diagrams

Another common graphical business process modelling technique is the Role-Activity Diagram (RAD) (Figure 3.3). The Role Activity Diagram was born out of the study of coordination by Holt *et al* [52] in 1983 and later adopted by the UK government-sponsored IPSE2.5 project during the late eighties. The focus was on “processes that involve the coordination of inter-related activities carried out by people in organisations” (i.e. business processes as we call them now). A RAD is a notation for describing the interaction between people as they perform collaborative activities. Like the flowchart, it is also simplistic but, unlike the flowchart, it **clearly defines roles** (see Customer, Sales Clerk, Delivery Operator of Figure 3.3). Although intuitively simple and easy to read, RAD can depict very complex processes [53, 54]. Like the flowchart, the RAD notation is easily understood by business people who can describe BPs in a very intuitive and natural way [55].

Figure 3.3: An example of a simplistic purchasing business process in RAD format. Src.: <http://www.instream-dev.co.uk/instream/rad.jsp>

3.1.2 Grammatical Flow Based Methods

As stated earlier, many contemporary flow-based BP modelling standards are either grammatical or diagrammatical in nature. A grammar is a set of formal language rules that describe how to form strings from the language's *alphabet* that are valid according to the language's *syntax*[56]. This section focuses on the **grammatical** flow based method, by briefly introducing: Pi-Calculus [37, 57] and Process Specification Language (PSL) [58, 59].

3.1.2.1 Pi-Calculus

Despite its academic origin [27], Pi-Calculus has become one of BPM practitioners' most attention getting terms [7]. As shown in Table 3.1, many popular BPM standards like XLANG, WSCI, BPML, BPEL and WS-CDL claim to be based on the Pi-Calculus. Although these claims have been challenged by BPM researchers[37], BPM and its predecessor workflow management are not simply an extension of the theory of Pi-Calculus [57].

Pi-Calculus in a nutshell

As a grammatical flow based method, the syntax of Pi-Calculus has been described as one that lets you represent [60]:

1. **processes**,
2. **parallel composition** of processes,
3. **synchronous communication** between processes through **channels**,
4. **creation of fresh channels**,

5. **replication of processes**, and
6. **non-determinism** (i.e. states where more than 1 resulting action can be described).

Put simply, **actions** (sequential, recursive or concurrent) are encapsulated in **processes**, which in turn run on **channels**, the media of communications among processes (e.g. to send data to a receiver or to receive a reply from another process, etc.). On these channels, **messages** (e.g. business information) are typically exchanged [25, 27]. More details and examples on Pi Calculus can be found in Appendix C.

Notable Strengths of Pi-Calculus in the Context of BPM

The strengths of Pi-Calculus in the context of BPM are [7, 37, 61]:

1. Robust expression of control flow

In Pi-Calculus, the sequential, parallel, conditional, and recursive behaviour of a process can be defined in a comprehensive yet concise manner. In the author's opinion, the most significant and powerful edge of Pi-calculus, over graphical formalisms is its ability to express recursive actions (commonly found in negotiation business processes, and will be covered in the methodology chapters later). This is currently a big limitation of typical BPM graphical notations.

2. Message-based communication

The heart of Pi-Calculus is its clean syntax and semantics for inbound and outbound messaging.

3. Adaptability and Mobility

Contemporary business processes (especially cBPs) cannot hardcode the specific addresses of where the underlying Web services are located and require the flexibility to express and change these addresses dynamically. Pi-Calculus' dynamic addressing capability enables control flow and message passing protocols of processes to be defined without being tied down by the exact addresses of the Web services.

4. Modelling path permutations of dynamic processes

Because of the preceding 3 points, there is a consensus [7, 37, 61] that Pi Calculus is especially adept at modelling (hard coding) the possible permutations of paths in dynamic processes (e.g. real-time situational dependent decisions). Strictly speaking, the process definitions in Pi Calculus are hard-coded before execution and not generated on-the-fly. Hence, it cannot be termed as a form of dynamic business process modelling. Rather, its strengths are in the representation of non-deterministic states in the processes (see page 16).

Limitations of the Pi-Calculus in the context of BPM

While Pi-Calculus is a natural choice for describing mobile and constantly-changing collaborating partners, it is not a natural choice for [7] describing abstract data types and describing states with rich or complex data structures.

In summary, Pi Calculus, with its capabilities to represent message passing through channels, and the composition of processes, is ideal for modelling non-deterministic processes dependent on real-time situation-dependent processes (i.e. dynamic processes). However, “dynamic processes” should not be mixed with “dynamic *formulation* of business process definitions”. Pi Calculus is unable to do the latter. A modelling methodology that addresses the challenge of *on-the-fly* generation of process definitions based on high-level business goals and constraints will be proposed in Chapter 9 and Chapter 10.

3.1.2.2 Process Specification Language

First proposed by the National Institute of Standards and Technology (NIST) in 2003, the Process Specification Language (PSL) is a neutral representation of manufacturing processes that supports automated reasoning [62]. Put simply, PSL is meant to reduce the need for developing point-to-point translation programs to facilitate communication between applications in a manufacturing context.

For it to act as a mediator, an ontology (i.e. a set of specialized terminology along with some specification of the meaning of terms in the lexicon [59]) is necessary. This ontology represents primitive concepts found across basic manufacturing, engineering, and business processes; some examples are “*Activity*”, “*Object*”, “*Timepoint*” and “*Relationship*”. These primitives are coordinated by functions (e.g. *beginof*, *endof*) and relations (*participates-in*, *between*, *before*, *occurring-at*) that are common to all types of manufacturing processes [59].

The mediation strengths of PSL was recognised and adopted by BPM interface standards such as the Business Process Definition Metamodel (BPDm) [63], which serves as the translator between graphical and grammatical BPM standards. An in-depth discussion of PSL syntax and features is not within the scope of this report. In short, PSL is an ontological representation of a specific field, namely, manufacturing processes. Despite its claim to be intuitive and easy to use, its adoption has been rather slow. Although it set out to be a process modelling formalism, it has become instead an ontology for manufacturing processes. The usage of ontologies to describe components and relationships between them is applied by the author in his approach in Chapter 9.

3.1.3 Flow-based Approaches in the Industry

Table 3.1: Contemporary BPM standards and their underlying flow-based theories

#	Contemporary BPM Standards	Pi Calculus	Activity Diagrams	Petri Nets	Process Specification Language
1.	BPML[64, 65]	X			
2.	WS-CDL[66, 67]	X			
3.	WSOI [68]	X			
4.	XLANG [69]	X			
5.	WSFL [28]			X	
6.	BPEL[29, 30]	X		X	
7.	YAWL [36]			X	
8.	BPMN[10, 11]		X	X	
9.	BPDM [63]				X
10.	UML[51, 70, 71]		X		

After reviewing flow-based methods, it is useful to understand their influence on the prominent flow-based BPM standards [13] in the industry. The major BPM industry standards that claim to be based on major grammatical or diagrammatical approaches described in earlier sections are shown in Table 3.1, which highlights 3 important observations about trends in BPM research:

1. Most of the contemporary BP modelling standards are mostly based on either the grammatical Pi-Calculus or the graphical Petri Nets (see Section 3.1.1.1). Of the 10 contemporary BPM industry standards listed, 5 are based on the Pi-Calculus and 4 are based on Petri Nets. Because of the influence of these 2 formalisms, the main advantages and limitations of both formalisms were covered in Sections 3.1.1.1 and 0.
2. It also hints at signs of the dangerous trend of “Yet Another Modelling Approach (YAMA)” as mentioned in [72]. This trend generated the plethora of modelling standards in the market today, but it stifles true progression in BP modelling knowledge. To be fair, some of these standards are used for different purposes. For example, WS-CDL (Web Service Choreography Description Language) is only used for Web Service Choreography (the ad hoc chaining of relevant web services) while BPEL (Business Process Execution Language) is used for Web Service Orchestration (pre-defined chaining of Web Services according to a definition). However, WSOI, BPEL, BPEL’s predecessor XLANG and BPML do have similar duplicating purposes. Thankfully, the industry is starting to show signs of consolidation towards BPEL.

3. At the time of writing, there are 96 types of formalisms listed in the Formal Methods Virtual Library database [73] which describe systems such as hardware, real-time programs, protocols), in terms of:

- the types of desired properties (e.g., correctness, deadlock freedom, timing, liveness),
- their underlying logical power (e.g., first-order, higher-order),
- their underlying semantic model (e.g., type of algebraic structure),
- and their usability (e.g., expressive power, tool support, scalability).

However, in reality only 3 of these 96 formalisms (i.e. Pi-Calculus, Petri Nets, and State Machines) have been applied to BPM research with impact. It appears worthwhile to explore the potential of other formalisms to close some knowledge gaps in current BPM knowledge (as future work or even as a future Masters research program).

After reviewing the flow-based approaches, we shall now investigate how communicative approaches impact business process modelling and contribute to the dynamic business process formulation research challenge.

3.2 The Communicative Approach

Compared to the flow-based approaches discussed in the earlier section, the **communicative approach** focuses on the more *natural* side of modelling business processes. Its approaches to modelling the ‘deep structure’ (discussed later in Section 3.2.3.5) of business process helped the author model the essential components of B2B Collaborations into a framework enabling the dynamic formulation of collaborative business processes in Chapter 9.

The communicative approach has its roots in Searle’s *Theory of Speech Acts* [74] in the 1930s, which expounded on how verbal or written conversations eventually led to real commitments. Linguistics (philosophy of language), grammatical relationships [75, 76] and Natural Language Processing [77] were the other pioneering influences of this approach. From the late 1980’s to the 1990’s the *Theory of Communicative Actions* [78] of the **communicative approach** was quite widely adopted together with the Theory of Speech Acts in the modelling of phases and processes of business transactions. A few examples include the Action Workflow [79] (Section 0), and the DEMO [80, 81] project (Section 3.2.3.5).

3.2.1 Rationale for Communicative Approaches

According to Chomsky [56], a grammar of syntax does not describe the meaning of the strings or what can be done with them in whatever context —only their form. In other words, current grammatical flow-based BPM techniques stated (section 3.1.2) are usually context-less.

Meaning and context can be handled by *Semantics* and *Pragmatics* respectively in the science of *Semiotics* (i.e. the study of symbolic systems). In the science of *Semiotics* there are 3 components in linguistic communication [82]:

1. *Syntax – Dealing with structure of symbols*
2. *Semantics – Dealing with the meaning of the structure of symbols*
3. *Pragmatics – Dealing with the context of their usage*

In the author's opinion, current BPM theory and applications are only addressing the *Syntax* component, leaving wide gaps in the *Semantics* and *Pragmatics* area. The current research climate also recognises this, and is actively pursuing a hybrid of *Semantics* and *Web Services* [83] to give: ***Semantic Web Services*** (See Section 5.1). Another semiotic-based approach is the ***Language/ Action Perspective (LAP)*** paradigm research, which fits snugly into the gaps experienced by flow-based methodologies [82]. In the context of BPM, LAP researchers were able to identify the essence of business processes [84, 85], generic models of transactions [86-89], frameworks for cross-organisational workflow integrations [90-94], etc. Some of these essential components and frameworks lay the foundation of the methods proposed in this thesis.

3.2.2 ***Language Action Perspective (LAP)***

Unlike Petri Nets and Pi-Calculus, the *Language Action Perspective (LAP)* BPM mainstream practitioners never explicitly adopted it as a modelling technique. In its twenty-odd year history, it remained largely of academic interest only [95]. LAP was first introduced in Information Systems (IS) design in 1980 by Flores and Ludlow [96], who argued that human beings are fundamentally linguistic beings. It was argued that *language* is not only the medium for exchanging information (as in reports, statements, fax, etc.) but also for *actions*, e.g. promises, orders, declarations, etc. This perspective became the bedrock for research into *Computer-Supported Cooperative Work (CSCW)* [97]. The author notes that the underlying concepts of CSCW [98] and BPMS design are similar as both support human office collaborations with IT systems.

3.2.2.1 ***Rationalistic approaches vs LAP approaches***

In *rationalistic* approaches like emails and flow-based workflow management systems, it is tacitly assumed that more information processing gives rise to more options and hence more effective decision making. However, organizational productivity is not only dependent on information, but *how* people get things done. More incoherent information from a growing number of systems will lead to information overload and confusion [99]. Furthermore, most current BPM approaches (e.g. UML AD, BPMN) cannot systematically and formally deal with the *pragmatics* of data (i.e. its use and effects) [100]. Flores *et al.* argued that computers would

be more effective if they recorded and tracked commitments, rather than simply moving information [82, 96, 99]. In contrast to rationalistic approaches, LAP deals with [82, 97]:

1. How language creates a ‘common reality’ for all the communicating partners (i.e. the ontology) (basis for the author’s methodology in Chapter 8), and
2. What people (and companies) do while communicating,
3. How these (communication) activities are coordinated through language.

3.2.2.2 The birth of the term “Language Action”

After the initial ideas of Flores and Ludlow in 1980 of “*Doing and Speaking in the Office*” [96], Goldkuhl and Lyytinen [101] coined the term “language action” in 1982 in “*A Language Action View of Information Systems*”. In their publication, Goldkuhl and Lyytinen asserted that the development of an information system is a social practice tied to a practical interest [102]. According to Weigand [102], “language action” was later adopted by Winograd and Flores in their 1986 book “*Understanding Computers and Cognition: A New Foundation for Design*” [103] to debate about the grandiose Artificial Intelligence pretensions of the late 1980’s. They postulated that the very idea that intelligence can be reduced to the smart processing of a set of explicit knowledge rules is untenable.

3.2.2.3 Importance of the human decision maker in the loop

In the second part of the book by Winograd and Flores [103], they proposed a new direction for design, namely human communication and coordination: *Instead of getting the human out of the ‘loop’, as AI was heading for, it is necessary to reaffirm the position and responsibility of the human decision maker in the loop.* This was later developed as the *Conversation for Action (CfA)* approach, on which their system The Coordinator is based upon. The author identifies with keeping *the human decision maker in the loop* in his proposed methods later in Chapter 9 and Chapter 10.

3.2.3 Applications of the LAP in BPM

LAP addresses the ‘pragmatic’ aspects of collaborations in real-life BPM, i.e. context. Hence, we can gain valuable insights into bridging high-level business goals with the commonly modelled low-level, syntactic operational tasks. Let us begin by exploring five high-impact LAP-based BPM approaches, starting with the Conversation for Action approach.

3.2.3.1 Conversation for Action (CfA) Approach

The Coordinator by F. Flores and T. Winograd [82, 99] was the first software application using the LAP approach. The Coordinator is a system for managing action in time, grounded in a theory of linguistic commitment and fulfilment of conversations. Conversations are

essentially temporal, both as a sequence of acts and in the wider context of a community or organization. In making a request or promise, the speaker brings into a shared domain of interpretation, **a set of conditions to be fulfilled through action in the future** [82].

Flores and Winograd designed the Coordinator based on a generic network of speech acts called the **Conversation for Action (CfA)**, which represents recurrent patterns of conversations in mutually understood domains associated with formally declared roles in a typical office [96]. Like Searles and Habermas, they also asserted that language conveys not only information but also commitment, and that people act by expressing assessments and promises. Figure 3.4 shows the CfA structure for action-oriented conversations. Arcs represent actions while nodes represent the states of conversations. *A* starts the conversation by making a **request** to *B* in state 1. In state 2, *B* can accept, reject or negotiate a change; *A* could cancel the request or negotiate a change of the initial conditions.

Figure 3.4: Conversation for Action (CfA) proposed by Winograd and Flores [104]

Following the normal course of the conversation, at the end states 5, 7, 8 and 9, the actors would have arrived at a consensus and fulfilment. It is also interesting to note that all the acts are speech acts in Searle's taxonomy of illocutionary forces (i.e. strength of commitments; more in Appendix D) and like a B2B collaboration protocol, each act is conditioned by its predecessors. Winograd and Flores view the CfA as a kind of "dance" in which particular linguistic steps move toward completion:

1. *if an action has been requested of you, you promise or decline;*
2. *if you have promised to complete the action, you report completion or revoke your promise;*
3. *if you have requested an action, you cancel your request, ask for a progress report, or declare that your conditions have been fulfilled and the action completed.*

The underlying basis in CfA is its claim that social action happens through language. The conversational *dance* is a social dance through which actions are committed and fulfilled. Executives in an organisation are part of a conversation network which the organisation is keen to maintain. The CfA was implemented as software known as the Coordinator. A deeper discussion of the Coordinator is not within the scope of this thesis but can be found in [104]. In summary, as the Coordinator is the first of its kind, it received its fair share of positive and negative feedback on the approach [105] from the public [106]. Most importantly, the CfA introduced a method for a more “direct” way to express high level human communication, a challenge at the heart of this Ph.D. research (see Section 3.3). CfA was the precursor to present-day LAP applications (e.g. SAMPO, DEMO, Action Workflow, BAT) which address the real needs of cooperative office work or in modern day “business process management”, and the basis for Chapter 8.

3.2.3.2 ***Speech Act-based information analysis Methodology with comPuter-aided tOols (SAMPO)***

Proposed by the Swedish team of Aumaraki, Lehtinen and Lyytinen from 1983 to 1988, the *Speech Act based information analysis Methodology with comPuter-aided tOols (SAMPO)* [107] was another pioneer application of LAP. Like Searle and Habermas, SAMPO also views organisational activities as a series of speech acts creating, maintaining, modifying, reporting, and terminating commitments. It **focuses on the abstract structure of these social interactions though (1) a modelling methodology and (2) an information system**. SAMPO claims to augment the *syntactic* and *semantic* conceptualizations of an information system, resulting in the modelling of *pragmatics* of business communication [107].

Modelling Business Processes via the SAMPO Methodology

To model organisations as systems of communicative action [108], SAMPO uses concepts of the *Speech Act Theory* and *discourse analysis*¹ [109] for the information analysis of organisations [110, 111]. The designer starts by considering two domains: the ***entity domain (ED)*** and the ***action domain (AD)***.

The ED comprises of *static* entities that persist over longer periods of time. From the author’s view, examples of such static entities can be the collaborators (roles), the standard list of operations in typical B2B collaborations. On the other hand, the AD *includes* dynamic entities called ***Acts*** [110, 111]. Two types of Acts are studied:

1. ***Instrumental acts (IACTS)*** accomplish changes in the ***ED***.

¹ In semantics, discourses are linguistic units composed of several sentences; in other words, conversations, arguments, or speeches. Discourse analysis is a branch of linguistic studies analyzing written, spoken or signed language but is not within the scope of this thesis.

2. **Speech acts (SACTS)** are defined as *symbolic deeds* that result in linguistic expressions having a "meaning" [110, 111]. SACTS form larger wholes called *conversations* or *discourses* and SAMPO introduces an analysis and modelling method for discourses in organisations.

Figure 3.5: An Example of a Conversation Graph for an Invitation Discourse. Src: [107]

After an analysis of the quality of the discourses (not in scope of report; more information in [107, 110]) in which coherency, completeness, and ambiguity are defined, SAMPO recommends modeling business processes with the following [108]:

1. **Discourse graph.** This graph describes the *overall structure* of the discourse, and discourse objects, their properties and relationships. It can be thought of as a partial "script" or "schema" for communications. Most importantly, the graph also defines *all conversation possibilities for each discourse participant*.
2. **Conversation graph** characterises the dynamic discourse features. It *describes the stages* and the *movements* in a conversation as well as the *conditions* (which in SAMPO are called *predicates*) that restrict and control the performance of acts.

Both the Discourse and Conversation Graphs describe networks of acts and help to detect [108].:

1. principles needed to set-up and control commitments,

2. inconsistencies in the co-ordination of commitments and
3. possibilities for organisational development that simplify communication and control mechanisms.

In relation to this research, even though SAMPO was not able to dynamically formulate semantically rich business processes on the fly, it has demonstrated how the use of LAP can help model both the syntactic (ED) and the semantic (AD) components of business processes. Also, the clarification between ED and AD in SAMPO highlighted the differentiation between static and dynamic domains. The execution of dynamic entities in AD impacts the static entities of ED.

3.2.3.3 Action Workflow Approach

Action Workflow [88], an underlying theory of a workflow management software by Flores' Action Technologies, appeared in a 1992 publication by Medina-Mora *et al.* It is one of the most cited proposals to model & manage workflow. Action Workflow is based on the Speech Act Theory and the CfA conversational pattern. It can be seen as [112]:

1. *a theory (a generic business framework),*
2. *an analysis and modelling method and*
3. *a supporting software tool*

This three-part approach is commonly observed in all LAP approaches, and acts as the basic approach by the author in solving the dynamic cBP formulation challenge in the later chapters. Figure 3.6 shows the basic sequence of actions in the action workflow loop. There are **always 2 roles**: an identified **customer** and a **performer**; the loop deals with a particular action that the **performer** agrees to complete to the satisfaction of the **customer**.

Figure 3.6: The Action Workflow Loop, the most fundamental transaction in all business processes, according to [88]

The action workflow loop is like an “atomic element” of business processes. By combining these loops, all the complex processes in organizations can be modelled (See Figure 3.7). The loop has four phases [88] (see Figure 3.6):

1. **Proposal** - The customer requests (or the performer offers) completion of a particular action according to some stated conditions of satisfaction.

2. **Agreement** - The two parties come to mutual agreement on the conditions of satisfaction, including the times by which further steps will be taken. This agreement is only partially explicit in the negotiations, resting on a shared background of assumptions and standard practices.
3. **Performance** - The performer declares to the customer that the action is complete.
4. **Satisfaction** - The customer declares to the performer that the completion is satisfactory.

At any phase, there may be additional actions, such as clarifications, further negotiations about the conditions, and changes of commitments by the participants. The structure is defined by the speech acts of participants, **not** the actual actions of individuals to meet the conditions of satisfaction. The key difference in the Action Workflow approach from other BPM (or workflow as it was known at that time) systems is the shift from the **task structure** to the **coordination structure**. In a more traditional workflow approach, actions of coordination are seen as one kind of task or as a flow of information between tasks. In Medina-Mora *et al's* perspective, tasks are defined by the requests and commitments expressed in the loops.

Figure 3.7: An example workflow loops making up the core structure of a Candidate Interview Review Process [88]

This shift is analogous to moving from a view of a network as a collection of nodes (with links between them) to seeing it as a collection of links (with shared nodes). Although all the elements are still there, the different starting point leads to different potentials for representing and supporting the activities [88].

The simple workflow loop structure is both general and universal. It is general in that it occurs whenever there is coordination among people, regardless of what they are doing. The words “customer” and “performer” apply to people within a single organization as well as outside the

organization (see Section 3.3; adopted by the author's research in Chapter 9 and Chapter 10). The loop structure is universal in that it is independent of any culture, language, or communication medium. There are endless variations in the specifics of how the steps are taken, what other loops are triggered, and how people respond to breakdowns within the loops, but the basic structure is the same.

3.2.3.4 Business as Action Theory (BAT) Framework

The fourth prominent LAP-based business process modelling approach is the *Business Action Theory*. The Business Action Theory is the successor of the *Business as an Action game Theory (BAT)* [112]; both of which were proposed by Goldkuhl and Lind [87, 113, 114].

Figure 3.8: An illustration of a BAT generic business framework in action. Src: [115]

The Business as Action game is graphically illustrated in Figure 3.8. This generic framework is strongly influenced by communicative action theories of Habermas and Searle, and the language game (discussed in brief later)[116]. According to Goldkuhl [115], there are **always two parties involved in a business transaction**: the **supplier** selling and a **customer** buying.

In each transaction, there are *communicative actions* and *"material" actions* (as e.g. delivering physical goods). Like in earlier discussions, the communicative actions in a business transaction like requests and commitments are also not mere information transfer;

they change the relations between seller and buyer and get their meaning from the prevailing business context.

How the BAT is influenced by the Language Game

As mentioned earlier, the generic framework in Figure 3.8 is partly influenced by the notion of language game [116], which is a powerful way to describe communication as actions performed and interpreted according to different inter-subjective rules. In the language game, there are rules for single acts and for relations between different acts as a whole. When we add in material acts to the language game's communicative action rules, we have the constructs of a business transaction. Action conditions **necessary for establishing a business transaction** (Figure 3.9) were also highlighted by Goldkuhl [87, 113] in a very detailed fashion.

Figure 3.9: The constituents of the business transaction in BAT. Src: [115]

Figure 3.9 provides us with a brief summary of the business transactions in BAT (which partly forms the basis for the author's framework in Chapter 8). BAT proposes that all business transactions are started by an *offer by the supplier* and *order by the customer*. The offer and the order together form a contract. Such a contract is a mutual commitment. The supplier's offer is based on capacity and know-how to produce products or services. The customer's purchase interest in the supplier's offers is based on some needs, which in turn can be intrinsic to the customer. In the author's opinion, the need for the supplier to evaluate its capacity and know-how and the customer's needs for the products or services mark the real starting point of transactions (more in Chapter 8).

According to BAT, in many cases, the customer negotiates with the supplier. An inquiry-based offer can lead to new questions and then a new offer and so forth. There can be bidding with

offers and counter-offers. The negotiation process leads to more complex situations and involves an elicitation of customer needs and also an investigation of the available and possible supplier product capacity and know-how. This negotiation process ceases when the offer is either accepted (e.g. an order) or rejected. If accepted, the supplier and customer enter into a contract expressing mutual commitments (as described above). It is important to view commitments made by the supplier are commitments of both future action and non-action.

After the contract agreement, the supplier will deliver the specified products and invoice the customer accordingly. The supplier will not demand payment that is different from an agreed amount. The latter parts of the business process involve fulfilment of commitments (delivery and payment) and related communicative action like invoicing. The invoice is a request to the customer to honour the commitment to pay. The delivered products can lead to customer satisfaction or dissatisfaction. There is also supplier satisfaction when receiving the payment. This is a necessity in business for any business activity. If the customer is not satisfied (i.e. there is perceived difference between the supplier's commitments and what he delivered), there might be claims made against the supplier. If the supplier does not get paid, there will likewise be payment claims.

Figure 3.10: A simplified version of the Business as Action game. Src: [115]

In summary, the business transaction is a process of interchanges between supplier and customer involving the creation and sustaining of business relations. There are different phases in this interchange process. Goldkuhl proposed four generic phases in the process of interchange as shown in Figure 3.9 and Figure 3.10, similar to the Action Workflow framework [115]:

1. **Proposal phase** – this phase has 3 sub-phases:
 - a. **Business identification phase**- The supplier initially identifies the potential market and customers for his products/services. This initial stage of the business transaction process. also entails identifying the needs of the customer leading to a purchase interest
 - b. **Exposure and contact search phase** - The supplier communicates the offers of sale of products to potential buyers. and actively searches for potential

customers. The customer can in a similar way try to get into contact with potential suppliers.

- c. **Contact establishment and negotiation phase** - After finding each other, the supplier and customer establish contact and exchange their business needs and expectations (offers and inquiries).

2. Commitment (contractual) phase

3. Fulfilment phase

4. Completion (acceptance/claim) phase

In the author's opinion, the BAT framework resembles the Action Workflow loop in the previous section. The main difference is that in BAT, the first phase of the workflow loop, the proposal is refined in subsequently (1) the business pre-requisite phase, (2) the exposure and contact phase and the (2) contact establishment and proposal phase. These three sub phases are also relevant to collaborative business processes, because unlike the Action Workflow loop which focuses on the customer, the BAT focuses on both the supplier and the customer. This symmetric consideration better reflects coordination in reality. However, BAT is more complex than the Action Workflow loop, since both its methodology and publications are very verbose and not easy to interpret.

3.2.3.5 Dynamic Essential MOdeling (DEMO) Framework

The **Dynamic Essential Modelling (DEMO) Framework** [84, 86, 117, 118], proposed by a group of researchers headed by Prof Jan Dietz of TU Delft in 1990 is the most direct and comprehensive application of LAP-related concepts in BPM. The DEMO framework is based on the theories of speech acts and communicative action, the Workflow Loop and the CfA, and has a long history of close to twenty years [119]. Its concepts influenced the theoretical foundation to this PhD research's challenges in B2B collaboration (see Section 3.3).

DEMO in a nutshell

In DEMO, *a social system comprises social individuals who enter into and comply with commitments*. The entering into and complying with commitments take place in a background of shared norms and values (i.e. *normative reality*), and is directly related to authority and responsibility. Similarly in organizations, every employee is allowed only to enter into commitments that they are responsible for or are authorized to undertake. These concepts are similar to LAP foundation theories mentioned earlier.

As shown in Figure 3.11, the functioning of organisations may be perceived from 3 levels of abstraction[108]:

Figure 3.11: 3 levels of abstraction of an organisation [85]

1. **Documental level** - At this level, an organisation is viewed as a group of actors that produce, store, transport and destroy documents. In other words, the documental level embodies the *substance and form* by which co-ordination becomes visible.
2. **Informational level** - At this level, one *abstracts from this substance and form* (i.e. documents) *the actual meaning*. The organisation is observed as a group of actors that send and receive information, and process this information in order to derive more information.
3. **Essential level** - At this level, an organisation is conceptualised as a group of actors that are engaged in the execution of business transactions. At the essential level, organisations are considered as networks of business transactions, which are composed of interrelated communicative acts.

Figure 3.12: The O-E-R Transaction, the essence/ deep structure of business process transactions
 DEMO proposed that at the *essential level*, the essential transaction is a pattern of activities performed by *two actors*: the *Initiator (I)* and the *Executor (E)* (see Figure 3.12). The **actors are roles in an organisation and not confined to persons**. An essential transaction (O-E-R Transaction) is composed of three phases:

1. the **Order phase** in which two actors come to an agreement about the execution of some future action;
2. the **Execution phase**, in which the negotiated action is executed, and
3. the **Result phase** in which the actors negotiate an agreement about the result as brought about in the preceding execution phase.

The successful execution of a transaction in the *Subject World* (the world of communication) results in a change in the *Object World* (the world of facts) in which the actors exist. The execution of a transaction can be described and consequently modelled at all three levels of abstraction. At the *essential level*, the transaction is described as a pattern of *performative communication* (i.e. communication that leads to the performing of actions relevant to the strength of commitment expressed). At the *informational level*, the execution of a transaction is described as the exchange of information (information flows), and at the *documental level* the materialisation of the transaction in tangible objects (documents, files etc.) is described.

Hence, true business process modelling should only consider the *essential level* [85]. It is interesting to note that current B2B systems and methods have only concentrated on the documental and informational levels. The essential level is overlooked by most proposals, and it is the author's intentions to address this gap (see Chapter 8).

Demonstration of the DEMO technique

As can be seen, DEMO concepts are extremely useful in distilling the deep structure (i.e. essential level) of business processes [85, 120]. Hence, when the DEMO method [84] is applied to the simple “*Buy product*” cross-enterprise business process, essentially the buyers and sellers are involved in the following steps:

1. *Buyer expressing an interest to buy an item from the seller*
2. *Seller quotes a price*
3. *Both agree on the price or negotiates a new price*
4. *Seller sends items over to buyer*
5. *Buyer makes payment.*

Although some steps may be missing (e.g. payment before delivery) the five steps essentially define the deep structure [85] (i.e. essential level) of the “buy process”. All five steps are necessary to complete the “buy process”; the absence of any one step invalidates the “buy process”. Also notice that all steps are independent of computer technicians, invoice formats, technical tools, clerks to enter data, trucks despatched, inventory management tools that support them. That is to say, this deep structure lays the foundation for engineering advanced cBP solutions (e.g. dynamic B2B collaboration) at the *informational* and *documental* levels.

Gaps in DEMO

Despite its clarity in the essential modelling of business processes, the DEMO methodology has some limitations:

1. No accompanying model of norms and values (i.e. normative reality)

The authority for roles to perform certain actions is based on the underlying social norms or in a business situation, the organizational hierarchy or duties of the role. If this is

captured, DEMO can be even more robust and can cover all aspects of Business Process modelling. Such normative reality can be readily modelled using ontology languages (e.g. OWL) to model the entire ontology of the organization hierarchy.

2. ***How are the basic O-E-R transactions linked up?***

Though there is the action model to link models up, the actual linkage, and the facilitation of the implicit linkages between the essential transactions are not explained.

3. ***The methodology is not easy to pick up without prior training.***

The methodology is not the usual innate way one to model business processes. To the author's best knowledge, there are industry certification courses to train people in the DEMO methodology [119, 121].

3.2.4 Automating the Negotiation Phase

While the above mentioned researchers provided the basis for extracting the most generic socially conscious way of expressing commitments through language, there was another group of researchers attempting to use the Speech Act Theory as an answer to B2B collaboration, but from an Artificial Intelligence agent oriented perspective. These proposals did not catch much attention, as they were (1) ahead of their time, in terms of popularity and adoption of XML and (2) the AI-based arguments that they assume. The in-depth study of these approaches is not in the author's intention. However, these approaches have triggered the author to further research into the AI field (See Chapter 6) and propose methods for expressing business goals in the high level in Chapter 9 and Chapter 10. To the author's best knowledge, there were two such languages:

3.2.4.1 Speech-Act-Based Negotiation Protocol (SANP)

Proposed around 1994, Chang and Woo's *Speech Act Based Negotiation Protocol (SANP)* [122] was a Natural Science and Engineering Research Council of Canada research project aimed at developing a negotiation protocol used in the so-called "Distributed Artificial Intelligence Systems". Their basis was Ballmer and Brennenstuhl's Speech Act Classification [123] and on negotiation analysis literature [122]. Unlike Habermas' and Searle's theories, Ballmer and Brennenstuhl's classification of speech acts was mainly argued as vague and had limitations due to their original works being based on German verbs [124, 125]. SANP is a communication based protocol, and not a computer-based system. The users of SANP need to incorporate their own reasoning mechanism in the program attached to the possible states in a negotiation in order to automate the negotiation process. Their protocol was implemented on an email toolkit named Strudel [122].

3.2.4.2 Formal Language for Business Communication (FLBC)

Proposed around 1997, the *Formal Language for Business Communication (FLBC)* [126] was a XML Document Type Definition (DTD) primarily proposed during the era when the B2B information exchange standard EDI was popular. Like the author of this thesis, Scott A. Moore and Steven O. Kimbrough, the creators of FLBC believed that electronic commerce must have available an **expressive and easily extendable “language” for automated communication**. Moore’s vision was that this language and its associated message interpretation system should be easily integrated with open standards in the industry (e.g., XML and X.12) so that programmers and analysts can more easily add it to agents, web pages, EDI programs, and databases. However, as there is a slant towards the Artificial Intelligence’s agent-oriented perspective of achieving B2B, this language may not be applicable in the near future.

3.2.5 Limitations of LAP and Related LAP Works

While speech acts and communicative actions facilitate the most “natural” way of modelling business processes, its application into the domain of computer systems via the LAP movement still has the following limitations as observed by the author:

1. ***The LAP frameworks did not cover transactions for collaborators with existing contracts***
While they represent most generic transactions, the frameworks discussed (e.g. CfA, Action workflow, BAT, DEMO) did not have considerations for suppliers and buyers with existing or long-running contracts.
2. ***Speech Acts are dependent on verbs, which in turn are dependant on language***
Searle’s speech acts are based on English verbs that portray commitments, and to translate this taxonomy of speech acts into another language may have its losses in meaning and expressivity. Likewise, Ballmer and Brennenstuhl’s Speech Act Classification [123] are based on German verbs and encountered similar information loss.
3. ***Absence of study in combinations of speech acts and concurrent/ parallel-running speech acts***
Currently, there is no study on the combination of speech acts as an action. This happens much in real life but in LAP applications, we are restricted to singular speech acts.
4. ***Lack of study of the modelling of whole enterprise and sector***
 - Scalability is not tested in the LAP applications. For example, if we get an extremely complicated business process to be modelled in the Action Workflow loop, the whole diagram will be very messy.
 - Level of details need to be introduced to simplify business processes at the highest level.
5. ***Storage and representation of the normative reality is need for true pragmatics***
The hierarchy of roles and the responsibilities and authorities of these roles are not captured but assumed to be understood.

3.3 Applications to Research

BPM is a broad topic, and there is currently a serious lack of treatise on the different schools of thought and a classification of its methodologies. This makes the author's challenge of dynamically formulating cBP's a gargantuan one. The only reasonable way to manage this challenge is by understanding how cBP's are modelled from the "essential level" (See Section 3.2.3.5), using techniques such as DEMO and many points considered in this chapter. Table 3.2 summarises how theoretical underpinnings covered in this chapter serve as foundations for knowledge contributions in the proposed methods in Chapter 8, Chapter 9, Chapter 10 and Chapter 11.

Table 3.2: Applications of Chapter 3's theories to PhD methodologies

Section	Theory	Assertions/ Basis / Arguments Raised	How it contributes	Impact Chapter
Flow-based approaches				
3.1.1	Graphical Flow-Based Methods	While the flow structures of the graphical approaches are easily comprehended via observation, many of the techniques are only simplifications of semantically-rich social processes into mere flows such as state machines.	The proposed dynamic cBP methodology in this thesis must contain semantic descriptions.	Chapter 9, Chapter 10 and Chapter 11
0	Pi-Calculus	Actions in Pi Calculus can be arranged sequentially, in parallel or conditional paths, or recursively.	Dynamic processes need to cater to the sequential, conditional, concurrent and recursive paths	Chapter 9 and Chapter 10
3.1.2.2	Process Specification Language	PSL works as a mediator, modeled as an underlying ontology representing primitive concepts found across manufacturing and engineering business processes	Established how ontologies can be used to describe not only the actions, but also the objects, timepoints and relationships between the concepts of business processes.	Chapter 9 and Chapter 10
Communicative approaches				
3.2.3.1	Conversation for Action (CfA) Approach	Proposed that the "human decision maker" must be in the loop (i.e. business process).	Established the need for inclusion of human decision makers in all BPM system.	Chapter 8
		In each request or promise, there is a set of conditions to be fulfilled through action in the future.	Established the need for conditions to be fulfilled when tasks are executed.	Chapter 9 and Chapter 10
3.2.3.2	Speech Act-based information analysis Methodology with computer-aided tools (SAMPO)	A designer of a BPM system must consider 2 domains: 1. Entity domain (ED) – static entities that persist over long periods of time. E.g. the required roles, the organization structure, etc. 2. Action domain (AD) – dynamic entities called acts that changes static entities in the ED.	<ul style="list-style-type: none"> Established that there are both static and dynamic entities in all business processes. This is not commonly acknowledged in many literature surveyed. Established the need for consideration of not only actions, but also all the static entities such as roles, organization structures. 	Chapter 9 and Chapter 10
		Their conversation graphs characterize the AD features and describes stages, movements in a conversation, as well as the conditions (called predicates) that restrict and control the performance of acts.	<ul style="list-style-type: none"> A generic representation of the conversations is needed. Established the need for conditions to restrict and control the performance of actions in a process. 	Chapter 9
3.2.3.3	Action	The action workflow works	Established the need for a BPM	Chapter 8

	Workflow Approach	because it has the following components: 1. A theory (a generic business framework) 2. An analysis and modeling method 3. A supporting software tool	approach to have a theoretical underpinning, a modeling method and a software. This is the foundation for the components in this research.	
		There are always 2 roles in the action workflow loop: 1. An identified customer 2. A performer	Established the truism of two fundamental business roles (buyer and seller) of every essential business collaboration/ transaction.	Chapter 8
		The action workflow loop consists of 4 phases: 1. proposal, 2. agreement, 3. performance and 4. satisfaction.	Generic business transactions contain phases where buyers and sellers enter into common understanding on the proposal, agreement, performance and satisfaction of a business goal.	Chapter 8
		“tasks are defined by requests and commitments expressed in loops”	This means that a business process is composed of generic granular actions.	
3.2.3.4	Business as Action Theory (BAT) Framework	There are always 2 parties involved in a business transaction: 1. a supplier selling 2. a customer buying	Established the truism of two fundamental business roles (buyer and seller) of every essential business collaboration/ transaction.	Chapter 8
		BAT Generic business framework contains actions and conditions necessary for establishing a business transaction (refer to Figure 3.9): 1. Proposal phase (3 sub-phases) a. Business identification b. Exposure and contact search c. Contact establishment and negotiation 2. Commitment phase 3. Fulfilment phase 4. Completion phase	<ul style="list-style-type: none"> Established how a process-oriented “phase approach” to modeling business processes enables the true modeling of the essential business process much more than the traditional functional model (See next point in table) The BAT framework has extended the action workflow with a more in-depth proposal phase, and the commitment phase. 	Chapter 8
3.2.3.5	Dynamic Essential MOdeling (DEMO) Framework	“a functional model is rigid and not adaptable to business climate changes required to existing business processes during a radical business process reengineering or design”	Debunks the traditional approaches used by industry standards such as EDI, RosettaNet for B2B Collaborations.	Chapter 9, Chapter 10 and Chapter 11
		DEMO methodology for extracting the essential ‘deep structure’ of a business process	With the essential business process extracted, we can understand the actual main goal of the process, and encourage a technology-independent design of business processes. This is something that the traditional flow-based approaches are still trying to do.	Chapter 8
		At the informational level, an organization is observed as a system of actors that send and receive information and performs calculations on this information in order to create derived information.		Chapter 8
		The O-E-R Transaction (essential transaction) is composed of 3 phases: 1. The Order phase 2. Execution phase 3. Result phase		Chapter 8

3.4 Chapter Summary

The business process management landscape is fragmented. An overview of the topics covered in Chapters 2 and 3 is compiled in Figure 3.13. Current BPM theory can be generally classified into 2 schools of thought:

3.4.1 Flow Based Approaches in BPM

The theoretical foundations for this school of thought are *process algebra* and formalisms like *Pi Calculus* and *Petri Nets*. Flow based approaches allow business processes to be visualized as networks of process flows by industry but they are inadequate in dealing with dynamic cBPs. For instance, the **graphical** flow-based methods verify the control flows of business processes, e.g. ensuring that tasks are not un-intentionally by-passed or deadlocked. However, graphical flows cannot be computer-automated unless they are first translated into **grammatical** flows.

In both types of flow-based approaches, goals and constraints are embedded in the models themselves and not explicitly stated; this is a major drawback when it comes to formulating dynamic cBPs. Besides, the mutual commitments of cBPs are not embodied in both types of flow-based approaches. Hence, the need for a “communicative” approach to do it.

3.4.2 Communicative Approaches in BPM

While *Speech Acts Theory* / *Theory of Communicative Action* articulate commitments, it is the *Language-Action Perspective* (LAP) that expresses the *relationships* among commitments. Therefore, LAP can potentially model the relationships among collaborative business processes as networks of commitments, after which the cBPs can be computer-automated. A collaborative biz process undergoes several *phases* before it is realized. However, the LAP perspective is unable to propose a comprehensive set of phases for any cBP. However, from the disparate phases proposed by the various LAP implementations, the author defined five essential phases of collaboration which will be discussed in greater detail in Chapter 8.

The next chapter reviews common B2B industry standards.

Figure 3.13: Summary of Chapter 2 and Chapter 3

Chapter 4. Industry B2B Standards

As the focus of this research is on cross-enterprise business processes (cBP's) and as cBP's are multi-disciplinary and practitioner-dominated, there is a need to review the current industry efforts to facilitate B2B collaboration. The concept of cBP is not new in the industry, having been around since the seventies in the form of Electronic Data Interchange (EDI) [127-130]. Many industry standards such as RosettaNet [131-133] and ebXML [134-137] have been proposed since then to capitalize on technological advances in the World Wide Web (WWW), and more recently, the Service Oriented Architecture (SOA) [83, 138-141]. Generally, these standards usually offer a detailed logical functional (recall Section 2.2.2) framework for cross-enterprise data exchange, in uniform message formats encapsulating these data.

The underlying aims of these industry standards are uniform: to facilitate efficient information exchange and to reduce supply chain lead-time, cost and errors encountered in paper-based transactions. This chapter briefly reviews the components and limitations of prominent **industry B2B standards** arising from which the author proposes an ontology as discussed in Chapter 9.

4.1 RosettaNet

RosettaNet [131-133] was launched in June 1998 by multi-national companies from the USA, Europe, Japan, Taiwan, China, Singapore, Malaysia, Thailand and Australia as an independent, non-profit consortium [131] to standardise supply chain interactions by creating interoperable cBP's within the electronics industry: *information technology, electronic components, and semiconductor manufacturing* [134]. RosettaNet has since grown beyond the electronics industry, with member companies representing USD 1.2 trillion dollars in annual revenue, and transacting billions of dollars within their trading networks using Partner Interface Process (PIP) specifications [131], which are system-to-system, XML-based dialogues. PIPs represent operational-level collaborative business processes like PIP3A1 (Request Quote) shown in Figure 4.1.

Figure 4.1: An example of a Cluster-Segment-PIP-Action breakdown in RosettaNet for the “3A1-Request Quote” PIP.

4.1.1 Scope of RosettaNet

In RosettaNet, cBP's are standardised by PIPs for many common B2B activities. Just like the example in Figure 4.1., each PIP defines how specific processes, running in two different partners' organizations, are standardized and interfaced across the entire supply chain.

PIPs include all business logic, message flow, and message contents to align the two business processes [132]. The entire scope of RosettaNet processes is divided into several **clusters** (highest level, depicting the business function) and **segments** (or groups of core business processes) containing all supply chain processes. There are seven clusters [132]:

1. *Partner Product and Service Review*
2. *Product Information*
3. **Order Management**
4. **Inventory Management**
5. *Marketing Information Management*
6. *Service and Support*
7. *Manufacturing*

A specific business process may be created from a single PIP or combination of PIPs contained within the segments of every cluster. This thesis focuses on Cluster 3 and 4, as these processes are the most commonly transacted business processes in supply chains [142, 143].

4.1.1.1 The RosettaNet Framework and its Components

RosettaNet distinguishes between internal (i.e. private) and external (i.e. public) business processes (see Figure 4.2). RosettaNet embodies a wide range of public processes through PIP specifications of information content and the sequence of peer-to-peer (P2P) message exchange between trading partners. Either partner can initiate a business process, addressing another partner directly.

Because supply chain partners establish a set of business interactions through PIPs, the semantics of the PIPs need to be accepted by all parties. This also means that all collaborators have to be treated as peers on an equal basis (a view supported by this thesis). The semantics (i.e. meaning) of business terms (e.g. billing statements, request for quotations, etc.) are defined within a **RosettaNet Business Dictionary (RNBD)**. Technical terms are defined in the **RosettaNet Technical Dictionary (RNTD)**. The **RosettaNet Implementation Framework (RNIF)** support PIP execution by (1) specifying *message-exchange protocols* for disparate software systems to execute RosettaNet PIPs cooperatively and (2) describing *the components of a RosettaNet business message*, the format of these messages, message bindings (i.e. add-ons) for proper message transport, security and message flows. As shown in Figure 4.2, the

combination of RNIF, RNTD and PIPs enable message creation and message processing across 2 trading partners. Figure 4.2 also shows how the RosettaNet Framework cleanly separates business processes and their “semantics” (via PIPs) from the functional (execution level) implementation of the business processes.

Figure 4.2: Overview of the RNIF and role of each RosettaNet component (Src: <http://www.ibm.com/developerworks/webservices/library/ws-rose2/index.html>)

4.1.2 Challenges in RosettaNet Adoption

While RosettaNet is indeed a very powerful B2B information exchange standard, it has its fair share of challenges [134]:

1. New versions are not backward compatible

RNIF version 2.0 is not compatible with version 1.1 as the newer version is significantly different. This creates a barrier for member companies to upgrade, as the change to a new version will mean a complete overhaul of IT systems.

2. Expensive to adopt

Small and medium-sized supply chain partners are not able to afford the US\$150,000 to US\$300,000 for each RosettaNet solution. This is a catch twenty-two situation, as without widespread adoption, the prices will remain high. However, to overcome this problem, RosettaNet has introduced alternative pricing structures; the **RosettaNet Basics Express Program (RNB)**, is a light-weight version which a small sub-set of PIPs as follows: *PIP 3A4 - Request Purchase Order*, *PIP 3A7 - Notify of Purchase Order Acknowledgement*,

PIP 3A8 - Request Purchase Order Change, PIP 3A9 - Request Purchase Order Cancellation, PIP 3B2 - Advanced Shipment Notification

3. ***Processes cannot be dynamically formulated***

The implementation and customisation of software that are needed to link companies to RosettaNet requires both high costs (e.g. consulting fees, licences, etc) and long setup times. This is not realistic in the current supply chain, and there is no technique that can enable its PIPs to link up dynamically based on the collaborators' needs. This thesis attempts to address this gap.

4. ***Lack of accompanying support tools***

There are not many compliance testing mechanisms for the software products, and affordable diagnostic tools that verify that the partner networks are "RosettaNet-ready". While it is a concern, this will not be included in the scope of the thesis as it is not crucial for solving dynamic cBP formulation.

4.1.3 ***Application to the Author's Research***

The author's research stands to benefit from the afore-mentioned overview of RosettaNet in the following ways:

1. ***Reference and B2B knowledge repository.*** The vast number of PIPs and segments in RosettaNet is a rich source of B2B business processes widely adopted by industry. The PIPs are a ready pool of operational level cBP tasks from which to build the proposed methodology in Chapter 9 and Chapter 10. A list of the RosettaNet tasks used for this research can be found in Appendix E.
2. ***Verification of results and cBP models.*** RN's taxonomy of business processes (i.e. PIPs) can be used to verify the dynamically generated cBP from the author's methodologies.

4.2 **Electronic Business using eXtensible Markup Language (ebXML)**

The *Electronic Business using eXtensible Markup Language (ebXML)* [136, 144] is another high-impact B2B industry standard. It was formalised in 2001 as a joint initiative between the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) and the Organization for the Advancement of Structured Information Standards (OASIS). Presently, it is a complete set of International Organisation for Standardisation (ISO) standards maintained by its two contributing organisations as follows:

- **ISO 15000-1:** ebXML Collaborative Partner Profile Agreement
- **ISO 15000-2:** ebXML Messaging Service Specification
- **ISO 15000-3:** ebXML Registry Information Model
- **ISO 15000-4:** ebXML Registry Services Specification

- **ISO 15000-5:** ebXML Core Components Technical Specification, Version 2.01.

ebXML has a large number of specifications and can be confusing for beginners [145]. Put simply, it is an electronic business framework which applies to business transacted in a global marketplace via the exchange of XML-based messages.

4.2.1 Differences of ebXML from RosettaNet

The prospect of expanding e-business to small businesses was the motivation behind ebXML . In fact, ebXML's stated objective was to make it possible for any business of any size in any industry to do business with any other business anywhere in the world. The initial hope was that the presence of an accepted international e-business standard would motivate small business software developers to support ebXML. Hence, compared to RosettaNet, ebXML:

1. are **general standards** which are not specific to any business (a.k.a. horizontal standards) while RosettaNet comprises **specific standards**, thereby making a thorough coverage (a.k.a. vertical standards).
2. can be adopted at very **low cost** as compared to RosettaNet.

4.2.2 Key Components of ebXML

Before trying to understand how ebXML works, it is useful to understand the functions of some of the key components of ebXML [136]:

- **Registry:** A central server that stores a variety of data necessary to make ebXML work. Amongst the information a Registry makes available in XML form are: *Business Process & Information Meta Models, Core Library, Collaboration Protocol Profiles, and Business Library*. Basically, when a business wants to start an ebXML relationship with another business, it queries a *Registry* in order to locate a suitable partner and to find information about requirements for dealing with that partner.
- **Business Processes:** Activities that a business can engage in. A Business Process is formally described by the **Business Process Specification Schema (BPSS)**. BPSS provides a schema for partially executable BP's defined through business-level constructs. BPSS defines cBP's in terms of a sequence of message exchanges (i.e. BPSS business transactions) and defined message contents. It provides a slightly higher-level modelling abstraction for BP's compared to BPM standards [13] such as BPEL and BPML, and can be mapped onto these standards to offer a seamless path to execution [146].
- **Collaboration Protocol Profile (CPP):** A profile filed with a Registry by a business wishing to engage in ebXML transactions. The CPP will specify some *Business Processes* of the business, as well as some *Business Service Interfaces* it supports.

- **Business Service Interface:** This facilitates transactions necessary in *Business Processes*. The *Business Service Interface* also includes the kinds of *Business Messages* the business supports and the protocols over which these messages might travel.
- **Business Messages:** The actual information communicated as part of a business transaction. A message contains multiple layers. At the outside layer, an actual communication protocol must be used (such as HTTP or SMTP). Other layers may deal with encryption or authentication.
- **Core Library:** A set of standard, commonly used "parts" that may be used in larger ebXML elements. For example, *Core Processes* may be referenced by Business Processes. The *Core Library* is contributed by the ebXML initiative itself, while larger elements may be contributed by specific industries or businesses.
- **Collaboration Protocol Agreement (CPA):** In essence, a contract between two or more businesses that can be derived automatically from the CPPs of the respective companies. If a CPP from company 'A' says "I can do X," a CPA between company A and its contract manufacturer 'B' says "We will do X together."
- **Simple Object Access Protocol (SOAP):** A W3C protocol for exchange of information in a distributed environment endorsed by the ebXML initiative. Of interest to ebXML is SOAP's function as an envelope that defines a framework for describing what is in a message and how to process it.

4.2.3 How Companies Collaborate in ebXML

Figure 4.3 illustrates a typical ebXML-based collaboration between two companies, 'A' and 'B' [145]:

Figure 4.3: ebXML Technical Architecture specification. Src: [145]

Step 1: Company A will first review the contents of an ebXML Registry, especially the Core Library. The Core Library (and maybe other registered Business Processes) will allow Company A to determine the requirements for its own implementation of ebXML (and whether ebXML is appropriate for their business needs).

Step 2: After analysing the information accessed from an ebXML Registry, Company A can choose to build or buy an ebXML implementation suitable for its anticipated ebXML transactions. The hope of the ebXML initiative was that ebXML-compliant software vendors will support all the elements of ebXML. An "ebXML system" can range from a pre-packaged software system to one as comprehensive as a commercial database system. Figure 4.3 suggests that Company A builds its own ebXML compliant software system while the Company B uses something like a pre-packaged ebXML compliant system.

Step 3: The next step is for Company A to create and register a CPP with the ebXML Registry. Company A might wish to contribute new Business Processes to the Registry, or simply reference existing ones. The CPP will contain the information necessary for a potential partner to determine the business roles of interest to Company A, as well as its associated protocols.

Step 4: Once Company A is registered, Company B can look at Company A's CPP (analogous to a selection from a catalogue) to determine that it is compatible with Company B's CPP and requirements. At that point, Company B should be able to negotiate a CPA automatically with Company A, based on (1) the conformance of the CPPs and (2) agreement protocols, given as ebXML standards or recommendations.

Step 5: Finally, the two companies begin actual business transactions. These transactions are likely to involve XML-format business messages conforming to ebXML standards and recommendations. During transactions, "real-world" physical activities will also be executed (e.g. the shipment of goods from one place to another, the rendering of services, etc.). ebXML has helped greatly in monitoring, and verifying these real-world activities.

4.2.4 Adoption of ebXML

Due to the widespread establishment of EDI in many private enterprises at the time of release of ebXML in 1999 and the proliferation of web-service related architectures in the 2000's, ebXML adoption was slow [135, 147]. However in recent years, it has gained momentum, especially among government organisations and public agencies such as the **Public Health Information Network (PHIN)** of the U.S. Center for Disease Control and Prevention (CDC) for the exchange of clinical and business messages related to epidemiology and public safety, **Hong Kong University's Centre for E-Commerce Infrastructure Development (CECID)** and **Port of Hong Kong** to exchange information on dangerous goods manifests at sea ports, and

The U.S. Department of Defense (DoD) to provide XML components approved for DoD software development, as well as reference data tables and codes, such as country and state codes.

The reason for the wide adoption by these public agencies is mainly due to the “open” model of ebXML, which is different from the so-called classic “hub-and-spoke model” employed by EDI, in which large, powerful customers (“hubs”) impose EDI specifications on their suppliers.

4.2.5 Limitations of ebXML

1. **High customisation leads to high costs and inefficiency-** To accommodate their EDI-ready big customers, suppliers installed EDI-compliant systems, often with different variations for each trading partner [147]. This is costly in terms of customization for each collaboration and is not efficient.
2. **Hub-and-spoke model leads to inequality -** With ebXML, many of the “hubs” are public agencies, regulated public utilities, or organizations subject in some way to public accountability. Therefore the “hubs” cannot impose harsh marketplace monopolistic conditions on the “spokes” as with past EDI systems. Rather than the classic hub-and-spoke, a popular ebXML arrangement is more of a consensual hub-and-spoke [147].
3. **Unable to formulate business processes on the fly -** ebXML allows two companies which have already customised their infrastructure to formalize collaborations, but is unable to allow new companies formalize collaborative business processes on the fly. Existing ebXML users, having customised their processes for specific business scenarios, cannot adapt to changing business goals. This research attempts to address these limitations found of current B2B standards.

4.2.6 Application to the implementation aspects of the author's research

ebXML has an *ebXML Common Business Processes Catalogue*, which is a comprehensive reference list of low-level operational tasks commonly found in B2B collaborations. This list is created from a survey of thousands of corporate and governmental institutions from many sectors (e.g. manufacturing, aerospace, pharmaceutical, defence, etc) which are members of the ebXML consortium. In the author's research, this catalogue will be synthesised with tasks listed in RosettaNet Cluster 3 and 4's PIPs and OAGIS reference list of BODs (see next section) to form a comprehensive pool of B2B tasks for the construction of an ontology to facilitate dynamic cBP formulation in Chapter 9. A list of the Common Business Process Catalogue tasks used for this research can be found in Appendix E.

4.3 Open Application Group Integration Specification (OAGIS)

First released in 1995, the Open Applications Group Integration Specification (OAGIS) [148, 149] was created by the Open Applications Group (OAGi) [150] to achieve interoperability between collaborators across similar and disparate supply chains. OAGi members include companies such as Lucent, Lockheed Martin, Sun Microsystems, Oracle, IBM, Boeing, IBM, Agilent, etc. OAGIS integrates collaborators through business messages in the form of **Business Object Documents (BODs)**, which are exchanged between software components between companies across supply chains and between supply chains.

To allow easy adoption by companies, OAGIS has **Business Scenarios** which identify the business applications, components being integrated and the BODs used [150]. As of the time of writing, there are 70 business scenarios for integration and a repository of more than 400 BODs.

4.3.1 Key Components of OAGIS

4.3.1.1 Business process (Scenarios)

Figure 4.4: A scenario describing a supply chain integration between the Manufacturing, Purchasing, Order Management, Billing, Shipping, and Financials departments. Src: <http://www.ibm.com/developerworks/xml/library/x-oagis/>

To encourage adoption OAGIS BODs (version 9.0), the specification introduces a set of commonly encountered business scenarios. Scenarios identify the business applications and components that are being integrated along with the BODs employed. Figure 4.4 illustrates one

such scenario. The *ProcessPurchaseOrder* BOD starts this scenario by communicating a customer's demand for an order. Each arc represents the applicable BOD in each stage of the scenario, and the "noun" relating to that BOD (e.g. the roles such as Order Management, Billing, etc).

Business scenarios capture the intended sequence of the messages, the dependencies, the scope, and the error handling that are addressed. OAGIS currently has 70 business process scenarios, and companies wanting to adopt OAGIS into their cBPs will have to find the solution that most closely matches their needs.

4.3.1.2 Business Object Documents (BOD)

OAGIS' object-oriented BOD messages define a common consistent Noun (or object) that has Verbs (or methods) that indicate the action to be performed by the Noun. By using this approach, OAGIS messages are very re-useable. Once the initial OAGIS BOD is read or produced, much of the code can be adapted to read or produce the next message. Figure 4.5 shows the components of a BOD, and Figure.4.6 shows a simple example of a *ProcessPurchaseOrder* BOD.

Figure 4.5: General structure of all BODs

Figure.4.6: An example of Structure of the ProcessPurchaseOrder BOD

BOD messages are communicated independent of the underlying communication mechanism. It can be used with simple Internet protocols such as the HTTP and the SMTP and can also be used in complex protocols such as SOAP, ebXML Transport and Routing, or any other B2B integration systems.

4.3.2 Limitations of OAGIS

While OAGIS' independence from the mechanism of communications may seem to encourage countless permutations of adoption, the consultant implementing OAGIS has to make sure that the communications will not be corrupted by translation problems and incompatibilities. Also,

it is not realistic to expect a consultant to be conversant with all 700 granular activities embodied in BOD. While the 60 scenarios may help to scope down to solutions that closely resemble a real-life problem, it can also have the detrimental effect of force-fitting business operations so that the OAGIS BODs can be accommodated. This is not an ideal situation as in reality, collaborating companies would prefer systems that fit their *as-is* processes.

4.3.3 OAGIS' Role in This Research

As mentioned earlier in Section 3, the 700 OAGIS BODs will be synthesised with task lists from RosettaNet Cluster 3 and 4's PIPs and ebXML Common Business Process Catalogue to form a comprehensive pool of B2B tasks, for the construction of an ontology which enables dynamic cBP formulation in Chapter 9 and Chapter 10. A list of the OAGIS tasks used for this research can be found in Appendix E.

4.4 Limitations of Current B2B Standards

All these standards force fit organizations into their specifications regardless of the purpose of the usage. As stated in a 2007 research report by Forrester Research [4], new B2B standards have to be drawn whenever changes are encountered. These do not facilitate dynamic business collaboration commonly found in today's fast paced world. Alternatives like service-oriented architecture (SOA) and Web services together with proper B2B process formulation could possibly overcome the issues and challenges of dynamic business partnerships[151, 152].

In the preceding sections, the current inability to bridge high-level strategic business goals to the low-level operational tasks is discussed. It is easy to find many web service literatures describing the modelling and execution of the flow of business processes at the operational level. However, few actually bridge these low level tasks to high level strategic tasks.

The inability to dynamically decompose compound business process tasks into primitive operational tasks for direct web service execution has also led the author to propose a method to address to these issues and they will be described in the later chapters of this report.

4.5 Legacy IT-Based Industry B2B Practices

4.5.1 Electronic Data Interchange (EDI)

The Electronic Data Interchange (EDI) was born out of the need for communications between different proprietary formats of collaborating partners. It has been the most widely used electronic business format since the 1970s ; some 80% of B2B transactions and 200,000 to 300,000 organisations adopt EDI in the world [134]. The main adopters of EDI are large corporations which had previously spent large sums of money to implement their EDI infrastructure before the turn of the millennium. However, compared to the Web-based

RosettaNet and ebXML, EDI is now a “legacy system”. There are two main forms of EDI; the American National Standards Institute *X12* standards [127] and the European *UN/EDIFACT* standards [130]. Two other specific standards that target the UK retail industry and the European automotive industry are the TRADACOMS standard developed by the ANA (Article Numbering Association) and the ODETTE standard. In 1987, the ISO adopted the EDIFACT standard. Both X12 and EDIFACT provide similar capabilities but are quite different in their structure and syntax, hence interoperability between these two standards is challenging.

Simply put, EDI serves to facilitate document exchange between companies [134]. As EDI is not the main focus of our discussion, this report will not expand on its infrastructure and data formats. More details and studies about EDI and its impact in industry can be found in [128, 129, 153, 154].

4.5.2 Vendor Managed Inventory (VMI) Systems

Widely adopted in the supply chain industry, **Vendor Managed Inventory (VMI)** [155, 156] is a collaborative business model in which a supplier meets consumer demand from a pre-agreed level of inventory at the different echelons (e.g. retailers, distributors, wholesalers, etc) in the supply chain of the product.

By getting consumer demand trends directly from the retailers and controlling the inventory levels at the different echelons, VMI makes it less likely for a stock-out to happen [157] and reduces overall inventory costs in the supply chain. VMI helps foster a closer understanding and responsiveness between the supply chain echelons through software and statistics to forecast and maintain the appropriate inventory in the supply chain.

4.5.3 The ICAM Definition Languages (IDEF) Standards Family

ICAM Definition Languages (IDEF) standards [158-160] are Federal Information-Processing Standards originally initiated by the U.S. Air Force Integrated Computer-Aided Manufacturing (ICAM) Architecture in the 1970s, developed in the 1980s, and later released into the public domain in the 1990s. *IDEF0 Function Modelling Method* (a.k.a. FIPS 183) [160] is a process-mapping standard consisting of a high-level map of an organization’s major business processes and a second level consisting of processes functionally **decomposed** into ever-finer sequences of activities (See Figure 4.7). It has been widely adopted in BPR activity models. Particularly related to the author’s research is the *IDEF3 Process Description Capture Method* [159], which provides a methodology for discovering, collecting, and documenting high-level, non-executable business processes. It has influenced many contemporary BPM Graphical Standards. One example is the modelling of 'swimlanes', which are diagrammatical rows (or alternatively, columns) often used to depict the responsibility of

roles. This concept is later applied in Role Activity Diagrams and BPMN (recall Chapter 3). IDEF3 provides an alternative to vendor-specific enterprise frameworks such as ARIS, Catalyst, and Zachmann.

Figure 4.7: An IDEF0 Decomposition Diagram

Another *IDEF standard* related to the author's research is the *IDEF5 Ontology Description Capture Method*, which is a set of schematics used to model ontologies. An ontology is a domain vocabulary complete with precise definitions of terms for consistent interpretation of the data used. The general ontology construction steps in IDEF5 include: (1) *Cataloguing the terms*; (2) *Capturing the constraints that govern how those terms can be used to make descriptive statements about the domain*; and (3) *Building the model*. While the IDEF5 standard cannot facilitate dynamic cBP formulation, the author has also adopted these guidelines in the building of his proposed ontology for dynamic cBP formulation in Chapters 8 and 9. More details of the IDEF 5 Method and its Languages can be found in its detailed 187-page report [158].

4.6 Chapter Summary

This chapter focused on dominant B2B information exchange standards: RosettaNet, ebXML and OAGIS. For this research, a comprehensive pool of essential B2B tasks was synthesised from industry reference models from OAGIS BODs, RosettaNet Cluster 3 and 4's PIPs and ebXML Common Business Process Catalogue. From the synthesised list of essential B2B tasks, an ontology that facilitates the formulation of dynamic cBPs to describe their relationships.

These B2B industry standards are such that they require business processes to be manually coded. More importantly, these standards, being designed for established business processes, are unable to cope with the frequent changes commonly encountered in B2B collaboration. They are really quite rigid and inflexible. Because these current B2B standards are unable to nimbly adapt to changing business goals, recent cBP research have attempted to address this short-coming. This will be discussed in greater detail in the next chapter.

Chapter 5. Cross-Enterprise Business Process (cBP) Research

Having discussed the industry standardisation efforts and practitioners' approach to managing cBP's in Chapter 4, we will now turn our attention to interdisciplinary cBP methodologies proposed by researchers. The rapid emergence of research in Service-Oriented Computing [138, 139, 161-163], Semantic Web [164-167], the Pragmatic Web [93, 168-170], dynamic formulations of cBP [171-173] and the *ad-hoc* matchmaking of cBP [44, 45, 174-177] underscore the fact that most current B2B information exchange systems deal with only the *syntactic* aspect of understanding and the *static* modelling of processes. In this chapter, we shall discuss and identify gaps in current and relevant cBP research attempting to address the semantic modelling gap and the dynamic formulation and composition of cBP.

5.1 cBP Research: Web Services and Service-Oriented Computing

5.1.1 Web Services

The World Wide Web Consortium (W3C) defines a Web service as “a software system designed to support interoperable machine-to-machine interaction over a network”. Compared to classical *static* HTML pages, Web services are programmed to be *dynamic*, therefore addressing the syntactic level of interoperability. In the author's own words, Web services are atomic programs identified by a Uniform Resource Identifier (URI), hosted on some server, performs a specific task or granular function, take in some inputs and produces an output (usually in XML), and is used (and reused) over the WWW by other Web applications which need that particular function. Web Services communicate with each other via Internet standards and can be combined to satisfy more complex goals or behaviour.

A simple example of Web Services in action is the iGoogle [178] page (<http://www.google.com/ig>) hosted by Google Inc. On iGoogle, a user can decide which features to add onto his 'customised' home page. Common features include *Calendar*, *Weather Forecasts*, *YouTube*, *Currency Converters*, *Gmail*, *News*, etc. Each of these features is actually a Web Service hosted on Google's and other companies' servers, and these services are **combined** quickly into a customised user home page.

Another Web service example is shown in Figure 5.1, which demonstrates how the SAP Enterprise Portal [179] can help, say a sales executive, customise his portal to display sales trends, reports and product tracking services *via iViews*. iViews are self-contained Web

documents or applications that are provided via a Uniform Resource Locator (URL) managed by the SAP portal framework. In the author's industry experience, the SAP Enterprise Portal can be easily customised by both IT specialists and end-users to combine information from other systems such as *Enterprise Resource Planning (ERP)* systems, *Human Resource Management (HRM)* systems, *SAP script pages (i.e. BSP)*, customised SAP programs (e.g. *Web Dynpro*) and even external Web applications hosted across the Web (see Figure 5.1).

Figure 5.1: Screen capture of the SAP Enterprise Portal, showing Web Service deployments via iViews

The above two examples demonstrate the ease by which the service-oriented approach can create Web applications. In contrast to the traditional centralised way of computing (e.g. client-server, mainframes, etc) where computing users access a central server for all program executions, the distributed and modular nature of Web services increases the likelihood, and need for collaboration across organisations. The Web Service way of programming spawned an increased emphasis on decentralised computing and architectures, and the most notable of all, the service-oriented architecture (SOA).

5.1.2 Service Oriented Architecture (SOA)

The **Service Oriented Architecture (SOA)** [138, 139, 161-163] is a framework that uses Web Services as building blocks for cross-enterprise and cross-application integration. To support the SOA, the service-oriented computing (SOC) paradigm utilizes (Web) services as fundamental building blocks for developing applications [140]. As shown in Figure 5.2, SOA

involves a clear definition of service *layers*, *functionality*, and *roles* [140]. In Figure 5.2, the lowest layer “basic services” describes low-level Web Services and their basic operations (i.e. publication, discovery, selection, and binding). The higher layers in the SOA pyramid provide additional support required for *service composition* and *service management*. These layers have given rise to niche research areas in web service composition [180-183] and management (especially in business process management).

Figure 5.2: The Layers within Service Oriented Architecture [140]

5.1.3 Web Service Composition Challenges

Before Web Services can gain more widespread use as dynamic building blocks purported by the SOA, we need to first overcome the complex challenges of dynamically combining disparate and behaviorally-different services. Such challenges are further compounded by the following [180-184]:

1. Limitations of naming (e.g. multiple Web Services with the same name) and differences in naming conventions by different collaborating parties,
2. Mismatches in input and output types, causing a need for mediation when joining the *output* of a service as the *input* of another service (or vice versa), and increased difficulty in chaining Web services together.
3. Difficulties in interpreting the context and definitions of Web service behaviour (e.g. the same description can mean different context, hence the current need for additional semantics)

4. Organizations concerned with the risks of exposing their web services for others to use, which may result in competitors spying on or inferring their business practices from the services exposed.

Despite the large number of publications attempting to address these problems, Web Service Composition research is still very much work in progress [185-187] and there is a sentiment that this area of research is reaching a plateau in progress [188] unless there is a relook at the current approaches.

5.1.3.1 Possible reasons for non-progress in dynamic Web Service Composition

In the author's opinion and observation, the reason for the plateau in service composition research is due to four main reasons:

1. Current approaches to composition research usually assume that the Web services can be easily discovered; this is not realistic in a competitive business environment.
2. The lack of real industry-wide implementation of dynamic web service composition. Simplistic examples of conference travel planning workflows and insurance claim workflows are commonly found in service oriented research publications but it is rare to find a realistically scalable industry-wide example.
3. Most research in Web Service composition and orchestration focus on reducing execution-time hiccups such as deadlocks, synchronisation and concurrency errors but overlooked the facilitation of high-level business goals and requirements, hence not addressing actual business needs. This problem is also recently highlighted by Stanford's Petrie in [188].
4. The current assumption that web services can intelligently combine themselves into more complex programs if they are guided by some artificial intelligence or are modeled as agents is unrealistic. This is analogous to an assumption that one day, the raw material parts, screws, nuts, and bolts will be intelligent enough to combine dynamically by themselves to form a piece of furniture. It is obvious that intelligent compositions are not realistic in the near future, as it means that the parts already have some idea of what they can possibly merge to become (i.e. trillions of possibilities), and how their shapes and sizes can combine to fit properly. This preposterous approach, or the author-coined "*bottom-up approach*" (see next section), is also the main stumbling block of Web Service composition researchers. If we observe the normal way of fulfilling a goal, e.g. making furniture, the process usually involves two steps: 1) planning and (2) execution. Most of the current research focuses on execution without the awareness of the need for a plan to be generated; dangerously assuming that simplistic underlying services contain such plans.

5.1.3.2 **Bottom-Up vs. Top-Down approach**

The “*bottom-up*” approach discussed in Point 4 in the previous section is a summary of the most common approach taken by Web service researchers [188]. This approach is contingent upon the assumptions that (1) Web services are easily discoverable from the entire WWW, (2) they can be combined seamlessly after being discovered, and that (3) the Web services can be combined as high level business processes towards achieving a high-level goal.

The assumption of easily discoverable Web services is not realistic in a competitive business environment, as there are many companies which do not wish to reveal its basic operations or open port accesses to their servers. Also, the assumption that the entire WWW is the sourcing ground for the required Web services is not a realistic and scalable one, as such an assumption will exhaust much computational time and power. Furthermore, as humans often describe similar things differently (i.e. different context and naming conventions), duplicate and overlapping descriptions of similar-intent web services can be expected.

The author feels that while SOA promises reusability and scalability, the advantages of reusability and flexibility over that of computational costs and search time over the Internet just to get a program dynamically generated must first be assessed. Currently, SOA works only in the simplistic and manually-configured applications such as the iGoogle and SAP Enterprise portal examples, but it is obvious with the above arguments, true dynamic Web service composition will take some time to come.

While the components used and levels of abstraction of both *bottom-up* and *top-down* approaches are mainly the same, the assumptions of the *bottom-up* approach greatly contrast those of the *top-down* approach.

The *top-down* approach assumes (1) that the applications and Web services need a *master-plan* derived from a goal, and (2) that complex high-level business goals can be decomposed into the right granularity (for web services to directly implement each task), and (3) that decomposed tasks will be able to sequence themselves in the appropriate control flows (e.g. splits, sequence, concurrency). Some earlier literature of the *top-down* approach [5, 163] tried to discover the optimum number of levels to decompose goals into, but recent approaches [172, 173, 187] have dismissed such intentions as unrealistic. There is simply no optimum number of levels to decompose to.

On top of alleviating the need to decompose a goal, a master-plan (i.e. a more detailed blueprint of requirements) from the top-down approach can allow more pointed and specific searches for the required Web services. The so-called master-plan significantly reduces search

space and search time for matching Web services, and promises to be more realistic for the achievement of dynamic Web service composition in the face of an increasingly complex Web.

5.1.4 Why Top-Down Approach was Adopted in This Research

The author's research addresses the challenges of decomposition and sequencing of tasks by **adopting the top-down approach** (See Chapter 8, 9, and 10) for achieving dynamic business process formulation. The approach was also chosen because it is:

1. More humanly-intuitive way of doing things: *to fulfil goals*. This also potentially means that it can address business goals more directly.
2. More realistic in terms of computational requirements and time. The reduction of the search space for matching Web services in the top-down approach will greatly increase the effectiveness of service-oriented approaches.
3. Able to complement current technologies. The creation of master-plans for each goal in a top-down manner can potentially fill the high-level void left unexplored by current Web service orchestration (i.e. the definition of the order and control flow of execution) approaches. More details can be seen in Chapter 11.

With the *top-down* approach, web service architectures can address the needs of their business users without the need to alter the current way of running their businesses. This is contrary to many modern day information systems, which expect end-users to conform to their architectures and graphical user interfaces and learn how to run their business processes in the way the systems are designed to run. Users no longer need to be frustrated by their constant need to learn and conform to the functionalities of new computer systems, but rather, have computer systems to conform to their goals, which is one of the grand aims of semantic Web.

5.2 cBP Research: Semantic Web

The next most prominent feature of cBP research is undoubtedly the **Semantic Web**. According to the seminal paper of Berners-Lee, Hendler and Lassila in 2001 [164], the Semantic Web “is not a distinct Web but **an extension of the current World Wide Web**, in which information is given **well-defined meaning**, better enabling computers and people to work in cooperation”. Currently, the essential property of the Web is its universality (i.e. ability to link objects via Universal Resource Identifiers (URI)). The ‘power’ of such hypertext links (or hyperlinks) is that anything can link to anything else. However, Web technology also does not discriminate between the scribbled draft and the polished document, or amongst cultures, languages, media, etc. This freedom of expression has since created problems like information overload and poor content aggregation [189].

To comprehend the reasons for the problems of the current Web, we revisit the basic purposes of *information* [164]:

1. For human consumption – e.g. videos, poetry, wiki pages, etc.
2. For machine consumption – e.g. databases, programs, sensors of outputs, etc.

The initial Web pages were designed for *human consumption*. Humans can read Web pages and understand them, but their inherent meaning is not stored in a way that enables computers to interpret them [164]. Many foresee that over time the information on the Web will be used not only for display purposes, but also for (1) interoperability and (2) integration between systems and applications.

One of the ways proposed to achieve these is the **Semantic Web**, which aims to enhance machine-to-machine exchange and automated processing via information that computers can “understand” [167, 190]. Tim Berners-Lee has a 2-part vision for the future of the Semantic Web [189]:

1. to make the Web a *more collaborative medium* (in line with the aims of this research)
2. To make the Web *understandable, and thus, more processible by machines*.

5.2.1 State of the Art in the Semantic Web

As a result, Semantic Web research has witnessed a proliferation of efforts attempting to solve two main areas:

5.2.1.1 Syntactic Representation - Flexible Storage of Data

The first area is *standardising flexible data storage*. This has already been addressed by the flexible *eXtensible Markup Language (XML)* [191], which lets everyone create their own tags in a structured way. XML enables data to be described from the user’s perspective, but in a standardised structured manner that allows it to be parsed efficiently.

XML has since become the *de facto* way to describe data transferred over the Web and is the underlying language for the creation of semantic languages such as the RDF (Resource Description Framework) [192], OWL (Web Ontology Language) [193] and WSMO (Web Service Modelling Ontology) [194, 195].

5.2.1.2 Semantic Representation - Description of Meaning in the Data

The second area is the *Semantics* (i.e. description of meaning) part of the Semantic Web. The foundational layer of the semantic description is predominantly addressed by the **Resource Description Framework (RDF)** [192] standard. RDF provides a clear set of rules to describe information and their relationships on the WWW. In RDF, each object has some properties (e.g. “field”, “is the father of”) with certain values. The combination of the description of such properties creates a natural way to describe the relationships between objects (a.k.a. concepts).

For example in Figure 5.3, we can see that *Database A* has some *fields*, and its *field number 5* is of the type “*postal code*”. We are able to succinctly model this relationship between the (1) *Database* concept, (2) its attribute *field number 5*’s *type*, and (3) another concept *postal code* in a meaningful way. This method of describing the relationship between one object and another is also known as an **RDF Triple**.

Figure 5.3: Example showing the logical components of an RDF triple

Each concept in a triple can be described flexibly by “Universal Resource Identifier” (URI) (i.e. building onto the “hyper-link” concepts in the current WWW). With Subjects, Objects and Verbs identified by URIs, the computer will be able to interpret the attributes of concepts and possibly **infer and derive** the indirect relationships between concepts.

For example, “(The individual which is a human with the name Richard) (is the parent of) (the individual which is a human of the name Mary)”, combined with “(The individual which is a human with the name Richard) (is a sibling of) (the individual which is a human of the name Peter)” will allow the **computer to infer** that “Peter” and “Mary” are relatives, and more specifically “Peter is the uncle of Mary”.

Multiple RDF descriptions and triples combine to form **ontologies**. In philosophy, ontology is a theory about the nature of existence of things. However, artificial intelligence and Web researchers have co-opted the term to mean a file or document that formally defines the relationships among terms [164]. The ontology of the Web has a taxonomy and a set of inference rules. Currently, the dominant representation of ontology is the **Web Ontology Language (OWL)** [193]. OWL provides a language for defining structured Web-based ontologies which allow a richer integration and interoperability of data among communities and domains.

Given these descriptive benefits, OWL is adopted by the author in his modelling of the complex hierarchical task network of B2B tasks, and their relationships to business process concepts such as roles, control flows, preconditions, etc. As such, the author will describe ontologies in detail in the next chapter in Section 6.2.5.

When the above mentioned semantic and syntactic challenges of the Semantic Web are addressed, we can expect a collaborative medium which passes understandable data. However, RDF has not (yet) been widely adopted in prominent Web technologies, and OWL is still work

in progress, albeit it is widely adopted by those researching into ontology and life sciences. Hence, the potential of Semantic Web research is still very promising.

5.2.2 Related Research: Semantic Web Services

An offshoot of Semantic Web research is Semantic Web Service, which aims to apply Web Services concepts to support the Semantic Web. Web Services communicate and collaborate by inputting and outputting XML files. Hence, if these XML documents can be augmented with semantics, the Semantic Web is realised. In simple words, we embed semantics into Web services and use them to pass semantic messages.

By so doing, the grand aim of Semantic Web Services (SWS) will be to facilitate the automation of service interactions such as: *discovery*, *composition*, *selection*, *invocation*, as well as *enactment and monitoring* in an open environment. Semantic Web Services can then be linked together to form Semantic Business Processes to cater to different business needs. Currently, there are several initiatives aiming at creating SWS, a few prominent ones being:

1. Sheth and Cardoso – dealing with SWS Quality-of-Service (QOS) [162, 196, 197] and SWS Composition [167, 190, 198]
2. Bussler – Dealing with B2B collaborations using SWS [165, 199] and via B2B Protocols [148, 200]
3. Hepp - Proposed a hybridisation of SWS and BPM to form the Semantic BPM. This is also adopted in the SUPER Integrated Project to be discussed in the later section. [201-203]

Figure 5.4: Comparing Web Paradigms

Figure 5.4 puts into perspective the relationship of Semantic Web Services to Web Services, to the WWW, and also the Semantic Web. As can be seen, Semantic Web Services has Interoperable Semantics as envisioned by the Semantic Web and at the same time, has Dynamic Resources like that of the distributed Web services. As the Semantic Web Services is

a developing technology, and therefore not a sound basis for the study of Dynamic Business Collaborations (the author's research), this report will not further investigate into SWS.

5.3 cBP Research: Pragmatic Web

5.3.1 Limitations of the Semantic Web

The term "The Pragmatic Web" was coined by a prominent Web service researcher, Singh [169, 170] in 2002 when he asserted that the challenges of the Semantic Web arise from (1) the difficulty of capturing semantics in a manner that is reusable across applications, (2) the priority of process over data, (3) the importance of interaction, and (4) the critical need for accommodating user context. Singh claims that overcoming these challenges takes researchers from the realm of semantics into the realm of pragmatics (recall that semiotics has three components: *syntax*, *semantics* and *pragmatics*). He also claims that when the vision of the Semantic Web is realized, it will be *via* the pragmatic Web. According to McCool [168], cofounder of the large-scale (semantic Web) RDF project TAP, "*Semantic Web will not be able to achieve the vision of Berners-Lee directly. This is because the Semantic Web is a complex format and requires users to sacrifice expressivity and pay enormous costs in translation and maintenance, it will never achieve widespread public adoption.*" The most problematic assumption is that context-free facts and logical rules would be sufficient [204].

5.3.2 Vision of the Pragmatic Web

The vision of the Pragmatic Web is thus to augment human collaboration effectively by appropriate technologies, such as systems for ontology negotiations, for ontology-based business interactions, and for pragmatic ontology-building efforts in communities of practice [168]. In this view, the Pragmatic Web complements the Semantic Web by improving the quality and legitimacy of collaborative, goal-oriented discourses in communities.

Hence, Pragmatic Web is **not** a buzzword. Its main aim is to complement the Semantic Web but not to replace it wholesale. Naturally, this vision did not resonate with many ardent Semantic Web researchers, but it was nonetheless later (2005) adopted by many LAP (recall Section 3.2.2) researchers like Schoop [168], Weigand [93, 205], Winograd [206], Goldkuhl [207], etc. Although the Pragmatic Web is still not mature, the author believes that it will attract greater attention once the weaknesses of the Semantic Web become apparent. To the author's best knowledge, the components of the Pragmatic Web are still not explicitly defined, but the author foresees them to be:

1. Goal Oriented Collaborations
2. Expression of Commitments
3. Context-awareness for flexible adaptation of ontology changes/ updates

As highlighted in Chapter 3, Points 1 and 2 are very relevant to B2B collaboration research and are already addressed by Language Action Perspective, while Point 3 (Context-awareness) will be described in the following section.

5.4 cBP Research: Context-Awareness

While semantics and ontologies continue to occupy centre stage in current BPM research, there is a school of thought that leans heavily on context awareness in collaborative business processes. The main difference between context and semantics/ontological matching is the argument that ontologies of different companies can never truly be matched.

Saidani and Nurcan [208] introduced a basic taxonomy of context which captures most common context-related knowledge and two structures for modelling and categorizing the context: (1) the context tree (CT) depicting contextual characteristics, and (2) the adapted CT (ACT) applicable to a specific domain. In our opinion, the ACT is not scalable and hence not practical to manage. Furthermore, its basis, the CT, needs to be validated with industry business processes.

Ardissono and Furnari [209] proposed the *Context Aware Workflow Execution* (CAWE) framework for the development of applications composing Web Service suppliers in context-sensitive workflows. With this Framework, context-sensitive workflows can be executed in conventional workflow engines.

Tan *et al* [210] visualized B2B collaborations aided by a *Context Aware Framework* (CAF) that categorizes contextual information into user (i.e. the company), temporal and location. In their view, most B2B transactions are dominated by the exchange of business documents which are rich in contextual information in the form of user, temporal and location.

Not everyone is excited about embellishing BPM with context. According to LAP researcher Weigand [211], “*computing machines, which are purposely designed to process symbols independent of their context, have no hope of becoming experts.*” His concern with the context-less fundamentals of computers is understandable.

In the author’s view, the future of context awareness very much depends on a radically new infrastructure founded on secure theoretical bases, and not just extensions of current architectures. History has shown that technological advances are often marked by a progress to what is convenient and natural to use (especially in computer science).

For instance, programming languages have evolved from structural to object-oriented programming and to the present day Web-based programming (i.e. assembly languages to FORTRAN to C to Java and .NET).

5.5 cBP Research: Dynamic cBP Formulation

This section discusses some of the prominent research projects relating to B2B collaboration.

5.5.1 CrossFlow Project

Started in 1999, the CrossFlow ESPRIT project is made up of a consortium of seven European institutions: *IBM France (coordinator), IBM Corporate Research Division, Zürich, Switzerland, University of Twente, Enschede, The Netherlands, GMD-IPSI Research Center, Darmstadt, Germany, KPN Research, Groningen, The Netherlands, Allianz Church & General (formerly AGFIL), Dublin, Ireland, and the SEMA Group, Madrid, Spain.*

CrossFlow aims to develop information technology for advanced process support in dynamic *virtual organizations* with “contract-based service trading”. As such, CrossFlow can be considered a project investigating the intersection of workflow management and electronic commerce technology (a.k.a. CBP in this report) [171, 212]. The “high-level support” for business processes that CrossFlow offers is obtained by abstracting services and offering advanced cooperation support. *Virtual organizations* are dynamically formed by contract-based matchmaking between service providers and consumers. This understanding will also form part of the basis of the author’s viewpoint of a generic CBP life-cycle to be discussed in Chapter 8. Notably, CrossFlow developments are driven by requirements from real-world scenarios and have had successful implementations in the logistics industry. The concept of Crossflow is summarised as follows [171, 212]:

- ***Service abstraction***

In virtual organizations, a partner does not require full operational details of other partners. Rather, a well-defined abstraction of their operation is usually sufficient to obtain an effective view on both data and processes. As partners in a virtual organization often have different IT platforms, a heterogeneous environment exists. This heterogeneity should be addressed by abstraction of the technical characteristics of partners. For both reasons, CrossFlow defines the interaction between organizations not in terms of their workflow systems, but on an abstract level above these systems. In the author’s observation, this is echoed by the “essential business process” view by Dietz [85] (recall Chapter 3).

- ***Cooperation support***

CrossFlow addresses three areas of advanced co-operation support functionality to complement basic workflow interoperability:

- **Quality of Service** monitoring tracks the progress of outsourced services, both online during service execution and offline to provide aggregate information.
- **Level of Control** underpins high-level cross-organizational transaction management and consumer-controlled process control over outsourced services.
- **Flexible Change Control** allows for dynamic changes to execution paths of services during their execution.
- **Contracts**

The basis for cooperation in virtual organizations is the contract, in which the encapsulated service and cooperation support services can be completely specified. Partially defined contracts are used by service providers to advertise their services and by service consumers to search for services. As such, the contract is the basis for dynamic partnerships. In the author's view, this is an important axiom.

1. *Architecture of CrossFlow*

Figure 5.5: The CrossFlow Architecture

The CrossFlow architecture [213] supports both contract making and contract (service) enactment. The architecture is based on commercial workflow management system technology, shielded from the CrossFlow technology by an interface layer.

2. *Contract making*

When a service provider wants to advertise a service that it can perform on another organization's behalf, it uses its contract manager to send a contract template to a trader. When a service consumer wants to outsource the enactment of a service, it uses a contract template to search for service providers via a trader. When a match between the consumer's requirements and provider's offer is found, an electronic contract can be made by filling in the template.

3. *Contract enactment*

Based on specifications in the contract, a dynamic contract and service enactment architecture is set up. The symmetrical architecture contains proxy gateways that control all communication and support services for advanced cooperation functionality. After contract completion, the dynamically created modules can be disposed of.

Despite the waning interest (perhaps due to the increased interest in the Semantic Web and the SOA), CrossFlow is logical, well-founded and practical in nature. CrossFlow also believes that the abstract service level (high level business processes) must not be dictated by technology but rather by the essential CBP's. The "Contracting" sub-topics in CrossFlow have provided the author a basis for the development of a deep structure of Collaborative Business Processes in the later sections.

5.5.2 *Matchmaking of Ad Hoc Processes*

The "matchmaking of *ad hoc* processes" is worthy of investigation as part of dynamic B2B collaboration research. In their first few publications on *ad hoc* process matchmaking [44-46], Wombacher *et al* argued that the decentralised nature of Web services pre-disposes them to the *ad hoc* matchmaking of different business processes. For example, matching the procurement business process from company *A* with the transportation business process of a third party logistics company *B*.

However, no techniques for *ad hoc* matchmaking are available at the moment. Wombacher *et al* proposed a lower (and more technical) level approach to match business process descriptions described by means of conjunctive finite state automata [44, 46] (Author's note: *finite state automata* is simply a state machine as a model of behaviour composed of a finite number of states, transitions between those states, and actions [214, 215]).

The ultimate goal of Wombacher's matchmaking technique was to solve the "service discovery" problem (recall Section 5.1.2) commonly encountered by the widely distributed Web services. The success of Web services mainly depends on the availability of tools facilitating usage of technology within the addressed B2B integration problems. According to them, the current concept for service discovery in Web service architectures is *via* the Universal Description Discover and Integration (UDDI) (i.e. the Yellow Pages of Web services) providing limited query functionality and not being able to deal with the multiple dimensions of a service, like for example semantic, business process, or quality of service aspects. Building on their technique, Wombacher *et al* also proposed a matchmaking engine known as IPSI process finder (IPSI-PF) to address the limitations of UDDI, and to provide a matchmaking definition and an engine focusing on process aspects. [176]

5.5.3 Semantics Utilised for Process Management within and between Enterprises (SUPER) Integrated Project

The *Semantics Utilised for Process Management within and between Enterprises (SUPER) Integrated Project* is a follow-up of another research project known as *Data Information and Process Integration with Semantic Web Services (DIP)*. The project consortium is made up of the following institutions from Europe :

- Software vendors (*SAP AG, IBM Research GmbH, IDS Scheer, Ontotext Lab*)
- Service providers (*Telefónica I+D, Hanival Internet Services GmbH, Telekomunikacja Polska (TP) (Polish Telecom), Nexcom Telecommunications, eTel Austria AG, National University Ireland in Galway*) and
- Research teams (*MIP, the Business School of the Politecnico di Milano, TU Eindhoven, Open University Milton Keynes, University of Stuttgart, Leopold-Franzens University of Innsbruck,)*

The overall objective of SUPER is to “raise Business Process Management (BPM) to the business level”. This objective requires that BPM is accessible at the level of semantics of business experts. To them, this can be achieved via the Semantic Web Services (SWS) technology. Their argument is that SWS offers the promise of integrating applications at the “semantic level”. SUPER also states that by combining SWS and BPM, SUPER will create “horizontal ontologies” which describe business processes in general and “vertical oriented ontologies” to support domain-specific annotation (e.g. telecommunication industry)[203, 216].

After reviewing the relevant publications, the author’s impression is that Semantic Web Services is the “silver bullet” for making B2B collaborations and BPM more human. There is no deeper investigation into whether or not Semantic Web Services are actually able to easily capture the inputs naturally from business experts. Besides, SWS is still a developing technology, with many issues still unaddressed (e.g. dynamic composition of SWS) and has not yet been adopted in the mainstream as the most easy-to-use BPM or B2B methodology. Equating the business level to the semantic level overlooks the fact that other aspects like pragmatics and business goals exist in the real world. Because of these intrinsic weaknesses of the SUPER project, the author did not pursue it further.

5.5.4 SemBiz Project

Started in 2004, the SemBiz project [217] is an Austrian consortium’s (Leopold-Franzens University Innsbruck, Vienna University of Technology, Hanival Internet Services GmbH, eTel Austria AG) effort that “*aims at bridging the gap between the business level perspective and the technical implementation level in Business Process Management (BPM) by semantic descriptions of business processes along with respective tool support.*”

The author is not convinced. To begin, its architecture as shown in Figure 5.6. makes no mention of the integration (except for the “Automatic Deployment Tool”) and the “ease” of usage when one uses the architecture. The architecture looks like a comprehensive collection of contemporary Semantic Web terminologies and buzzwords, has no novel features and shows no natural way to bridge the business-IT gap. Furthermore, their publication list (e.g. [218, 219]) on their project Web site are not really pertinent to this architecture. They claimed that the main problem in BPM is the “human expertise needed” to transcend business needs and technical implementation, and their approach is to base their semantic framework on the Web Service Modelling Ontology as a basis for describing business processes. To the author, this viewpoint is naive as it effectively equates business to Semantic Web Services.

Figure 5.6: The SemBiz Architecture

In an effort to describe high-level business processes, SemBiz aims at developing techniques for (1) querying business process spaces to discover appropriate business processes for specific objectives, and (2) composing business processes out of existing process fragments. The author feels that this is not feasible, as the assumption of a pre-existing business process space must be first defined or set up, just like those in the early days of EDI. Furthermore, “composition” of the business processes still needs technical expertise. This means that it contradicts its grandiose goals of bridging the dichotomy between business and technical implementations. In the author’s opinion, the SemBiz project exemplifies the current lack of focus in the BPM field. To the author’s best knowledge, no prototype or application has been reported.

5.5.5 A*STAR IMSS Ont-CAF Project

The “*Ontology-augmented, Context-aware Web Services Framework for Agile Enterprise Collaboration*” (Ont-CAF) project [163] is a Singaporean collaboration project consisting of [220] the *Singapore Institute of Manufacturing Technology (SIMTech)*, *Nanyang*

Technological University (NTU), National University of Singapore (NUS) and Massachusetts Institute of Technology (MIT), USA.

The main aim of the research is to develop a low cost, Web-service based methodology and system that facilitates dynamic collaborations amongst companies. This project is logically divided into three stages (See Figure 5.7):

Figure 5.7: The decomposition of high level business processes into executable services via 3 stages in the IMSS Ont-CAF project. Src: [163]

1. Stage 1: Formation of business processes

In this stage, high level goal-oriented business processes are formulated dynamically, either *via* some rules, or *via* referencing a business process repository with generic business process templates. This thesis directly contributes to this stage.

2. Stage 2: Matching and selection of services

After the high-level business processes are formed, they will be decomposed and the activities in the business processes will be matched with the relevant service profiles that atomic activity is linked to. The profiles linked together will be able to be executed in Stage 3.

3. Stage 3: Services mapping and composition

At this stage, the service profiles that are matched would have formed a business process that can be described and executed by appropriate Web services. The Web services will then be executed into operational business processes.

The underlying supporting features of the 3 stages are:

1. Context-aware framework that articulates the contextual differences between collaborating partners
2. Ontology matching algorithm that enables semantic matching of Web services

3. Trust-filtering algorithm that pre-qualifies the buyer and supplier, and also the reliability of the service.
4. Web Service and Business Process Quality of Service (QoS) assessment features.

The Ont-CAF project started in June 2006 and was completed in February 2010 with a validation of procurement, order fulfilment and cross-enterprise leasing business processes of a regional pallet-leasing company, KHL Pte Ltd. More information of the Ont-CAF project can be found on: <http://vei.simtech.a-star.edu.sg/imss>

5.6 Applications to Research

Table 5.1 summarises the how the key cBP research reviewed motivated the PhD contributions in the proposed methods in Chapter 8, Chapter 9, Chapter 10 and Chapter 11.

Table 5.1: Applications of Chapter 5's discussions to PhD methodologies

Section	Assertions/ Basis / Arguments Raised	How it contributes	Impact Chapter
Web Services and Service-Oriented Computing			
5.1.1	Web Services	Advocates the atomic, modular Web-based software systems which can be combined to perform a more complex business process or task. This encourages reuse and also easy integration of available resources.	Chapter 11
5.1.2	Service Oriented Architecture (SOA)	The architecture defines the layers, roles and challenges for a truly service-oriented approach to Web-application design. Main challenges lie in the area of service composition and service management.	Chapter 9, Chapter 10 and Chapter 11
5.1.3.2	Bottom-Up vs. Top-Down approach	The discussions in this section highlighted the key difference between the author's approach (i.e. top-down) to solving SOA's challenges and common Web service research approaches (i.e. bottom-up).	Chapter 6, Chapter 7, Chapter 8 and Chapter 9.
Semantic Web			
5.2.1.2	Semantic Representation - Description of Meaning in the Data	Enables the author to model the complex relationship between primitive tasks, compound tasks, roles, documents, constraints and many other components usually found in business processes. Such complex data cannot be easily encapsulated singularly by current technologies such as Petri Net-based technologies. Ontologies also enable the merging of knowledge and the inference of new knowledge.	Chapter 9
Dynamic cBP Formulation Research			
5.5.1	CrossFlow Project	Demonstrates the achievability of a cross enterprise integration scenario via contracts. However, its technologies do not consider the prior discovery and contact establishment phases. Despite this, CrossFlow's architecture and framework demonstrate how an awareness of the cBP phases can bring about targeted solutions to cBP.	Chapter 8

5.7 Chapter Summary

To recapitulate, dynamic B2B collaboration pre-supposes the ability to compose high level business processes starting from strategic goals and then decomposing them into lower-level executable business tasks. Web services can represent business tasks while the semantic Web expresses their relationships.

Web services are basic building blocks that can be dynamically composed to create high level inter- and intra-enterprise business processes. However, present obstacles prevent this from happening: e.g. *representation of context*, *ascertaining the level of trust*, etc. Also, Web services research is stagnating because of (1) the assumption of easy discovery of services, (2) narrow focus on low-level execution, at the expense of high-level composite tasks, and (3) the assumption of a “*bottom-up*” approach in solving high level B2B problems.

The “*bottom-up*” approach assumes that low-level primitive Web services can be combined into composite services that address complex high-level issues. On the other hand, the “*top-down*” approach starts from a high-level goal, and decomposes the tasks into primitive low-level tasks. The success of the “*bottom-up*” approach is contingent upon low-level services ‘knowing’ their higher purpose while the success of the “*top-down*” approach is in the dynamic decomposition of the high level goal into low-level tasks of the appropriate granularity.

As B2B collaborations begin at the strategic level, the “*top-down*” approach seems to be a more pragmatic choice. The “*top-down*” approach drastically reduces the search space of Web services as well as the computational effort to match Web services. However, the decomposition of high level goals remains a challenge. A rich description of *meaning of data* is also needed so computer algorithms can ‘know’ how to decompose and chain tasks. The decompositions of cBPs tasks and the expression of their relationships are best addressed by RDF triples and ontologies such as OWL. This thesis adopts the “*top-down*” approach and addresses the decomposition challenge in Chapter 8, Chapter 9 and Chapter 10. Such task decomposition algorithms, and ways to best encapsulate task relationships, are described in the following chapter.

Chapter 6. Potential Techniques for Dynamic cBP Formulation Research

This chapter highlights the underlying nature of the research problem, and the key research challenges confronting dynamic cBP formulation. Related techniques which individually address each research challenge will be introduced and a hybridization of these techniques to address the formulation of dynamic cBP will be explained.

6.1 Nature of Research Problem

From the preceding chapters, we saw the importance of the “top-down” approach for achieving dynamic cBP formulation. As shown in Figure 6.1, such an approach will require (1) the collection of collaboration goals and constraints, (2) automated decomposing the goals into granular tasks for relevant roles, and (3) automated chaining tasks together to form meaningful cBPs. These three steps, also known as the *planning phase*, can be done manually (static) or automatically (dynamic).

Figure 6.1: Steps required in top-down approach for dynamic planning of cBP models

The predominant industry practice is to *manually* perform the planning phase via standards such as BPMN or UML Activity Diagrams, culminating in the execution of the models. However, according to Table 6.1, such manual BP modelling approaches are mainly *static*. *Static approaches* were also identified as one of the main obstacles to the *dynamic integration of business processes* in [5].

Table 6.1 also puts into perspective how a complete *dynamic* approach will require the **dynamic** achievement of **both** the *planning phase* activities and *execution phase* activities. Present cBP research predominantly focuses on the *execution*, overlooking the *planning phase* of BP modeling. This thesis attempts to address the dynamic *planning phase* gap.

Table 6.1: Status and differences between planning and execution phase

	Fulfilment of business goals via business process modelling and management

Phase and activities Approach	Planning Phase	Execution Phase
	<ul style="list-style-type: none"> • Collection of User Goals and Requirements • Decomposing goals into granular tasks • Chaining of tasks to form meaningful Business Process Models 	<ul style="list-style-type: none"> • Execution of generated Business Process Models • Monitoring and optimising business processes
Static Approaches (Current)	<i>Manually done, techniques widely available</i>	<i>Dynamically done, techniques widely available</i>
Dynamic Approaches (Thesis goal)	<i>Dynamically done , absence of techniques</i>	<i>Dynamically done, techniques widely available</i>

Hence, this chapter attempts to introduce a brief overview of related techniques which may individually offer solutions to the following challenges to achieving *dynamic planning phase*:

1. Dynamic decomposition of goals into granular tasks (See Section 6.2).
2. Description of relationships between compound tasks and primitive tasks, and how tasks are related to the roles involved in B2B collaboration. (See Section 6.2.5 and 6.3).
3. Ensure that decomposed granular tasks are dynamically composed with appropriate control-flow structure mirroring real-life business processes, and that they can be executed in the Web *via* Web services (see Section 6.4).

It is important to know that these techniques cannot individually address this thesis' research objective. Hybridising their desirable characteristics enabled the author to address the dynamic cBP formulation challenge (Chapter 8 to Chapter 11). Hence it is important for us to review the underlying techniques which influence the author's methods, starting with methods which enable dynamic decomposition of goals into granular tasks.

6.2 Hierarchical Task Network (HTN) Planning

6.2.1 Understanding the Nature of Artificial Intelligence Planning

Artificial Intelligence Planning (or AI Planning) is concerned with the *automatic synthesis of action strategies (a.k.a. plans)* from a description of actions, sensors (of the applied environment), and goals [221]. AI Planning **contrasts** two other approaches to ‘intelligent’ behaviour: the *programming approach*, where action strategies are defined manually, and the *learning approach*, where action strategies are inferred from experience [221].

6.2.1.1 Main components of an AI Planning methodology

From the Hubble Telescope to programs which play the game of bridge, AI Planning has many applications. Applications which adopt the AI planning methods were known as “general problem solvers” in the sixties. For a general problem solver to work effectively, its **domain** of application must be clearly defined. Otherwise, the general problem solver will not be able to provide the user 2 of its main ingredients: a *general modelling language* for describing the problem domain, and *general algorithms* for solving the problem [221].

6.2.1.2 “Sequence of Actions” vs. “Action Strategies”

Different problems require different solutions. The solution to a problem like travelling from Point A to B will require a *sequence of actions*, while the solution to a medical diagnosis of a patient’s symptoms may be an *action strategy* determining the tests and checks to perform as a function of the observations gathered [221, 222].

In *classical planning*, the initial state is known and the transition between states is deterministic. Hence, the solution is a *sequence of actions*, where the plan maps states from the initial state all the way to the ending goal state. In *conformant planning*, on the other hand, an action maps a state into a set of possible successor states, and the task is to find a sequence of actions that lead to the goal for any possible transition and initial state.

Business processes are not just simple sequences of actions, but have complex possible routes and control flow paths which can only be determined during runtime. Hence, they are better represented by action strategy solutions much like that of conformant planning.

While the action strategy solution is identified by the author as one that resembles dynamic cBP formulation needs, it is worthy to note that there is still an absence of action strategy planning methods (a contrast to a large abundance of classical sequence of action methods) for direct application into business process domains. Hence, the author has to augment methods

producing action strategies with modifications to the model (i.e. adding in workflow patterns (See Section 6.4.1)) for his proposed solution in Chapters 9 and 10.

6.2.1.3 **Planning vs. Scheduling vs. Searching**

It is important to differentiate between planning, searching [223] and scheduling [224]. While from a simplistic point of view they all take in a goal and provide a solution, they are very different in purpose and usage. While search algorithms return results that match the attributes a user is trying to find, it is unable to do the following planning characteristics: (1) unify goal and action representation to allow selection, (2) divide-and-conquer by ‘sub-goaling’, and (3) relax requirements for sequential constructions of solutions. In planning, actions are modeled to have requirements (i.e. prerequisites) and consequences that can constrain applicability in a given situation. Such considerations are not even considered by searching.

Unlike planning, scheduling did not gain prominence in AI until the 1980’s when Fox et al began work on the ISIS constraint-directed scheduling system [225-227]. When comparing between planning and scheduling, a simple definition is that planning defines what needs to be done and scheduling defines when to do what. Baker [228] and Pinedo [229] define scheduling as the problem of assigning limited resources to tasks over time to optimize one or more objectives. This means that the focus of scheduling is on optimizing time and resources, and it involves choices about which resources to use for each given task. Hence, scheduling’s focus is different from planning.

6.2.2 **Classical Planning**

With the above justifications, we have justified the application of AI planning for solving one of the thesis’ research challenges in “decomposition of compound tasks into primitive tasks”. This section focuses on classical planning, a loose grouping for most techniques proposed from the 1960’s to 2000.

In a classical planning problem, the objective is to achieve a given *set of goals*, usually expressed as a set of positive and negative literals in the propositional calculus. The initial state of the world, a.k.a. *initial conditions*, is also expressed as a set of literals. The possible actions are characterized by parameterized templates which contain a set of *preconditions* which must be true before the action can be executed and a set of effects that the action will have on the world. Both the *preconditions* and *effects* can be positive or negative literals. These parameterized templates are also known as STRIPS (Stanford Research Institute Problem Solver) *operators*. While many techniques extend STRIPS, there are serious limitations in STRIPS such as the lack of descriptions for resources used in a task, the lack of

ability to model uncertainty (due to a rigid set of simple positive or negative literals) and the fact that actions are modeled as instantaneous and uninterruptable.

AI planning techniques mostly fall into one of the following categories [224]: Classical Planning, Hierarchical Task Network (HTN) planning and Decision-Theoretic Planning. In this thesis, Decision-theoretic Planning, and two other prominent groups, case-based planning and reactive planning, will not be discussed, because they are less relevant to the cBP formulation domain. For a comprehensive overview of planning techniques the reader may wish to refer to survey articles such as [222, 230-234]. However, to understand why HTN Planning was adopted for dynamic cBP formulation, we have to understand the background and limitations of the following *classical planning* techniques.

6.2.2.1 Forward State Space Search

The Forward State Space Search (FSS) [224, 235, 236] is the most straightforward approach to AI Planning. The planner starts with the initial conditions describing the world, and chooses an operator whose preconditions are satisfied in that state, and constructs a state by adding the effects stored in the operator and removing any proposition that is the negation of an effect. Search continues until a state is found where all goals are attained, resulting in a sequence of states which will lead us from the initial conditions to the state of matching the goal.

However, FSS has its limitations. In FSS, the number of permissible actions in any state is often very large, resulting in an explosion in size of search space [224, 235, 236]. This can pose major performance problems because it considers even completely irrelevant actions. This is likened to the limitations of the “bottom-up” approaches in SOA research. For FSS to be successful, strong heuristic guidance must be provided so that only a small fraction of the search space is explored.

6.2.2.2 Backward Chaining

Backward chaining is the direct opposite of FSS, where the planner works backwards from the goal state [224, 235, 236]. This removes the problems associated with examining irrelevant actions, but it is not without its problems: there are often times it is not "obvious how to generate a description of the possible predecessors of the set of goal states".

6.2.2.3 Partial-Order Planning

What distinguishes partial-order planning from the other two is that it is not totally-ordered as we see in forward and backward chaining. Instead, partial-order planning enables us to "take advantage of problem decomposition." The algorithm "works on several sub-goals independently, solves them with several sub-plans, then combines the sub-plans" [224, 235,

236]. In addition, such an approach also has the advantage of flexibility in the order in which it constructs the plan. That is, the planner can work on 'obvious' or 'important' decisions first, rather than being forced to work on steps in chronological order.

Despite the promise, partial order planners have not met with much practical success [232, 233, 236]. This is because this methodology, although has a branching factor much lower than the FSS, still experiences a considerably large search space and requires considerable bookkeeping. Most of the planners proposed so far also have generally been limited to solving problems that involve a small order of a dozen actions.

6.2.3 Difficulties of Classical Planning and Partial Order Planning

In some real-world domains such as B2B collaborations, the FSS, backward chaining and the partial order planning can cause substantial difficulty. According to Nau et al [235], this is mainly because if a planner is searching backward from the goal or is developing a partial-order plan, it often does not know the input state for each of its planning operators. This makes it difficult to reason about those planning operators. For example in Figure 6.2, we cannot tell whether or not Step s3 is executable unless we know whether Step s2 comes before s3, whether it comes before Step s1, whether $w=v$, and whether $u=v$.

Figure 6.2: A partially ordered plan

To overcome this, standard AI planners requires each planning operator to have its constraints and effects represented as a list of atoms/ literals. However, representing operators in this way will make it very difficult for the AI planner to perform the following important aspects of realistic planning [235]:

1. Doing complex numeric calculations (e.g. computing probabilities, monetary calculations).
2. Interacting with external information sources (e.g. sensors, CAD modelers)
3. Dealing with imperfect information domains or multi-agent domains such as negotiations.

6.2.4 Hierarchical Task Network Planning

6.2.4.1 Why HTN Planning for Dynamic cBP Formulation?

Given the limitations of 'traditional' planning methods, there is a need for a method which will address complex real-life scenarios such as B2B collaborations. According to Smith et al [224],

virtually all planning systems that have been developed for practical applications make use of the Hierarchical Task Network (HTN) planning [237, 238] technique [239-241].

The main difference between the previously introduced AI planning techniques and HTN Planning is that HTN Planning is about reducing *high-level* tasks down to the primitive tasks, while classical planning is about assembling actions to attain goals. In HTN planning, the objective is usually specified as a high-level task to be performed, rather than a conjunction of literals to be attained.

This means that HTN planning's high-level task to be performed is equivalent to high-level business goals discussed in this thesis. HTN Planning also offers us not only a method to address the research challenge of decomposition from high-level compound tasks into primitive tasks, but also gives us a way to ensure that the Web services performing the granular tasks can be composed together with minimal hassle.

6.2.4.2 HTN Planning in a Nutshell

Human beings plan in a goal-directed way, and often divide the large goal into intermediate sub-goals or processes which can solve the problems. Likewise, HTN planning [242-244] proceeds by recursively expanding (decomposing) high-level tasks into networks of lower level tasks that accomplish the high-level task.

Figure 6.3: Different 'methods' for the travel compound task to decompose

In HTN planning, a *goal* to a *problem* is realised via a *plan* of simple steps generated from the dynamic decomposition of a hierarchical network of *compound tasks* and *primitive tasks* of a *domain*. During the decomposition and chaining of tasks, the HTN Planning Algorithm matches the *constraints* defined by the user with the *criteria* of the appropriate *method* that describes how this particular task can be further decomposed to its *sub-tasks* (See Figure 6.3). Tasks are stored as a network that describes the parent-child relationship of the tasks. There are two types of tasks: *compound tasks* which can comprise lower level compound tasks, and lowest level *primitive tasks* in the network. As such, primitive tasks do not have any methods for decomposition.

Figure 6.4: A travel problem represented as a HTN. [245]

After the HTN planning algorithm traverses through the HTN recursively decomposing tasks according to the matching methods, a result (or plan) is generated for “travelling from University of Maryland (UMD) to Massachusetts Institute of Technology (MIT)” (See Figure Figure 6.4). Through HTN planning, tasks can be decomposed and sequenced given some goals (e.g. *travel (UMD, MIT)*). This can potentially bridge high-level cBP tasks into low-level granular, operational tasks ideal for Web service execution. Planning is complete when the resulting network contains only primitive tasks and the set of constraints is consistent. This result is also known as a plan (Figure 6.5).

Figure 6.5: A plan generated by the HTN algorithm decomposing *travel(x, y)* [245]

6.2.5 Why HTN Planning alone is insufficient for Dynamic cBP Formulation?

Despite its robustness in decomposing high level tasks and its resolving of a prominent research challenge (i.e. decomposition), HTN planning cannot be applied directly to the dynamic cBP formulation problem wholesale. This thesis proposes the hybridization of HTN planning with other methodologies covered in the next few sections. The need for hybridization with other methodologies such as Ontologies and Workflow Patterns is mainly due to the following HTN planning limitations [239-241]:

1. Problem descriptions, methods and the task networks of a certain domain are traditionally described in a format known as the PDDL (Planning domain description language) [246, 247]. However, PDDL lacks clearly defined semantics for the decomposition methods and it causes a difficulty in judging or evaluating things such as consistency and completeness.
2. As the strength of the system is highly dependent on the skills and experience of the person who has pre-defined the methods in the domain, methods have to be developed to cover all possibilities. This is not easy, and changes to the domain can cause adaptation problems in the PDDL description.
3. HTN planners are often seen as ‘brittle’, as they are unable to handle tasks not explicitly anticipated by the designer.

The first and second limitations have been successfully overcome by the usage of ontologies to describe planning domains and problem descriptions in the place of PDDL. The third limitation currently has no clear solution, but the idea of adopting an augmented HTN planning methodology seems to be a very pragmatic solution to the dynamic cBP formulation problem. Ontological descriptions not only address the semantic description gaps and the lowering of dependence on the robustness of method definitions, they also promise to be Web accessible and allows inference and creation of new data from existing data. Let us now discuss how ontologies are applicable to the nature of this thesis’ research problems.

6.3 Ontologies

The previous section has shown that the original planning domain and problem description language PDDL may not be sufficient for a Web-based, semantically rich description of complex B2B collaboration tasks. RDF and Ontologies are the ideal candidates for addressing these limitations.

6.3.1 A Brief Overview of Web Ontologies

As such, this thesis’ proposed methodology to describe the problem and planning domain of dynamic business process formulation is based on Web ontologies. As mentioned in Section 5.2.1.2, RDF and ontologies enable one to semantically describe relations between different concepts for the Web.

Web ontologies are usually conceptual models of (or some aspect of) the world; they define vocabulary describing various aspects of the domain being modelled and explicitly specify the intended meaning of that vocabulary [248] so machines can “understand” the information they

parse and pass. In other words, such explicit declarations of meanings of each concept (resource) and how they are related in each ontology model equips machines with the ability to recognise and “make sense” of the meaning of the things described [248, 249].

At the time of writing, there are two dominant ontology description languages: WSMO (Web Service Modeling Ontology) and OWL (Web Ontology Language) (Section 5.2.1.2). While both are the dominant standards for describing ontologies, the author decided to utilise OWL for its larger and wide-spread industry adoption and declared support from global software consortiums, i.e. in the interests of higher long-term potential. This, however, does not demean the semantic capabilities of WSMO. For a detailed introduction to ontologies and the Semantic Web, the reader may refer to Section 5.2.1 and [248].

6.3.2 Why Model HTN's as Ontologies Described in OWL?

As mentioned in Section 6.2.4, HTNs store relationships between *compound tasks*, *primitive tasks*, *methods* and *problem description* in the form of ontologies. Traditional techniques of HTN planning usually store their knowledge base, problem and domain descriptions in the form of text-based files (e.g. PDDL [247, 250]).

In contrast to current XML-based files like OWL, traditional text-file-based formats like PDDL are usually not amenable to current Web-based processing and Web-dependent business environments. Furthermore, current techniques adopting classical HTN tools (e.g. SHOP2, XPlan) [251, 252] require prior translation into the classical domain before planning can happen; resulting in losses in translation. As such, there is a need to represent HTN's knowledge in more Web-friendly formats.

What is needed is a well-structured language that can represent HTN knowledge and the relationships between concepts, and to facilitate Web-based applications. OWL is the ideal candidate for these needs. To the authors' best knowledge, there is also no facility for easy merging of information with other files to create new knowledge in the older representations of HTN's. Modelling HTN's as ontologies in OWL also allows them to merge with other ontologies to create new knowledge.

In order to represent taxonomies within and across enterprises, Semantic Web researchers have proposed several ontological representations. Ontology captures a taxonomy which defines classes of objects and their inter-relationships. Ontologies hence represent concepts and their inter-relationships; these can be hierarchical relations, whole-part relations, or any other meaningful relationship. Figure 6.6 shows an example of these relationships. The diagram illustrates the divisions and sub-divisions in an organisation.

Figure 6.6: An example of an Ontology of an Organisation from the viewpoint of Fox *et al.*[253]

Classes, subclasses and relations are very powerful building blocks for the Web. We can express a large number of relationships among entities by assigning properties to classes which are then inherited by their subclasses. For example, an address may be defined as a type of *location*, whereas *city codes* may be defined to apply only to *locations*, and so on. Hence, if *city codes* must be of the type “city” and *cities* generally have Web sites, we can discuss the Web site associated with just a city code even if there are no databases linking the city code directly to a Website (i.e. inference) [164]. In the author’s opinion, classes (and subclasses) provide meaningful inferences and linkages between associated entities, thus paving the way for a robust planning domain description supporting dynamic B2B collaborations.

An ontology may define the rule “*If a city code is associated with a state code, and an address uses that city code, then that address has the associated state code.*” With this definition, a program (e.g. a semantic Web service) can then readily infer that the Cornell University address, being in Ithaca, must be in New York State, which is in the U.S.A and therefore should be formatted into U.S.A. addressing standards [164]. The computer program does not truly understand this information, but with ontology, it can now manipulate the terms so that they are meaningful to the user.

6.3.3 Related enterprise-related ontology research projects

There are many applications of ontology (e.g. in retail, in healthcare, etc.) but not research into **Enterprise Ontology** which has gained momentum in the last decade. Here are some examples of the prominent ones:

1. Wand and Weber proposed to use the philosophical discipline of Ontology for building and maintaining information systems. Based on the work by Bunge [254, 255], they provided a set of “*Bunge-Wand-Weber*” (BWW) models for the analysis of Information Systems’ modelling tasks [256, 257]. Rosemann and Green [256] applied the BWW

model to the development of the meta-models for information systems, with particular interest in BPM execution languages.

2. As part of the “*Toronto Virtual Enterprise (TOVE)*” project, Gruninger, Fox, and several others developed formal ontologies for various aspects of an enterprise [253, 258, 259]. While much of their work can likely be reused for dynamic cBP formulation, their ontologies are not expressed in current ontology languages like OWL.
3. Recently, Dietz postulated a comprehensive ontological approach to enterprise modelling [260]. His works do not address Semantic Web services, but focus on extracting the essentials of business organizations so as to manage the essential business processes (instead of harping on the technology). Due to his background in LAP (recall Section 3.2.2), contributions to contemporary challenges like process composition in Services-oriented Architectures (SOA) and intelligent queries to the process space are naturally limited.

6.4 Consideration of Execution Phase, Lower-Level Requirements

HTN Planning provides a methodology for dynamic synthesis of plans via recursive decompositions of goals and high-level compound tasks into primitive tasks, and ontologies provide the method to describe the knowledge base of common B2B collaboration tasks used by the HTN planning algorithm to perform decompositions matching business goals and criteria. A combination of these two methods will result in plans which are guidelines for Web service composition and execution.

These plans will also have to “cater to” the needs of the low level executions of each activity, and the manner at which they are strung together. Workflow patterns [35, 261] give us a systematic classification of recurring control flow patterns (much like software design patterns) found in business processes and when augmented into the ontology describing the planning domain and problem description, we will be able to create dynamically synthesized ‘plans’ of business process tasks containing the appropriate control flows at each phase.

Besides taking care of the control flows in the execution of the activities (i.e. how primitive tasks are strung together), we need to enable direct execution via existing techniques for BP execution.

6.4.1 Embedding Control Flows with Workflow patterns

Workflow Patterns [35, 261], which are a collection of generic, recurring process constructs, may be re-combined (assembled) into new business processes. The *workflow patterns*, headed by van der Aalst *et al* [35, 261], borrowed ideas from the hugely popular *Software Design*

Patterns [262] movement (i.e. general reusable solutions to commonly occurring problems in programming) in Software Engineering. (Note: To align to the terminology of the publications, the author will be using ‘workflow’ to describe business processes only in this section.)

Before the advent of workflow patterns, many processes were designed too hastily, very often without the benefit of experience. Design patterns in BPM can facilitate the construction of better processes [7]. Workflow patterns do not only provide a list of reusable modules of process control flow [35] but also a checklist to assess the modelling capabilities of commercial workflow management systems, and most recently for the evaluation of business process languages and notations [263-265].

Table 6.2: Workflow patterns

Category	Pattern	Explanation
Basic Control Flow	Sequence	Execute two or more activities in sequence
	Parallel Split	Execute two or more activities in any order or in parallel.
	Synchronization	Synchronize two or more activities that may execute in any order or in parallel; do not proceed with the execution of the following activities until all these preceding activities have completed; also known as barrier synchronisation.
	Exclusive Choice	Choose one execution path from many alternatives based on data that is available when the execution of the process reaches the exclusive choice
	Simple Merge	Wait for one among a set of activities completes before proceeding; it is assumed that only one of these activities will be executed; typically, these activities are on different paths stemming from an exclusive choice or a deferred choice.
Advanced Branching & Synchronisation Patterns	Multiple-Choice	Choose several execution paths from many alternatives.
	Synchronising Merge	Merge many execution paths; synchronize if many paths are taken; do the same as for a simple merge if only one execution path is taken.
	Multiple Merge	Wait for one among a set of activities to complete before proceeding; if several of the activities being waited for are executed, the simple merge fires each time that one of them completes.
	Discriminator	Wait for one among a set of activities to complete before proceeding; if several of the activities being waited for are executed, the discriminator only fires once.
	N-out-of-M Join	Same as the discriminator but it is now possible to wait until more than one of the preceding activities completes before proceeding by setting a parameter N to some natural number greater than one.
Structural Patterns	Arbitrary Cycles	Do not impose any structural restrictions on the types of loops that can exist in the process model
	Implicit Termination	Terminate an instance of the process if there is nothing else to be done
Multiple Instances (MI)	MI without synchronisation	Generate many instances of one activity without synchronizing them afterwards.
	MI with “a priori” known design time knowledge	Generate many instances of one activity when the number of instances is known at the design time (with synchronization)
	with “a priori” known runtime knowledge	Generate many instances of one activity when a number of instances can be determined at some point during the runtime (as in FOR loop but in parallel)
	MI with no “a priori” runtime knowledge	Generate many instances of one activity when a number of instances cannot be determined (as in WHILE loop but in parallel)
State-based patterns	Deferred Choice	Execute one of a number of alternatives threads. The choice which thread is to be executed is not based on data that is available at the moment the execution reaches the deferred choice, but is rather determined by an event (e.g. an application user selecting a task from the work-list, or a message being received by the process execution engine).
	Interleaved Parallel Routing	Execute a number of activities in any order (e.g. based on availability of resources), but do not execute any of these activities at the same time/simultaneously. Most BPMS vendors are not able to depict this properly.
	Milestone	Allow a certain activity at any time before the milestone is reached, after which the activity can no longer be executed.

Cancellation Patterns	Cancel activity	Stop the execution of an enabled activity. This pattern is also known as 'kill activity'.
	Cancel case	Stop the execution of a running process. This pattern is also known as 'kill process'.

The theoretical basis for workflow patterns is Petri Nets [24]. In workflow pattern nomenclature, processes have activities which may have many different paths. The patterns are grouped into six main categories [7, 35, 261] as summarised in Table 6.2. The above-mentioned constructs in Table 6.2 form the *workflow control flow patterns*. The reader should note that the basic control flow patterns are the ones adopted by the author in Section 9.5.2. Future work will explore how the other five categories can be augmented into the cBP hierarchical task ontologies for dynamic cBP formulation.

However, there are also other aspects of business processes to be addressed. In the author's opinion, this method may be too operationally focused, and biased towards flow control at the expense of other important factors of BPM like the fulfilment of business objectives. To be fair, the research done by van der Aalst *et al* focussed on operational processes in the first place [34]. Only in recent years, van der Aalst's team has moved on to propose new sets of patterns (2004-2006) as follows:

1. **Workflow Data Patterns** [266] - the various ways in which data is represented and utilised in workflows
2. **Workflow Resource Patterns** [267] - various ways in which resources are represented and utilized in workflows
3. **Exception Handling Patterns** [268] - patterns-based classification framework for characterising exception handling in workflow systems.

The technical discussion of the above mentioned patterns is not within the scope of this report. However, the reader should note that the three new sets of patterns have not yet matured.

6.4.2 Output Format for the Synthesized Plans

For the plan (output) produced from a HTN planner to work in a service-oriented environment, it needs to be encoded in a way which can link current Web services and is able to support the workflow patterns enforcing the control flows of the plan's primitive tasks.

In the author's literature review, it was found that the Business Process Execution Language (BPEL) was an ideal candidate for describing the plans for the following reasons:

1. **Most popular and does not have any serious competitors in the industry** [7, 98, 135, 136]. This means that companies considering products supporting BPEL can be assured of stability and minimal risk of obsolescence. With its adoption in systems of most major software vendors, users of products from small BPMS vendors also do not need to worry about portability issues.

2. ***Allows programmers to focus on processes rather than low-level programming constructs.*** In comparison with conventional programming languages such as Java, BPEL provides powerful mechanisms for typical business process interactions such as long-term transactions, asynchronous messaging and parallel activities. This means that it would require much more effort and lines of code to express the process in a conventional programming language [269].
3. ***Makes use of web services paradigm.*** This capitalises on the dynamic and highly-adaptive nature of web services. BPEL incorporates a number of specialized features for web services development including direct support for XML data definition and manipulation, a dynamic binding mechanism based on explicit manipulation of endpoint references, a declarative mechanism for correlating incoming messages to process instances, which is essential for asynchronous communication. As such, BPEL may be seen as an attractive alternative to conventional programming languages when it comes to developing web services [269].

6.4.2.1 BPEL in a nutshell

Business Process Execution Language (BPEL), an XML-based language for specifying business processes in the Web Service environment, is a collective term for both its versions [30, 270]:

1. *Business Process Execution Language for Web Services (BPEL4WS) ver. 1.1*
2. *Web Service Business Process Execution Language (WS-BPEL) ver. 2.0*

It is also currently the most influential execution standard in the market. BPEL is used together with Web Service Description Language (WSDL) and other related technologies. This means that BPEL is used to define how the business process is built from invocations of already-built Web services, and the kind of interactions with external participants the process is involved with. Technically, BPEL can be seen as an XML-based programming language for Web Service compositions. In-depth understanding of BPEL requires software development competence as well as knowledge of the Web Service technologies it is built on. The first version, BPEL4WS 1.0, was originally submitted to OASIS WSBPEL Technical Committee [30] by Microsoft and IBM in July 2002. BPEL4WS combined properties from Microsoft's Web Service Flow Language (WSFL) [28] and IBM's XLANG [69] (See Figure 6.7).

Figure 6.7: The Evolution of BPEL

The revised version of BPEL4WS (i.e. version 1.1) [270] has been widely adopted by tool vendors. In version 2.0, the language was renamed to WS-BPEL which was approved as an OASIS standard in 2007 [30]. In this thesis, the acronym BPEL is generally used to refer to both versions because the older version cannot be considered obsolete. Clear distinction between the versions is made whenever necessary. The new version involves syntactic changes and improved alignment with other XML streamlining technologies such as *XPath* [271]. Hence, it is important to note that newer versions of BPEL processes are **not backward-compatible**. From the viewpoint of the creators of BPEL [270], business processes can be described in 2 ways [272]:

1. **Executable business processes** model actual details and behaviour of a participant in a business interaction.
2. **Business protocols**, in contrast, use process descriptions that specify the mutually visible message exchange behaviour of each of the parties involved in the protocol, without revealing their internal behaviour and details. The process descriptions for business protocols are called **abstract processes** [273].

In other words, the full implementation logic of the process is defined via *executable processes*, while only the message exchange between process participants (i.e. business protocols) is included for the *abstract processes*.

The business protocols described by abstract BPEL processes closely resemble the nature of plans (of primitive tasks) generated by HTN planners. This is because their message exchange behaviours are devoid of the implementation details such as server location and port numbers, and such details are not needed in the planning phase of dynamic cBP formulation. For this strong resemblance, abstract BPEL is noted as a highly possible candidate for the proof of concept for the proposed hybrid methodology of this thesis.

As the details of BPEL syntax are not the main focus of this research, further details of the BPEL syntax and structure will not be discussed but can be found in Appendix B.

6.5 Chapter Summary

Right now, the planning of cBPs is not fully automated, because cBPs are manually modelled through such languages as BPMN, UML and the RAD. This is not acceptable because of the rapidly changing supply chain scenarios. Because of these rapidly changing supply chain scenarios a method is needed to automate the entire process starting from high level goals and constraints, to decomposing compound tasks into granular executable tasks and finally chaining them together. The chained executable tasks are then automatically executed by methods such as BPEL, thereby accomplishing dynamic cBP formulation.

Dynamic cBP formulation is contingent upon addressing **three challenges**: (1) the dynamic decomposition of goals into granular tasks, (2) describe the relationships between compound tasks and primitive tasks, and (3) describe the relationships between tasks and roles involved.

The **first challenge**, dynamic decomposition of goals into tasks, can be addressed *via* hierarchical task network (HTN) planning. HTN planning starts from its inputs, user goals and constraints, to generate its output “plans”, which are sequential chains of tasks that progressively achieve the stated goal. HTN planning was chosen amongst other AI planning techniques as it is goal-oriented, applies a top-down approach, enables effective decomposition of tasks, and is adopted by almost all real-life planning software.

In HTN Planning, task networks and the methods to decompose the tasks are stored in a so-called “problem domain description language (PDDL)” file. Because of the PDDL files, HTN planning can only generate sequential tasks at one level, and not the complex, nested control flows of tasks of real-life cBPs. HTN Planning also cannot describe relationships between B2B roles and business documents transacted across tasks. Worse, as the PDDL file is just a structured text file, it cannot merge knowledge and cannot be deployed on the Web, the medium of choice for global B2B collaboration. In light of the above limitations of PDDL, an alternative must be sought.

As for the **second and third challenges of dynamic modelling**, ontologies can address these two challenges. Current ontological languages such as RDF, OWL and WSMO were reviewed. OWL emerged as the best candidate for replacing PDDL and for facilitating dynamic cBP formulation, as it is scalable, encompasses RDF descriptions of relationships, and is most widely adopted by practitioners.

By themselves HTN planning and ontologies cannot solve dynamic cBP formulation. They need to be hybridized together for Web-based B2B collaborations. The original HTN planning

algorithm need to be extended to consider control flow constructs, and to read from ontologies instead of the traditional PDDL files in order to generate “plans”.

The dynamically synthesized “plans” of primitive tasks must next be able to exploit current Web technologies such as Web services. To effect the correct control flow and decision routes, workflow patterns were also incorporated into ontological HTN’s. Abstract BPEL was also chosen as the language of choice for the output “plans” generated.

With these in place, we are now primed to summarize the gaps uncovered in the literature review and articulate the sub-objectives of this research in the following chapter.

Chapter 7. Research Issues and Directions

In Chapters 2-6, a detailed overview of BPM and B2B collaboration research and practice and the key terminologies were presented. To the author's best knowledge, such a detailed treatise and classification of BPM literature has not been performed before.

This chapter discusses (1) the approach taken by the author in his investigations, (2) a high level summary of the research contributions of Chapters 2-6, and (3) a 'genealogy' of BPM's historical developments. Following from these, we will review the gaps of current literature in depth, and establish the objectives of the author's research.

7.1 Research Roadmap

As mentioned in the Introduction chapter, the *goal of this research* is to discover a technique that enables agile B2B collaborations through the *dynamic formulation of cross-enterprise business processes based on high level user business goals and criteria*.

Figure 7.1: Summary of author's approach towards addressing the research goal

Figure 7.1 shows a diagrammatical overview of the author's approach to addressing the research goal and the chapters which are linked to the progression from literature review to methodology. As shown in this roadmap, the author investigated into the states of the art in

BPM and B2B collaboration research and practice, and then attempted to explore techniques which address the sub-behaviour of dynamic formulation of cBP.

The literature review of these 3 areas of study were covered in Chapters 2 to 6, and Chapter 7 (this chapter) summarises the gaps relevant to this research, and bridge the literature review with the research methodologies presented in Chapters 8-11.

7.2 BPM Landscape

7.2.1 Research Contributions of the BPM Landscape

The BPM Landscape (see Figure 7.2) is a non-exhaustive taxonomy² of the key theories, best practices, industry standards and B2B concepts in BPM. The classification of BPM concepts clarified many previously loose terminologies and articulated the key BPM features. The BPM landscape:

1. Affords a bird's eye view of the key BPM areas, those topics that are gaining popularity, losing steam or adopted as an industry standard as of the time of writing.
2. Clarifies the current confusion and misinterpretations of BPM terminology. For example, BPM standards and notations are now classified into graphical, interchange, execution and diagnosis standards (refer to Appendix B for details).
3. Highlights missing linkages, knowledge gaps, and prospects for improvement.

7.2.2 Organisation of the BPM Landscape

The BPM Landscape depicts research efforts, theory, best practices in BP modelling and industry standards. As shown in Figure 7.2, the author has divided the BPM Landscape into 2 broad domains:

1. *The Western Half* consists of **industry**-related terms; information more relevant to practitioners and solution developers.
 - a. **Standards** – A plethora of BP modelling notations, languages and standards exist, but a proper classification is absent. This part of the BPM Landscape makes sense of the differences between graphical standards, interchange standards, execution standards and diagnosis standards (see Appendix B).
 - b. **Best Practices** – Current industry best practises in BP modelling, and the different business process types and components are featured here. The different types of business processes and architectures are also presented. Many of these topics were covered in Chapter 2 and Chapter 3. Architectures

² The author intends to create a Creative Commons project for a community development for the BPM landscape after his PhD.

and enabling tools are not discussed in this thesis as it is not within the scope of this research.

2. ***The Eastern Half*** consists of research efforts, current trends and potential research (not just confined to this research).
 - a. ***Theoretical Foundations*** – The theoretical underpinnings of BPM have been classified into 3 schools of thought, of which the flow-based and communicative perspectives were discussed in Chapter 3. The Artificial Intelligence approach is not discussed in this report as it is not within the scope of this research.
 - b. ***B2B Process Integration*** – Industry practices and research initiatives in B2B collaboration/ cBP areas discussed previously in Chapter 4 and Chapter 5 are presented.

7.2.3 Topics with research potential

The BPM Landscape also attempts to indicate, at the time of writing this report, the so-called ‘hot’ research topics and those waning in influence. To do so, four symbols are adopted:

- ***Up and Coming*** – Topics recently gaining widespread attention.
- ***Losing Steam*** – Topics that are legacy items/ obsolete/ waning in influence.
- ***Most Popular*** – The “de facto” standards used by most practitioners/ researchers.
- ***Very Promising*** – Research topics that hold promise.

Besides knowing the key concepts of BPM and the status of research topics, researchers must be able to appreciate the theoretical developments and knowledge progression of the field. A study of the *evolution* of BPM in the next section is timely.

Figure 7.2: The BPM Landscape

7.3 The BPM Research Genealogy

BPM is a multi-disciplinary field (*viz* organisation management, mathematics, computer science, linguistic computation, etc.). There is a need to appreciate the seminal BPM works, their influences and progression unto the current status today. Such an exercise can also identify potential research trends. The ***BPM Research Genealogy*** (Figure 7.3) supplements the BPM landscape in addressing the “***BPM fundamentals***” foundation by documenting the seminal works and their subsequent developments in a comprehensive timeline.

7.3.1 The Scientific Development of BPM

From Figure 7.3, it is clear that Business Process Management has its foundations in organisation management, process theories and even communicative theories like the theory of speech acts and the theory of communicative actions. Figure 7.3 reveals three primary thrusts of scientific BPM developments:

1. ***The Formalism Track*** focuses on research into process theory and algebras which have their roots in the formalisms of communicating systems. It can be observed how the 2 key theories (Petri Nets and Pi Calculus) eventually influenced developments in the recent years.
2. ***The Business Process and Workflow Management Track*** champions scientific management and the BPM discipline. It has its roots in value chain networks and business process reengineering in the challenges of increased globalisation.
3. ***The Language Action Perspective Track*** focuses on designing systems that are natural to the way humans communicate. Linguistics, natural language and semiotics are employed to encapsulate the natural behaviour of business collaborations and workflows. This track also summarises the developments of the Communicative Approach.

7.3.2 Chronological Progression of BPM Research

Besides a track-wise chronological view, the BPM Research Genealogy reveals trends in each decade starting from the 1930's:

- ***1930's and 1960's : Basic Research***

Key seminal theories in formalisms depicting processes (e.g. Lambda-calculus, Petri Nets (recall Section 3.1)), and philosophical underpinnings about the communicative nature of the office environment and communication (Austin's “How to Do Things with Words” and Searle's Speech Acts Theory) were proposed.

Figure 7.3: The Genealogy of Business Process Research

- ***1980's: Emergence of Globalisation= Management Theories***

Globalisation rendered the management of businesses and their processes more complex. This gave rise to management paradigms like Value-Chains. LAP researchers, who were much ahead of their time, also started proposing IT solutions based on the Speech Acts Theory and the successor Theory of Communicative Action to support “cooperative work” in business office environments. This also means that the actual pioneers of applying IT to BPM are the LAP pioneers.

- ***1990's: Application of IT into BPR and Workflow***

With the increased computerisation of business practices, the application of IT systems like Workflow Management systems and LAP systems like Action Workflow increased. In the early nineties, Hammer and Champy, after observing the ingenuity of the Japanese manufacturers, proposed the Business Process Reengineering concept with a hint about how IT can support this paradigm. This kick-started the proliferation IT-based BPR and WfM proposals.

- ***2000's: Standardisation and Consolidation; Quest for higher forms of business representations***

The BPM field is saturated with BPM languages and notations. At the same time as attempts to standardise BPM languages and consolidate BPM knowledge gained momentum, many started to realise the absence of scientifically grounded design methodologies for business processes.

In the other school of thought, LAP researchers finally found an application area to apply the communicative aspects of business processes: the Pragmatic Web.

Just as many ideas and paradigms were bred from the needs of different eras. BPM researchers need to understand ***this era's needs: adapting to the demands of accelerated globalisation backed by the proliferation of Web-based technologies [4].***

While many Web based applications have made information transfer more efficient, a new business-IT paradigm that can naturally depict businesses is needed. In the 2000's the accelerating scale and pace of globalisation and the growing maturity of Web technologies have rendered static, operational-level business processes largely obsolete. Web based technologies, being distributed, offer the best chance of coping with fast, dynamic changes. In the author's opinion, dynamic business collaboration is in the vanguard of BPM research.

The ***BPM Landscape*** and the ***BPM Research Genealogy*** underpin BPM fundamentals. We next move on to reviewing the gaps of the literature relevant to the main research problem.

7.4 Summary of Gaps in BPM Research

This section focuses on gaps which have not been addressed by current and past research. While some progress in BPM research has been made, many crucial issues remain unresolved.

7.4.1 *No universal BPM terminology*

A common problem in BPM is the absence of universal terminologies [7, 37, 274-276]. Terms are used loosely to represent distinct scope and feature differences [277]. One example is the confusion between *business process reengineering (BPR)*, *workflow management (WfM)* and BPM. Such confusion has led to mismatched (or worse, wrong) BP solutions being implemented. This was addressed in Chapter 2.

7.4.2 *Too many duplicating BPM languages and standards*

Because of a lack of understanding of the fundamentals of BPM, new languages and standards proposed often contain duplicating features [274], and loosely claim to be based on formalisms like Pi-Calculus [7] and Petri Nets [104]. Much of the above may be traced to the software industry, which is notorious for new terminologies and buzzwords. Though not directly within the scope of the research problem, the BPM Landscape and Appendix B attempts to provide an authoritative taxonomy of BPM terminology to resolve the issue.

7.4.3 *Inadequate basic research in business process essentials and formulation*

Present BPM representations are grounded in flow-based approaches, which are based on computer science process theories like state machines, Petri-Nets and Pi Calculus. As discussed in Section 3.1, flow-based methods are only good for modelling and controlling flow as they are usually low-level and, being void of context, are indifferent to the highly communicative and dynamic BP's in real life [278, 279]. This gap was discussed in Chapter 3.

7.4.4 *Lack of high level business process representations*

To the author's best knowledge, graphical notations like flowcharts and role-activity diagrams may not be the best way to represent *real* business processes. Much inter-personal communication comes to pass before a business deal is sealed. Researchers must be wary of the "IT-proposal" trap and actually observe how *real* business processes are formulated. When we can understand the essence of business processes and extract the most natural way to mimic real-life business operations (e.g. decomposition of tasks from high-level goals), we would have done for business processes what object-oriented programming did for software development and what relational databases did for data management.

7.4.5 Too much focus on operational processes

Flow-based methods are adept at modelling the control of flow, but current techniques are still unable to link strategic, management level and cross-enterprise business processes effectively. Graphical notations can model all levels of business processes, but they are best suited for operational level business processes because they cannot embody higher level semantics. There is a real need to represent the higher level dynamic business information in strategic, management and cBP's. This was discussed in Chapter 3 and later addressed in Chapter 8.

7.4.6 Gap between business analysts and IT implementers

Presently, only operational business processes are automated; there are few tools that can model high level business processes. Currently, business analysts have to model business processes *via* graphical notations like flow charts and role-activity diagrams before IT implementers can convert these notations into verbose syntax, to automate the biz processes. How such strategic business processes are decomposed into management business processes and eventually decomposed into bite-sized operational business processes (or *tasks*) has not been fully investigated. Chapter 6 highlighted this phase as the planning phase, and discussed the main challenges which are later addressed from Chapter 8 to Chapter 11.

7.5 Summary of Gaps in Current B2B Systems and Approaches

Chapters 4-6 highlighted some **pressing issues** concerning systems and techniques supporting B2B collaborations:

7.5.1 High setup and licensing costs

While B2B information exchange standards like EDI and RosettaNet support cBP's, they usually come at exorbitant setup and licensing costs. For example, the cost of a RosettaNet template for a singular collaborative business process costs between S\$150,000 to S\$300,000 [133, 280, 281]. Just imagine the financial burden to a small company which may easily need 20 to 30 solutions. Apart from high costs to the organisation, potential collaborators may be deterred as well. These are typically small suppliers. Hence, there is a pressing need for a low-cost, non-exclusive open standard collaboration platform just like the Web.

7.5.2 Static B2B setups not flexible and not responsive

Besides being expensive, cBP's modelled in B2B standards are usually static in nature, and are unable to adapt to sudden changes on short notice (recall Chapter 6). A dynamic and agile method to model cross-enterprise business processes is needed. This is directly addressed by the author's dynamic proposals in the methodology chapters later in this thesis.

7.5.3 Lack of Research into Dynamic B2B Approaches

7.5.3.1 Overlooking the Planning Phase of B2B Collaboration

Current research seldom differentiate between the *planning* and the *execution* phases of an entire B2B collaboration cycle. Many of the current literature such as the Semantic Web Services and the Matchmaking of *Ad Hoc* processes do not explicitly differentiate between *planning* and *execution* but focus on *programming* solutions which address a dynamic B2B collaboration. As mentioned in Section 6.2.1, plain programming will never be able to address real-life scenarios if continued in their current approach. This thesis proposes a novel technique in Chapter 8, Chapter 9 and Chapter 10 addressing the current *planning* phase gap of B2B collaboration solutions, thereby enabling dynamic on-the-fly collaborations.

7.5.3.2 Current Emphasis of the Bottom-Up Approach

The Semantic Web and the SOA approaches generally belong to the bottom-up approach and are distinctly different from the way humans approach complex problems, i.e. divide-and-conquer of goals and sub-goals. This was highlighted in Section 5.1.3.2 and will be addressed in Chapter 8, Chapter 9 and Chapter 10.

7.5.4 Current B2B standards' emphasis on functional perspective of business processes

It was shown in Chapter 4 that current B2B standards such as RosettaNet, ebXML and OAGIS have focused on classifying common B2B tasks into scenarios, functions or departments. The advantage of this is easy customisation for the IT consultant and process owners. However, it creates an “over-the-wall”, silo mentality when modelling B2B collaborations. BP's in reality require cross-functional/ cross-department processes to realize B2B goals. For example, in typical “*Procurement*” processes, the inventory department needs to provide the current inventory status and the finance department the budget available in one transaction. The “*cross-department*” or “*process-oriented*” perspective of B2B collaboration tasks is something which the author believes are more aligned to the realities of supply chain business processes. Chapters 8 and 9 provide a method which promotes the process-oriented perspective.

7.5.5 Limited semantic representations in current B2B standards and systems

Many current B2B standards also cannot succinctly and accurately describe the semantic knowledge of (1) how cross-enterprise business tasks are linked to each other, (2) how business goals are decomposed into more granular tasks, (3) how tasks and goals are linked to

business documents, (4) which actors are authorized to transact on the tasks, and (5) the organizational structure of collaborating companies. An ontological modelling methodology which succinctly addresses these limitations is proposed in Chapter 9.

7.5.6 Lack of open access collaboration framework

Outside of B2B information exchange standards, it is difficult to discover potential collaborators and formulate public business processes *on-the-fly*. A methodology that allows sellers (big and small) to advertise their products and services can facilitate the search for matching buyers. This facility should not be confined to a membership within a consortium like the RosettaNet, but open to all. While the methodology proposed in the later chapters promises an open access framework, an industry-wide participation is required to prove the validity of an open-access framework which is more economical than existing methods. While such industry-wide validation is not attempted due to the constraints of his research, the author was able to validate his approach against the real-life industry processes of a regional pallet-leasing company.

7.6 Research Goal and Sub-Objectives

The above gaps underscored the urgency of the **objective of this thesis**, namely:

To discover, implement, test and verify a technique that enables agile B2B collaborations through the dynamic formulation of cross-enterprise business processes based on high-level user business goals and criteria.

This is in anticipation of the impending wide-spread increased dependence on the Web as a medium for B2B collaboration [2, 4], and promises to reduce the dependence on current high-cost time-consuming methods which do not automate the *planning* phase in fulfilling business goals.

7.6.1 Formulation of Research Questions

This objective requires an emphasis on the following planning-phase activities:

- 1. Establish user goals and requirements, with consideration of the scenarios' constraints.**
- 2. Dynamically formulate cross-enterprise business processes which address the goals, and integrate collaborating companies *on-the-fly*.**

These two phases can be interpreted as the following research challenges (RC) and their associated research questions (RQ):

- **RC 1: Encapsulation of high-level business goals and constraints**

- RQ 1.1: How many types of fundamental business goals are there in B2B collaborations?
- RQ 1.2: What are the critical criteria and constraints in B2B collaboration?
- RQ 1.3: Is there a more human natural way of expressing these high-level goals and constraints?
- **RC 2: Decomposition of high-level goals into granular executable tasks**
 - RQ 2.1: What methodologies exist to decompose compound tasks into primitive tasks?
 - RQ 2.2: Are there any current modelling techniques to represent decompositions?
- **RC 3: Composition of the granular tasks in the appropriate sequences and control flows, performed by the appropriate roles**
 - RQ 3.1: How do we ensure that the business process' control flows are faithfully reproduced when the granular tasks are composed together?
 - RQ 3.2: What modeling augmentations/ techniques must be applied to ensure that the composed primitive tasks can be easily executed by existing execution-phase technologies such as Web Services?

7.6.2 Scope and Domain of Research

This research focuses on solving the problems encountered in the pursuance of a global service-oriented architecture, and hence assumes Web services as the main building blocks of business processes across collaborating companies.

The research pertains to predominantly public processes, as public processes are fairly standard regardless of industry and domain. They are a sound basis for dynamic business process formulation research. As a result, reference data for industrial B2B tasks in cross-enterprise business processes will be extracted from RosettaNet's Cluster 3 and Cluster 4, ebXML's common business process catalogue, and the OAGIS list of BOD's.

This thesis will focus on creating a solution for the *planning phase* of dynamic formulation of cBP's rather than execution phase which has been well researched. The author intends for his methodology to be independent of the underlying technologies, and envisions his method for generating dynamically-formed cBP's residing above the layer of execution-level languages for Web service execution. Some examples of these execution-level languages include BPEL and WS-CDL. However, BPEL is chosen for the purpose of implementation of the proof-of-concept, and also because of its wide-spread adoption and long term potential.

7.7 Chapter Summary

The Chapter presents the BPM landscape (a taxonomy of BPM research topics) and the author's research roadmap.

Following the BPM landscape, this chapter discussed the BPM Genealogy since the advent of serious study into business processes as a research discipline. Over the decades, BPM theories developed along three paths: the *formalisms* path, *business process and workflow management* path, and the *language-action perspective* path. This research is based on *business process and workflow management* path, and the *language-action perspective*.

The BPM genealogy traced the evolution of BP management theories, through IT automation of BP execution, culminating in the most pressing need of this decade: *Rapidly adapting to the demands of accelerated globalisation using Web-based technologies*.

The gaps in the research literature helped the author formulate the research objective, challenges and associated research questions, namely modeling high-level business goals and constraints, decomposing them into granular executable tasks, and composing granular tasks in the appropriate sequences and control flows.

Part II: Method, Implementation and Results

Chapter 8. Proposed Conceptual Foundations and CBP Framework

This chapter establishes theoretical underpinnings of cross-enterprise business processes (cBP's), and introduces the author's proposed *Essential cBP (ECBP) Framework*, the cBP axioms inferred from the ECBP Framework, and finally, a treatise on the types of B2B goals and high-priority constraints. Together with the axioms, the ECBP sets the foundation for the author's proposed hybrid approach to solving dynamic cBP formulation in Chapters 9-11.

8.1 The Essential cBP (ECBP) Framework

The proposed *Essential Cross-Enterprise Business Process (ECBP) Framework* (see Figure 8.2) identifies the generic phases in almost all B2B collaborations. The author derived this framework of generic B2B phases through a literature review of B2B phases modelled in LAP approaches and the DEMO approach (recall Section 3.2.3.5).

DEMO stipulates that beneath the complexities and increasing load of information experienced in business processes, there is always an *essential model* [85] which generically holds true for most scenarios. Within this *essential model*, each step is crucial for the achievement of the business goal, and the removal of any step in the model will cause the entire process to fail. The ECBP is the *essential model* for all types of B2B collaborations.

8.1.1 How the ECBP Framework is Derived

Figure 8.1: How ECBP is derived, and how its contributions serve as basis for BOWL

As shown in Figure 8.1, the proposed ECBP Framework has its roots in the collaboration frameworks found in LAP approaches like *Conversations for Action (CfA)*, *SAMPO*, *Action Workflow*, *Business-As-Action Theory* and *Dynamic Essential Modelling (DEMO)*.

LAP approaches (recall Section 3.2) are based on the Speech Acts Theory, which states that all speech acts have some degree of commitments expressed by the speaker. LAP not only computerised the Speech Acts Theory, but also placed the different types of commitments, also known as illocutionary forces, into meaningful conversation graphs. These conversation graphs generically represent B2B frameworks. For example, *CfA* models how a ‘request’ commitment can trigger the next action of ‘promise’ or ‘reject’ and so on (recall Figure 3.4). While each of the actions have different levels of commitments, we know that each action, as a result of commitments will trigger the progression of the business process transaction to the next task, till the end task is reached.

All five LAP approaches present conversation frameworks as the foundation of their modelling of collaborative business processes. Likewise, the author uses the ECBP as the foundational framework for his attempt at modelling dynamic B2B collaboration. From a high level, the LAP approaches overlap to describe the (1) *Discovery* of collaborators (buyers or suppliers), (2) *Establishment of Contact* between suppliers and buyers, (3) *Negotiation* of terms, (4) *Contract Agreement* or acceptance, and (5) *Contract Fulfilment*. These five phases were further validated to be essential through the application of DEMO steps, and the author’s interviews with 6 industry representatives from the manufacturing industry (described in detail in Chapter 12).

Table 8.1: ECBP Phases Distilled from LAP Approaches

B2B phase LAP Approach	Discovery	Establish Contact	Negotiation	Contract Agreement	Contract Fulfilment
CfA		✓	✓	✓	
SAMPO	✓	✓		✓	
Action Workflow			✓	✓	✓
BAT			✓	✓	✓
DEMO		✓	✓	✓	✓

While each of these five phases is variously found in the CfA approach and the four other frameworks, **none of them completely embody all five phases**. CfA is lacking *Discovery* and *Contract Fulfilment*; *Negotiation* and *Contract Fulfilment* are missing in SAMPO; *Discovery* and *Establish Contact* phases are missing in both Action Workflow and BAT, while the DEMO method lacked the *Discovery* phase.

8.1.2 Other Influences From LAP

Table 8.2 below summarizes the other key contributions of LAP approaches into the construction of the ECBP framework.

Table 8.2: Summary of contributions from LAP methods to the ECBP Framework

Section	Theory	Assertions/ Basis / Arguments Raised	How it contributes to ECBP
3.2.3.1	Conversation for Action (CfA) Approach	Proposed that the “human decision maker” must be in the loop (i.e. business process).	Established the need for inclusion of human decision makers in all BPM system.
3.2.3.2	Speech Act-based information analysis Methodology with computer-aided tools (SAMPO)	Their conversation graphs characterize the AD features and describes stages, movements in a conversation, as well as the conditions (called predicates) that restrict and control the performance of acts.	<ul style="list-style-type: none"> • A generic representation of the conversations is needed. • Established the need for conditions to restrict and control the performance of actions in a process.
3.2.3.3	Action Workflow approach	The action workflow works because it has the following components: 1. A theory (a generic business framework) 2. An analysis and modeling method 3. A supporting software tool	Established the need for a BPM approach to have a theoretical underpinning, a modeling method and a software. This is the foundation for the components in this research.
		There are always 2 roles in the action workflow loop: 1. An identified customer 2. A performer	Established the truism of two fundamental business roles (buyer and seller) of every essential business collaboration/ transaction.
		The action workflow loop consists of 4 phases: 1. proposal, 2. agreement, 3. performance and 4. satisfaction.	Generic business transactions contain phases where buyers and sellers enter into common understanding on the proposal, agreement, performance and satisfaction of a business goal.
		“tasks are defined by requests and commitments expressed in loops”	This means that a business process is composed of generic granular actions.
3.2.3.4	Business as Action Theory (BAT) Framework	There are always 2 parties involved in a business transaction: 1. a supplier selling 2. a customer buying	Established the truism of two fundamental business roles (buyer and seller) of every essential business collaboration/ transaction.
		BAT Generic business framework contains actions and conditions necessary for establishing a business transaction (refer to Figure 3.9): 1. Proposal phase (3 sub-phases) a. Business identification b. Exposure and contact search c. Contact establishment and negotiation 2. Commitment phase 3. Fulfilment phase 4. Completion phase	<ul style="list-style-type: none"> • Established how a process-oriented “phase approach” to modeling business processes enables the true modeling of the essential business process much more than the traditional functional model (See next point in table) • The BAT framework has extended the action workflow with a more in-depth proposal phase, and the commitment phase.
3.2.3.5	Dynamic Essential Modeling (DEMO) Framework	DEMO methodology for extracting the essential ‘deep structure’ of a business process	With the essential business process extracted, we can understand the actual main goal of the process, and encourage a technology-independent design of business processes. This is something that the traditional flow-based approaches are still trying to do.
		At the informational level, an organization is observed as a system of actors that send and receive information and performs calculations on this information in order to create derived information.	
		The O-E-R Transaction (essential transaction) is composed of 3 phases: 1. The Order phase 2. Execution phase 3. Result phase

Figure 8.2: The Essential cBP Framework

8.1.3 Phases in the ECBP Framework

The ECBP Framework shown in Figure 8.2 conceptually and essentially represents the Essential Level of abstraction of all cBPs. There are two general collaborator roles: the *Buyer* (*a.k.a. Customer*) and the *Seller* (*a.k.a Supplier*). As stated in the Scope of Work (Section 1.2.2), the aims of the thesis is to performing the *planning*, i.e. the generation of the plans (action strategies) that can enable the eventual execution of each phase by human or machines. The focus is on generating plans which do not neglect or overlook the phases unknowingly. Hence, the Essential CBP framework is a crucial representation of the essential components of any typical B2B collaboration. The Essential CBP framework mainly consists of five phases:

8.1.3.1 Discovery Phase

This stage is started by a buyer's need to buy and a seller's intention to sell some product or service. This simple relationship is the basis of the economy and business relations. The aim is to gain mutual benefit (e.g. profits, reputation, long term collaborations, etc) amongst the collaborators. Figure 8.2 can be viewed in two rows, with the top representing the buyer and the bottom representing the seller. The buyer has a business goal (e.g. to buy some raw materials) subject to constraints. This business goal of the buyer must be matched with a supplier's capabilities (and best offers). Potential sellers are then identified and contacted (via any media) for the start of the next stage.

8.1.3.2 Establish Contact Phase

The contact of a potential partner for collaboration will mark the beginning of the next phase of the cross-enterprise business process. Some examples of this transition in reality are the Request for Quotation (RFQ) business process. When a supplier supplies the quotation, the buyer will negotiate the best deal given its constraints.

8.1.3.3 Negotiation Phase

Negotiation may take one of 3 forms:

Fully Manual Negotiation – This is the classical (and most used) negotiation technique. It can range from a simple phone call or email to the settling of a negotiated deal in the golf course or restaurant. IT researchers often ignore this, when they should be facilitating the transfer of the collaborator's data from Discovery to the Contract Fulfilment phase so business processes can be automatically decomposed once a negotiation is confirmed. In other words, after some protracted negotiations, the CEO needs only tell his staff "Let us buy 1000 items from Company B". His staff should be able to select Company B from a list of contacts generated

after the Discovery stage and trigger the modeling and execution of lower-level supporting business processes that realize the high level goal set by the CEO.

Semi-Automated Negotiation – This is supported by computer software. The IT system will do the laborious task of discovering and matching of the buyer and seller’s wants, but the decision to enter into a contract will still be left to human users. In this method, a buyer specifies the intended items to be purchased and the terms and conditions (e.g. acceptable price range, location, time, etc). After the software captures this info, it returns with the selected sellers. The software increases efficiency but the employee makes the vital decisions. This mode is the most realistic human-computer interaction to support B2B collaborations. In fact, some of the current techniques are already doing this. e-Auctioning portals, Reverse Auctioning software are some examples.

Fully Automated Negotiation – In this mode, the buyer inputs information about the targeted product, specific price limits, and related constraints (e.g. geographical, monetary, time, etc), and the IT system takes over without human intervention. This is not going to be feasible in complex collaborations (e.g. building of a skyscraper) but will be useful for simple transactions (e.g. buying 10 cartons of stationery). IT researchers aiming to fully automate negotiations via “agents” must realise this difference or they will propose ideas which are not easily adopted (e.g. the agent-oriented agreement technologies, or the speech-act theory-based SANP [122] and FLBC [126]).

8.1.3.4 Contract Agreement Phase

The Negotiation phase’s output will be the **contracts** (i.e. “deal” arcs in the diagram) that bind the buyer and the seller. With the contract, the formation of a virtual enterprise is legally established, with each collaborating company fulfilling its role in this newly formed partnership.

8.1.3.5 Contract Fulfillment Phase

After the deal is struck (i.e. contract established and agreed), the buyer and the seller are now part of a virtual enterprise seeking to fulfil the contract (which is simply the business goal for each collaborator). To fulfil this high level goal (i.e. the contract), the collaborators have recourse to Business Process Management. Within the organization, the Business Process Management life cycle [1] stages of “business process design”, “business process enactment”, “system configuration”, “diagnosis” will be enacted to support the fulfilment of the contract. The high level goal must first be decomposed into operational level business processes. One of the ways is MIT Process Handbook’s way of listing down the supporting “*Design*”, “*Buy*”,

“Make”, “Sell” and “Manage” processes [9, 282] and then decompose them. The vision of the author is that IT systems can facilitate this decomposition of processes.

8.1.4 Buyers and Sellers

These five phases of the Essential CBP represents the core *Essential Transactions* (Section 2.1.1) between the Buyer and the Seller. As can be seen, many of the current IT capabilities are ideally suited to each stage in the ECBP Framework. For example, the Semantic Web [164] and ontologies [193, 216, 254-256, 260] can now aid in the discovery stage; LAP methods can aid in Process Formulation; Negotiation protocols, game theory applications can aid in negotiation stage and so forth.

The above five phases actually exist to fulfil **two essential business goals: *to buy* or *to sell***. This view is also supported by all five LAP approaches reviewed.

There are many alternative terms for the “buy” goal, such as procurement, outsourcing, request for tender, etc. The same can also be said for “sell”. This is true even for manufacturing or designing outsourcing, where companies are basically buying or selling some sub-products or sub-goods to make a profit. Thus, we are now able to identify the highest level goals that will fulfil B2B collaborations: “*to buy*” and “*to sell*”. Most importantly, we now observe that both business goals can be decomposed into five phases described above.

8.2 cBP Axioms Derived From ECBP Framework

Arising from an understanding of the ECBP Framework, the author is able to infer four axioms to govern all CBP’s. These axioms enables the author to formulate the essential goals, and the also enabled the author to identify the trigger points for the transition between the ECBP phases. The axioms are:

Axiom 1: *All business processes arise from symbiotic collaborations where the buyer buys a product or service that a seller sells, each gaining some benefits.*

All collaborations arise from the business goal to achieve some benefits (e.g. making a profit, getting the required raw materials to make some product to make a profit in return, etc.).

Axiom 2: *There are at most 5 main phases (Discovery, Establish Contact, Negotiation, Contract Agreement, Contract Fulfilment) in any cross-enterprise business processes.*

These five phases are ever-present in all collaborations. Most of the works from the Business Process Management (BPM) and Business Process Re-engineering field (for example [8, 14, 261, 283]) focus on how the company’s functional units (or departments) serve their roles in

the newly established virtual enterprise, which is formed and sustained throughout the duration of the contract. Before the end of the contract, there are many Action Workflow Loops that creates satisfaction of lower level supporting business processes. The fulfillment of a contract will also mean the end of a virtual enterprise.

Axiom 3: *Essentially, at the ultimate highest level of abstraction, there are only two cross-enterprise goals: “Buy” and “Sell”.*

Regardless of whether it is the transportation processes or the manufacturing business processes, the highest-level business processes that trigger the creation of these business processes are always “Buy” and “Sell”. From the author’s view, Buy and Sell are “multi-illocutionary-force” speech acts which contain elements of *Assertives*, *Declaratives*, *Commissives* and *Directives* (Refer to Section 3.2.2 and Appendix D). These “composite speech acts” form the highest point of business intention that is succinct, and yet contain many pragmatics that allow the inference and eventual construction of operational business processes in the lower levels of decomposition.

The MIT Process Handbook [9, 282] lists five essential business processes (i.e. *Buy*, *Sell*, *Make*, *Manage*, *Design*). In the author’s view, *Design*, *Make*, *Manage* can never be mixed with the processes *Buy* and *Sell* as the highest level of business processes.

Axiom 4: *The start of the contract agreement stage in the CBP life cycle always triggers the decomposition into operational business processes, which are eventually modelled, executed and diagnosed by the BPM discipline.*

The end of *negotiations* of terms and constraints form the start of the *contract agreement*. Very often, the signing of the contract by the collaborators (e.g. sell semiconductor probe equipment to a semiconductor plant) will mean the start of job dissemination, process design (both explicitly and implicitly) and role ownership in the organization in order to achieve sub-goal of “Sell”. The decomposition process stated here is still a manual task in reality, with room for further investigations.

8.3 Key Uses of the EcBP Framework

The ECBP framework and the governing axioms facilitate the following:

8.3.1 Simplifying B2B Collaboration Modelling

The ECBP conceptual framework has simplified the modelling and understanding of complex B2B collaborations without compromising on the essential stages and transactions.

8.3.2 B2B Collaboration Systems and Architecture Design

When researchers and practitioners design B2B Collaboration tools, they can be guided by the deep structure of CBPs: the roles (buyer and seller) and the interactions and transitions across the ECBP phases.

As such, software and architectural designs (i.e. informational and documental levels) will be concise but complete. When the practitioners focus on the essential interactions and transitions, they will naturally be able to design easily-customisable B2B tools that can cater to all companies. Aside from systems, software designers can also adapt the 3 stages of the ECBP into their architecture designs.

8.3.3 Refining Current ECBP research

The ECBP Conceptual Framework also enables us to evaluate the missing gaps in current research and industry efforts to promote real-time B2B collaborations. Table 8.3 compares current B2B standards, methods and projects against the ECBP stages (grouped into three groups: (1) Discovery and Establish Contact, (2) Negotiation and (3) Virtual Enterprise, which consists of Contract Establishment and Fulfilment).

It was found that there were currently no B2B methods that have extensively covered all ECBP stages, and most significantly, there is least coverage on the Negotiation stage. We shall now discuss these industry and research B2B efforts in detail:

Table 8.3: Using the ECBP Framework to Evaluate Current B2B Methods

Current B2B Methods	ECBP Stages Achieved/ Covered		
	<i>Discovery and Establish Contact</i>	<i>Negotiation</i>	<i>Virtual Enterprise (Contract Establishment and Fulfilment)</i>
EDI			✓
RosettaNet	✓		✓
ebXML	✓		✓
OAGIS	✓		✓
BAT		✓	
CrossFlow		✓	✓
SemBiz	✓		✓
SUPER IP			✓

8.3.3.1 **EDI from the Perspective of the ECBP Framework**

EDI mainly handles the *document exchange* between companies, and hence can only cover the collaboration transactions described in the *Virtual Enterprise* Stage of the ECBP framework. This revelation not only uncovers EDI's focus, but also exposes the need for EDI to look into *Discovery*, *Establish Contact* and *Negotiation*. Meaningful extensions need to be added to make EDI fully ready for supply chain realities discussed earlier. The ECBP evaluation has also debunked the misconception that EDI is a full-fledged B2B system.

8.3.3.2 **RosettaNet from Perspective of the ECBP Framework**

With its rich infrastructure, it is evident that RosettaNet is very well-designed for the Informational and Documental levels. With PIPs (recall Section 4.1) that can deal with 'Request for Quotations' and 'Advanced Shipment Notification', it is evident that RosettaNet covers both the transactions in the *Discovery and Establish Contact*, and the *Virtual Enterprise* Stage of the ECBP Framework (see Table 8.3). To make RosettaNet a world-view B2B solution and *dynamic*, extensions to the Negotiation stage have to be added. While the Discovery and Establish Contact Stages are supported, RosettaNet is very expensive and usually costs between USD 100,000 to 300,000 per solution. This is an obstacle to small and medium enterprises. This also means that the number of potential RosettaNet users (collaborators/ partners) is limited to a privileged domain, which is not desirable in an increasingly globalised world.

8.3.3.3 **ebXML from the Perspective of the ECBP Framework**

From the perspective of the ECBP framework, ebXML covers mainly the Discovery stage, the establishment of the contract and the Virtual Enterprise stage (See Table 8.3). One can observe that there is little automation of the transactions of ECBP's Negotiation stage. This is because, like the example mentioned above, collaborators are integrated *via* a mere matching of the queries and the closest CPP that satisfies the query.

8.3.3.4 **Related B2B research**

It is also evident that BAT, CrossFlow, SemBiz and the SUPER project all exhibit gaps in one or more of the ECBP framework.

8.4 **Constraints and Criteria in B2B Collaborations**

When companies collaborate, they not only express their goals, but also consider constraints which bind their collaboration decisions. As postulated in Chapter 7, this research employs a goal driven, top-down approach which decomposes high-level business **goals and constraints** into business processes containing granular tasks suitable for Web service execution. Hence,

before proposing the methodology in Chapter 9 and Chapter 10, it is perhaps useful to understand the types of constraints considered by collaborators in the supply chain.

Table 8.4: Constraints considered in B2B Collaboration based on 1966 Dickson study. Src:[284]

Rank	Factor	Mean Rating	Evaluation
1	Quality	3.508	Extreme importance
2	Delivery	3.417	
3	Performance history	2.998	
4	Warranties and claim policies	2.849	Considerable importance
5	Production facilities and capacity	2.775	
6	Price	2.758	
7	Technical capability	2.545	
8	Financial position	2.514	
9	Procedural compliance	2.488	
10	Communication system	2.426	Average importance
11	Reputation and position in industry	2.412	
12	Desire for business	2.256	
13	Management and organization	2.216	
14	Operating controls	2.211	
15	Repair service	2.187	
16	Attitude	2.120	
17	Impression	2.054	
18	Packaging ability	2.009	
19	Labor relations record	2.003	
20	Geographical location	1.872	Slight importance
21	Amount of past business	1.597	
22	Training aids	1.537	
23	Reciprocal arrangements	0.610	

After (1) reviewing seminal publications on vendor selection criteria [284-286], (2) interviewing HP procurement staff (and ex-staff) about the Hewlett-Packard TQRDCEB Model³, and (3) further interviewing industry representatives from Motorola, Dell, Meiban Plastics Singapore (a plastic parts manufacturer), and LHT Holdings Ltd (a regional, listed pallet leasing company) (Interview questionnaire attached in Appendix G) for the list of top-priority criteria, the author could conclude that all interviewed industry representatives viewed

³ TQRDCEB stands for Technology, Quality, Responsiveness, Delivery, Cost, Environment (Responsibility), and Business (Ethics). It is a set of performance expectations and measurements used by HP and other large US companies (e.g. GE, Seagate, Agilent Technologies, etc.) in qualifying new suppliers, and managing the ongoing long-term business relationship with existing suppliers. (See Appendix F for more details)

the top eight criteria of the twenty three (23) listed (See Table 8.4) in Dickson's study [284] as of either extreme or considerable importance.

The 1966 Dickson study is highly cited as the baseline for later vendor selection criteria methods and papers [286], and was based on a questionnaire sent to 273 purchasing agents and managers selected from the membership list of the USA's National Association of Purchasing Managers. The list included purchasing agents and managers from the United States and Canada. A total of 170 (62.3%) responses were received. Table 8.4 also summarizes the findings of Dickson's study regarding the so-called importance of the 23 criteria for vendor selection.

It was found in the interviews that, depending on (1) the type of industry, (2) collaboration mode (e.g. a new supplier versus an established qualified supplier), and (3) the process taken to produce and deliver the product (e.g. Just-In-Time), the order of importance of the 8 "top" criteria will change. Hence, the author's methodology must not have any bias towards or against any of the criteria. During the interviews, it was also observed that the industry representatives unanimously agreed that the order of priority also differs according to the type of product, and whether it is a made-to-order, made-to-stock. Amongst the criteria, the most commonly mentioned top three criteria that were mentioned has to be the Price, Quality and the Technical Capability of the collaborator.

Constraints versus Criteria

It is useful to note that in this thesis, *constraints* and *criteria* mean the same factors of collaboration, but are used from different viewpoints. *Constraints* are provided or expressed from the collaborator's viewpoint, while *criteria* are guiding values which are stored in models, and used by the algorithm or system to compare against the user-input *constraints*. In situations which do not involve the description of any proposed system or algorithm, the terms *constraints* and *criteria* can be used interchangeably.

8.5 Chapter Summary

Essential Cross-Enterprise Business Process (ECBP) Framework, the cBP Axioms, and fundamental B2B goals and constraints are the theoretical underpinnings of the research contributions of Chapters 9-11.

The ECBP Framework identifies the five generic phases and key tasks found in almost all B2B collaborations: (1) *Discovery* of collaborators (buyers or suppliers), (2) *Establishment of Contact* between suppliers and buyers, (3) *Negotiation* of terms, (4) *Contract Agreement* or acceptance, and (5) *Contract Fulfilment*.

The five generic phases may be found in five LAP approaches, however none of the approaches embody all five phases. CfA lacks *Discovery* and *Contract Fulfilment*; *Negotiation* and *Contract Fulfilment* are not found in SAMPO; *Discovery* and *Establish Contact* phases are not found in both Action Workflow and BAT, while the DEMO method did not consider the *Discovery* phase.

The ECBP framework facilitates B2B collaboration research because it (1) simplifies the modelling and understanding of complex B2B collaborations without compromising on the essential stages and transactions (2) reveals the inability of current B2B standards to model the five generic cBP phases and (3) designing concise but complete software systems.

Four axioms are proposed: (i) the highest level goals are *Buy* and *Sell* (ii) there are only two cross-enterprise goals: “Buy” and “Sell (iii) there are five main B2B phases, and (iv) higher order BPs may be decomposed into operational business processes. As current industry approaches are unable to address all the ECBP phases, they are therefore unable to accommodate dynamic cBP.

From the ECBP and axioms, the highest level B2B goals are established as Buy and Sell. Alongside these goals are eight key criteria for B2B collaboration which are also presented. Together, the framework, axioms, goal and criteria form the basis for a novel methodology for B2B collaboration proposed in the ensuing chapters.

Chapter 9. An Ontology Which Enables Dynamic cBP Formulation

The theoretical foundations presented in Chapter 8 forms the basis of the novel hybridized approach to solve the challenge of dynamic cross-enterprise business process formulation, presented in this chapter (Chapter 9) and Chapter 10. This hybrid approach primarily extends the task decomposition techniques of HTN planning [221], which requires the following elements (recall Section 6.2) (See Figure 9.1):

1. A **modelling language** for describing the problem domain (i.e. BOWL), and
2. An **algorithm**, which refers to the model to solve the problem (i.e. Genesis Algorithm).

As discussed in Section 6.2, HTN planning and ontologies cannot bring about dynamic cBP formulation by themselves. Hence, the author improved the original HTN planning concept shown in Figure 9.1, which displays how the Business-OWL and the Genesis Algorithm facilitate dynamic cBP formulation.

Figure 9.1 also highlights the key differences between the HTN Planning and the author's proposed Genesis+BOWL system. The original HTN planner and the PDDL files can only generate simple single-level plans of sequential tasks, and the PDDL files are not able to encapsulate many more constructs found in business processes in reality (e.g. roles, control flows, etc). Hence, there is a need to modify the algorithm (See Chapter 10) and remodel the PDDL as an ontology with added extensions. These extensions are not just technical extensions but are necessary descriptions of relationships between business process components in reality.

The ontology also serves as a well-structured model that is able to represent HTN knowledge and the relationships between concepts, and to facilitate Web-based applications. OWL is the ideal candidate for these needs.

To the author's best knowledge, there is also currently no facility in the older representations of HTN's for easy merging of information with other files to create new knowledge. Modelling HTN's as ontologies in OWL enables ontologies to be merged to create new knowledge. It also enables the Genesis algorithm to reference directly into the Web-ready Business-OWL (BOWL) ontology (More details in Chapter 10).

9.1 Rationale for Creating BOWL

BOWL models cBP's as a hierarchical ontology of decomposable business tasks and an ontology of possible methods (recall Section 6.2) for all compound tasks. Hence, BOWL is

primarily a knowledge base for an augmented HTN Planning algorithm (i.e. Genesis algorithm [172] which decomposes B2B compound tasks on-the-fly to form a meaningful cBP for a user-stated goal and set of business criteria (e.g. item price, order quantity, lead time, etc) [173].

Figure 9.1: The hybrid approach, its basis and its technical contributions

Aside from addressing the six gaps identified in Section 7.4, BOWL also attempts to address the following:

9.1.1 Lack of dynamic cBP task decomposition methodologies

To the author's best knowledge, there are currently no techniques to synthesise workflows from high-level goals and criteria. Current BP modelling research mainly focuses on the *execution* modelling of processes using BPMN (*Business Process Modelling Notation* [10]) and BPEL (*Business Process Execution Language* [30]), but neglects *planning*, i.e. how workflows are derived to address goals and criteria.

9.1.2 Limitations of Traditional HTN Planning Problem and Domain Description Files

Traditional techniques of HTN planning usually store their problem and domain descriptions in the form of text-based files (e.g. the Planning Domain Description Language (PDDL) [247, 250]). In contrast to current XML-based files, such traditional text-file-based formats are usually unsuitable for current Web-based processing and Web-dependent business environments. Furthermore, current techniques adopting classical HTN tools (e.g. SHOP2, XPlan) [251, 252] require prior translation into the classical domain descriptions before planning takes place; resulting in losses in translation.

9.2 Creating BOWL

The creation of BOWL was contingent upon the creation of both (1) the hierarchical network of compound and primitive tasks, and (2) the network of methods which are related to the compound tasks.

9.2.1 Creating the Hierarchical Task Ontology

9.2.1.1 Manual Synthesis of Tasks

To demonstrate dynamic cBP formulation using industry-based reference models, BOWL was created from synthesizing of hundreds of cBP tasks listed in three major B2B information exchange standards: *ebXML*, *OAGIS* and *RosettaNet*. The consolidated task list is a UNION of all common OAGIS, ebXML, RosettaNet B2B tasks in the following industries: wireless communication [287], computer hardware [288], manufacturing [289] and plastic injection moulding [290]. The manual synthesis took the author about six months to complete as there were several rounds of manual checking for completeness and correctness.

The current tasks in BOWL are from the business processes commonly found in *sales and marketing, inventory management, procurement and order management, logistics and payment*. Other functions that are highly customised, for example: *manufacturing, design and engineering* will be sequenced in the author's future work. RosettaNet was chosen as the base as it has the most up-to-date and most comprehensive reference model amongst the three

standards. It has also a very large user base. Also, RosettaNet PIPs (recall Section 4.1) from Cluster 3 (Order Management) and Cluster 4 (Inventory Management) were used as the foundational tasks for the master list as they represent the majority of supply chain processes.

Tasks from ebXML and OAGIS reference models which are *semantically similar* to the RosettaNet task found in the base list, but which have different task names were discarded after manually checking for similarities in both semantics and control flow (See Appendix E).

Tasks from ebXML and OAGIS **not** found in the RosettaNet list were duly added manually to the master list. Each step in creating the BOWL final task list was later carefully audited for correctness against industry standards' reference lists several times by the author. When the final task was completed, it was further audited and validated by 4 interviewed industry experts from the wireless communication [287], computer hardware [288], manufacturing [289] and plastic injection moulding [290] industries.

During the synthesis, tasks which were added to the master list were also tagged with the relevant executor: Buyer Role, Seller Role, and Web Service Role. Tasks can have a combination of roles. For example, a task can be both executed by a human decision maker (e.g. buyer) and/ or automated by a Web service. This is to facilitate easier Web service composition experiment in the author's related project found in [291].

9.2.1.2 *Sorting and Categorising into 5 B2B Phases*

After the synthesis of tasks into a master list, the tasks were sorted, denoted and categorised into 5 main ECBP phases mentioned in Chapter 8 (See Figure 9.2 and Appendix E): ***Discovery, Establish Contact, Negotiation, Contract Agreement, and Contract Fulfilment*** tasks because:

- Current standards are restrictively categorized according to *functions* (e.g. order management, sales and marketing, etc), rather than in ECBP *phases* which flow together to form a true B2B collaboration.
- The author's proposed five ECBP phases give a better process-oriented overview of B2B collaborations.
- Such a grouping will enable an easier sorting and creation of the hierarchical task networks.

The sorted list can be found in Appendix E, and the sorted list of tasks is then modelled visually using Microsoft Visio as shown in Appendix E. Visio diagrams were used for easier validation of the task hierarchies with industry experts and checking against the industry reference models. They were also beneficial for discussions with the author's colleagues.

Primitive tasks and compound tasks in each phase were then segregated and their hierarchical relationships (decomposition and parent-child) were drafted in logical diagrams such as Figure 3, which shows an example of the Discovery phase (*Note: this was meant to guide the author in sorting parent child relationship and not the complete look of the ontology*). In total, BOWL contains a **total of 341 tasks** networked in a HTN fashion (see Table 9.1). Of these, the *Contract Fulfilment* phase contains the most number of tasks.

Figure 9.2: Synthesising B2B collaboration task lists from B2B standards into five hierarchies of tasks for each collaboration phase

Table 9.1: Breakdown of B2B tasks in BOWL

Phase(s)	Number of tasks within phase
Discovery	17
Establish Contact	55
Negotiation	50
Contract Agreement	18
Contract Fulfilment	198
Top-level compound tasks	3 (i.e. B2BCollaboration, Buy, Sell)
Total:	341 tasks in BOWL

9.2.2 Creating and Linking the Methods Ontology

A hierarchical task network of parent and child tasks cannot work as a standalone. Just like the supply chain reality, compound tasks modelled into BOWL can be decomposed by multiple permutations, given different sets of requirements. For each compound task, its relevant *method* (more explanations in Section 9.4.2) is logically modelled in the diagram like those in

Figure 9.3. These methods are extracted from control flow documentations of the reference models found in RosettaNet PIP specifications. More details of the modelling of methods will be covered in Section 9.4.2.1.

Figure 9.3: Overview of the Visio version of the Discovery Phase HTN

are supported by several sub-classes (e.g. PlanningCriteria) and classes describing Actors and Documents found in business processes.

As shown in Figure 9.5 and Figure 9.6, these are embedded in methods. For Sections 9.4.1 to 9.4.6, readers are encouraged to cross-refer to the logical representation of BOWL tasks and methods in Figure 9.3, Figure 9.5, Figure 9.6 and Figure 9.7 to gain a visual overview and understanding.

9.4.1 Tasks

There are both *compound tasks* and *primitive tasks* in BOWL. *Compound tasks* can be further decomposed into lower level tasks. BOWL makes ***no explicit distinction*** between *compound tasks* and *primitive tasks*; both are modelled as *tasks*. The way to differentiate them is to observe the existence of “hasMethod” properties within the task definition (See Figure 9.5). Compound tasks have one or more “hasMethod” properties since they can be decomposed in one or more permutations while a primitive task does not have any *method*. Figure 9.5 shows a simple definition of the “Buy” compound task and its properties (Note that it has two *methods*).

```
<Task rdf:ID="B-Buy">
  <hasMethod rdf:resource="#MethodBuy.A"/>
  <hasMethod rdf:resource="#MethodBuy.C"/>
  <hasActor rdf:resource="#BuyerActor"/>
  <hasParent rdf:resource="#B2BCollaboration"/>
</Task>
```

Figure 9.5: Simple task description in BOWL

9.4.2 Methods and their components

9.4.2.1 Modelling task decomposition permutations

Methods are constructs that **facilitate dynamic decomposition** by distinguishing between (1) “describing parent-child task relationships”, and (2) “describing the possible permutations a parent task can decompose into some or all of its children tasks”. The former has been handled by the HTN described in Section 9.4.1. The latter is handled by a network of “*Method*” descriptions (individuals) which are linked to their respective compound tasks.

```

<Method rdf:ID="MethodBuy.A">
  <hasControlFlowPattern rdf:resource="#Sequence"/>
  <hasSubtasks rdf:resource="#D-Discovery"/>
  <hasSubtasks rdf:resource="#EC-EstablishContact"/>
  <hasSubtasks rdf:resource="#N-Negotiation"/>
  <hasSubtasks rdf:resource="#CA-ContractAgreement"/>
  <hasControlFlowPattern rdf:resource="#SequenceEnd"/>
  <hasProductType rdf:resource="#Goods"/>
  <hasProductType rdf:resource="#Services"/>
  <hasProcessType rdf:resource="#Manufacturing"/>
  <hasProcessType rdf:resource="#ServiceProvision"/>
  <hasSourcingMode rdf:resource="#ReverseAuction"/>
  <hasSourcingMode rdf:resource="#TraditionalSourcing"/>
  <hasCollaborationMode rdf:resource="#NewSupplier"/>
  <hasBusinessGoal rdf:resource="#BusinessGoalBuy"/>
</Method>

```

Figure 9.6: Methods for the task “Buy”

Methods complement the task network by telling the dynamic business process decomposition algorithm which subtasks to decompose into for each particular matching set of business goal and criteria (see Chapter 10). In BOWL, each method belongs to one single compound task, and stores one possible decomposition path corresponding to matching criteria input by the user. Hence, a compound task can have **more than one** method for it to be decomposed.

9.4.2.2 Components encapsulated in Methods

Before explaining how BOWL enables decomposition from goals to tasks, it is crucial for the reader to understand the following components in method (*cross refer with* Figure 9.6 and Figure 9.7):

1. **Business goal (BG)**– There are primarily two highest-level business goals for each B2B collaboration: *Buy* and *Sell*.
2. **Business Criteria** – There are two types of criteria:
 - a. *Supervised criteria* are constraints decided or described *by the business users*. They describe real-world constraints that determine how business processes are to be formulated and executed. Through the author’s interviews with experts from the plastic injection moulding industry, the electronics industry and the computer industry [29-32] (recall Section 8.4), **eight key supervised criteria were identified to be suitable for planning purposes** (See Table 9.2). Figure 9.6 shows an example of a method containing some of the supervised criteria listed in Table 9.2 as method properties.

Table 9.2: Supervised Criteria and their range of values

No.	Supervised Criteria	Range of values
1	Collaboration Mode (CM)	{new supplier (NS), existing supplier with contract (ESC), existing supplier without contract (ESW), new buyer (NB), existing buyer with contract (EBC), existing buyer without contract (EBW)}
2	Product Type	{goods (Gds), services (Svc)}

	(PType)	
3	Product Name (PName)	{any product name}
4	Quantity (Qty)	{1, ..., n}
5	Item Price (Pr)	{\$1, ..., \$n} in US Dollars
6	Order Lead Time (LT)	{1 workweek (ww), ..., n ww}
7	Sourcing Mode (SM)	{Reverse Auction (RA), Traditional Sourcing (TS)}
8	Process Type (Proc)	{Manufacturing (Mfg), Service Provision (SP)}

b. **Unsupervised criteria** are back-end criteria that are important internal Boolean conditions and system variables storing temporary values aiding the proper execution of a business process transaction. Examples found in BOWL include the Boolean-valued “PO Rejected” (PO_Rej), and “Winning Tender” (WT).

3. **Subtasks** – The possible subtasks that can be decomposed from a compound task the method belongs to. Subtasks are denoted in *methods* that yield different permutations of subtasks.
4. **Control Flow Pattern** – The original HTN Planning algorithm assumes that decomposed tasks are executed from left-to-right in a sequence. This is unrealistic in real-life business processes, where many types of control flow patterns (e.g. sequence, concurrent tasks, splits, loops) exist. Hence in BOWL, as shown in Figure 9.7, the subtasks in methods can take **one or several (nested) number of control flows** adopted from the workflow patterns research [35, 261] and BPEL’s structured actions [30]. Some examples are:
 - a. *SEQUENCE* – Tasks executed in a sequentially-ordered manner.
 - b. *FLOW* – Tasks are executed concurrently.
 - c. *PICK* – Select one task from many possible tasks.
 - d. *REPEATUNTIL* – Tasks are repeated until the specified condition is true.
 - e. *FOREACH* – Tasks that can be performed in Flow or in Sequence controlled by a specified counter variable.
 - f. *IF-ELSE* – Execute 1 of 2 child tasks based on a condition.

[EC] Establish Contact	
METHOD EC.A	{CC= ((BG == Buy) ^ (CM == NS v ESW) ^ (SM == TS)) [SEQUENCE (FLOW (SEQUENCE (EC1, EC2, EC5), EC6), EC7))]
METHOD EC.B	{CC= ((BG == Sell) ^ (CM == NB v EBW) ^ (SM == TS)) [SEQUENCE (FLOW (SEQUENCE (EC3, EC4), EC6), EC8))]
METHOD EC.C	{CC= ((BG == Sell) ^ (CM == NB v EBW) ^ (SM == RA)) [SEQUENCE (EC9, EC10))]
METHOD EC.D	{CC= ((BG == Buy) ^ (CM == NS v ESW) ^ (SM == RA)) [SEQUENCE (EC11))]
METHOD EC.E	{CC= ((BG == Buy) ^ (CM == NS v ESW) ^ (SM == RA) ^ (TTM == False)) [SEQUENCE (EC12))] Parent = Buy v Sell Actor = B v S v WS

Figure 9.7: Logical snapshot of task “EC-Establish Contact” with methods EC.A, EC.B, EC.C and EC.D

Together, **each set of business goals, criteria, subtasks and control flow patterns** tells the dynamic cBP formulation algorithm **one possible method** to decompose a compound task.

9.4.3 Actors of Tasks

After tasks and methods, we move our attention to actors. BOWL tasks can be acted upon by human (i.e. Buyer/ Seller) or non-human actors (e.g. Web services or any modular program). As shown in the example in Figure 9.7, we can see that task EC has 3 possible actors: Buyer or Seller or Web Service (i.e. $B \vee S \vee WS$).

9.4.4 Effects

As in HTN Planning, *effects* are changes to the real-world after the execution of a task. There are three possible kinds of effects:

- **Supervised Effect (sEff)** – Updates to the values of user-input-related supervised criteria. For example, the collaboration mode (CM) supervised criteria value “New Supplier (NS)” is updated to “Existing Supplier with Contract (ESC)” when there is a transition from the Contract Agreement to Contract Fulfilment phase.
- **Unsupervised Effect (uEff)** – Necessary back-end changes to the values of unsupervised criteria used to signify transaction status change.
- **Document Effect (dEff)** – These depict the creation of new business documents (NewDocument), the updating or approval of a document (DocumentActedOn), and the archival of a document (DocumentArchived).

9.4.5 Organization-Defined Thresholds

These are quantitative *thresholds* that are defined subjectively by the company. An example is the definition of low cost. ‘Low cost’ to a large multi-national company may not be similar to a small business’ understanding of ‘low cost’. Organization-defined thresholds such as “*LowCostCeiling*” and “*MaximumLeadTime*” enable the decomposition algorithm to decompose and chain only tasks which are relevant to the defined thresholds of the company. An example will be a quotation received by a small company from a supplier. If the amount exceeds the small company’s definition of low-cost, the process would require a step which escalates the transaction for higher management decisions.

9.4.6 Primitive Actions

Primitive actions (PA’s) are not to be confused with primitive tasks. PA’s describe the most primitive, generic type of action a task will perform (e.g. receive, reply, etc). Hence, PA’s are

found only in primitive tasks. They are the feature closely linked to Web service execution level. As Web services are usually described by the input-behaviour-output properties, our proposed methodology links behaviour via PA's. Since primitive tasks are the leaf-node tasks which will be executed by Web services, the primitive activities will adopt the PA's described in WS-BPEL[30]: *Invoke, Receive, Reply, Assign, Validate, Throw, reThrow, Wait, Compensate, CompensateScope, Exit, Empty, Terminate*.

On top of these, the author understands that there will be tasks that require manual intervention even at the primitive levels. Hence, another “primitive action” at the primitive action level is the “*HumanTask*” PA, for tasks such as “*Approve Purchase Order*”. The “*HumanTask*” PA will then signal the need for a synchronous Web Service which requests and receive human decisions on the approval of purchase orders.

9.5 How Business Goals Are Decomposed Into Primitive Tasks

We now explain how the components of BOWL enable dynamic business process decomposition via a simple example of a Buyer trying to source for a New Supplier via the Reverse Auction (RA) sourcing mode. Figure 9.8 is a simplified overview of the decomposition process, starting from the high level process and goal “Buy”. The *Buy* compound task has a few methods and because the constraints of the user was for a ‘new supplier’ via ‘reverse auction’, method *Buy.B* was the matching method. *Buy.B* will contain guidelines for it to be decomposed by the algorithm into sub-tasks ‘*Discovery*’, ‘*Establish Contact*’ and ‘*Contract Agreement*’. The compound task ‘*Negotiation*’ was skipped because it was not required in a reverse auction mechanism.

Figure 9.8: Simplified overview of how BOWL tasks are decomposed to the Primitive Task level

The algorithm then moves deeper into the sequence of tasks, starting from ‘Discovery’. The decomposition sequence can also be found in the *Discovery* task in the earlier Figure 9.3. As discovery is yet another compound task, it decomposes with the method that has criteria matching the user constraints of NS and RA.

This resulted in *Discovery* being decomposed into ‘Create Tender Request’, and further into primitive tasks ‘Create Request for Tender’, ‘Send Request for Tender’ and then ‘Receive Request for Tender’ in a ‘SEQUENCE’. The primitive tasks will be recorded into the output abstract BPEL file with the appropriate primitive action equivalent for Web service invocation.

After which, the algorithm will back track and look for the next sub-task of *Discovery*, i.e. *Establish Contact*. This continues recursively until all the relevant primitive tasks are decomposed and chained according to the appropriate control flows.

More in-depth technical details of the algorithm will be discussed later in Chapter 10.

9.6 Technical Merits of BOWL

As standalones, HTN planning and ontologies are unable to fulfil the dynamic synthesis of business processes from goals and criteria. Hence, BOWL and Genesis hybridises the advantages of HTN planning and ontological descriptions to create the following key contributions:

- A reproducible technique of modelling cBP’s for dynamic business process formulation

- The 5 ECBP phases allow easier classification and modelling of HTN tasks, resulting in business processes which are process-oriented; more aligned to supply chain realities.
- Compared to traditional HTN planning, BOWL adds workflow control flow features by appending properties (e.g. “*hasSubTask*”, “*hasControlFlowPattern*”) in *Methods* to allow the Genesis algorithm to recognise the relevant control flow structure, decomposition and order of subtasks decomposed from a compound task.
- The *Supervised Criteria* and *Unsupervised Criteria* enable business constraints to be described and facilitate automatic execution of processes dependent on human input and decisions.
- *Methods* allow one to model cBP’s dynamically, and rather than the more restrictive traditional parent-child modelling found in current BPM standards. Methods are also customised to contain business goals so that there is a direct linkage to the decomposed tasks.
- Ontologies allow relationships between *Tasks*, *Methods*, *Actors*, *Effects*, *etc* to be described semantically and richly.
- At the primitive task level, *Primitive Actions* directly links of tasks to Web Services, enabling ease of execution.

9.7 Chapter Summary

To achieve goal-oriented, “top-down” dynamic formulation of cBP’s, a hybrid HTN-planning methodology consisting of two key ingredients is needed: (1) an extended HTN planning *algorithm* that decomposes real-life cBPs, and (2) the “*planning domain description*”, a *knowledge base* which efficiently stores:

- semantic relationships between compound and primitive tasks,
- permutations for decomposing each compound task,
- constructs for embedding control flows into output cBPs and govern decision routes taken during actual cBP executions
- B2B roles involved for each task.

This chapter discusses the rationale for and key research contributions of the *knowledge base*, known as the **Business-OWL (BOWL)**. Formerly, planning domain descriptions were modelled in text-based, Web-unready PDDL. BOWL, being written in the semantic Web language OWL, enables Web-friendly domain descriptions which bypasses cumbersome translations that traditional AI planners require in order to work in the WWW.

BOWL, by adopting the ECBP framework as its essential process, is *process-oriented*, rather than the rigid *functional perspective* of business processes commonly found in current B2B standards’ reference models. BOWL also models (1) nested control flows governing the tasks,

and (2) roles required to execute or make key decisions at each primitive task, thus upgrading the traditional HTN planning.

BOWL was created by synthesising common cBP tasks found in three dominant B2B information exchange standards' reference models: RosettaNet, ebXML and OAGIS into one master list. The common cBP tasks were also categorised into the five phases of the ECBP framework discussed in Chapter 8. In all, a total of 341 tasks are modelled as an extended HTN in BOWL.

The main features of BOWL include tasks (compound and primitive), methods (permutations of task decompositions), the actors of the tasks, the effects after executing a task, the subjective values stored as organisation-defined thresholds (e.g. definition of “high-price”), and the primitive actions which mirror the atomic actions performed by Web services orchestrated by BPEL. These features form the classes, properties and individuals found in a highly complex BOWL file, modelled using the Protégé OWL ontology editor. Some novel features such as the embedding of control flows around the task properties within the OWL file had to be entered manually.

The extended HTN planning algorithm which decomposes and chains tasks achieves dynamic cBP formulation process via referencing BOWL. The way dynamic decomposition and chaining of tasks is achieved will be discussed in the next chapter.

Chapter 10. Enhanced HTN Planning Algorithm for Dynamic cBP Formulation

This chapter proposes and presents Genesis⁴, an extended HTN planning algorithm which generates cross-enterprise business processes given business goals and constraints. Genesis' main purpose is to automatically decompose compound tasks from goals into granular tasks, and then chain them together in the appropriate control flows.

10.1 Genesis Algorithm in a Nutshell

```

1  procedure Genesis( B, {C})
2      Create empty F
3      Look at the topmost task, t' in B
4      If B is empty
5          Return empty F
6      Else
7          DecomposeAndChainTasks(t', B, {C}, F)
8          Return F
9      End If
10 end Genesis
11
12 procedure DecomposeAndChainTasks(t, B, {C}, F)
13     count number of methods in t
14     If there are no methods
15         Mark task t as primitive task for Web service execution
16         Extract actor of task t
17         Write t in F
18     Else
19         While there are unprocessed methods for t
20             Select the next method m of t
21             Check supervised criteria {msc} of m against user-input criteria {C}
22             If {msc} matches {C}
23                 check number of control flows in m
24                 While there are still control flows in m
25                     Read the outermost control flow ocf
26                     Write the start of the ocf in F
27                     For each subtask st in ocf
28                         DecomposeAndChainTask( st, B)
29                     Write the end of the ocf in F
30                 End While
31             End If
32         End While
33     End If
34 end DecomposeAndChainTasks

```

Figure 10.1: Simplified Genesis Procedure

Section 9.5 has given us a high level overview of the Genesis algorithm, and how it reads from the BOWL knowledge base. Figure 10.1 presents a simplified procedure for the Genesis algorithm. In the beginning, the algorithm takes in **two inputs**: (1) the *BOWL* (*B*) knowledge base and (2) a set of *user criteria* {*C*}. {*C*} is a container of **business goals**, & **supervised**

⁴ The name Genesis was chosen because Genesis, in biblical context, depicts the beginning of Creation, and the realisation of a Master Plan according to God. Similarly, the author envisions Genesis as the beginning of the creation of business processes.

criteria and **organisation-defined thresholds** (recall Section 9.4) collected by the prototype’s GUI (discussed later in Chapter 11).

With **B** and **{C}**, and starting from **the topmost compound task** in the loaded BOWL file (i.e. *B2BCollaboration*), the algorithm recursively decomposes the compound tasks in the HTN in **B** (Figure 10.1: Line 7). The decomposition process of each compound task (Figure 10.1: Lines 12-34) is described in the following section.

10.1.1 Task Decomposition and Sub-Task Chaining

Whenever the Genesis algorithm encounters a task, the algorithm will perform the following (cross refer with Figure 10.1, Figure 10.2 and Figure 10.3):

1. Scan for the number of methods found in the compound task under consideration. (Figure 10.1: Line 13)

Figure 10.2: How matching Method “EC.B” decomposes the compound task “EC” into sub-tasks chained by nested control flows.

2. **If there are no methods**, the current task is a **primitive task**. This primitive task is recorded into output file *F*. The actor of the task is also extracted. This is an important extension over the traditional HTN planning algorithm, which does not consider roles performing tasks decomposed. In this research, although *F* is written as an Abstract BPEL file, it can be any Web service orchestration language. (Figure 10.1: Line 14-17)

Finding the Matching Method for the Compound Task Under Consideration

3. **If more than one method are found**, the current task is a **compound task**. (Figure 10.1: Lines 18-33), the algorithm has to find the method which matches the user's input in $\{C\}$. BOWL is designed such that there is one and only one matching method for a set of $\{C\}$ at each compound task. This removes ambiguity in the real-time decision making by the algorithm. This is also a trait inherited from HTN Planning.

For each method of the currently considered compound task, the algorithm checks if it is the matching decomposition method by comparing the equivalence of its **criteria values** against each **relevant value in the user-input set $\{C\}$** as shown below in Figure 10.2.

The example shown in Figure 10.2 demonstrates how *method EC.B* requires the user criteria set $\{C\}$ to have the values $\{Business\ Goal\ (BG) == Sell, Collaboration\ Mode\ (CM) == New\ Business\ (NB)\ or\ Existing\ Business\ with\ Contract\ (EBW), Sourcing\ Mode\ (SM) == Traditional\ Sourcing\ (TS)\}$.

If $\{C\}$ has the same values for each supervised criteria, we have a matching method (Figure 10.1: Line 21-22). When the matching method is found, the algorithm will trigger the decomposition of the task under consideration into its sub-tasks and control flows via the matching method.

Decomposition of Compound Task

4. When the matching method is found, the algorithm will replace the compound task (e.g. EC) with its subtasks and their relevant control flows in output file *F*, thus decomposing the compound task into its subtasks. (Figure 10.1: Line 23-31)

The example in Figure 10.2 demonstrates the decomposition of task EC into its subtasks without neglecting the nested control flows recorded in the method's properties in the BOWL ontology. This is (1) in line with control flow structures found in real-life cBPs, and (2) an extension to the original HTN planning algorithm, as the original algorithm only considers single-level tasks, executed in a left-to-right sequence.

Chaining of Decomposed Sub-Tasks

5. The algorithm will then recursively extract the outermost control flow and write the relevant subtasks and nested control flows into **F**. The chaining of decomposed tasks into the relevant control flows is a unique contribution to the original HTN planning algorithm. (See decomposition diagram in Figure 10.2). (Figure 10.1: Line 21-22)

Recursive Decomposition of Tasks

6. For each sub-task (e.g. EC3, EC4, EC6, EC8), repeat Steps 1 to 5.
7. Recursively, these sub-tasks, chained together with the appropriate nested control flows, are marked into the **output file F**. This goes on until a nested control flow structure of primitive tasks (i.e. business process) is derived. An example of the output file F can be found in the next chapter in Figure 11.3.

10.1.2 How Genesis Algorithm traverses through BOWL's HTN

Figure 10.3: How the tasks are decomposed and chained during the Genesis algorithm execution

As shown in the top left of Figure 10.3, **Genesis** traverses through BOWL top down, decompose the hierarchical tasks recursively, and sequences primitive tasks across all five B2B collaboration phases.

The recursion will result in the algorithm reaching the leaf nodes, i.e. primitive tasks, that have supervised criteria conditions which match the user's {C}. This is denoted diagrammatically with the nodes marked 'X'. The top right of Figure 10.3 shows the resultant dynamically formulated business process (i.e. **output F**) required for the business goal and criteria.

10.2 Output and eventual Web service matching and execution

The final F , a nested control flow of primitive tasks, is an abstract BPEL file which enables us to easily “plug in” Web services for eventual business process execution. There are two types of BPEL files: *abstract* and *executable* BPEL. The former contains only *actors*, *task sequences*, *control flows* and *invocation points*, while the latter is a complete and executable BPEL file that includes implementation details such as *server locations*, *etc.* BPEL (in our case, WS-BPEL) was selected as the demo language as it is one of the industry’s most influential BPM languages [293].

While the generated abstract BPEL definitions are not ready for direct execution in engines, this “top-down” approach has significantly reduced the search space and computation efforts required for Web service matching and composition as compared to many “bottom-up” current techniques in SOA. This abstract BPEL serves an important skeleton for further appending of information by the author’s related research projects (in context-awareness, trust evaluation, partnerLink information, etc) [291]. Further details on the eventual business process execution can be found in [291].

10.3 Contributions of the Genesis Algorithm

The Genesis algorithm is universally applicable as a planner for BOWL’s designed for other domains and context. This means that BOWL’s can be designed and deployed for a context different from B2B Collaborations, and the Genesis algorithm referencing it will still work. One must also note that, BOWL (especially the methods within) was designed so that the Genesis Algorithm will always provide a solution/ plan. Traditional HTN planners which are not as specific as Genesis, may yield zero solutions after execution.

Together with BOWL, the Genesis algorithm gives us a goal-oriented, top-down approach to building processes as compared to bottom-up approaches such as Web service compositions and business process mining (i.e. mining business processes from audit trails). As mentioned in Section 5.1.4, a top-down approach will reduce the computational and search space of the possible Web services, as the scope of matching Web services will be greatly reduced when we decompose into the specific, primitive tasks.

The Genesis algorithm considers the planning of nested control flows when it is decomposing compound tasks. This is a significant extension to overcome the limitations of traditional HTN planning algorithms, which assume that tasks in output plans are executed via a single-level left-to-right sequence of tasks.

The Genesis algorithm also considers roles for decomposed and chained tasks. Roles are generically grouped into three main types: Buyer role, Seller role and Web service role. This is because tasks typically can either be automated by computer systems such as Web services, or require decision making or updates from human roles from both Buyer companies and Seller companies. The embedding of roles into each primitive task in the final output depicts real-life cross-enterprise business processes, and extends the traditional HTN planning algorithm which did not consider task roles for plans generated.

The achievement of the Genesis algorithm also enables us to bridge high-level goals with low-level primitive tasks. Not only are the low-level primitive tasks chained in the appropriate control flows, they can also be easily appended by executable Web services for the business process to be executed. To the author's best knowledge, this has not been achieved dynamically before.

10.4 Chapter Summary

This chapter introduces how the Genesis algorithm decomposes and chains tasks into cBPs during its execution. The algorithm collects the user goals and constraints and starts referencing BOWL from the top-most task, i.e. *B2BCollaboration*, which can be decomposed into *Buy*-scenario tasks and also *Sell*-goal scenario tasks.

Task decomposition is recursively performed at each compound task deeper and deeper into lower levels of the hierarchy of tasks. When the supervised criteria embedded in *methods of the compound task under consideration* matches (in equivalence) the user-expressed goals and criteria, the compound tasks are then decomposed into primitive tasks which are then chained by nested control flow constructs. At the same time, roles are also embedded into each primitive task.

Eventually, the Genesis algorithm generates a cBP containing nested control flows of primitive tasks which can then be easily executed by Web services or even agents. The cBP tasks achieve the desired business goal.

The Genesis algorithm overcomes the limitations of traditional HTN planning algorithms, which cannot dynamically generate cBPs as they (1) do not consider actors of tasks, and (2) unable to deal with decision-making routes which are frequently encountered in cBP.

The following chapter discusses how the Genesis algorithm and the BOWL knowledge base realise dynamic cBP formulation in a service-oriented environment.

Chapter 11. Implementation, Verification and Validation

To demonstrate how the research questions in Chapter 7 were resolved, and how the proposed hybrid approach in Chapters 8, 9 and 10 realise dynamic cBP, the author implemented BOWL and the Genesis algorithm as a Java Web application.

This chapter describes the details of the implementation of both the prototype and the BOWL ontology, culminating in the verification and validation of the prototype against industry reference models and best practices in industry. Let us begin by reviewing the architecture of the Genesis prototype.

11.1 Implementing the Genesis System

11.1.1 Architecture of the Genesis System

As shown in Figure 11.1, the Genesis prototype [172] consists of:

1. **Client- A Web Graphical User Interface (GUI)** – a JSP (Java Server Pages) page which interacts with the user and passes information to and from the server (see Figure 11.2).
2. **Server** – The other two components of *Genesis* are housed within a server:
 - a. **Genesis Algorithm** – housed within a Java Servlet
 - b. **Business-OWL (BOWL)** – a knowledge base referenced by the Java Servlet, located within the server. There are 2 BOWL representations: (1) a **Visio file** which was used by the author to graphically design and plan the decomposition structure of tasks (see Appendix H), and (2) the actual **modified OWL file** edited via Protégé (OWL editor) (see Appendix I).

As shown in the top left hand corner of Figure 11.1, the Genesis GUI takes in the user's (buyer's or seller's) business goals, the underlying criteria (e.g. product name, quantity ordered, price per unit, etc) and company-defined thresholds (e.g. definition of "low-cost") via a form shown in Figure 11.2. These user inputs will be passed from the Web browser to the Java Servlet (Step 1 in Figure 11.1) hosted on an Apache Tomcat server.

Genesis' users can be both suppliers and buyers. It is intended to generate the relevant business process for their collaboration with their buyers and suppliers respectively. Starting from the **topmost task** in BOWL (i.e. B2Bcollaboration), the inputs (*BUY/SELL and set of constraints shown in Figure 11.2*) are recursively compared with the relevant *supervised criteria* of methods of tasks found in BOWL (**Step 2** in Figure 11.1), and the correct control flow of all tasks required to fulfil the input goals and criteria is generated.

Figure 11.1: Architecture of Genesis Prototype

The screenshot shows the Genesis prototype input GUI. The interface includes a header with the 'GENESIS' logo and navigation links: '-|- Generate cBPs -|- Annotated BPEL File -|- Ontologised HTN'. A sidebar on the left lists 'Latest Updates' and 'Some additional features:'. The main content area is divided into several sections:

- User Company Detail:** Includes fields for Company Name (Acme Inc.), Address (Street: Jurong West St. 61, Postal Code: 640625, Country: Singapore).
- Supervised Criteria:** Includes a Business Goal dropdown (Select Buy or Sell), Product Type (Goods/Services), Product Name (Pallet), Quantity (500), Item Price (2), Lead Time (1), Process Type (Manufacturing/Service Provision), Sourcing Mode (Reverse Auction/Traditional Sourcing), and Collaboration Type (Please select the Business Goal first).
- Company Defined Threshold:** Includes Low Cost Ceiling (1200) and Maximum Lead Time (2).

Annotations on the screenshot:

- A red box highlights the 'Business Goals' section, which includes the 'Supervised Criteria' and 'Company Defined Threshold' sections.
- A green box highlights the 'Constraints' section, which includes the 'Supervised Criteria' and 'Company Defined Threshold' sections.
- A red box highlights the 'Company Defined Threshold' section.
- A green box highlights the 'Company Defined Threshold' section.

Buttons at the bottom include 'Reset!' and 'Generate Collaborative Business Process'.

Figure 11.2: Screenshot of Genesis prototype input GUI, and the components included in the GUI

We use the *dom4j* API (www.dom4j.org/) to parse the XML-based BOWL, and for writing relevant tasks into an abstract BPEL file. The detailed Java codes for the Genesis system can be found in Appendix J.

11.1.2 Types of Outputs of Genesis System

After the algorithm decomposes tasks in BOWL according to the user goals and criteria, three types of dynamically generated outputs are generated for the Genesis GUI:

11.1.2.1 Annotated Abstract BPEL for Eventual SOA Implementation

The first output of Genesis is an annotated abstract BPEL file. Put simply, the *abstract BPEL file* contains the relevant control flows, task sequences and Web service requirements without the execution details (e.g. exact server locations hosting each service). In other words, it can also be described as containing the blue-print (or in AI planning, the “plan”) for the tasks required, and the way they are linked for meaningful execution.

```
<process name="BPEL Process" targetNamespace="http://imss.simtech.a-star.edu.sg/Genesis/B-OWL"
  suppressJoinFailure="yes">
  <ontcaf:collaboratorsFile location="http://localhost:8081/Genesis/Files/ontcaf.xml"/>
  <partnerLinks>
    <partnerLink name="#BuyerActor" myRole="Buying" partnerRole="Selling" partnerLinkType=""/>
    <partnerLink name="#SellerActor" myRole="Selling" partnerRole="Buying" partnerLinkType=""/>
    <partnerLink name="#WebServiceActor" myRole="" partnerRole="" partnerLinkType=""/>
  </partnerLinks>
  <sequence>
    <sequence>
      <sequence>
        <flow>
          <sequence>
            <invoke name="#D1.1.1- ContactReferrals" partnerLink="#WebServiceActor or
              #BuyerActor"/>
            <invoke name="#D1.1.2- ObtainPerspectivePartnersContactInfo"
              partnerLink="#WebServiceActor or #BuyerActor"/>
          </sequence>
          <invoke name="#D1.2- SearchBizDirectory" partnerLink="#WebServiceActor or
            #BuyerActor"/>
          <invoke name="#D1.3- SearchInternet" partnerLink="#WebServiceActor or
            #BuyerActor"/>
          <invoke name="#D1.4- VisitingTradeShows" partnerLink="#BuyerActor"/>
        </sequence>
        <sequence>
          <invoke name="#D1.5.2- ReceiveProductCatalogInfo"

```

Figure 11.3: Screen Capture of the Abstract BPEL output generated by Genesis

As shown in Figure 11.3 above, the abstract BPEL file contains three roles recorded as “PartnerLinks”: “BuyerActor”, “SellerActor” and “WebServiceActor”. This is important, as this provides the declaration for the roles required at each task. Some tasks which can be automated by Web services will be labelled as “partnerLink=WebServiceActor”. This is a temporary placement information for the relevant role or server to look for the actor and scope down on the search space for relevant services, as the abstract BPEL file is expected to be augmented or overwritten with more exact implementation details to form an executable BPEL file.

Figure 11.3 also shows how control flows are embedded in a nested way in the BPEL file. In the middle of the figure, a sequence of tasks (“D1.1.1- ContactReferrals”, “D1.1.2-Obtain PerspectivePartnersContactInfo”, “D1.2-SearchBizDirectory”, and so on) were embedded in the “Flow” control flow tag. This means that this sequence of tasks will be executed in parallel with yet another sequence embedded in the “Flow” tag further down the file.

The abstract BPEL file shows how the tasks necessary for fulfilling the expressed goals and constraint will be executed, but a graphical overview of the breakdown of the required tasks will be better for debugging and verification.

11.1.2.2 Displaying Nesting of Control Flows and Parent-Child Task Relations

Figure 11.4: Text based outputs of the Genesis Prototype

Hence, besides the Abstract BPEL file, there is also a graphical *tree structure breakdown* of the dynamically generated required business process tasks in a hierarchical tree diagram (see Figure 11.5). As shown in Figure 11.5, the user has a choice between a Full Tree Diagram or just a List of Primitive Tasks, both generated via JavaScript.

In the Full Tree Diagram (see Figure 11.5), the order of tasks, roles (e.g. [WS/B], [B], etc) and the control flows (shown in parentheses, e.g. #Flow, #Sequence, etc.) are all displayed succinctly. The right-most tasks are the primitive tasks and are hence embedded with information about the executor of the task (e.g. WS (i.e. Web Service), B (Buyer) or S (Seller)).

In the List of Primitive Task (see Figure 11.6), only the primitive tasks required for the scenario are listed. Each primitive task in the list has an accompanying hyperlink to the OWL file describing the semantic descriptions of the Web service required. OWL was chosen over the commonly used WSDL as WSDL is void of the semantic descriptions needed for the prototype's related module in the Ont-CAF project.

In conformance with good software engineering practices, a progress bar was coded (see Figure 11.7) to show the decomposition progress and the tasks decomposed during the execution of the algorithm so that the user waiting time (approximately two to three minutes) is tolerable. During the writing of the abstract BPEL file and the preparation for the tree structure, the Genesis prototype also logs down the recursive decomposition process in the form of a text-based event log.

Figure 11.5: Screen Capture of the Tree structure output of Genesis

Figure 11.6: List of Primitive Tasks (i.e. Web Services) Required to Fulfill the User's Goals and Constraints

Figure 11.7: Progress bar to reduce user waiting frustration

11.1.2.3 Graphical Display For Visual Inspection of Generated Decision Routes

While the tree structure may be sufficient for some users, it is rather inadequate for complex process flows involving intricate control flows and multiple levels of task. A graphical display is a better alternative.. Hence, there is a third output option - a **graphical display** - of the control flows of cBP's tasks (see Figure 11.12).

The author made use of GraphViz to draw the graph based on the information from a *.DOT* file. GraphViz is a package of open source tools initiated by AT&T Research Labs for drawing graphs specified in the *.DOT* language script. The software is licensed under the Common Public License. It was chosen after an evaluation against related graph visualisation tools like *Prefuse*, *yFILES*, *aiSee* and *Udraw*. The *.DOT* language is a plain text graph-description language. It is a simple way of describing graphs that both humans and computer programs can use. GraphViz runs as a program on server, translating the *.DOT* file into a graphical diagram.

Since the annotated abstract BPEL file has an XML format, in order to visualize the BPEL file, we need to translate the data into a *.DOT* file for GraphViz. Here, the author used the *XLST* tool to transform the BPEL output into the DOT format. After the Genesis algorithm generates the BPEL output file, the file is automatically translated into the DOT version for visualization.

Figure 11.8: Overall process of generating graph from the BPEL file

While GraphViz is able to generate graphs from *.DOT* files on-the-fly, it has a limited set of symbols relative needed to annotate business processes. Hence, there is a need for workarounds when expressing control flows.

Translating Abstract BPEL Control Flow Tags to GraphViz Notations

Currently, the syntax of BPEL file contains four main types of control flow tags: *<sequence>*, *<flow>*, *<pick>* and *<invoke>*. The three tags *<sequence>*, *<flow>* and *<pick>* define the

type of control flow. The `<invoke>` tag contains the name of primitive task. The GraphViz workarounds for each tag are summarized in Table 11.1.

Table 11.1: A summary of GraphViz notations for current control flow tags in the generated Abstract BPEL

Tags	Example	Description	Graph model
invoke	<code><invoke name=T1 /></code>	Execute the task T1
sequence	<code><sequence></code> <code><invoke name=T1 /></code> <code><invoke name=T2 /></code> <code><invoke name=T3 /></code> <code></sequence></code>	Execute all the tasks T1, T2 and T3 in the given order.
flow	<code><flow></code> <code><invoke name=T1 /></code> <code><invoke name=T2 /></code> <code><invoke name=T3 /></code> <code></flow></code>	Execute all the tasks T1, T2 and T3 concurrently.
pick	<code><pick></code> <code><invoke name=T1 /></code> <code><invoke name=T2 /></code> <code><invoke name=T3 /></code> <code></pick></code>	Execute only one task from T1, T2 and T3.

Workarounds for Task Name Labels

Each primitive task is presented as a node, which is an oval shape that has the name of task inside. The task's name originally has the format like *"EC2.1.1-QueryMarketInfo"*. However, since GraphViz cannot recognize the "." and "-" characters, the author had to change the format of the task's name into *"EC2_1_1_QueryMarketInfo"* for display.

Workarounds for Pick and Flow Control Flows

The arrows in the graph show the type of control flow of tasks to be executed. There are two types of arrows: the solid line ($\longrightarrow$) and the dash line ($--\longrightarrow$) arrows. Since the fundamental flow of graph shape of the *pick* and *flow* tags are identical (i.e. both graphically shows a split into parallel tasks and an eventual merge into a single task), the dash line arrow is used to distinguish the *pick* control flow.

Workarounds for Graphical Flows of Split and Merge Workflow Patterns

As shown in a simplified example in Figure 11.9, in situations when there is a split or a merge, GraphViz would display all possible paths of each individual node. This is undesirable in the context of business process modelling, and would result in very complicated sets of arcs in a highly nested business process. Hence, the author used implemented workaround codes using XSLT tools to modify the .DOT file to depict the resulting workflows like the one on the right side of Figure 11.9.

Figure 11.9: Overcoming control flow representation limitations of GraphViz

The tail of the Flow of T1 and T2 is also the head of the Flow of T3, T4 and T5. Hence, there is a need to denote this overlap as a ‘black dot’ as a workaround to depict both the tail and the head of the different control flows.

Workarounds for Recursive Nesting of Task Control Flows

Another problem resolved was the unpredictability of the number of levels of nested tags in each generated annotated Abstract BPEL. The author designed a solution by using a recursive method to traverse all the levels without the need to know how deep the tag-structure is. This recursive code is not the top priority of the research, but can be further referenced as BPELtoDOT.xsl in Appendix K.

Corresponding Abstract BPEL, DOT file and the GraphViz output

Figure 11.10’s corresponding .dot file and the GraphViz output are found in Figure 11.11 and Figure 11.12 respectively.


```

<?xml version="1.0" encoding="UTF-8" ?>
- <process xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/" xmlns:ontcaf="http://www.ntu.edu.sg/ontcaf" name="BPEL Process"
  targetNamespace="http://imss.simtech.a-star.edu.sg/Genesis/B-OWL" suppressJoinFailure="yes" xmlns:tns="http://www.ntu.edu.sg">
  <ontcaf:collaboratorsFile location="http://localhost:8081/Genesis/Files/ontcaf.xml" />
  <ontcaf:servicesFile location="http://localhost:8081/Genesis/Files/services.xml" />
  <partnerLinks>
    <partnerLink name="#BuyerActor" myRole="Buying" partnerRole="Selling" partnerLinkType="" />
    <partnerLink name="#SellerActor" myRole="Selling" partnerRole="Buying" partnerLinkType="" />
    <partnerLink name="#WebServiceActor" myRole="" partnerRole="" partnerLinkType="" />
  </partnerLinks>
  <sequence>
    <sequence>
      <sequence>
        <sequence>
          <Invoke name="D3_1_CreateRequestForTender" partnerLink="#WebServiceActor or #BuyerActor" />
          <Invoke name="D3_2_SendRequestForTender" partnerLink="#WebServiceActor or #BuyerActor" />
        </sequence>
      </sequence>
    </sequence>
    <sequence>
      <pick>
        <Invoke name="EC11_1_TenderTermsMet" partnerLink="#BuyerActor" />
        <Terminate name="EC11_2_FailedTender" partnerLink="#BuyerActor" />
      </pick>
    </sequence>
    <sequence>
      <sequence>
        <HumanTask name="CA2_ChooseWinningTender" partnerLink="#BuyerActor" />
        <Invoke name="CA3_NotifyTenderResult" partnerLink="#WebServiceActor or #BuyerActor" />
        <Invoke name="CA5_AwardContract" partnerLink="#BuyerActor" />
      </sequence>
      <sequence>
        <HumanTask name="CA2_ChooseWinningTender" partnerLink="#BuyerActor" />
        <Invoke name="CA3_NotifyTenderResult" partnerLink="#WebServiceActor or #BuyerActor" />
        <Terminate name="CA6_TenderRejection" partnerLink="#BuyerActor" />
      </sequence>
    </pick>
  </sequence>
</process>

```

Figure 11.10: Sample Abstract BPEL file segment which corresponds to the .dot and the graph figures below

```

2  digraph {
3      start ->D3_1_CreateRequestForTender;
4      D3_1_CreateRequestForTender ->D3_2_SendRequestForTender;
5      D3_2_SendRequestForTender -> {EC11_1_TenderTermsMet;EC11_2_FailedTender;}
6                                  [style=dotted];
7      {EC11_1_TenderTermsMet;EC11_2_FailedTender;}->EC11_2_FailedTenderconverge;
8      EC11_2_FailedTenderconverge[shape=point];
9      EC11_2_FailedTenderconverge->
10                                     {CA2_ChooseWinningTender;CA2_ChooseWinningTender;}
11                                     [style=dotted];
12      CA2_ChooseWinningTender->CA3_NotifyTenderResult;
13      CA3_NotifyTenderResult->CA5_AwardContract;
14      CA2_ChooseWinningTender->CA3_NotifyTenderResult;
15      CA3_NotifyTenderResult->CA6_TenderRejection;
16      {CA5_AwardContract; CA6_TenderRejection;}
17                                     ->converge;
18                                     converge[shape=point];
19                                     converge
20      ->end
21  }
22

```

Figure 11.11: Corresponding .dot file from the Abstract BPEL

In the example in Figure 11.12, if the Tender fails (*EC11-2*), the business process transaction traverses through the dot stage and then follows the path on the right: *CA2* (choosing another winning tender), *CA3* (notifying the tender result to ALL tenderers), and then *CA6* (updating the status of the seller whose tender was rejected). If the Tender terms are met (*EC11-1*), the transaction should take the path on the left side (*i.e.* *CA2*, *CA3*, *CA5*).

Figure 11.12: Graphical visual output generated by Genesis using Graphviz tools, corresponding to the earlier Abstract BPEL and .dot file diagrams

11.2 Adapting to Different Sets of Goals and Criteria

This section focuses on demonstrating how Genesis adapts to different buying goals and business criteria by generating for each set of goals and criteria a unique set of decomposed and chained tasks. Due to some criteria accepting strings as inputs (e.g. Product Name), there can be countless scenarios of business goals and criteria which a user can input. Hence for demonstration, the author chose two very different made-up scenarios summarised in Table 11.2:

1. The first sample displays a scenario where a **manufacturer sources** and **purchases ball bearings** from a **new supplier** via **traditional sourcing** tasks.
2. The second sample, on the contrary, displays how a **construction company** outsources its **scaffolding works** to a new **sub-contractor** via **reverse auctions** (i.e. tendering).

For each scenario, *Genesis* can dynamically generate specific sequences of tasks which can enable the users to fulfil their goals. (Note: More details on the stress testing for branch coverage of the prototype will be introduced later in Section 11.4)

Table 11.2: Business Goals and Criteria of 2 Selected Sample Scenarios

Sample Scenario 1: Buying via Sourcing (Bulk purchase of ball bearings from a new supplier)	
Business Goal	BUY
Business Criteria	
• Product Type	Ball Bearings
• Product Name	Ball Bearing Model 100
• Quantity	10 000
• Item Price	USD 1.00
• Lead Time	10 workweeks
• Process Type	Goods and Manufacturing
• Sourcing Mode	Traditional Sourcing
• Collaboration Mode	New Supplier
Sample Scenario 2: Buying via Reverse Auction (Requesting for tenders from sub-contractors to perform building construction support works)	
Business Goal	BUY
Business Criteria	
• Product Type	Construction Works
• Product Name	Scaffolding Setup
• Quantity	10
• Item Price	USD 20 000
• Lead Time	10 workweeks
• Process Type	Goods and Service Provision
• Sourcing Mode	Reverse Auction
• Collaboration Mode	New Supplier

```

#B2BCollaboration (#Sequence)
#B-Buy (#Sequence)
#D-Discovery (#Sequence)
#D1-SearchForPotentialSuppliers (#Flow)
#D1.1-ViaReferrals (#Sequence)
#D1.1.1-ContactReferrals
#D1.1.2-ObtainPerspectivePartnersContactInfo
#D1.2-SearchBizDirectory
#D1.3-SearchInternet
#D1.4-VisitingTradeShows
#D1.5-ProductCatalogInfo (#Sequence)
#D1.5.2-ReceiveProductCatalogInfo
#D4-PotentialPartnerListFound
#EC-EstablishContact (#Sequence)
#Flow
#Sequence
#EC1-DecideSellersToContact
#EC2-ContactSellers (#Sequence)
#EC2.1-QueryOrderCostInfoRequest (#Flow)
#EC2.1.1-QueryMarketingInfo
#EC2.1.2-QuerySalesPromotionAndRebateInfo
#EC2.1.3-RequestPriceAndAvailability
#EC2.2-QueryProductInfoRequest (#FlowEnd)
#EC5-EvaluateSupplierResponses
#SequenceEnd
#EC6-PersonalVisit
#FlowEnd
#EC7-RequestForQuote (#Sequence)
#EC7.1-SendRequestForQuote
#EC7.2-ReceiveQuote
#EC7.3-ChangeQuote
#EC7.4-NotifyOfQuoteAcknowledgement
#N-Negotiation (#Sequence)

```

... (217 more tasks)

Figure 11.13: Snippet of resulting tasks in Scenario 1's cBP

Figure 11.13 and Figure 11.14 show snippets of the list of tasks required for both Scenarios 1 and 2 respectively.

For the traditional sourcing scenario, *B2BCollaboration* task is decomposed into *Buy* task, which further decomposes into the *Discovery* tasks for sourcing of new suppliers, the *Establish Contact* tasks and followed by *Negotiation* tasks (See Figure 11.13). Control flows are also written into the output (e.g. *Sequence*, *Flow*, etc) encompassing the sub-tasks belonging to a particular compound task. For example, compound task “*D1-SearchForPotentialSupplier*” has the *FLOW* control flow for its subtasks *D1.1*, *D1.2*, *D1.3*, *D1.4* and *D1.5*. This means that any of the *D1*'s subtasks' execution will fire the business process flow to the next task, “*D4-PotentialPartnerListFound*”.

On the other hand, the *Reverse Auction* scenario in Figure 11.14 yielded a different set of tasks as output. Tasks for *creating tenders*, *sending tenders* are now found in *Discovery* task and *Establish Contact* tasks instead of sourcing-related tasks.

```
#B2BCollaboration (#Sequence)
#B-Buy (#Sequence)
#D-Discovery (#Sequence)
#D3-CreateTenderRequest (#Sequence)
#D3.1-CreateRequestForTender
#D3.2-SendRequestForTender
#EC-EstablishContact (#Sequence)
#EC11-EvaluateTenders (#Pick)
#EC11.1-TenderTermsMet
#EC11.2-FailedTender
#CA-ContractAgreement (#Sequence)
#CA2-ChooseWinningTender
#CA3-NotifyTenderResult
#CA5-AwardContract
#CA7-AccountSetup (#Sequence)
#CA7.1-RequestAccountSetup
#CF-ContractFulfillment (#Flow)
... 108 more tasks...
```

Figure 11.14: Snippet of resulting tasks in Scenario 2's cBP

11.3 Successful Application into A*STAR IMSS Ont-CAF Project

The Genesis prototype was also successfully implemented and integrated into the Ont-CAF project, which aims to develop a low cost, Web-service based methodology and system that facilitates dynamic collaborations amongst companies (recall Section 5.5.5). To achieve its goal, the Ont-CAF project is logically divided into three stages:

1. Formulation of abstract business processes from business goals

2. Matching and selection of suitable Web services for each task in the abstract business processes, via ontology matching algorithms with consideration of context, and level of trust for the candidate Web services.
3. Mapping and composition of Web services.

The Genesis prototype addressed Stage 1, i.e. generate abstract business process definitions in an augmented abstract BPEL format given a set of user goals and constraints. The Genesis input GUI was also deployed as the front-end of the Ont-CAF demo system. The augmented BPEL file contains annotations which assist the steps in Stage 2 and 3. The integrated Ont-CAF project was successfully completed in February 2010, and its full implementation demo, known as Atlas, can be found in <http://imss.simtech.a-star.edu.sg/Atlas/> (*Login = imss; password = simtech*).

More importantly, the integration with the Ont-CAF project has further proven the capabilities of the Genesis system in dynamically formulating cBPs given some goals and constraints.

11.4 Prototype Verification Procedure and Results

Naturally, there are no tools for automatic verification of a dynamically generated cBP from Genesis. This also means a need for manual verification. The manual verification has to cover three main areas:

1. Correctness of the structure of BOWL
 - a. Tasks are in the appropriate phases
 - b. Compound tasks can be decomposed by at least one method
 - c. There should be no situation where a set of inputs will not be processed by a method of a compound task.
2. Correctness of the generated cBP via the Genesis algorithm in the prototype
 - a. Tasks are correctly decomposed and included into the generated cBP
 - b. Correct routing of business tasks in the generated cBP
 - c. Appropriate control flow patterns generated
3. The Genesis algorithm is able to provide a solution, covering all possible paths of the BOWL, given all possible input scenarios.

There are 341 tasks (both compound and primitive), and 174 methods to be verified. While the verification has to be executed manually, there is a need for a more efficient and effective method to verify the correctness of the system.

With respect to this, the author decided to adopt *path testing* [294] techniques from software engineering best practices. In path testing, all possible sequences of steps executable by the software are paths. This concept of paths is akin to the recursive task decomposition process of

the Genesis algorithm. Many of the paths overlap at certain stages of execution, and hence, there exists a minimum number of paths that can be tested to cover all the implementation routes of the software. This results in a highly efficient method for verification of correctness of all possible routes within the software's execution, as shown in the following section.

11.4.1 *Path Testing*

To test the robustness of Genesis, there is a need to stress test the Genesis system. Typical software stress testing for mission-critical software usually requires both (1) branch coverage and (2) load testing on the software.

In our case, *path testing* allows the author to verify full branch coverage of both the algorithm and BOWL. Achieving 100% branch coverage means that every branch in a program (and its knowledge base) has been executed at least once under the designed test scenarios. On top of that, the author also tested the system for *fault tolerance* by testing it against ‘illegal’ parameters (i.e. mismatched data types, input values outside range).

However, *load testing* will not be conducted in this thesis, as this performance testing has to be executed by a large number of users and large volume of concurrent transactions. There are three reasons for not applying load testing:

- It is not achievable without the support from industry consortiums.
- Achieving load testing does not add mileage to demonstrate the achievement of the research objective, which is to achieve dynamic cBP formulation given some high level goals and constraints.
- Load testing is only applicable and needed when the prototype becomes commercialised.

Hence, for verification, the author focused on achieving full branch coverage via path testing and fault tolerance tests.

11.4.1.1 *Reducing Factors*

The HTN of tasks and their related methods in BOWL form a massive network of complex paths. However, on closer observation, some subsets of the different paths overlap. Hence, knowing the maximum number of scenarios to test is a fruitless affair, but knowing the minimum number of scenarios to effectively cover the whole of the HTN and also test the methods would be both effective and efficient. In other words, full branch coverage can still be achieved by a realistic yet minimum number of scenarios.

To get this minimum number of scenarios, we need to recall the eight planning criteria/constraints discussed in Section 9.4.2.2 (e.g. CM, SM, PType, Proc, etc.). Each of these

constraints has its own range of values and a resulting large number of permutations when considered together. To reduce the number of permutations, there is a need to merge and prioritize some of the 8 planning criteria, i.e. reducing the number of factors considered.

One of the ways to reduce the number of permutations is to reduce the number of factors. Factors (in our case, the business constraints) that are ‘implicitly similar’ can be treated as being the same.

For example, the author realised that Business Goal (Buy and Sell) implicitly represented by Collaboration Mode. For example, “*CM = New Supplier*” represents “*BG = Buy*” or in other words, the Buyer is looking for new supplier. When the goal is “*Buy*”, the Collaboration Mode has to be one of the “*Supplier*”-related ones. The same can be said for the “*Sell*” goal.

Also, the author has found that criteria “*Product Type*” and “*Process Type*” are related by three relationships shown in Figure 11.15. As such, tasks for an organization that does “*Manufacturing*” and offers “*Services*” are not possible (i.e. you cannot manufacture services *per se*), and hence unavailable in BOWL.

Figure 11.15: Relationships between Product Type and Process Type

Lastly, the criteria “*Price*” and “*Quantity*” are always considered as a pair. The total cost derived from the price multiplied quantity values is often the bottom-line consideration for businesses. Thus, they can be merged as one criterion (i.e. one factor).

11.4.1.2 Minimum Scenarios Needed For Complete Branch Coverage

After having reduced the number of factors, the author was able to minimise the possible scenarios into 14 comprehensive scenarios which will cover all paths traversed in BOWL by the Genesis algorithm.

Table 11.3 shows the 14 minimum scenarios derived to ensure (1) the correctness of the structure of BOWL, (2) the correctness of the generated cBP via the Genesis prototype, and (3) that the Genesis algorithm is able to provide a solution, covering all possible paths of the

BOWL, given all possible scenarios of user input. These 14 scenarios also result in a 100% branch coverage testing for the system, i.e. fully verified.

Table 11.3: 14 scenarios that can verify BOWL and Genesis via path testing

Scenario	1	2	3	4	5	6	7
CM	NS	NS	NS	NS	NS	NS	NB
PType and Proc	Svc and SP	Svc and SP	Gds and SP	Gds and SP	Gds and Mfg	Gds and Mfg	Svc and SP
SM	TS	RA	TS	RA	TS	RA	TS
BG	According to CM	According to CM	According to CM	According to CM	According to CM	According to CM	According to CM
Pname	Product A	Product A	Product A	Product A	Product A	Product A	Product A
Qty x Pr	1000	1000	1000	1000	1000	1000	1000
LT	10ww	10ww	10ww	10ww	10ww	10ww	10ww
Scenario	8	9	10	11	12	13	14
CM	NB	NB	NB	NB	NB	ESW	EBW
Ptype and Proc	Svc and SP	Gds and SP	Gds and SP	Gds and Mfg	Gds and Mfg	Gds and Mfg	Gds and Mfg
SM	RA	TS	RA	TS	RA	TS	TS
BG	According to CM	According to CM	According to CM	According to CM	According to CM	According to CM	According to CM
Pname	Product A	Product A	Product A	Product A	Product A	Semiconductor	Semiconductor
Qty x Pr	1000	1000	1000	1000	1000	1000	1000
LT	10ww	10ww	10ww	10ww	10ww	10ww	10ww

11.4.1.3 Path Testing Verification Process

With the 14 scenarios, the verification process was executed in the following order:

1. Generating the cBPs from the inputs stipulated by each of the 14 scenarios, and checking if there are any discrepancies against the RosettNet, ebXML, and OAGIS industry reference models.
2. The author exhaustively traced through the Visio HTN diagram, and the BOWL ontology file, to ensure that all tasks and methods have been accounted for. For all 14 scenarios, the results were tabulated using a check list for all tasks and all methods (screen capture in Figure 11.16 and full lists found in Appendix L. A3-sized print-outs showing examples of the manual tracing of task decompositions in BOWL (**bold** = “task traversed” and grey = “task skipped”) can be found in Appendix M. The author made sure that no boxes were left “grey” at the end of testing all the 14 scenarios. In other words, every task or method **must** be passed by at least 1 scenario to be verified.

No.	Methods	Scenario														No.ofScenarios	Verified in Visio
		1	2	3	4	5	6	7	8	9	10	11	12	13	14		
1	MethodB2B.A	✓	✓	✓	✓	✓	✓									6	✓
2	MethodB2B.B							✓	✓	✓	✓	✓	✓			6	✓
3	MethodBuy.A	✓		✓		✓										3	✓
4	MethodBuy.B		✓		✓		✓									3	✓
5	MethodBuy.C													✓		1	✓
6	MethodBuy.D	✓		✓		✓										3	✓
7	MethodSell.A							✓		✓		✓				3	✓
8	MethodSell.B								✓		✓		✓			3	✓
9	MethodSell.C														✓	1	✓
10	MethodSell.D							✓		✓		✓				3	✓

Figure 11.16: Screen capture of part of the *Method* checklist

3. Edit and debug BOWL file accordingly.

Any mistakes spotted and changes are remedied in both the Visio format and the BOWL file via Protégé, an ontology editor. Manual updates are necessary because there are currently no tools available for updating such changes.

4. **Lastly, match the manually traced paths (in the Visio version of BOWL) with the Genesis generated cBPs for each scenario.** This checks the robustness of the algorithm. More details are found in the check list in Appendix L.

As a result of this process, **all 333 tasks and 174 methods** have been tested and were verified by 14 generated scenarios, achieving full branch coverage.

11.4.1.4 Fault Tolerance Tests and Limitations Remedied

On top of path testing, the author also tested the robustness of the prototype by fault tolerance tests. Fault tolerance testing is performed by entering illegal values into each parameter in the GUI. Table 11.4 shows the summary of types of illegal input entered. The values entered depend on the type of GUI control (e.g. radio buttons, textboxes).

This testing stretches the system, and exposed the need for a “runtime validation” function. The “runtime validation” of input data on the GUI checks for erroneous input and is done using JavaScript. Using JavaScript, the input page can be coded to adjust the parameters directly to the user’s selection in some fields. For example, there should not be any combination of *Product type: Services* together with *Process Type: Manufacturing* because logically, *Services* cannot be manufactured. When the user chooses *Services* as the product type, the JavaScript automatically adjusts by disabling the *Process Type: Manufacturing* and automatically choose *Service Provision* as the process type.

After the user tries to commit the input by clicking on the “Generate Collaborative Business Process” submit button at the bottom of the GUI page, the JavaScript will trigger a “runtime

validation” of the user inputs keyed in for erroneous input (i.e. wrong type, out of range). Window pop-outs alerting the user of the specific erroneous input will be triggered upon detection of error during the “runtime validation”. Such a feature acts as a gatekeeper to reduce parameter input errors and ensures that the inputs passed into the algorithm to be of the legal ranges and types.

Table 11.4: Input parameters and illegal input entered for robustness testing

No.	Input Parameter	Type of GUI Control	Correct Range of Values	Illegal Input Entered
1	Business Goal	Dropdown	Buy/ Sell	Nothing selected
2	Collaboration Mode (CM)	Radio Button	{new supplier (NS), existing supplier with contract (ESC), existing supplier without contract (ESW), new buyer (NB), existing buyer with contract (EBC), existing buyer without contract (EBW)}	Nothing selected
3	Product Type (PType)	Textbox	{goods (Gds), services (Svc)}	Numbers, Symbols
4	Product Name (PName)	Textbox	{any product name}	Numbers, Symbols
5	Quantity (Qty)	Textbox	{1, ..., n}	Alphabets, Negative Numbers
6	Item Price (Pr)	Textbox	{1, ..., n} in US Dollars	Alphabets, Negative Numbers
7	Order Lead Time (LT)	Textbox	{1 workweek (ww), ..., n ww}	Alphabets, Negative Numbers
8	Sourcing Mode (SM)	Radio Button	{Reverse Auction (RA), Traditional Sourcing (TS)}	Nothing selected
9	Process Type (Proc)	Radio Button	{Manufacturing (Mfg), Service Provision (SP)}	Nothing selected
10	Low Price Ceiling	Textbox	{1, ..., n}	Alphabets, Negative Numbers
11	Maximum Lead Time	Textbox	{1, ..., n}	Alphabets, Negative Numbers

11.4.1.5 HTN Planning-Related Limitation Revealed via Verification

The verification exercise revealed a limitation in the cBP’s inherited from HTN planning. Infinite loops during runtime may occur if some cBP tasks are automated to handle loop executions (see example in Figure 11.17). Recalling the differentiation between the *planning* and *execution* phases of BPM in Chapter 6, Genesis mainly handles the planning of cBPs and not the execution phase. As such, during the *planning* phase, there is no way anyone, let alone the algorithm, can predict how many times an automated loop will be executed during actual runtime. The worst scenario will result in infinite loops during execution, despite the cBP planned showing the correct decision routes and correct control flows. This limitation was also highlighted in earlier HTN planning related works in [295].

To overcome the possibility of infinite loops during runtime, the author set the maximum number of automated loops to be 10. There is no theoretical backing for the choice of 10 as the

limit, but most of the real-life negotiations (according to industry experts interviewed) do not reach 10, let alone go past 10 loops. The workaround was proven to be effective for the Genesis' integration with the IMSS Ont-CAF project mentioned in Section 5.5.5. Such loops are also most commonly found in the Negotiation such as the one in Figure 11.17.

Figure 11.17: Examples of loops which can be properly formulated into the output cBP but may result in infinite loops during runtime

11.5 Validation against Industry Processes

In the preceding sections, we have demonstrated that the ontological structures proposed within BOWL (*e.g. methods, tasks, constraints, primitive actions, preconditions, etc*) have enabled the Genesis algorithm to dynamically generate business processes on-the-fly.

We have also demonstrated that within a domain, BOWL's descriptions of HTNs of bite-sized tasks and their associated methods are enough to enable the Genesis algorithm to generate a suitable output (*i.e. a business process*) which will fulfil different scenarios given different input sets of goals and constraints.

The intelligence of the Genesis algorithm to produce the outputs was verified to be robust, consistent and error free in Section 11.4. The next aim of the research is to prove that the BOWL's constructs and modelling technique of atomic tasks and methods will enable the Genesis algorithm to generate a business process which will fulfil a different domain in a real-life scenario.

The author applied the prototype for the **leasing** goals encountered by a regional pallet manufacturing and leasing subsidiary company, Kim Hiap Lee Company (Pte) Limited (KHL) and its customers. KHL is one of the top two pallet leasing companies in the South East Asia region, and its processes would be highly representative of the best practices of its industry. Leasing goals and processes were chosen for validation of Genesis for two reasons:

- Increased complexity - Leasing is a process which combines elements of **both** buying (*i.e. KHL's customer buying the right to use KHL's rental pallets within a timeframe*), and selling (*i.e. KHL's customer intending to sell its products to its own customer*) within the same business process. For example, a customer looking to lease pallets from (*i.e. Buy*) KHL is actually using the leased pallets to transport its sold products to its customer (*i.e. Sell*) in turn.
- New unique processes specific to the pallet leasing industry - According to KHL, leasing has 5 unique leasing processes which are not found in typical "Buy-Sell" B2B collaborations. These 5 processes (more details in Section 11.5.1.2) coexist with common B2B tasks. The tasks, control flows and decomposition methods of the 5 processes can be simply added into BOWL, without affecting the original BOWL ontology. Because no customisation to the BOWL ontology classes are needed, if the extended BOWL can still enable Genesis to formulate cBPs for leasing, it would have proved that BOWL and Genesis are truly dynamic and not rigidly fixed for specific Buy and Sell goals only, *i.e. can be applied to any other context requiring some form of collaboration*.

11.5.1 Validation Background

KHL is a subsidiary company of Singapore Exchange-listed LHT Holdings Limited (LHT), a regional timber company with over 20 years of history and one of the largest manufacturers of high quality wooden pallets, boxes and crates in Singapore. LHT has 6 subsidiary companies that provide a wide range of products and services (see Figure 11.18). Some examples of its products include wooden crates, and pallets, plastic pallet, solid timber, technical wood, etc. Some examples of services are waste disposal, industrial packaging, and heat treatment.

Figure 11.18: Subsidiary companies of LHT Holdings Limited

LHT's major customers are from Singapore, Japan, Malaysia, Taiwan and the Middle East, including Fortune 500 companies such as Philips, Hewlett Packard Group, and South-East Asian supermarket giant Dairy Farm Group.

11.5.1.1 Why KHL?

KHL was selected because of its scale of operations, its market influence and share, and its unique leasing processes. It was also the only company which was willing to share with the author generic information about its business processes, with agreement to non-disclosure of its sensitive data. In August 2009, the author visited and interviewed the KHL management and staff to ascertain KHL's business processes. KHL basically rents pallets, and hence, the main focus of the validation scenario will be on *pallet leasing business processes*. Its operations predominantly span across the manufacturing and supermarket supply chain network, including all 4000 outlets of the Dairy Farm Group's Cold Storage and Giant supermarkets in South East Asia.

11.5.1.2 Introduction to KHL's Pallet Leasing Business Processes

Generically, KHL's pallet-leasing business mainly involves three main roles: KHL itself, KHL's customers and distribution centres (DC). As shown in Figure 11.19, the business process usually starts when a customer issues a purchase order, and the KHL sales team subsequently generates a corresponding *hire order* to lease a certain number of pallets. *Hire Orders* are cBP's unique to **leasing** B2B collaborations as typical buy-sell collaborations involve the purchase order.

Figure 11.19: Typical Flow of KHL Pallet Leasing Processes

After a *Hire Order* is received, *Issue Notes* are generated. This means that KHL is ready (i.e. has enough stock) to despatch pallets to the customer. When the customer returns the leased pallets to KHL, *Retrieve Notes* will be generated. *Exchange Notes* are generated when the customer hands the pallets to the distribution centre, which subsequently updates KHL's pallet inventory. When the customer uses moves their products in KHL's pallets to another KHL customer, a *transfer note* is generated. This will update KHL's inventory as well as both customers' account. A typical life cycle of a KHL pallet is shown in Figure 11.20 below.

Figure 11.20: Pallet leasing life cycle

As mentioned, there are 5 main business processes involved in pallet leasing; **Hire Order**, **Issue Note**, **Retrieve Note**, **Exchange Note**, and **Transfer Note**. As the standard B2B collaboration tasks modelled in BOWL do not cater to these 5 processes, the required tasks and methods were modelled into a new extended version of BOWL for the leasing business domain. Despite the addition of contents into BOWL, the Genesis algorithm was still able to produce all the processes, validating the thesis' hybrid approach. Before discussing the validation process, let us briefly describe these processes:

11.5.1.3 Hire Order

As shown in Figure 11.19, after receiving confirmation from the customer, usually in the form of a purchase order, KHL generates a new *Hire Order* for the customer. Hire Orders contain the customer's data, the date and time to despatch the pallets, the order quantity, pallet type and size, etc.

***Important Note:** Figure 11.19 to Figure 11.24 are process models modelled from KHL's provided information of as-is processes. The author is unable to release the original documentations due to non-disclosure agreements and will use these processes to validate against the generated cBPs. However, the flows and the tasks, if generated accurately by Genesis, will be sufficient to validate the reproducibility and robustness of the proposed hybrid methodology.*

11.5.1.4 Issue Note

An issue note is generated when the pallets are despatched to the customer. Because the pallets are delivered by lorry, more than one issue note may be generated; for instance, due to the lorry's capacity a single hire order is sub-divided into several issue notes for the same customer or different issue notes for different customers whose orders are conveyed in the same lorry.

After generating the issue note, KHL's inventory records and customer account are updated accordingly. There is a possibility that the customer may reject the pallets either in part or altogether:

1. When only part of the consignment of pallets is rejected, the issue note is revised when the rejected pallets return to KHL.
2. When the entire consignment of pallets is rejected, the issue note will be voided and a new issue note is generated later.

A typical issue note business process is summarised in Figure 11.21.

Figure 11.21: Issue Note business process

Figure 11.22: Retrieve Note business process

11.5.1.5 Retrieve Note

As shown in Figure 11.22, whenever KHL retrieves the pallets from the customer or the customer returns the pallets to KHL, with or without advance notification, a retrieve note is generated. Upon receipt of the returned pallets, KHL's inventory and customer account are updated accordingly. There is also a possibility that pallets may be lost or damaged. In either case, KHL and the customer have to agree on the relevant charges. After the agreement is met, KHL's inventory will be updated to reflect the decommissioned pallets.

11.5.1.6 Exchange Note

As mentioned earlier, the customer may exchange some of its pallets with those in the distribution centre. Therefore, the status of the customer's account remains unchanged, only those of the distribution centre and KHL need to be updated. This is accomplished through exchange notes (see Figure 11.23).

Figure 11.23: Exchange Note business process

11.5.1.7 Transfer Note

Transfer notes (Figure 11.24) are generated when a KHL customer physically transfers the pallets to another KHL customer when a company's products are shipped on the pallets to another company. In KHL's point of view, there is no need for detail transaction between the two customers; only the inventory and customer accounts need be updated.

Figure 11.24: Transfer Note business process

11.5.2 Validation Goal

The main goal of the validation process is to prove that:

1. The same task modeling approach and components of BOWL can be applied to a different domain, to a different goal of Leasing.
2. The same Genesis algorithm works for different types of BOWL files, even in an industrial setting.
3. The outputs automatically generated by the Genesis algorithm are highly similar to the actual real-life processes used in KHL.
4. The outputs (i.e. leasing business processes) produced were generated accurately, and highly relevant to real-life processes, even though it was tested against a large number of sets of goals and constraints, and an insignificant modeling effort was invested into adding about twenty leasing specific tasks into the generic BOWL.

The expected results of the validation are:

1. Successful and accurate dynamic decomposition of leasing goals into the underlying granular tasks.
2. The tasks are generated in a flexible way, meaning that different criteria and goals will result in different sets of results (without compromising on achievability of goals).

3. The tasks generated are chained by the correct control flow structures which will enable both the “leaser” and the “leasee” to achieve their goals.
4. The dynamically formulated cBP are vetted to be accurate by KHL management or personnel.

11.5.3 Validation Protocol

11.5.3.1 Constants

During validation, the following were unchanged:

1. Despite considering leasing processes, **the modelling methodology for key concepts of BOWL** (*i.e. compound and primitive tasks, actors, criteria, effects, company defined thresholds, etc*) is unchanged. In other words, the **ontological design of BOWL** remains intact. To ensure the most accurate and faithful validity check of the BOWL techniques proposed in Chapter 8, Chapter 9 and Chapter 10, all 341 tasks for the original BOWL were adopted into the validation, together with the leasing business process.
2. The **Genesis algorithm** remains intact and was in no way modified or customised for the leasing processes. Hence, this means that despite having to encompass more goals, we still deploy the same dynamic cBP formulation modelling methodology and tools. If results are generated as expected (*i.e. mimicking real-life processes*), the algorithm is validated to be generic enough for all cBP scenarios, but yet can be used for differing context of B2B collaborations.
3. Figure 11.25 shows a screen capture of the Genesis system applied for the leasing context, with leasing goals but much of its **supervised criteria** left intact. The criteria remains unchanged as leasing, being a B2B collaboration, still requires the same set of criteria for B2B collaborations.

11.5.3.2 Validation Variables

The following information was included into the BOWL knowledge base with little effort:

1. *LeaseOut* goals for suppliers (*i.e. KHL*) and *LeaseFrom* goal for buyers. The *LeaseOut* goals will focus on tasks needed by the pallet supplier, which the *LeaseFrom* goals will generate tasks which “leasee” companies need during pallet leasing. They are modelled using the same techniques proposed in Chapter 9, *i.e. as decomposable high-level compound tasks*, each with their own sets of decomposition methods. Each goal is expected to result in a different cBP.

2. All 5 types of leasing business processes were generically added into the original BOWL knowledge base. The decomposition methods and the compound-primitive task relations were added in generically. If the results can encompass these steps as per the different requirements, we would have proven the success of the validation.

11.5.3.3 Industry Expert Validation

To further validate the accuracy of the results generated dynamically, the author also conducted two rounds of interviews with Mr Thomas Yeo, manager of KHL and expert on leasing processes, before and after the leasing validation exercise to check the validity of the processes generated.

GENESIS

-|- Generate cBPs -|- Annotated BPEL File -|- Ontologised HTN -|- Help -|- Contact us -|- Log out -|-

Quick Links

- [Procurement and Order Management](#)
- [Implementation: LHT Pallet Leasing Scenario](#)

Last update
Monday, 12th October 2009

Scenario 2: LHT Pallet Leasing Scenario

Scenario Description:
Besides the typical buy/sell scenario in globalised trade, there is a special case of cross-enterprise collaboration: Leasing. Leasing is a mixture of business process involving buying/ selling and also, returns of the hired goods when their leasing dates are up. The demo demonstrates another use of the Business-OWL (BOWL) and how together with the Genesis algorithm, the system is able to generate cross-enterprise business processes on the fly.

User Company Detail

Company Name
LHT Holdings Limited

Address
Street: 27 Sungei Kadut Street
Postal Code: 729335
Country: Singapore

Supervised Criteria

Business Goal LeaseOut	Product Type ☒ Goods ☐ Services	For KHL use only Minimum Lease Price (in US\$) 1200 Longest Lead Dispatch Time (in workweeks) 2
Product Name Pallet	Quantity 500	
Item Price (in US\$) 2	Lead Dispatch Time (in workweeks) 1	
Process Type ☐ Manufacturing ☒ Service Provision	Sourcing Mode ☐ Reverse Auction ☒ Traditional Sourcing	
Collaboration Type ☐ New Leaser ☐ Existing Leaser Without Contract ☒ Existing Leaser With Contract		

Reset **Generate Collaborative Business Process**

Figure 11.25: Genesis system for dynamic creation of leasing business processes

11.5.4 Validation Highlights and Results

Let us now study the dynamic decomposition results of the LeaseFrom and LeaseOut goals, and how they measure against the actual real-life processes described in Section 11.5.1.2.

11.5.4.1 High Similarity against Real-Life Processes

Figure 11.26 shows the validation process embarked by the author. About 20 tasks and their related methods were added into the BOWL ontology (Step 2). The addition was a simple appending of text into the BOWL text file, and was included to introduce new leasing-specific tasks. However, it is of utmost importance to note that the rest of the original BOWL remained intact. The introduced tasks are not hard-coded tasks, but are atomic tasks (e.g. fill form, delivery, etc) which can be chained in several different permutations to form the more meaningful leasing goals. The author envision that as more and more industries take part in using BOWL, they will be able to add in their industry specific atomic tasks into BOWL.

Figure 11.26: Validation process

The similarity of the generated processes were manually inspected against KHL's industry real life processes. As shown in Table 11.5, it was found that all business processes were generated with a similarity of 100% to their respective original real-life leasing goals. This shows that the same task modeling approach and components of BOWL can be applied to a different domain, and to a different goal of Leasing. It also shows that the same Genesis algorithm works for different types of BOWL files, even in an industrial setting.

Table 11.5: Validation results were highly similar to real-life processes

Original KHL process	Percentage accuracy of generated process against real-life process (%)
Hire Order	100
Issue Note	100
Retrieve Note	100
Exchange Note	100
Transfer Note	100

Furthermore, Table 11.5 showed that the outputs automatically generated by the Genesis algorithm are highly similar to the actual real-life processes used in KHL. The outputs (i.e. leasing business processes) produced were generated accurately, and highly relevant to real-life processes, even though it was tested against a large number of sets of goals and constraints, and an insignificant modeling effort was invested into adding about twenty leasing specific tasks into the generic BOWL.

As a final conclusion to the validation, the similarity comparison results were further validated by KHL's manager Mr Thomas Yeo.

11.5.4.2 Future Validation Work

Ideally, BOWL and genesis has to be applied and validated in an entire industrial sector with several hundreds, if not thousands of collaborators, preferably via an industrial consortium. However, due to the reluctance of several companies approached, and the economic crisis coinciding with the duration of the experiments (2008 to 2009), the author faced several difficulties gaining the participation of companies for a timely completion of his research.

Despite this, the approach and validation results in this section has already surpassed the application and validation of several related research projects (e.g. Charles Petrie of SRI International (refer to Section 12.4.1), Matchmaking of Ad Hoc Processes (refer to Section 5.5.2), the SUPER project (refer to Section 5.5.3), the SemBiz project (refer to Section 5.5.4)), which are at best, prototypes simulating simplistic processes, and not validated within real-life B2B scenarios and processes.

After the completion of this research, there should be further work to apply the BOWL modeling constructs and in-depth industry-wide testing against an entire supply chain of an industrial sector.

Let us now discuss two examples from the validation results in detail.

11.5.4.3 An example of the LeaseOut goal achieved

Figure 11.27 shows how the LeaseOut goal and its criteria were input into the Genesis system. Key components of the resulting cBP output generated by Genesis are shown in Figure 11.28 and Figure 11.29. Observing Figure 11.28, it can be seen that the HireOrder business process tasks were generated. The HireOrder tasks “receive Hire Order” for the supplier were found in the Negotiation stage as it is the stage where the potential customer negotiates the terms for the hire.

Quick Links

- Procurement and Order Management
- Implementation: LHT Pallet Leasing Scenario

Last Update: Monday, 12th October 2009

Scenario 2: LHT Pallet Leasing Scenario

Scenario Description:
 Besides the typical buy/sell scenario in globalised trade, there is a special case of cross-enterprise collaboration: Leasing.
 Leasing is a mixture of business process involving buying/ selling and also, returns of the hired goods when their leasing dates are up. The demo demonstrates another use of the Business-OWL (BOWL) and how together with the Genesis algorithm, the system is able to generate cross-enterprise business processes on the fly.

User Company Detail

Company Name LHT Holdings Limited	Address Street: 27 Sungei Kadut Street Postal Code: 729335 Country: Singapore
---	---

Supervised Criteria		For KHL use only
Business Goal LeaseOut	Product Type ☒ Goods ☐ Services	Minimum Lease Price (in £38) 1200
Product Name Pallet	Quantity 500	Longest Lead Dispatch Time (in month/year) 2
Item Price (in £38) 2	Lead Dispatch Time (in month/year) 1	
Process Type ☐ Manufacturing ☒ Service Provision	Sourcing Mode ☐ Reverse Auction ☒ Traditional Sourcing	
Collaboration Type ☐ New Leaser ☐ Existing Leaser Without Contract ☒ Existing Leaser With Contract		

Reset! Generate Collaborative Business Process

Figure 11.27: Screen capture of LeaseOut goal and the input criteria during validation

Upon “HireOrderConfirmation”, there will be a “Pick” (i.e. decision split) for accepting, rejecting or renegotiating the hire order. These tasks bear full resemblance to the real-life Hire Order process discussed earlier, and are of the correct control flows. Each of these tasks can be executed eventually by Web services in a service-oriented architecture.

Figure 11.28: Hire Order business process tasks dynamically formulated in the correct control flow and order

Another example in Figure 11.29 below shows the later part of the output cBP, where the Issue Note leasing BP tasks are generated and integrated into a normal B2B collaboration, in the contract agreement phase, just before the contract fulfilment phase. It is also important to note that the decision route (recall the Issue Note decision routes and related steps in Figure 11.21) in the Issue Note BP was also generated as a “Pick” control flow encapsulating tasks which mirror the real-life Issue Notes.

Figure 11.29: Issue Notes and Exchange Notes business process tasks dynamically generated for LeaseOut Goal

These examples have demonstrated how (1) dynamic decomposition of leasing goals into the underlying granular tasks can be found in the output, (2) the tasks are generated for different criteria and goals without compromising on achievability of goals, and (3) the tasks generated

are also chained by the correct control flow structures which will enable both the “leaser” and the “leasee” to achieve their goals.

On top of the checks against the original KHL leasing business processes, the dynamically formulated cBP are vetted to be accurate by KHL management or personnel. The KHL process for leasing processes was also integrated into the IMSS Ont-CAF project’s Atlas system, resulting in a fully executable business process which performs the LeaseOut goal.

11.5.4.4 An example of the LeaseFrom Goal Achieved

For the LeaseFrom goal, the input screen is the same as Figure 11.27, except for the differing goal stated. It was also found that the resulting cBP contained the successful formulation and decomposition of leasing BP tasks. In the example of the resulting tasks generated for the “Leasee”, the company leasing from KHL, it was shown that in the contract fulfilment phase, we can see that the customer has a “Pick” (decision split) between Exchange Note, Transfer Note and Retrieve Note BPs, reflecting the real-life business processes discussed in Section 11.5.1.2. Each leasing business process generated also contained the correct order and control flows of tasks, thus validating the BOWL+Genesis hybrid methodology.

The screenshot displays the Genesis web application interface. On the left, a 'Quick Links' sidebar contains links for 'Procurement and Order Management' and 'Implementation: LHT Pallet Leasing Scenario'. The main content area shows a hierarchical tree of tasks for the LeaseFrom goal. The tree is organized into three main sections: Exchange Note, Transfer Note, and Retrieve Note. Each section contains a list of tasks with their IDs and descriptions. The Exchange Note section includes tasks like #CF11-ExchangeNote, #CF11.1-ReceiveOnlineExchangeNote, and #CF11.2-IssuePallets. The Transfer Note section includes tasks like #CF12-TransferNote, #CF12.1-ReceiveOnlineTransferNote, and #CF12.2-IssuePalletsForTransfer. The Retrieve Note section includes tasks like #CF13-RetrieveNote, #CF13.1-SendRetrieveNote, and #CA8.2-UpdateInventory. The interface also features a top navigation bar with links for 'Generate cBPs', 'Annotated BPEL File', 'Ontologised HTN', 'Help', 'Contact us', and 'Log out'.

Figure 11.30: Exchange Notes, Transfer Notes and Retrieve Notes tasks successfully generated for LeaseFrom Goals

11.6 Chapter Summary

This chapter discusses the foundations of the ECBP framework, and the hybrid ontology-HTN planning approach in the form of the Genesis prototype which realises dynamic goal-driven formulation of business processes.

The system was implemented as a Java-based Web application that can be accessed by both types of collaborators (i.e. the buyer or the seller). The business goals and constraints are input *via* a GUI, and the Genesis algorithm generates dynamically formed cBP's, in three cBP format containing the *actors*, *control flow logic* and *decompositions* of tasks needed to achieve the stated goal and criteria.

The first cBP format is the ***annotated abstract BPEL format***, which contains the flow of tasks and the roles associated with each task. This abstract BPEL file can be further augmented with more details such as the exact Web service description files and their location, etc.

The second cBP format was a ***graphical tree-structure format*** which shows the levels of decomposed tasks and nested control flows of the cBPs generated.

The third cBP format was a ***dynamically generated graph of tasks*** required by the input goals and constraints. Together, the second and third formats offer the user a graphical method to troubleshoot errors and comprehend the flows of the tasks involved in the output cBP.

The output cBPs were verified against industry reference models such as RosettaNet, ebXML and OAGIS as follows. The full branch coverage of both the algorithm and BOWL were verified *via path testing*. Every branch in a program (and its knowledge base) was executed at least once under 14 test scenarios comprising combinations of goals and criteria. Furthermore, both the algorithm and BOWL were tested for *fault tolerance* by entering 'illegal' parameters (e.g. mismatched data types, input values outside range). However, *load testing* was not conducted in this research, as it requires a large volume of concurrent transactions associated with a large number of users. There are three reasons for not applying load testing: (1) achieving load testing does not add mileage to demonstrate dynamic cBP formulation given some high level goals and constraints; (2) the lack of support from industry consortiums and (3) load testing is only necessary when the prototype is to be commercialised.

The Genesis and BOWL system was validated by applying it to a pallet leasing company Kim Hiap Lee Timber Co. Pte Ltd (KHL). The Genesis and BOWL system generated cBP's which were similar to KHL's five pallet leasing business processes. The tasks required and the control flows governing the decision routes of the leasing processes were generated as expected.

Part III: Discussions and Conclusion

Chapter 12. Discussions and Key Research Contributions

This chapter discusses how the research contributions of the hybrid HTN planning-Ontology approach addressed the knowledge gaps highlighted in Part I of this thesis.

12.1 Outline of Thesis Research Contributions

12.1.1 General Overview of Contributions

Read from left to right, Figure 12.1 shows an outline of the key research contributions of this thesis. The boxes shaded in *olive green with single-point outlines* represent current concepts, while the *pink boxes with bold outline* show the significant milestones achieved during the research. Arcs with labels pre-fixed by plus signs (i.e. “+”) indicates modifications to existing techniques or literature to result in contributions. The research challenges (RCs) and questions (RQs) discussed in Chapter 7 are also shown in Figure 12.1.

These RC’s and RQ’s are reproduced from Section 7.6:

- **RC 1: Encapsulation of high-level business goals and constraints**
 - *RQ 1.1: What are the highest-level business goals in B2B collaborations?*
 - *RQ 1.2: What are the top priority criteria considered in B2B collaboration?*
 - *RQ: 1.3: Is there a more “human natural” way to express these high-level goals and constraints?*
- **RC 2: Decomposition of high-level goals into granular executable tasks**
 - *RQ 2.1: What methodologies to decompose compound tasks into primitive tasks exist?*
 - *RQ 2.2: Are there any current modelling techniques to represent dynamic decompositions?*
- **RC 3: Composition of the granular tasks in the appropriate sequences and control flows, performed by the appropriate roles**
 - *RQ 3.1: How do we ensure that the business process’ control flows are faithfully reproduced when the granular tasks are composed together?*

Figure 12.1: Outline of Thesis Research Contributions

- *RQ 3.2: What modeling augmentations/ techniques must be proposed to ensure that the composed primitive tasks can be easily executed by existing execution-phase technologies such as Web Services?*

The highlights of Figure 12.1 are:

1. The ECBP framework embodying the 4 cBP axioms and 5 B2B collaboration phases addresses RC 1.
2. The modified HTN planning algorithm (Genesis algorithm) and the enhanced Web-ontologised Planning Domain Description (BOWL) together generate dynamic cBP's (annotated abstract BPEL). They address RC 2 and RC 3.

12.1.2 Addressing Research Challenge 1- Encapsulation of High-level Business Goals and Constraints

Research Challenge 1 (RC1) is achieved via the application and extension of concepts from LAP and Speech Acts Theory. The resulting ECBP framework in Chapter 8, addressed the following research questions:

12.1.2.1 Highest level business goals in B2B collaborations

Instead of attempting to classify the highly context-dependent and almost infinite number of business goals types, the author applied LAP to identify any generic goals which hold true for any type of business process collaboration. This is a significant milestone initiative, as goals are seldom modelled in contemporary BPM methods; goals are usually implied from the names of business processes. Guided by the DEMO method, the author isolated the two highest-level and essential goals of ALL business collaborations: **Buy and Sell**. Regardless of the names and terminologies used (e.g. procurement, order fulfillment, tendering, outsourcing, advertising, etc), they only express two symbiotic goals: *To BUY goods or services for some need, and to SELL goods or services to make some profit*. Sometimes, items or services are bought to make something which can eventually be sold to someone to make a profit. Buy and Sell are the two highest-level, fundamental goals for ALL B2B collaborations.

12.1.2.2 Finding the “top priority” criteria for each B2B collaboration

After knowing the highest level goal, we need to know the high-priority criteria considered in B2B collaborations. In order to answer RQ1.2 and to check the validity of such a question, the author conducted a literature review (recall Section 8.4) and several on-site interviews with the following industry representatives:

1. Mr. Thomas Yeo – Manager, Kim Hiap Lee Company (Pte) Ltd Singapore (Regional pallet leasing, supply chain industry), 2009

2. Mr. Basil Sim – Manager, Meiban Mold Manufacture Pte. Ltd. (Plastics injection molding manufacturing), 2008
3. Ms. Serina Lim – Procurement Officer, Meiban Mold Manufacture Pte. Ltd. (Plastics injection molding manufacturing), 2008
4. Mr. Albert Tan – Manager, Dell Singapore, Pte. Ltd (Computer manufacturing), 2008
5. Mr. TM Chua – Manager, Motorola Electronics Pte. Ltd., Singapore (Wireless communication device manufacturing), 2008
6. Mr. Bobby Koh – Founder, Managing Director, Endela Trading & Manufacturing Pte. Ltd. (Singapore regional fastener manufacturing), 2008

While the author recognizes that the ideal assessment of B2B collaboration criteria was to conduct a comprehensive consortium/ industry-wide survey, there were practical difficulties in securing the cooperation of companies.

Nevertheless, the comments and suggestions from the six industry representatives were very much corroborated by Dickson [284], whose work was based on the responses of 170 USA-based purchasing executives as discussed in Section 8.4.

In a nutshell, both the author's and Dickson's findings revealed that the top criteria for B2B collaboration depend on (1) the type of industry, (2) collaboration mode (e.g. a new supplier versus an established qualified supplier), and (3) the process taken to produce and deliver the product (e.g. Just-In-Time).

The industry representatives interviewed unanimously agreed that the order of priority also differs according to the type of product, e.g. whether it is made-to-order or made-to-stock. Amongst the criteria, the most commonly mentioned top 3 criteria that were mentioned were *Price*, *Quality* and the *Technical Capability* of the collaborator. As technical capability is a subjective qualitative indicator, it was not included as a constraint. However, ontological descriptions (e.g. BOWL) can accommodate technical capability context in the future.

Eight key criteria for B2B collaboration were identified (*See Table 9.2: Supervised Criteria and their range of values*) as being consistently the most important. They are *Collaboration Mode*, *Product (Type and Name)*, *Quantity*, *Item Price*, *Order Lead Time*, *Sourcing Mode*, and *the Process Type*. These eight criteria were adopted in Genesis and BOWL presented in Chapters 9 and 10.

Contribution: Generically, the top 8 quantifiable criteria considered by companies during B2B collaborations are Item Price, Order Lead Time, Collaboration Mode, Product (Type and Name), Quantity, Sourcing Mode, and the Process Type.

12.1.2.3 The Need for a Generic B2B Collaboration Framework

The research questions RQ1.1 and RQ1.2 also led the author to question if there is a generic framework which holds true for all B2B collaborations and which embodies the core of all B2B collaborations. The ECBP framework was derived from CfA, SAMPO and DEMO frameworks, which are based on the speech-act theory / language-action perspective approaches.

The author realised, after a review of Communicative Approaches (e.g. CfA, DEMO, SAMPO, etc) and industry interviews, that the 5 phases of the ECBP framework. i.e. “*Discovery*”, “*Establish Contact*”, “*Negotiation*”, “*Contract Agreement*” and “*Contract Fulfilment*” (Section 8.1), are not addressed by any current solutions (recall discussions in Section 8.3).

Contribution: The ECBP framework, derived by the steps in DEMO and from the theoretical frameworks of CfA, SAMPO and DEMO, represents the generic 5 phases involved in all B2B contributions (Discovery, Establish Contact, Negotiation, Contract Agreement, and Contract Fulfilment). Solutions built on such frameworks will cater to all necessary phases of B2B collaboration, thus enabling dynamic cBP formulation.

Because the ECBP framework also puts the tasks into perspective against the phases, we can now classify tasks into the relevant phases. Such a classification greatly helped the author in the synthesis of industry reference models while creating BOWL; this is important as it also infers that we can now reduce the consideration spaces when looking for Web services to execute each decomposed task. For example, if we know that the task is actually an *Establish Contact* task, we would just need to search in the domain of Establish Contact-related Web services, greatly reducing the load for searching into services from other phases. It is also good to note that the modelling of business processes for dynamic cBP formulation Genesis and BOWL was done using the ECBP framework and the contributions in Chapter 8 as a foundation.

12.1.2.4 Seeking a more-natural way to express high-level goals and constraints

The author embarked on investigations early in his PhD into the *communicative approaches* in BP modeling in an attempt to answer research question RQ 1.3. There was some promise in the way an act to commit (commonly found in B2B collaborations) can be easily quantified by speech acts theory into its role, illocutionary force (type of commitment), and the strength/degree of commitment. However, despite the promise of modeling high-level processes and commitments, this approach could not allow the author to address research challenge 2, i.e. the dynamic decomposition of high-level goals into granular executable tasks. In the end, the investigations into communicative acts were not fruitless, as it allowed the author to discover

DEMO to extract the so-called essential business process, and also to have references to create the framework needed for designing B2B systems.

Bearing in mind the success of previous LAP works, LAP as standalone methods can express high-level goals and constraints, hence addressing this RQ1.3. However, they cannot help the dynamic planning or synthesis of tasks without the solving of RQ2.1 and RQ2.2 discussed below. With the solving of RQ2.1 and RQ2.2 via the hybrid HTN planning-Ontological method, we now have a way to link high level goals to decomposed tasks. With this approach, we can perhaps now propose a *Speech-Act theory-based* business process goal and constraint expression or modeling as future work (See Chapter 13).

12.1.3 Addressing Research Challenge 2 - Decomposition of High-level Goals into Granular Executable Tasks

Research Challenge 2 (RC2) seeks solutions for the **decomposition of high-level goals into granular executable tasks**. As shown in Figure 12.1, hierarchical task network planning and ontologies influenced most of the methods addressing RC2 (and its questions RQ2.1 and RQ2.2).

12.1.3.1 What methodologies decompose compound tasks into primitive tasks?

To the author's best knowledge, there are no methodologies which directly address dynamic formulation of collaborative business processes. After breaking the objective into research questions such as RQ2.1, the author was able to find individual solutions for each research question.

HTN planning was identified as a potential approach for solving the dynamic task decomposition sub-problem, but it was also noted that it cannot handle BPM as a standalone. This is because plans generated by HTN planning are single-level sequences of tasks without control flows and decision routes. Extensions need to be made to enhance its algorithm's consideration for control flows and roles found in real-life business processes, and to enable the modified HTN planning to produce plans directly for SOA implementation.

To address RQ2.1, the following contributions were made to the original HTN planning algorithm (also denoted in Figure 12.1):

Contribution: The algorithm was modified to read the control flow definitions while checking for the matching decomposition method, and then embed the control flows governing the decomposed tasks into the final output cBP. The control flows were adopted from basic control flows classified in Workflow Patterns.

Contribution: The algorithm was extended to embed actors for each primitive tasks during the decomposition process. The original HTN planning outputs were devoid of actors, because of an assumption that the plans are solely executed by the executer of the planner. This is not the case in real-life B2B collaborations, which involves 2 fundamental roles: Buyers and Sellers.

Contribution: The algorithm was modified to read from Web-ontologised PDDL files written in extended-OWL, instead of the usual text-based, Web unready PDDL formats. The direct access to the extended OWL files enable the direct generation of tasks described for execution in Web environments.

12.1.3.2 Any current modeling techniques available for describing dynamic decompositions?

RQ2.2 seeks a modeling methodology which enables dynamic decompositions of tasks for business process formulation. As shown in Part I (Literature Review), contemporary BPM modeling techniques only enable the modeling (programming) of business processes, without considering the planning phase. To the author's best knowledge, there are currently no methods which encourage the modeling of BPs for dynamic generation of different permutations of business processes.

Through the study for HTN planning, it was shown that its algorithm could decompose high level tasks into granular ones as the algorithm refers to a knowledge base described in the Problem Domain Description Language (PDDL) (see example in Figure 12.2). However, as the PDDL was not intended for addressing cBPs but for sequential generic tasks executed by one role (i.e. the user or machine activating the planner), its modeling only needs to consider output single-level sequences of tasks without decision routes and other control flows.

PDDL (for a blocks worlds example)	A method in BOWL
<p>Src: http://www.ida.liu.se/~TDDA13/labbar/planning/strips/bw.pddl</p> <pre> (define (domain blocks_world) (:requirements :strips) (:predicates (on-table ?x) (on ?x ?y) (clear ?x)) (:action MoveToTable :parameters (?x ?y) :precondition (and (clear ?x) (on ?x ?y)) :effect (and (clear ?y) (on-table ?x) (not (on ?x ?y)))) (:action MoveToBlock1 :parameters (?x ?y ?z) :precondition (and (clear ?x) (clear ?z) (on ?x ?y)) :effect (and (clear ?y) (on ?x ?z) (not (clear ?z)) (not (on ?x ?y)))) (:action MoveToBlock2 :parameters (?x ?y) :precondition (and (clear ?x) (clear ?y) (on-table ?x)) :effect (and (on ?x ?y) (not (clear ?y)) (not (on-table ?x)))))</pre>	<pre> <rdf:Description rdf:about="#MethodCF6.2.2.3.A"> <hasProcessType rdf:resource="#ServiceProvision"/> <rdf:type rdf:resource="#Method"/> <hasProcessType rdf:resource="#Manufacturing"/> <hasProductType rdf:resource="#Goods"/> <hasProductType rdf:resource="#Services"/> <hasControlFlowPattern rdf:resource="#Sequence"/> <hasSubtasks rdf:resource="#CF6.2.2.3.1- SendTransportationProjection"/> <hasControlFlowPattern rdf:resource="#SequenceEnd"/> <hasSourcingMode rdf:resource="#ReverseAuction"/> <hasPriceXQuantity rdf:resource="#HighPriceXQuantity"/> <hasCollaborationMode rdf:resource="#ExistingSupplierWithContract"/> <hasLeadTime rdf:resource="#LongLeadTime"/> <hasBusinessGoal rdf:resource="#BusinessGoalBuy"/> <hasLeadTime rdf:resource="#ShortLeadTime"/> <hasPriceXQuantity rdf:resource="#LowPriceXQuantity"/> <hasSourcingMode rdf:resource="#TraditionalSourcing"/> </rdf:Description></pre>

Figure 12.2: Comparing the structure and syntax of PDDL and BOWL

From observing the PDDL syntax in Figure 12.2, it is evident that PDDL has no descriptions for consideration of task actors or control flows. To make it encompass control flows, the author decided to adapt the PDDL into an ontological format, as ontologies enable semantic descriptions of relationships between tasks, methods, criteria and effects (see Figure 12.2).

This leads us to the following contributions:

Contribution: PDDL concepts were adopted into OWL ontological descriptions (see Figure 12.2), with extensions. This means that the descriptions for relationships between primitive tasks, compound tasks, methods, criteria and effects are ontologised, making its structure more ready to generate outputs for SOA Web service orchestration, e.g. BPEL.

Contribution: The OWL-based problem domain description file was extended to enable modeling of nested control flows, defined as properties of methods belonging to compound tasks. Ordinary OWL cannot embed nested control flow information. Modifications were done to encompass sub-tasks with control flows via added metadata structures for control flows.

Contribution: The OWL-based problem domain description file was extended to enable inclusion of task actors during the algorithm's dynamic decomposition and sequencing. This enables it to represent real-life, multiple role business processes and enables both human and computer-automated execution (e.g. Web services) of tasks formulated.

The above contributions enabled the author to address the dynamic task decomposition challenge. We will now explore how to compose the decomposed task in a way most reflective of business processes in reality.

12.1.4 Addressing Research Challenge 3- Composition of granular tasks into appropriate control flows for appropriate roles

With the granular tasks dynamically decomposed in RC2, RC3 investigates how to best chain these decomposed tasks in the appropriate sequences and control flows, and also ensure that they are performed by the appropriate roles.

12.1.4.1 Ensuring that business process' control flows are considered during composition of granular tasks

As shown in Figure 12.1, RQ 3.1 is addressed by the embedding of Workflow Pattern properties within the methods in BOWL. Each permutation of decomposition stored in the methods will not only instruct the algorithm how to decompose the tasks, but also how to string them together with the appropriate workflow patterns (e.g. sequence, pick, etc).

However, as the original OWL file cannot support the embedding of tags to depict nesting of control flows, there was also a need to modify the metadata structure of OWL for this

contribution. The algorithm was also significantly modified to consider the control flows of the decomposed sub tasks upon each decomposition.

Contribution: Basic control flows in the Workflow Patterns field are embedded into the modified OWL file, enabling the algorithm to recursive model nesting of complex control flows for an entire cBP.

12.1.4.2 Modeling augmentations/ techniques to ensure composed primitive tasks can be easily executed by existing execution-phase technologies

Within RC3, RQ3.2 proposes modeling augmentations/ techniques that ensures composed primitive tasks can be easily executed by existing execution-phase technologies such as Web Services. The following contributions (listed in Figure 12.1) ensure that the output Abstract BPEL file can be easily implemented in an SOA environment.

Contribution: During the creation of BOWL, a reference model list of cBP tasks was synthesized from three major B2B information exchange standards, and modeled in a process oriented, hierarchical task network which enables dynamic cBP formulation. The tasks are also of an acceptable level of granularity for direct Web service execution.

Contribution: To enable the output primitive tasks generated to be executed directly by Web services, basic actions from BPEL (e.g. “invoke” of Web services) were included as properties of tasks in BOWL. This also enables a smooth, translation free process when the abstract BPEL output file is further augmented into executable BPEL files.

12.2 How the Research Addressed BPM Gaps

On top of addressing the three main research challenges, the contributions in this thesis have addressed gaps identified for both BPM and B2B research. We can now make some observations on the fundamental obstacles to dynamic cBP formulation:

12.2.1 Absence of universally agreed BPM terminologies

A common problem in BPM is the absence of universal terminologies [7, 37, 274-276]. Terms are used loosely to represent distinct scope and feature differences [277]. One example is the confusion between *business process reengineering (BPR)*, *workflow management (WfM)* and BPM. Such confusion has led to mismatched (or worse, wrong) BP solutions being implemented. This thesis, through Chapter 2, attempts to address this gap.

12.2.2 *Presence of many duplicating languages and standards*

Because of a lack of understanding of the fundamentals of BPM, new languages and standards proposed often contain duplicating features [274], and loosely claim to be based on formalisms like Pi-Calculus and Petri Nets [7],[104]. Much of the above may be traced to the software industry, which is notorious for new terminologies and buzzwords. There is a strong need to overcome this knowledge obstacle. The author believes that an authoritative taxonomy of BPM terminology is needed to resolve the issue. The author's efforts in classifying and reviewing the literature review can be found in [13] and Appendix B.

12.2.3 *Current State of BP Modeling Theory*

Whether it is within or between enterprises, business process modeling is the most fundamental, as it is the conceptual basis for workflow or business process owners to translate the real-life processes into automated executable ones. Hence, the understanding the current approaches and trends in business process modeling had to be covered in this thesis.

Chapter 3 states that there are two main schools of thought in business process modeling: *Flow-based approaches* and *Communicative approaches*. Many *flow-based approaches* are based on computer science process theories like state machines, Petri-Nets and Pi Calculus. While ideal for modelling and controlling flow, they are usually too low-level, void of context, and are indifferent to the highly communicative and contextual business processes in reality (see Section 3.1). This also goes to show that there is inadequate basic research in business process essentials and formulation [278, 279]. If we can understand the essence of business processes and extract the most natural way to mimic how real-life business operations are formulated, we would have done for business processes what object-oriented programming did for software development and what relational databases did for data management. Communicative approaches such as DEMO in Section 3.2.3.5 offer us a way of extracting this essential business process.

As mentioned in the review in Chapter 3, the graphical and grammatical flow-based approaches are the most dominant best practices in the industry while the communicative approaches, while semantically rich, are less popular. However, in recent years, there is an increasing trend towards the Semantic Web, and also the abstraction of high-level tasks from low-level details (e.g. cloud computing, especially Software as a Service (SaaS)). These trends are not explicitly stated as communicative approaches *per se*, but it is obvious that with the strong foundations in flow-based approaches such as BPEL, BPM researchers are attempting to bridge the “goal-to-task” or “business-to-technical” semantic divide via more annotations, and new methodologies such as semantic Web services (see Section 5.2.2).

It is also interesting to note that all presented communicative approaches were able to address the complex communication and processes in real-life business processes. This is mainly because they always start from establishing a generic B2B collaboration framework, e.g. conversation graphs, SAMPO, DEMO (*cf.* Section 3.2.3).

Hence, current movements do not just need robust control flows and exhaustive annotations for describing semantics (like that of the flow-based approaches), there is also a strong need to begin design the methodology from a foundational B2B framework (like those in the communicative approach). The author's proposed ECBP framework in Chapter 8 aims to do this.

12.2.4 *Overlooking the Planning Approach to BP Modeling*

Business process modeling involves two steps: (1) *creating* a business process model in relation to the goals, and (2) *executing* the business process model. The challenge of current BPM is automating the processes, and this involves some form of computing. In order for the computer to execute the inputs of the human user (e.g. a business process model), it can collect the inputs via three possible methods identified by AI researchers: *programming*, *planning* and *learning*.

As discussed in Chapter 6, *programming* involves the user inputting *manually* a model or code for the computer to understand, while *planning* involves the automatic generation of the model or code for different set of goals based on some pre-defined relationships and rules, and *learning* involves the extraction of models or codes from historical data or archived transactions.

Not surprisingly, current predominant methods for BPM (such as BPEL and BPMN) represent the *programming* paradigm. On the other hand, this thesis proposes the *planning* approach, which dynamically creates the model, based on some predefined rules (e.g. task decompositions, relationships between roles and tasks, etc). The field of business process mining [296-299], which involves the formulation and understanding of business processes from the intelligent extraction of processes are reminiscent of the *learning* approach.

Compared to planning and learning paradigms of creating models and code, programming (with its manual requirements) is the least adaptive to changes in requirements. Both flow-based and communicative approaches mainly focus on the modeling and execution via programming, with little emphasis on planning. Hence, the lack of tools to *dynamically* formulate business processes for supply chain changes is quite expected. This also means that current, programming-based approaches in BPM are unable to adapt to rapid changes to requirements in supply chain demands. Goal-driven approaches lack planning capabilities

This thesis introduced a goal-driven planning approach to the problem of automatic creation of business processes. It is noteworthy to know that while there are some works in goal-based approaches [300, 301], these tend to focus on *programming* goals and tasks, and not surprisingly, lacked considerations for the planning phase. The proposed method in this thesis (i.e. Genesis and BOWL) is perhaps one of the first (to the author's best knowledge) towards the goal-driven **and** planning approach.

12.2.5 *Top-down vs. Bottom-Up approaches*

The generalized terms “top-down” and “bottom-up” approaches introduced in Section 5.1.3.2 sums up the predominant ways of fulfilling business goals. In contrast to the current dominant “bottom-up” approaches (e.g. semantic Web services), this thesis proposes a top-down approach, which produces an abstract BPEL file which can fulfill the goal and constraints input by the user. Logically, this reduces the computational considers for the Web services that are to be composed together, as there would be reduced requirements on deciphering control flows before and after each task, roles appropriate, the sequence of tasks, the types of Web services required for each primitive task. Without these information from such a “blueprint”, there would be infinite things to consider for the composition of Web services. Hence by being a “bottom-up” approach, current approaches such as dynamic Web service compositions face a tremendously large task in reasoning and decision making.

12.3 How the Research Addressed B2B Gaps

Apart from business process modeling paradigms, many factors relating to B2B information exchange standards and methods also impede the achievement of dynamic cBP formulation. This section describes the obstacles and how this thesis attempts to address most of them.

12.3.1 *High setup and licensing costs*

While B2B information exchange standards like EDI (Electronic Data Interchange) and RosettaNet support cBP's, they usually come at exorbitant setup and licensing costs. We recall that the cost of a RosettaNet template for a singular collaborative business process costs between S\$150,000 to S\$300,000 [133, 281]. Just imagine the financial burden to a small company which may easily need 20 to 30 solutions. Apart from high costs to the organisation, potential collaborators may be deterred as well. There are typically small suppliers. Hence, there is a pressing need for a low-cost, non-exclusive open standard collaboration platform just like the Web. The proposed BOWL+Genesis approach, together with the Ont-CAF project, attempts to equip companies with a low-cost easily adopted method for dynamic business process formulation.

12.3.2 *Static setups not flexible and responsive*

Besides being expensive, cBP's modelled in B2B standards are usually static in nature, and are unable to adapt to sudden changes on short notice. A dynamic method such as Genesis will enable companies to remodel cross-enterprise business processes when needed.

12.3.3 *Current B2B standards' emphasis on functional perspective of business processes*

Current B2B standards such as RosettaNet, ebXML and OAGIS (Open Applications Group Integration Specification) [21] have focused on classifying common B2B tasks into scenarios, functions or departments. The advantage of this is easy customisation for the IT consultant and process owners. However, it creates an "over-the-wall", silo mentality when modelling B2B collaborations. BP's in reality require cross-functional/ cross-department processes to realize B2B goals. For example, in a typical "*Procurement*" processes, the inventory department needs to provide the current inventory status and the finance department the budget available in one transaction. As the "*cross-department*" or "*process-oriented*" perspective of B2B collaboration tasks is something which the author believes to be more aligned with the realities of supply chain business processes, the ECBP framework (see Chapter 8) and the creation of BOWL (see Chapter 9) considered all phases of B2B collaborations.

12.3.4 *Limited semantic representations in current B2B standards and systems*

Many current B2B standards also cannot succinctly and accurately describe the semantic knowledge of (1) how cross-enterprise business tasks are related to each other (apart from the categories they fall into), (2) how business goals are decomposed into more granular tasks, (3) how tasks and goals are linked to business documents, (4) which actors are authorized to transact on the tasks, and (5) the organizational structure of collaborating companies. A methodology which succinctly addresses these limitations is urgently needed. The proposed BOWL in Chapter 9 attempts to address this gap.

12.4 Comparisons with Related Work

Apart from the works reviewed in Chapter 6, there are other related works which further support the author's claim of the emerging awareness for the need of dynamic cBP formulation and the potential of addressing dynamic cBP formulation in BPM via planning and semantic descriptions such as ontologies.

12.4.1 **Planning Processes with Web Services**

A sign of emergence of the awareness in using planning as a business process modeling methodology can be found in the late-2009 work by Dr. Charles Petrie of Stanford Research Institute [295]. In this paper, it was claimed that “planning is an important approach to developing complex applications composed of web services, based upon semantic annotations of these services”. He showed the necessity of using planning technology by giving an example of web service composition that cannot be solved with flow-based approaches.

It was also noted that when you adopt planning as a way to dynamically synthesis business processes or even do Web service composition, the problems usually considered in the dominant flow-based approach literature (e.g. prevention of deadlocks, ensuring proper message exchange, etc) there is actually a very negligible need to worry about deadlocks or similar low-level problems. In a view similar to my observations in Chapter 6, 7, 8 and this chapter, his work attempted to debunk the current Web service composition researchers’ focus on flow-based theories such as Petri Net and Pi Calculus (*cf.* similar arguments in Section 3.4.1).

Even though he did not adopt a hybrid planning-ontological approach like this thesis, his experiments on the effectiveness of planning for a very similar problem proved to be an **important backing for this thesis**. It strongly supports the author’s views and arguments for the usage of planning methodologies to address dynamic formulation of cBPs in an SOA environment.

12.4.2 **Semantics of Business Vocabulary and Business Rules (SBVR)**

Recently in February 2008, the Semantics of Business Vocabulary and Business Rules (SBVR) [302] Version 1.0 Specification, a new formal logic with a natural language interface, received final approval at the December 2007 OMG meeting in Burlingame, California, and its specifications are now available on the OMG website [303]. The SBVR allows a “Business Community” (e.g. a company) to define the concepts, facts, and rules of its daily business operations and business processes. Given these definitions, the Semantics of Business Vocabulary (i.e. Taxonomy, Thesaurus, and Ontology) and the Business Rules (i.e. Operative Business Rules) can be created. SBVR facilitates the expression of business rules in natural textual structure English expressions like “*It is prohibited that a barred driver is a driver of a rental*”. “Barred driver”, “driver” and “rental” in this example are identified as symbols in the Business Vocabulary of that company. These symbols are then parsed through language rules and facts of logical formulations (e.g. “It is prohibited...” or “It is obligatory...”) are formed.

Facts of logical formulations are represented as Objects (a la object oriented programming). These objects are eventually written as XML definitions.

SBVR is a landmark milestone for OMG and is heavily influenced by the Business Rules research [304, 305], as it allows the non-graphical natural language to be used as a notation for its predominantly graphical standards (e.g. UML). The SBVR is dubbed to be used in other OMG standards like BPDM and Knowledge Discovery Metamodel (KDM) [302].

In the early days of the business process reengineering revolution, we witnessed the tradition business process modeling methodology, which mandates analysts to model graphical flows which will be translated into executable workflows running on business process engines. In the recent years, however, we witnessed the emergence of fact-oriented, rule-based approaches such as SBVR [] by the OMG (Object Modeling Group). Using enterprises policies as a basis, rules are described using natural language to model how these policies guide the achievement of business objectives. These clear, unambiguous descriptions in natural language can then be translated into other business process modeling representations for execution. This is a hallmark for business process modeling, as the standardisation of SBVR signifies the formal start of the awareness of natural language as a more semantically-aligned language over graphical models such as BPMN (Business Process Modeling Notation) [10, 11] and UML Activity Diagrams [70], which are adept at modeling flows, but are often void of considerations of semantics and constraints happening in reality. While SBVR is attempting to use natural language to formally model rules, Genesis attempts to model goals, and with the goal-modeling, utilise goal-driven techniques to generate business processes on the fly.

12.4.3 Web Service Composition Methods using Traditional HTN-Planners

The research closest to Genesis are methods using **HTN planners for Web Service compositions** [251, 252]. Generally, these approaches take in Web services that are initially defined in the semantic Web service description language, OWL-S, as inputs and then translates the Web service descriptions into traditional HTN planning domain descriptions (e.g. PDDL). After the translation, the adopted HTN planner (i.e. SHOP2 [252], XPlan [251]) will further decompose compound tasks into atomic tasks. This results in abstract non-executable Web service composition plans. However, it is important to note that these approaches still **fail to address the high-level business goals and collaboration criteria** frequently encountered in real-life. Furthermore, they still require manually defined, operational-level OWL-S Web service descriptions as the starting point for decomposition. These methods also require **additional steps translating** OWL-S to a domain description before the HTN planners can work. This is inefficient and may cause unnecessary restrictions due to the differences in the

nature of OWL-S descriptions and a planning domain description. On the other hand, the Genesis approach ontologized the domain (i.e. B2B collaboration tasks) descriptions directly as BOWL, and hence is not restricted to such translation restrictions.

12.4.4 **Ontological and Reference Model Approaches**

Process reference models conceptually (1) capture the “*as-is*” state of a process and [306], (2) benchmarks best practices and (3) characterize the management practices and software solutions that result in such optimal performance [307, 308]. Dominant reference models include the *Supply Chain Operations Reference Model (SCOR)* [307], *Design Chain Operations Reference Models (DCOR)* [308], the *APQC Process Classification Framework (PCF)* [309], and the BP creation steps and repository found in the *MIT Process Handbook* [282].

SCOR is based on 5 high-level management business processes: ***plan, source, return, make, deliver and return***. It postulates that the cross enterprise linkage of these five fundamental business processes make up the supply chain (*from the perspective of the collaboration from your supplier’s supplier to your customer’s customers*). Like SCOR, DCOR has three levels of detail and the topmost level is also represented by 5 main management processes: ***Plan, Research, Design, Integrate, and Amend***.

The MIT Process Handbook, mainly active from the late nineties till the mid 2000’s, attempts to **organize useful knowledge about business** through (1) the collation of generic models of typical business activities in different businesses, (2) specific case examples of unusual business practices, and (3) frameworks for classifying all this knowledge [310]. It is available in two media forms: **The book version** [282] mainly focuses on research in process design best practices [311]. **The online version**, also known as the “***eBusiness Process Handbook***” [312], is a subset of the book’s contents, and facilitates the creation of new business models guided by the basic “*design, buy, make, manage, sell*” formalism. An example is the combination of “*sell online*” and “*sell books*” to form the Amazon online bookstore business model.

Another related work is the PLIB Ontology [313], which unifies B2B e-commerce standards such as RosettaNet and ebXML through a dictionary model. This approach promotes the use of a meta-dictionary model which provides a uniform representation of ontologies that may be exchanged and integrated to constitute a common shared ontology. However, their research failed to mention practical applications and adoptions for their meta-dictionaries.

12.4.5 Other Related Approaches

Soffer and Wand's goal-driven business process modelling approach [301] rightly pointed out that many current techniques are based on graphical notations and are driven by experience and practice rather than by theoretical foundations. Their proposed model enables a systematic analysis of goal reachability, including identification of possible reasons for failure and possible design corrections.

Another related work is the research on business process choreography by Jung et al [314]. In this approach, an interface protocol to represent Workflow Management Coalition (WfMC) [315]'s three interoperability patterns (*chain*, *nested* and *synchronised*) is proposed. Their approach allows the collaboration of existing processes, usually managed by an enterprise's own workflow management system. While they approach B2B collaboration from the business level, much of their methodology is still manual, not scalable and lacks the dynamic business process integration capability.

12.5 Limitations of Study

The modelling technique used to create BOWL can be applied into different industries requiring B2B collaborations, can develop their own BOWL for dynamic business process creation. Because of this, the Genesis approach is also contingent upon the creation of a well-validated BOWL for a particular industry. The quality and comprehensiveness of the BOWL modelling will determine the quality of the business processes generated. This is also known as '*brittleness*', which is an inherent problem in HTN planning [224]. However, this can be minimised by a true industry- or sector-wide collaboration and aggregation of industry data.

Aside from brittleness, the BOWL and Genesis hybrid method inherited another AI planning limitation: the differences in the actual environment between *planning time* and *execution time*. There are two main sub problems from the perspective of this research:

1. From the time of receiving the goals to the actual generation of the plans, there may be some changes to the requirements or the conditions of the real environment. As the current Genesis algorithm should generate a plan within 3-5 minutes, the differences which may occur may not be too drastic to impact the operation. The user, when faced with changes to criteria during the planning phase, will just have to "re-plan" a new cBP for the updated criteria. When the system is deployed in heavy load environments, it may require a longer planning time. As of now, a simple reformulation of cBP's via the Genesis system will suffice.
2. As mentioned in Section 11.4.1.5, there is also the inability to predict the number of loops experienced during executions of cBPs. If the entire cBP was to be automated,

there needs to be termination of the loop, to prevent infinite loops from occurring during runtime. There is currently no solution for this except to preset the maximum number of loops in BOWL. This was also recently identified by Petrie in 2009 [188].

HTN planning, like most AI planning techniques, was meant to handle single user planning of execution steps. Therefore, the generated plans usually assume a single actor (i.e. the user) and do not handle collaborative partners. BOWL and Genesis has changed this by adding roles into the plans, i.e. cBPs. The proposed technique solves the most fundamental one-to-one collaboration, but has not handled tiers beyond the two collaborators in contention. For example, the supplier of the supplier, and the buyer of the buyer and so on.

While we have achieved the dynamic formulation of processes across organisations, we need to address three present limitations as future work:

1. Decomposition of tasks within organisations.
2. Enhancing the modelling of preconditions and effects of each business task in BOWL.
3. Inability to add new, change existing or customizable the list of criteria for each B2B context.

Chapter 13. Conclusions and Future Work

13.1 Highlights of the research

A hybrid ontology-HTN planning approach is proposed to dynamically formulate cross-enterprise business processes based on high-level user business goals and criteria. In order to do this, three (3) sub-objectives/ research challenges (RC) were identified and addressed:

- ***RC1 - encapsulation of high-level business goals and constraints,***

Two essential B2B goals (*Buy and Sell*) and eight B2B criteria namely *Item Price, Order Lead Time, Collaboration Mode, Product (Type and Name), Quantity, Sourcing Mode, and the Process Type* were identified from Language/ Action Perspective (LAP) collaborative frameworks (one of which extracts essential tasks in business processes), and best practices in supplier selection, in industry.

Arising from the two essential B2B goals and eight B2B criteria, an *Essential Collaborative Business Process (ECBP) Framework* is proposed. The ECBP Framework embodies five (5) generic phases of any cBP collaboration: *Discovery, Establish Contact, Negotiation, Contract Agreement and Contract Fulfilment*. Each of these five phases contains essential ‘steps’ which are complete and generic. Through the *ECBP Framework*, a set of commonly-adopted cBPs was established from reference models such as RosettaNet, ebXML and OAGIS.

- ***RC2- decomposition of high-level business goals into granular executable tasks***

Hierarchical task network (HTN) planning takes in goals and a set of criteria and dynamically decomposes high-level goals into granular executable tasks without the need to ‘know’ any predefined levels to decompose into. In HTN Planning, the decomposition algorithm which refers to a knowledge base modelled as a HTN. Within the HTN knowledge base, tasks are modelled as compound and primitive tasks. The algorithm which decomposes compound tasks into primitive ones compares the user’s criteria against the values of criteria stored in the *methods* of decomposition. The *method* that best fits the user criteria is chosen, generating single-level sequences of primitive tasks. However, the original HTN planning cannot distinguish between authorized roles of a business task (*a.k.a.* ‘actors’) and are devoid of control flow structures. These are two severe limitations in real-life B2B collaborations, which involve 2 fundamental roles (*Buyer and Seller*) and nested control flow governing the decision routes of tasks.

Hence, there is a need in this research to enhance the original HTN planning algorithm into the proposed Genesis algorithm. Basic control flows from Workflow Patterns were embedded in

the original HTN planning algorithm. Furthermore, actors were embedded in each primitive task during decomposition.

The original HTN Planning algorithm only understands Problem Domain Description Language (PDDL) format. As PDDL is not Web ready, HTN cannot generate executable tasks in the Web environment, and so OWL was chosen to substitute for PDDL. OWL enables semantic descriptions of cBP components. This new HTN knowledge base is known as Business-OWL (BOWL). In BOWL, PDDL concepts were migrated into OWL ontological descriptions of the relationships among primitive tasks, compound tasks, methods, criteria and effects. As a result, the ontologized structure is ready to generate outputs for SOA Web service orchestration, e.g. BPEL.

However, the structure of OWL does not allow nested control flows to be described succinctly around primitive tasks. Hence, BOWL extended the original OWL structures to accommodate nested control flows, thus enabling the Genesis algorithm to recursively write the complex control flows. Lastly, BOWL enhanced the task descriptions in traditional PDDL to include task actors.

During the creation of the BOWL ontology, a reference model list of cBP tasks was synthesized from three major B2B information exchange standards (RosettaNet, ebXML and OAGIS), and modelled into a process-oriented, hierarchical task network which enables dynamic cBP formulation.

- ***RC3- chaining of granular tasks in appropriate control flows, performed by the appropriate roles***

To enable these output primitive tasks to be executed directly by Web services, basic actions from BPEL (e.g. “invoke” of Web services) were included as properties of tasks described in BOWL.

The hybrid ontology-HTN Planning methodology was eventually implemented in a prototype called Genesis. Genesis is a Web-based application which takes in user goals and criteria *via* a GUI. Genesis generates three outputs formats; (i) abstract BPEL format for further appending of information for Web service execution, (ii) graphical flow format for visual inspection, and (iii) a tree-structure format depicting the task decomposition process. The prototype was able to generate cBPs according to different business goals. For example, the cBPs generated for a reverse auction mode of collaboration were different from the cBPs generated for a conventional source-and-buy mode of collaboration.

The hybrid ontology-HTN Planning methodology was verified by manual path testing. Path testing checks the full branch coverage in both the Genesis algorithm and the BOWL

knowledge base. The tests involved fourteen combinations of different goals and criteria for all 341 tasks and their associated methods. Following these tests, the prototype was validated *via* five leasing business processes of a regional pallet leasing company. The five pallet-leasing business processes generated by the BOWL and Genesis algorithm were found to be identical to the as-is business processes of the Company.

Therefore, the research demonstrates that, starting from high level business goals and criteria, it is possible to decompose these high level compound business tasks into granular executable tasks, appropriately chained by relevant control flows.

In the course of the author's research, several areas for future research were identified. These are covered in the following sections.

13.2 Future Work

Three substantive spin-off opportunities from this research are as follows:

13.2.1 *Modeling Transition Points between Public and Private Processes*

The current Genesis output is in the form of *cross-enterprise*, or *public business processes* across two collaborating companies. While this is undoubtedly the first step to dynamic B2B collaboration, further research on the linkages between these public business processes to the collaborating companies' (often confidential) private business processes is necessary. This will enable a seamless dynamic integration between companies.

One obstacle to achieving such seamless dynamic B2B integration is the wide variety of private business processes. Organisation structures and corporate culture vary from company to company. Hence, the same business processes (e.g. submitting a purchasing request) may be approached by different companies in varying approval routes, by different roles or even named differently across different companies. On the contrary, public (cross-enterprise) business processes, are rather standard across different industry sectors, e.g. purchase orders, invoices, etc.

Currently, private-public-private business process integrations are managed manually or automated via customised B2B software integrations. However, as these are inefficient, very time consuming and require high setup costs, customisation is not the answer to the demands of the supply chain. Instead of attempting to exhaustively cover all permutations of private business processes, one possible way is to focus on the "transition points" (TP) between public and private business processes. TP's act as the "middleware" between the Genesis output abstract BPEL file and the internal business processes of each collaborator. The process is

likened to the Universal Serial Bus (USB) port, which allows various types of hardware to be integrated seamlessly with personal computers.

As such, there is a need to investigate into proposed *Transition Points (TP)*, and if so, what types of TP's exist, and how they can integrate public business processes like the Genesis' output into a company's internal business processes. Such information can then be modelled as an OWL ontology which complements BOWL, thus enhancing the knowledge of the current system. Initial studies by the author revealed that research into Service Interaction Patterns [316] may be able to provide clues to dynamic integration of public-to-private process transitions.

13.2.2 Enhancing BOWL with Task Input-Behaviour-Output Description Ontologies

Each primitive task in Genesis' output abstract BPEL file is matched to Web services *via* the matching of *Web service name* and *description* with the values described in the abstract BPEL. While the method works, such matching can potentially become slow when there are many users. Therefore, a more efficient “primitive task-to-Web service” matching is needed.

To pre-empt this potential problem, more information about the *input* (i.e. preconditions and values needed), *behaviour* and *output* (i.e. effects to the values and status of the transaction) of each primitive task is needed so as to better pinpoint the exact Web service to match. Currently, to the author's best knowledge, a comprehensive treatise on the input-behaviour-output descriptions does not exist.

The *input-behaviour-output* information for all tasks descriptions in BOWL can be modelled into a separate OWL file. When integrated with BOWL, the *input-behaviour-output* information for each primitive task can serve as the semantic annotation for each task and can be used to scope down to an exact Web service that matches the properties being described in the input-behaviour-output file.

To create an ontology which describes task input-behaviour-output, we need to accomplish the following three tasks:

1. Taxonomise the generic types of *inputs* and the preconditions commonly found in primitive B2B tasks. To the author's best knowledge, there is currently no definitive taxonomy on B2B task inputs and preconditions.
2. Taxonomise the generic types of *outputs* and their effects (e.g. on document values, on supervised criteria values, etc.) commonly found in primitive tasks. To the author's

best knowledge, there is currently no definitive taxonomy on B2B task inputs and preconditions.

3. Identify and categorise the types of Web service functions and *behaviour* commonly found in primitive tasks. Clues for this can be found in the primitive action categories found in Web service orchestration standards such as WS-BPEL.

When all three categorisations are derived, we will be able to design an ontology which can complement BOWL's ontology during the Genesis algorithm's generation of cBP's. The tasks in the resulting abstract BPEL file will also be described more richly.

13.2.3 *Measurement of Quality of CBPs Generated*

While the current thesis focuses on generating cross-enterprise business processes, there is currently no way to measure 'how good' a recommended business process generated is. A way to measure the strength and the relevance to the real-life scenario will be desirable, and will definitely enhance the proposed BOWL+Genesis approach in this thesis.

13.2.4 *High-Level Language to Describe Goals and Criteria*

While the collection of necessary parameters *via* a JSP GUI is sufficient for the proof of concept of the Genesis system's algorithm, the GUI does not maximise the potential of Genesis. The GUI is unable to provide a flexible design of business processes. It has the following limitations:

- The number of options in the form is limited. They are insufficient to describe various types of user's requests and constraints.
- The user may wish to declare more features, descriptions about the products but the form filling method does not support this.
- The requests are sometimes complicated and long and edited from time to time in order to reflect changing business' needs in real life. The need of saving the previous request for later editing is essential. However, the current GUI does not support that purpose.
- Commands cannot be nested, and thus modeling of the buyers' buyer and the suppliers' supplier business process integration cannot be performed dynamically.

An alternative to the present Genesis GUI is a high-level language which enables flexible expressions of high-level business goals and constraints in order to effect the eventual formulation of collaborative business processes. This is likened to the Structured Query Language (SQL) which abstracts the high-level goals of database management from the low-level transactions carried out within the database. The proposed language should include the

rules and grammar syntax that allows a user to express business goals and criteria in an intentionally-simple English-like structured language. As such, the user can easily design the business requests in a more creative and freer way. The user's input will then be sent to Genesis for dynamic cBP formulation. The high-level language should consist of commands for (1) formulation of collaborative business processes, and (2) configuration of criteria and server configuration.

To test the feasibility of such an idea, the author created a simple version of such high level language – *CollaGen (Collaboration Generator Language)*. CollaGen's syntax definition in Extended Backus-Naur Form (EBNF) can be found in Appendix N. In CollaGen, each user's query contains two main components: the business goal and list of criteria. The business goal can be BUY or SELL (example in Figure 13.1) with the respective collaborative type (eg. new buyer, existing buyer with contract, new supplier).

```
SELL ('dining table', 'furniture', 'steel')
TO existing_buyer_without_contract
CONSTRAINT
[
  ITEM_PRICE = 100.01,
  QUANTITY < 1000,
  PROCESS_TYPE = Service_Provision,
  LEAD_TIME < 1,
  SOURCING_MODE = reverse_auction,
  LOCATION = {Singapore, Malaysia, Indonesia, Laos, Thailand},
]
```

Figure 13.1: An example of SELL command in CollaGen syntax

Initial trials were carried out and a text command input console was developed for storing, editing and reuse of CollaGen syntax. High-level goals and criteria expressed in CollaGen were able to generate cBP's *via* Genesis as expected. Despite the achievements of Buy and Sell commands, many other features need to be developed: for instance (1) customisation of configuration files for adding, removing or editing of criteria for other context, (2) manipulation of BOWL, and (3) the creation, edition and deletion of Web services on a server.

References

- [1] W. M. P. van der Aalst, *et al.*, "Business Process Management: A Survey," in *Business Process Management: International Conference, BPM 2003*, Eindhoven, the Netherlands, 2003.
- [2] J. B. Hill, *et al.*, "Magic Quadrant for Business Process Management Suites, 2007," Gartner Research 2007.
- [3] T. Payne, *et al.*, "Hype Cycle for Supply Chain Management, 2009," Stamford, CT, USA G00169602, 30 July 2009 2009.
- [4] J. R. Rymer and C. Moore, *The Dynamic Business Applications Imperative: The Principles Of "Design For People, Build For Change" Will Anchor A New Generation Of Business Applications*: Forrester Research, 2007.
- [5] C. Goh, *et al.*, "Key issues in manufacturing enterprise integration-the Singapore perspective," 2005, pp. 390-395.
- [6] W. M. P. van der Aalst, *et al.*, "Business Process Management: A Survey," in *International Conference on Business Process Management (BPM 2003)*, Berlin, 2003, pp. 1-12.
- [7] M. Havey, *Essential Business Process Modeling*, 1st ed. Sebastopol, CA 95472, USA: O'Reilly Media, Inc., 2005.
- [8] D. Georgakopoulos, *et al.*, "An overview of workflow management: From process modeling to workflow automation infrastructure," *Distributed and Parallel Databases*, vol. 3, pp. 119-153, 1995.
- [9] T. W. Malone, *et al.*, "Tools for Inventing Organizations: Towards a Handbook of Organizational Processes," *Management Science*, vol. 45, pp. 425-443, 1999.
- [10] OMG. (2004, 14 September 2007). *Business Process Modeling Notation (BPMN)*. Available: <http://www.bpmn.org/>
- [11] P. Wohed, *et al.*, "On the Suitability of BPMN for Business Process Modelling," *Business Process Management*, pp. 161-176, 2006.
- [12] R. K. L. Ko, "A computer scientist's introductory guide to business process management (BPM)," *ACM Crossroads*, vol. 15, pp. 11-18, 2009.
- [13] R. K. L. Ko, *et al.*, "Business Process Management (BPM) Standards: A Survey," *Business Process Management Journal*, vol. 15, 2009.
- [14] M. Hammer and J. Champy, *Reengineering the corporation : a manifesto for business revolution*. New York: HarperBusiness, 1993.
- [15] M. Hammer and J. Champy, *Reengineering the Corporation: A Manifesto for Business Revolution*. New York, United States of America: Harper Business, 1993.
- [16] T. H. Davenport, *Process innovation : reengineering work through information technology*. Boston, Mass.: Harvard Business School Press, 1993.
- [17] M. A. Ould, *Business Process: Modelling and Analysis for Re-engineering and Improvement*. Baffins Lane, Chichester, England: John Wiley & Sons Ltd, 1995.
- [18] R. N. Anthony, *Planning and Control Systems: A Framework for Analysis*: Harvard Business School, 1965.
- [19] R. N. Anthony, *et al.*, *Management control systems*: Irwin Homewood, Ill, 1995.
- [20] C. K. Prahalad and G. Hamel, "The core competence of the corporation," *Harvard Business Review*, vol. 68, pp. 79-91, 1990.
- [21] M. Rowell, "OAGIS: A Canonical Business Language," *XML Journal*, vol. 3, 2002.
- [22] I. Davies, *et al.*, "How do practitioners use conceptual modeling in practice?," *Data and Knowledge Engineering*, vol. 58, pp. 358-380, 2006.
- [23] A. Church and S. C. Kleene, "Formal definitions in the theory of ordinal numbers," *Fundamenta mathematicae*, 1936.
- [24] C. A. Petri, "Kommunikation mit Automaten," PhD, Rheinisch-Westfälisches Institut f. instrumentelle Mathematik an d. Univ, 1962.
- [25] R. Milner, *A Calculus of Communicating Systems*: Springer-Verlag New York, Inc. Secaucus, NJ, USA, 1982.

- [26] R. Milner, "Elements of interaction: Turing award lecture," *Communications of the ACM*, vol. 36, pp. 78-89, 1993.
- [27] R. R. Milner, *Communicating and Mobile Systems: The Pi Calculus*: Cambridge University Press, 1999.
- [28] F. Leymann, "WSFL–Web Services Flow Language," in *IBM Software Group, Whitepaper*, ed, 2001.
- [29] R. Farahbod, *et al.*, "Specification and Validation of the Business Process Execution Language for Web Services," *Abstract State Machines*, vol. 3052, pp. 79–94, 2004.
- [30] OASIS. (2007, 14 September 2007). *Web Services Business Process Execution Language (WSBPEL)*. Available: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
- [31] P. Huber, *et al.*, "Hierarchies in coloured Petri nets," *Advances in Petri Nets 1990*, pp. 313-341, 1990.
- [32] K. Jensen, "Coloured petri nets," *Petri nets: central models and their properties*, pp. 248-299, 1992.
- [33] K. Jensen, *Coloured Petri nets: basic concepts, analysis methods, and practical use*: Springer Verlag, 1996.
- [34] W. M. P. Van Der Aalst, "The application of Petri nets to workflow management," *Journal of Circuits, Systems and Computers*, vol. 8, pp. 21-66, 1998.
- [35] W. van der Aalst, *et al.*, "Advanced Workflow Patterns," in *Cooperative Information Systems*, ed, 2000, pp. 18-29.
- [36] YAWL. (2007, 2 Feb 2008). *Yet Another Workflow Language (YAWL) Homepage*. Available: <http://www.yawl-system.com/>
- [37] W. M. P. van der Aalst, "Pi calculus versus Petri nets: Let us eat “humble pie” rather than further inflate the “Pi hype”," *Unpublished paper*, 2005.
- [38] A. Gill, *Introduction to the Theory of Finite-state Machines*: McGraw-Hill New York, 1962.
- [39] J. E. Hopcroft and J. D. Ullman, "Formal languages and their relation to automata," *Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA*, p. 262, 1969.
- [40] P. Andrew, *et al.*, "Presenting Windows Workflow Foundation," 2005.
- [41] K. Scribner, "Microsoft® windows® workflow foundation step by step," 2007.
- [42] T. Cai, *et al.*, "DartFlow: A Workflow Management System on the Web using Transportable Agents," *Dartmouth College, Hanover, NH*, 1996.
- [43] D. Wodtke and G. Weikum, "A formal foundation for distributed workflow execution based on state charts," *Proceedings of the 6th International Conference on Database Theory*, pp. 230-246, 1997.
- [44] A. Wombacher, *et al.*, "Matchmaking for business processes," Newport Beach, CA, USA, 2003, pp. 7-11.
- [45] A. Wombacher, *et al.*, "Matchmaking for business processes based on choreographies," Taipei, Taiwan, 2004, pp. 359-368.
- [46] A. Wombacher, *et al.*, "Matchmaking for business processes based on conjunctive finite state automata," *International Journal of Business Process Integration and Management*, vol. 1, pp. 3-11, 2005.
- [47] F. B. Gilbreth and L. M. Gilbreth, *Applied Motion Study: A Collection of Papers on the Efficient Method to Industrial Preparedness*: Sturgis & Walton company, 1917.
- [48] F. B. Gilbreth and L. M. Gilbreth, "Process Charts and Their Place in Management," *Mechanical Engineering*, vol. 70, pp. 38–41, 1922.
- [49] S. White, "Introduction to BPMN," BPMI2004.
- [50] N. Russell, *et al.*, "On the suitability of UML 2.0 activity diagrams for business process modelling," *Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling-Volume 53*, pp. 95-104, 2006.
- [51] M. Dumas and A. H. M. ter Hofstede, "UML Activity Diagrams as a Workflow Specification Language," *Uml 2001: The Unified Modeling Language: Modeling Languages, Concepts, and Tools: 4th International Conference, Toronto, Canada, October 1-5, 2001: Proceedings*, 2001.

- [52] A. W. Holt, *et al.*, "Coordination system technology as the basis for a programming environment," *Electrical Communication*, vol. 57, pp. 307-14, 1983.
- [53] Y. H. Cho, *et al.*, "Role-based approach to business process simulation modeling and analysis," *Computers & Industrial Engineering*, vol. 35, pp. 343-346, 1998.
- [54] D. G. Wastell, *et al.*, "A methodology for business process redesign: experiences and issues," *The Journal of Strategic Information Systems*, vol. 3, pp. 23-40, 1994.
- [55] M. A. Ould, *Business processes : modelling and analysis for re-engineering and improvement*. Chichester: Wiley, 1995.
- [56] N. Chomsky, "On certain formal properties of grammars," *Information and control*, vol. 2, pp. 137-167, 1959.
- [57] J. Pyke and R. Whitehead, "Does better math lead to better business processes?," Novembre, Workflow Management Coalition2003.
- [58] M. Grüninger and C. Menzel, "The process specification language (PSL) theory and applications," *AI Magazine*, vol. 24, pp. 63-74, 2003.
- [59] C. Schlenoff, *et al.*, "The Process Specification Language (PSL): Overview and Version 1.0 Specification," *ISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD (2000)*, 2000.
- [60] J. M. Wing, "Title," unpublished|.
- [61] D. Sangiorgi and D. Walker, *The π -calculus: a Theory of Mobile Processes*: Cambridge University Press, 2001.
- [62] C. Schlenoff, *et al.*, "The essence of the process specification language," *Transactions of the Society for Computer Simulation*, vol. 16, pp. 204-16, 1999.
- [63] J. H. Frank, *et al.*, "Business processes definition metamodel concepts and overview," *IBM's Proposal in Response to the OMG's RFP for a Business Process Definition Metamodel*, pp. 202004-05, 2004.
- [64] W. M. P. van der Aalst, *et al.*, "Pattern-Based Analysis of BPML (and WSCI)," *Queensland University of Technology, Brisbane: QUT Technical Report FIT-TR-2002-05*, 2002.
- [65] A. Arkin, "Business Process Modeling Language (BPML)," *BPML.org*, 2002.
- [66] A. Barros, *et al.*, "A Critical Overview of the Web Services Choreography Description Language (WS-CDL)," *BPTrends*, vol. 2005, 2005.
- [67] N. Kavantzaz, *et al.*, "Web Services Choreography Description Language (WS-CDL)," *Specification*, vol. 1, 2004.
- [68] A. Arkin, *et al.*, "Web Service Choreography Interface (WSCI) 1.0," *Standards proposal by BEA Systems, Intalio, SAP, and Sun Microsystems*, 2002.
- [69] S. Thatte, "XLANG: Web Services for Business Process Design," ed, 2001.
- [70] OMG, "UML 2.0 Superstructure Specification," 2004.
- [71] OMG, "Unified Modeling Language (UML) Resource Page," 2004.
- [72] J. L. H. Oei, "The Meta Model Hierarchy: A Framework for Information Systems Concepts and Techniques," 1992.
- [73] J. Bowen. The Formal Methods Virtual Library [Online].
- [74] J. R. Searle, *Speech acts: an essay in the philosophy of language*. London,: Cambridge U.P., 1969.
- [75] B. T. Pentland, "Building process theory with narrative: From description to explanation," *Academy of Management Review*, vol. 24, pp. 711-724, Oct 1999.
- [76] B. T. Pentland, "Sequential variety in work processes," *Organization Science*, vol. 14, pp. 528-540, Sep-Oct 2003.
- [77] E. Black, *et al.*, "Procedure for quantitatively comparing the syntactic coverage of English grammars," *Proceedings of the workshop on Speech and Natural Language*, pp. 306-311, 1991.
- [78] J. r. Habermas, "The theory of communicative action," ed London: Heinemann, 1984, pp. xlii, 465.
- [79] R. Medina-Mora, *et al.*, "The ActionWorkflow approach to workflow management technology," in *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '92)*, Toronto, Ont., Canada, 1992, pp. 281-288.

- [80] V. E. Van Reijswoud, *et al.*, "Communicative action-based business process and information systems modelling with DEMO," *Information Systems Journal*, vol. 9, pp. 117-138, 1999.
- [81] J. L. G. Dietz, "DEMO: Towards a discipline of organization engineering," *European Journal of Operational Research*, vol. 128, pp. 351-363, 2001.
- [82] T. Winograd, "A language/action perspective on the design of cooperative work," presented at the Proceedings of the 1986 ACM conference on Computer-supported cooperative work, Austin, Texas, 1986.
- [83] OASIS, "Service Oriented Architecture Reference Model," Technical report, OASIS, June 20052005.
- [84] J. L. G. Dietz, "The what and why of modelling business processes," *Dynamic Enterprise Modeling*, 1996.
- [85] J. L. G. Dietz, "The deep structure of business processes," *Communications of the ACM*, vol. 49, pp. 58-64, 2006.
- [86] J. L. G. Dietz, "Generic Recurrent Patterns in Business Processes," *Business Process Management: International Conference, BPM 2003, Eindhoven, the Netherlands, June 26-27, 2003: Proceedings*, 2003.
- [87] G. Goldkuhl, "The six phases of business processes: business communication and the exchange of value," 1998.
- [88] R. Medina-Mora, *et al.*, "The action workflow approach to workflow management technology," *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pp. 281-288, 1992.
- [89] H. Weigand and W. J. van den Heuvel, "Meta-patterns for electronic commerce transactions based on the Formal Language for Business Communication (FLBC)," *International Journal of Electronic Commerce*, vol. 3, pp. 45-65, 1998.
- [90] J. L. G. Dietz and N. Habing, "A meta ontology for organizations," *Workshop on Modeling Inter-Organizational Systems (MIOS), LNCS*, vol. 3292, 2004.
- [91] S. A. Moore, "Testing speech act theory and its applicability to EDI and other computer-processable messages," *System Sciences, 1996., Proceedings of the Twenty-Ninth Hawaii International Conference on*, vol. 2, 1996.
- [92] D. Te'eni, "The language-action perspective as a basis for communication support systems," *Communications of the ACM*, vol. 49, pp. 65-70, 2006.
- [93] H. Weigand, *et al.*, "B2B Negotiation Support: The Need for a Communication Perspective," *Group Decision and Negotiation*, vol. 12, pp. 3-29, 2003.
- [94] H. Weigand and W. J. van den Heuvel, "Cross-organizational workflow integration using contracts," *Decision Support Systems*, vol. 33, pp. 247-265, 2002.
- [95] K. Lyytinen, "The Struggle with the Language in the IT - Why is LAP not in the Mainstream?," in *9th International Working Conference on the Language-Action Perspective on Communication Modelling (LAP 2004)*, Rutgers University, The State University of New Jersey, New Brunswick, NJ, 2004.
- [96] F. Flores and J. J. Ludlow, "Doing and Speaking in the Office.," in *Decision Support Systems: Issues and Challenges*, 1980, pp. 95-118.
- [97] M. Schoop, "An introduction to the language-action perspective," *ACM SIGGROUP Bulletin*, vol. 22, pp. 3-8, 2001.
- [98] P. Wilson, *Computer Supported Cooperative Work: An Introduction.*: Kluwer Academic Pub. , 1991.
- [99] F. Flores, *et al.*, "Computer Systems and the Design of Organizational Interaction," *ACM Transactions on Office Information Systems*, vol. 6, pp. 153-172, 1988.
- [100] E. Lehtinen and K. Lyytinen, "Action based model of information system," *Information Systems*, vol. 11, pp. 299-317, 1986.
- [101] G. Goldkuhl and K. Lyytinen, "A language action view of information systems," *Ginzberg, M., Ross, C. A. (Eds., 1982): Proceedings of the 3rd international conference on information systems, TIMS/SMIS/ACM*, 1982.
- [102] H. Weigand, "LAP: 10 years in retrospect," in *The Language Action Perspective on Communication Modelling*, Kiruna, Sweden, 2005.

- [103] T. Winograd and F. Flores, *Understanding Computers and Cognition: A New Foundation for Design*: Intellect Books, 1986.
- [104] T. Winograd and F. Flores, *Understanding computers and cognition : a new foundation for design*. Norwood, N.J.: Ablex Pub. Corp., 1986.
- [105] L. Suchman, "Do Categories Have Politics? The Language/Action Perspective Reconsidered," *Human Values and the Design of Computer Technology*, 1997.
- [106] R. P. Carasik and C. E. Grantham, "A case study of CSCW in a dispersed organization," *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 61-66, 1988.
- [107] E. Auramäki, *et al.*, "A speech-act-based office modeling approach," *ACM Transactions on Information Systems (TOIS)*, vol. 6, pp. 126-152, 1988.
- [108] V. E. van Reijswoud and H. B. F. Mulder, "Speech Act Based Communication and Information Modelling with DEMO," *Australasia Journal of Information Systems*, vol. 6, pp. 89-102, 1998.
- [109] Z. Harris, "Discourse analysis," *Language*, vol. 28, pp. 1-30, 1952.
- [110] E. Auramäki, *et al.*, "Modelling offices through discourse analysis: the SAMPO approach," *The Computer Journal*, vol. 35, pp. 342-352, 1992.
- [111] E. Lehtinen and K. Lyytinen, "SAMPO project: A speech act based IA methodology with computer aided tools," *Rep. WP-3, Dept. of Computer Science, Univ. of Jyväskylä*, 1983.
- [112] H. Weigand, *et al.*, "Modelling Electronic Commerce Transactions--A Layered Approach," 1998.
- [113] G. Goldkuhl, "Generic Business Frameworks and Action Modelling," presented at the Proceedings of the First International Workshop on Communication Modeling - The Language/Action Perspective., Tilburg, Netherlands, 1996.
- [114] M. Lind and G. Goldkuhl, "Generic Layered Patterns for Business Modelling," *Proceedings of the Sixth International Workshop on the Language-Action Perspective on Communication Modelling (LAP 2001)*, 2001.
- [115] G. Goldkuhl, "Generic business frameworks and action modelling," Tilburg, Netherlands, 1997.
- [116] L. Wittgenstein, *Philosophical investigations*: Basil Blackwell, London, 1953.
- [117] J. Dietz, *et al.*, "A Conceptual Framework for the Continuous Alignment of Business and ICT," *Whitepaper Technical University Delft & Cap Gemini*, 1999.
- [118] J. L. G. Dietz, "Business modelling for business redesign," Wailea, HI, USA, 1994, pp. 723-32.
- [119] K. Ven and J. Verelst, "The Adoption of DEMO: A Research Agenda," *Advances in Enterprise Engineering III*, pp. 157-171, 2009.
- [120] J. L. G. Dietz, "The deep structure of business processes," *Communications of the ACM*, vol. 49, pp. 59-64, 2006.
- [121] R. Ettema and J. Dietz, "ArchiMate and DEMO--Mates to Date?," *Advances in Enterprise Engineering III*, pp. 172-186, 2009.
- [122] M. K. Chang and C. C. Woo, "A speech-act-based negotiation protocol: design, implementation, and test use," *ACM Trans. Inf. Syst.*, vol. 12, pp. 360-382, 1994.
- [123] T. Ballmer and W. Brennenstuhl, *Speech Act Classification. A study in the Lexical analysis of English speech activity verbs*. vol. 8: Springer Verlag, Berlin-Heidelberg-New York, 1981.
- [124] J. Allwood, "A Critical Look at Speech Act Theory," *Logic, Pragmatics and Grammar. Lund: Studentlitteratur*, pp. 53-69, 1977.
- [125] J. Allwood, "Review of Ballmer, T. and Brennenstuhl, W. Speech Act Classification," *Journal of Semantic*, vol. 1, pp. 287-290, 1982.
- [126] S. O. Kimbrough and S. A. Moore, "On automated message processing in electronic commerce and work support systems: speech act theory and expressive felicity," *ACM Transactions on Information Systems (TOIS)*, vol. 15, pp. 321-367, 1997.
- [127] ANSI, "X12, Electronic Data Interchange X12 Standards," ed: Draft Version, 1979.

- [128] C. L. Iacovou, *et al.*, "Electronic Data Interchange and Small Organizations: Adoption and Impact of Technology," *MIS Quarterly*, vol. 19, pp. 465-485, 1995.
- [129] S. Scala and R. McGrath Jr, "Advantages and disadvantages of electronic data interchange: an industry perspective," *Information and Management*, vol. 25, pp. 85-91, 1993.
- [130] UN/EDIFACT. (1990, 14 September 2007). *United Nations Directories for Electronic Data Interchange for Administration Commerce and Transport*. Available: http://www.unece.org/trade/untdid/texts/d100_d.htm
- [131] RosettaNet, "RosettaNet Implementation Framework: Core Specification," 1998.
- [132] RosettaNet Program Office, "Overview - Cluster, Segments, and PIPs (version 02.10.00)," ed, 2007.
- [133] M. Sundaram and S. S. Y. Shim, "Infrastructure for B2B exchanges with RosettaNet," *The 3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, pp. 21-22, 2001.
- [134] B. K. Gibb and S. Damodaran, *ebXML: Concepts and Application*: John Wiley & Sons, Inc. New York, NY, USA, 2002.
- [135] B. Hofreiter, *et al.*, "ebXML: status, research issues, and obstacles," *Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems, 2002. RIDE-2EC 2002. Proceedings. Twelfth International Workshop on*, pp. 7-16, 2002.
- [136] OASIS and UN/CEFACT, "ebXML Technical Architecture Specification v1.04," ed: OASIS, UN/CEFACT, 2001.
- [137] UN/CEFACT and OASIS, "ebXmL, Xmlbpss Version 1.01 Technical Specification," ed, 2002.
- [138] T. Erl, *Service-oriented architecture: concepts, technology, and design*: Prentice Hall PTR Upper Saddle River, NJ, USA, 2005.
- [139] M. P. Papazoglou, "Service-oriented computing: Concepts, characteristics and directions," 2003, pp. 3-12.
- [140] M. P. Papazoglou and D. Georgakopoulos, "Service-oriented computing," *Communications of the ACM*, vol. 46, pp. 25-28, 2003.
- [141] M. P. Papazoglou, *et al.*, "Service-oriented computing: State of the art and research challenges," *COMPUTER-IEEE COMPUTER SOCIETY-*, vol. 40, p. 38, 2007.
- [142] Q. Dai and R. J. Kauffman, "Business models for Internet-based B2B electronic markets," *International Journal of Electronic Commerce*, vol. 6, pp. 41-72, 2002.
- [143] J. Yang and M. P. Papazoglou, "Interoperation support for electronic business," *Communications of the ACM*, vol. 43, pp. 39-47, 2000.
- [144] OASIS and UN/CEFACT, "Business Process Specification Schema v1.01," ed: OASIS, UN/CEFACT, 2001.
- [145] D. Mertz. (2001, 28 Feb 2008). *Understanding ebXML*. Available: <http://www.ibm.com/developerworks/xml/library/x-ebxml/>
- [146] I. Ghalimi and D. McGoveran, "Standards and BPM," BPM.COM, Commentary2005.
- [147] A. Kotok, "Adoption of ebXML: Hiding in Plain Sight," ed: Creative Commons, 2007.
- [148] C. Bussler, "B2B Protocol Standards and their Role in Semantic B2B Integration Engines," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 2001.
- [149] D. Chen and F. Vernadat, "Standards on enterprise integration and engineering—state of the art," *International Journal of Computer Integrated Manufacturing*, vol. 17, pp. 235-253, 2004.
- [150] M. Rowell. (2003, December 5). *The Open Applications Group Integration Specification*. Available: <http://www.ibm.com/developerworks/xml/library/x-oagis/>
- [151] K. Mous, *et al.*, "High-level Business Processes for Agile B2B Collaboration."
- [152] P. S. Tan, *et al.*, "Issues and Approaches to Dynamic, Service-oriented Multi-enterprise Collaboration," in *Industrial Informatics, 2006 IEEE International Conference on*, 2006, pp. 399-404.
- [153] R. I. Benjamin, *et al.*, "Electronic data interchange: How much competitive advantage?," *Long Range Planning*, vol. 23, pp. 29-40, 1990.

- [154] T. Mukhopadhyay, *et al.*, "Business Value of Information Technology: A Study of Electronic Data Interchange," *MIS Quarterly*, vol. 19, pp. 137-156, 1995.
- [155] D. Simchi-Levi, *et al.*, *Managing the Supply Chain: The Definitive Guide for the Business Professional*: McGraw-Hill, 2004.
- [156] H. Tempelmeier, *Inventory Management in Supply Networks -- Problems, Models, Solutions*, : Norderstedt, 2006.
- [157] H. L. Lee, *et al.*, "The bullwhip effect in supply chains," *Sloan Management Review*, vol. 38, pp. 93-102, 1997.
- [158] P. C. Benjamin, *et al.*, "IDEF5 Method Report," *Knowledge Based Systems, Inc.*, vol. 21, p. 1994, 1994.
- [159] R. J. Mayer, *et al.*, "Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report," *Knowledge Based Systems Inc.(KBSI)*, September, 1995.
- [160] F. Pubs, "Integration definition for function modeling (IDEF0)," *Federal Information Processing Standards Publication*, vol. 183, 1993.
- [161] B. Orriens, *et al.*, "A Rule Driven Approach for Developing Adaptive Service Oriented Business Collaboration," in *Service-Oriented Computing - ICSOC 2005*, ed, 2005, pp. 61-72.
- [162] A. Sheth, *et al.*, "QoS for Service-Oriented Middleware," *Proceedings of the 6th World Multiconference on Systemics, Cybernetics abd Informatics (SCI02)*, pp. 528–534, 2002.
- [163] P. S. Tan, *et al.*, "Issues and Approaches to Dynamic, Service-oriented Multi-enterprise Collaboration," in *2006 IEEE International Conference on Industrial Informatics (INDIN 2006)*, Singapore, 2006, pp. 399-404.
- [164] T. Berners-Lee, *et al.*, "The Semantic Web," *Scientific American*, vol. 284, pp. 28-37, 2001.
- [165] C. Bussler, *et al.*, "A conceptual architecture for semantic web enabled web services," *ACM SIGMOD Record*, vol. 31, pp. 24-29, 2002.
- [166] J. Cardoso, *et al.*, "Academic and Industrial Research: Do their Approaches Differ in Adding Semantics to Web Services," *Semantic Web Process: powering next generation of processes with Semantics and Web services*, J. Cardoso and SA, Editors, pp. 14-21, 2005.
- [167] J. Cardoso and A. Sheth, *Semantic Web Services, Processes and Applications*: Springer, 2006.
- [168] M. Schoop, *et al.*, "The pragmatic web: a manifesto," *Communications of the ACM*, vol. 49, pp. 75-76, 2006.
- [169] M. P. Singh, "The Pragmatic Web: Preliminary Thoughts," *Proc. of the NSF-EU Workshop on Database and Information Systems Research for Semantic Web and Enterprises, April 3-5, Amicalolo Falls and State Park, Georgia*, 2002.
- [170] M. P. Singh, "The pragmatic web," *Internet Computing, IEEE*, vol. 6, pp. 4-5, 2002.
- [171] P. Grefen, *CrossFlow: Cross-organizational Workflow Management in Dynamic Virtual Enterprises*: University of Twente Host; Centre for Telematics and Information Technology, University of Twente, 2000.
- [172] R. K. L. Ko, *et al.*, "Genesis–Dynamic Collaborative Business Process Formulation Based on Business Goals and Criteria," presented at the IEEE Service Cup 2009, Los Angeles, California, 2009.
- [173] R. K. L. Ko, *et al.*, "Dynamic Collaborative Business Process Formulation via Ontologised Hierarchical Task Network (HTN) Planning," presented at the 7th IEEE International Conference on Web Services (ICWS 2009), Los Angeles, California, 2009.
- [174] J. C. Corrales, *et al.*, "BPEL processes matchmaking for service discovery," Montpellier, France, 2006, pp. 237-54.
- [175] W. Jian and W. Zhaohui, "Similarity-based Web service matchmaking," Orlando, FL, USA, 2005, pp. 287-94.

- [176] A. Wombacher, *et al.*, "IPSI-PF: a business process matchmaking engine," San Diego, CA, USA, 2004, pp. 137-45.
- [177] Z. Zhile, *et al.*, "A comprehensive framework for dynamic Web services integration," Zurich, Switzerland, 2006, p. 10 pp.
- [178] E. Auchard, "Google expands personalization with iGoogle," ed: Reuters, 2007.
- [179] A. Goebel and D. Ritthaler, *SAP enterprise portal*: Galileo Press, 2005.
- [180] S. Derong, *et al.*, "Resolving heterogeneity of Web-service composition in network manufacturing based on ontology," *International Journal of Computer Integrated Manufacturing*, vol. 20, pp. 222-33, 2007.
- [181] J. Dorn, *et al.*, "Web service discovery and composition for virtual enterprises," *International Journal of Web Services Research*, vol. 4, pp. 23-39, 2007.
- [182] L. Padgham and W. Liu, "Internet collaboration and service composition as a loose form of teamwork," *Journal of Network and Computer Applications*, vol. 30, pp. 1116-1135, 2007.
- [183] M. Younas, *et al.*, "An efficient composition of Web services with active network support," *Expert Systems with Applications*, vol. 31, pp. 859-69, 2006.
- [184] Y. Chunyang, *et al.*, "Publishing and composition of atomicity-equivalent services for B2B collaboration," Shanghai, China, 2006, pp. 351-60.
- [185] Z. Maamar, "Toward an agent-based and context-oriented approach for Web services composition," *IEEE Transactions on Knowledge and Data Engineering*, pp. 686-697, 2005.
- [186] B. Srivastava and J. Koehler, "Web service composition-current solutions and open problems," 2003.
- [187] D. Wu, *et al.*, "Automating DAML-S web services composition using SHOP2," *The SemanticWeb-ISWC 2003*, pp. 195-210, 2003.
- [188] C. Petrie, "Planning Process Instances with Web Services," presented at the ICEIS 2009 International Conference on Enterprise Information Systems, Milan, Italy, 2009.
- [189] M. C. Daconta, *et al.*, *The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management*: Wiley Publishing Inc, Indianapolis. Indiana, 2003.
- [190] J. Cardoso and A. Sheth, "Introduction to Semantic Web Services and Web Process Composition," *Semantic Web Process: powering next generation of processes with Semantics and Web services*, J. Cardoso and AP Sheth, Editors, pp. 1-13, 2005.
- [191] T. Bray, *et al.*, "Extensible Markup Language (XML) 1.0," *W3C Recommendation*, vol. 6, 2000.
- [192] O. Lassila and R. R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification," *W3C Recommendation*, vol. 22, pp. 2004-03, 1999.
- [193] S. Bechhofer, *et al.*, "OWL Web Ontology Language Reference," *W3C Recommendation*, vol. 10, pp. 2006-01, 2004.
- [194] J. De Bruijn, *et al.*, "Web service modeling ontology (WSMO)," *W3C member submission*, vol. 3, 2005.
- [195] C. Feier, *et al.*, "Towards intelligent web services: The web service modeling ontology (WSMO)," 2005, pp. 23-26.
- [196] J. Cardoso, "Quality of Service and Semantic Composition of Workflows," 215, Department of Computer Science. , University of Georgia, Athens, GA, 2002.
- [197] J. Cardoso, *et al.*, "Quality of service for workflows and web service processes," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 1, pp. 281-308, 2004.
- [198] J. Cardoso and A. Sheth, "Semantic E-Workflow Composition," *Journal of Intelligent Information Systems*, vol. 21, pp. 191-225, 2003.
- [199] C. Bussler, *B2B Integration: Concepts and Architecture*: Springer, 2003.
- [200] C. Bussler, "The Role of B2B Protocols in Inter-Enterprise Process Execution," *Technologies for E-Services: Second International Workshop, TES 2001, Rome, Italy, September 14-15, 2001: Proceedings*, 2001.

- [201] M. Brodie, *et al.*, "Semantically Enabled Service-Oriented Architectures: A Manifesto and a Paradigm Shift in Computer Science," *DERI-Digital Enterprise Research Institute, Tech. Rep. TR-2005-12-26, Dec*, pp. 2005-12, 2005.
- [202] M. Hepp, "Semantic Web and Semantic Web Services: Father and Son or Indivisible Twins?," *IEEE INTERNET COMPUTING*, pp. 85-88, 2006.
- [203] M. Hepp, *et al.*, "Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management," *ICEBE*, vol. 5, pp. 535–540, 2005.
- [204] R. McCool, "Rethinking the Semantic Web, Part 1. ," *IEEE Internet Computing*, pp. 86–88, 2005.
- [205] H. Weigand, "Two Decades of the Language Action Perspective," *Communications of the ACM*, vol. Vol. 49, pp. 45-46, 2006.
- [206] T. Winograd, "Designing a new foundation for design," *Communications of the ACM*, vol. 49, pp. 71-74, 2006.
- [207] G. Goldkuhl, "Action and media in interorganizational interaction," *Communications of the ACM*, vol. 49, pp. 53-57, 2006.
- [208] O. Saidani and S. Nurcan, "Towards Context Aware Business Process Modelling," *The 8th Workshop on Business Process Modelling, Development, and Support (BPMDS'07, (in association with CAISE'07), Springer Verlag (pub), June 11-12, 2007, Trondheim, Norway*, 2007.
- [209] L. Ardissono, *et al.*, "A Framework for the Management of Context-Aware Workflow Systems," presented at the Third International Conference on Web Information Systems and Technologies (WEBIST 2007), 2007.
- [210] P. S. Tan, *et al.*, "Context-enabled B2B Collaborations," presented at the IEEE International Conference on Services Computing, 2007 (SCC 2007), 2007.
- [211] H. Weigand, "Two Decades of the Language Action Perspective," *Communications of the ACM*, vol. Vol. 49, pp. 45-46, May 2006.
- [212] M. Koetsier, *et al.*, "Contracts for Cross-Organizational Workflow Management," presented at the Proceedings of the First International Conference on Electronic Commerce and Web Technologies, Lecture Notes In Computer Science, London, Greenwich, 2000.
- [213] Y. Hoffner, *et al.*, "An architecture for cross-organisational business processes," *Advanced Issues of E-Commerce and Web-Based Information Systems, 2000. WECWIS 2000. Second International Workshop on*, pp. 2-11, 2000.
- [214] A. Wombacher, *et al.*, "Transforming BPEL into annotated deterministic finite state automata for service discovery," San Diego, CA, USA, 2004, pp. 316-23.
- [215] A. Wombacher and B. Mahleko, "Ad-hoc business processes in Web services," 2003, pp. 101-105.
- [216] M. Hepp and D. Roman, "An Ontology Framework for Semantic Business Process Management," *Proceedings of Wirtschaftsinformatik*, vol. (1) 2007, pp. 423-440, 2007.
- [217] SemBiz. (2004, 12 Jan 2008). *SemBiz Project*. Available: <http://www.sembiz.org/index.html>
- [218] H. Tran, *et al.*, "View-based and Model-driven Approach for Reducing the Development Complexity in Process-Driven SOA," presented at the Int. Conf. on Business Process and Services Computing (BPSC), Leipzig, Germany, 2007.
- [219] Z. Yan, *et al.*, "BPMO: Semantic Business Process Modeling and WSMO Extension," presented at the The IEEE International Conference on Web Services 2007 (ICWS2007), Salt Lake City, Utah, USA, 2007.
- [220] A*STAR, *et al.* (2005, 20 Feb 2008). *An Ontology-augmented, Context-aware Web Services Framework for Agile Enterprise Collaboration*. Available: <http://vei.simtech.a-star.edu.sg/imss/>
- [221] H. Geffner, "Perspectives on artificial intelligence planning," 2002, pp. 1013-1023.
- [222] M. Ghallab, *et al.*, *Automated Planning: theory and practice*: Morgan Kaufmann Publishers, 2004.

- [223] D. Knuth, *The art of computer programming. Vol. 3, Sorting and Searching*: Addison-Wesley Reading, MA, 1973.
- [224] D. Smith, *et al.*, "Bridging the gap between planning and scheduling," *The Knowledge Engineering Review*, vol. 15, pp. 47-83, 2000.
- [225] M. Fox, "ISIS: a retrospective," *Intelligent scheduling*, pp. 3-28, 1994.
- [226] M. Fox and S. Smith, "ISIS: A knowledge-based system for factory scheduling," *Expert systems*, vol. 1, pp. 25-49, 1984.
- [227] M. Zweben and M. Fox, *Intelligent scheduling*: Morgan Kaufmann Pub, 1994.
- [228] K. Baker, *Elements of sequencing and scheduling*: Kenneth R. Baker, 1995.
- [229] M. Pinedo, *Scheduling: theory, algorithms, and systems*: Springer Verlag, 2008.
- [230] J. Blythe, "Decision-theoretic planning," *AI Magazine*, vol. 20, p. 37, 1999.
- [231] C. Boutilier, *et al.*, "Decision-theoretic planning: Structural assumptions and computational leverage," *Journal of Artificial Intelligence Research*, vol. 11, p. 94, 1999.
- [232] T. Dean and S. Kambhampati, "Planning and scheduling," *CRC Handbook of Computer Science and Engineering*, 1996.
- [233] D. Weld, "An introduction to least commitment planning," *AI Magazine*, vol. 15, p. 27, 1994.
- [234] D. Weld, "Recent advances in AI planning," *AI Magazine*, vol. 20, p. 93, 1999.
- [235] D. Nau, *et al.*, "Control strategies in HTN planning: Theory versus practice," 1998, pp. 1127-1133.
- [236] S. Russell, *et al.*, *Artificial intelligence: a modern approach*: Prentice hall Englewood Cliffs, NJ, 1995.
- [237] E. Sacerdoti, *A structure for plans and behavior*: Elsevier New York, 1977.
- [238] A. Tate, "Generating project networks," 1977, pp. 888-893.
- [239] D. Nau, *et al.*, "AI planning versus manufacturing-operation planning: A case study," 1995, pp. 1670-1676.
- [240] A. Tate, *et al.*, *O-Plan2: an open architecture for command, planning and control*: Citeseer, 1992.
- [241] D. Wilkins, "Can AI planners solve practical problems?," *Computational Intelligence*, vol. 6, pp. 232-246, 2007.
- [242] D. E. Wilkins, *Practical planning: extending the classical AI planning paradigm*: Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1988.
- [243] K. Erol, "Hierarchical task network planning: Formalization, analysis, and implementation," Ph.D., University of Maryland College Park, United States -- Maryland, 1995.
- [244] D. Nau, *et al.*, *Automated Planning: Theory & Practice*: Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2004.
- [245] D. S. Nau, *et al.*, "Control Strategies in HTN Planning: Theory Versus Practice," 1998, pp. 1127-1133.
- [246] M. Ghallab, *et al.*, "PDDL—the planning domain definition language," *AIPS-98 planning committee*, 1998.
- [247] D. McDermott, *et al.*, "PDDL-the planning domain definition language," *The AIPS-98 Planning Competition Comitee*, 1998.
- [248] I. Horrocks, "Ontologies and the semantic web," *Communications of the ACM*, vol. 51, pp. 58-67, 2008.
- [249] J. Heflin and J. Hendler, "A Portrait of the Semantic Web in Action," *IEEE INTELLIGENT SYSTEMS*, pp. 54-59, 2001.
- [250] M. Fox and D. Long, "PDDL2. 1: An extension to PDDL for expressing temporal planning domains," *Journal of Artificial Intelligence Research*, vol. 20, pp. 61-124, 2003.
- [251] M. Klusch, *et al.*, "Semantic Web Service Composition Planning with OWLS-Xplan," 2005.

- [252] E. Sirin, *et al.*, "HTN planning for Web Service composition using SHOP2," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 1, pp. 377-396, 2004.
- [253] M. S. Fox, *et al.*, "An organization ontology for enterprise modelling," *Simulating Organizations: Computational Models of Institutions and Groups*, pp. 131-152, 1998.
- [254] M. A. Bunge and M. Bunge, *Ontology*: D Reidel Pub Co, 1977.
- [255] Y. Wand and R. Weber, "Mario Bunge's Ontology as a Formal Foundation for Information Systems Concepts," *Studies on Mario Bunge's Treatise*, pp. 123-149, 1990.
- [256] M. Rosemann and P. Green, "Developing a meta model for the Bunge-Wand-Weber ontological constructs," *Information Systems*, vol. 27, pp. 75-91, 2002.
- [257] R. Weber, "Ontological foundations of information systems," 1997.
- [258] M. S. Fox and M. Gruninger, "On Ontologies and Enterprise Modelling," *International Conference on Enterprise Integration Modelling Technology*, vol. 97, 1997.
- [259] M. S. Fox and M. Gruninger, "Enterprise Modeling," *AI Magazine*, vol. 19, pp. 109-121, 1998.
- [260] J. L. G. Dietz, *Enterprise Ontology: Theory and Methodology*: Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006.
- [261] W. M. P. van der Aalst, *et al.*, "Workflow Patterns," *QUT Technical report*, vol. FIT-TR-2002-02, 2002 2002.
- [262] E. Gamma, *et al.*, *Design patterns: elements of reusable object-oriented software*: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1995.
- [263] W. M. P. van der Aalst, "Patterns and XPDL: A Critical Evaluation of the XML Process Definition Language," *QUT Technical report*, FIT-TR-2003-06, <http://www.citi.qut.edu.au/pubs/ce-xpdl.pdf>, Queensland University of Technology, Brisbane, 20032003.
- [264] P. Wohed, *Pattern-based Analysis of UML Activity Diagrams*: Beta, Research School for Operations Management and Logistics, 2004.
- [265] P. Wohed, *et al.*, "Pattern-Based Analysis of BPEL4WS," *Department of Computer and Systems Sciences, Stockholm University/The Royal Institute of Technology, Sweden*, 2003.
- [266] N. Russell, *et al.*, "Workflow Data Patterns," *QUT Technical report*, FIT-TR-2004-01, *Queensland University of Technology, Brisbane*, 2004.
- [267] N. Russell, *et al.*, "Workflow Resource Patterns," *BETA Working Paper Series, WP 127*, *Eindhoven University of Technology, Eindhoven*, 2004.
- [268] N. Russell, *et al.*, "Exception Handling Patterns in Process-Aware Information Systems," *BPM Center Report BPM-06-04*, *BPMcenter.org*, 2006.
- [269] W. M. P. van der Aalst, *et al.*, "Life After BPEL," *WS-FM*, vol. 3670, pp. 35-50, 2005.
- [270] T. Andrews, *et al.*, "Business Process Execution Language for Web Services, Version 1.1," *Specification*, *BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems*, 2003.
- [271] M. Koskela and J. Haajanen, "Business Process Modeling and Execution: Tools and Technologies Report for the SOAMeS Project," *VTT Technical Research Centre of Finland*2007.
- [272] R. Shapiro, "A Comparison of XPDL, BPML, and BPEL4WS," *xml.coverpages.org*, 2002.
- [273] M. Havey, "Essential Business Process Modeling," 1st ed Sebastopol, CA 95472, USA: O'Reilly Media, Inc., 2005, pp. 16-17.
- [274] J. Mendling and G. Neumann, "A Comparison of XML Interchange Formats for Business Process Modelling," *Workflow Handbook 2005*, 2005.
- [275] J. Pyke. (2007, XPDL - The Silent Workhorse of BPM. Available: <http://www.bpm.com/FeatureRO.asp?FeatureID=232>
- [276] W. M. P. van der Aalst, "Don't go with the flow: Web services composition standards exposed," *IEEE Intelligent Systems*, vol. 18, pp. 72-76, 2003.

- [277] J. B. Hill, *et al.*, "Findings: Confusion Remains Regarding BPM Terminologies," *Gartner Research*, vol. ID Number: G00155817, 10 March 2008 2008.
- [278] R. K. L. Ko, *et al.*, "Business Process Management (BPM) Standards: A Survey," *ACM Computing Surveys, Association for Computing Machinery, New York, USA. Impact Factor: 4.130*, 2008 [submitted for review on 10 April 2008].
- [279] S. S. G. Lee, *et al.*, "A Categorization of Business Process Management Standards," *Production and Operation Management Journal, Production & Operations Management Society, Impact Factor: 2.516*, 2008 [submitted for review on 31 March 2008].
- [280] M. Pusnik, *et al.*, "Evaluation of technologies for business process automation," *Informatica*, vol. 26, pp. 373-80, 2002.
- [281] S. Damodaran, "B2B integration over the Internet with XML: RosettaNet successes and challenges," *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pp. 188-195, 2004.
- [282] T. W. Malone, *et al.*, *Organizing Business Knowledge: The MIT Process Handbook*: Mit Press, 2003.
- [283] F. Flores and J. J. Ludlow, "Doing and Speaking in the Office," *Decision Support Systems: Issues and Challenges*, pp. 95-118, 1980.
- [284] G. Dickson, "An analysis of vendor selection systems and decisions," *Journal of purchasing*, vol. 2, pp. 5-17, 1966.
- [285] L. de Boer, *et al.*, "A review of methods supporting supplier selection," *European Journal of Purchasing & Supply Management*, vol. 7, pp. 75-89, 2001.
- [286] C. A. Weber, *et al.*, "Vendor selection criteria and methods," *European Journal of Operational Research*, vol. 50, pp. 2-18, 1991.
- [287] T. M. Chua, "Private Communication," ed: Motorola Electronics Private Limited, 2008.
- [288] A. Tan, "Private Communication," ed: Dell Singapore Private Limited, 2008.
- [289] B. Koh, "Private Communication," ed: Endela Trading and Manufacturing Private Limited, 2008.
- [290] B. Sim, "Private Communication," ed: Meiban Plastics Private Limited, 2008.
- [291] D. N. Le, *et al.*, "Generating Request Web Services from Annotated BPEL," in *Computing and Communication Technologies, 2009. RIVF '09. International Conference on*, 2009, pp. 1-8.
- [292] H. Knublauch, *et al.*, "The Protege OWL Plugin: An Open Development Environment for Semantic Web Applications," *LECTURE NOTES IN COMPUTER SCIENCE*, pp. 229-243, 2004.
- [293] R. K. L. Ko, *et al.*, "Business Process Management (BPM) Standards: A Survey," *Business Process Management Journal*, vol. 15 [in press], 2009.
- [294] R. S. Pressman, *Software Engineering: A Practitioner's Approach*: McGraw-Hill, Boston, 2000.
- [295] C. Petrie, "Planning Process Instances with Web Services," 2010.
- [296] R. Agrawal, *et al.*, "Mining process models from workflow logs," *Advances in Database Technology—EDBT'98*, pp. 467-483, 1998.
- [297] M. Dumas, *et al.*, *Process-aware information systems: bridging people and software through process technology*: Wiley-Blackwell, 2005.
- [298] W. Van Der Aalst, *et al.*, "Business process mining: An industrial application," *Information Systems*, vol. 32, pp. 713-732, 2007.
- [299] B. Van Dongen, *et al.*, "The ProM framework: A new era in process mining tool support," *Applications and Theory of Petri Nets 2005*, pp. 444-454, 2005.
- [300] P. Kueng and P. Kawalek, "Goal-based business process models: creation and evaluation," *Journal, Vol*, vol. 3, pp. 17-38, 1997.
- [301] P. Soffer and Y. Wand, "Goal-driven multi-process analysis," *Journal of the Association for Information Systems*, vol. 8, pp. 175-202, 2007.
- [302] OMG. (2008, 1 Feb 2008). *Semantics of Business Vocabulary and Business Rules (SBVR)*. Available: <http://www.omg.org/docs/formal/08-01-02.pdf>

- [303] OMG. (2008). *OMG Business Modeling Specifications*. Available: <http://www.omg.org>
- [304] M. Bajec and M. Krisper, "A methodology and tool support for managing business rules in organisations," *Information Systems*, vol. 30, pp. 423-443, 2005.
- [305] J. Hall, "Business Rules Boot Camp (Tutorial)," presented at the 6th European Business Rules Conference (EBRC 2007), Dusseldorf, Germany, 2005.
- [306] P. Fettke, *et al.*, "Business Process Reference Models: Survey and Classification," *Workshop on Business Process Reference Models (BPMR 2005)*, pp. 1-15, 2005.
- [307] Supply Chain Council, "Supply Chain Operations Reference-Model (SCOR) Version 8.0," ed: Supply Chain Council (SCC), 2007.
- [308] Supply Chain Council, "Design Chain Operations Reference Model Version 1.0," ed: Supply Chain Council (SCC), 2007.
- [309] APQC, "APQC Process Classification Framework (PCF) Version 5.0.2," ed: APQC, Houston, Texas 77024-7797, USA, 2008.
- [310] G. A. Herman and T. W. Malone, "What is in the Process Handbook?," in *Organizing Business Knowledge: The MIT Process Handbook*, T. W. Malone, *et al.*, Eds., ed Cambridge, Massachusetts: The MIT Press, 2003, pp. 221-258.
- [311] T. W. Malone and K. Crowston, "The Interdisciplinary Study of Coordination," *ACM Computing Surveys*, vol. 26, pp. 87-119, 1994.
- [312] Phios Corporation and MIT Sloan, "eBusiness Process Handbook," ed. One Broadway Suite 600 Cambridge, MA 02142 USA: Phios Corporation, 2001.
- [313] Y. Aklouf, *et al.*, "PLIB ontology: A mature solution for products characterization in B2B electronic commerce," *International Journal of IT Standards and Standardization Research*, vol. 3, pp. 66-81, 2005.
- [314] J. Jung, *et al.*, "Business process choreography for B2B collaboration," *Internet Computing, IEEE*, vol. 8, pp. 37-45, 2004.
- [315] WfMC. (1995, 20 Jan 2008). *The Workflow Management Coalition (WfMC)*. Available: <http://www.wfmc.org>
- [316] A. Barros, *et al.*, "Service Interaction Patterns," ed, 2005, pp. 302-318.

Related Publications and Awards

This research has resulted in the following awards and publications:

- **Awards**

- **Top Prize**, 4th IEEE Services Cup (Service Computing Contest) 2009, Los Angeles, California.
 - Technical paper and on-site demo of Genesis system emerged as top-prize winner from 13 entries from all over the world.
 - Technical Paper: Ryan K. L. Ko, Stephen S. G. Lee and Andre Jusuf, Genesis – Dynamic Collaborative Business Process Formulation Based on Business Goals and Criteria, IEEE Services Cup 2009, 6-10 July 2009
- **Best Student Award 2009**, Agency for Science Technology and Research (A*STAR) – Singapore Institute of Manufacturing Technology (SIMTech), 2010.

- **International Peer-Reviewed Journal**

- Ryan K L Ko, E W Lee and Stephen S. G. Lee, Business-OWL (BOWL) – A Hierarchical Task Network Ontology for Dynamic Business Process Decomposition and Formulation, **IEEE Transactions on Services Computing**, Accepted (25 December 2010), In print.
- Ryan K. L. Ko, Stephen S. G. Lee and E. W. Lee, Business Process Management (BPM) Standards: A Survey Business Process Management Journal, Emerald Publishing, Vol. 15 No. 5, Pp. 744-791, 2009.
- Ryan K. L. Ko, A Computer Scientists Introductory Guide to Business Process Management (BPM), ACM Crossroads, Vol. 15 Issue 4, pp. 11-18, ACM Press, 2009.

- **International Peer-Reviewed Conference**

- Ryan K L Ko, T L Hua, Angela E S Goh, Stephen S G Lee, E W Lee, CollaGen - High level language for dynamic cross-enterprise business process integration, Conference Paper in submission.
- Ryan K. L. Ko, Stephen S. G. Lee, E. W. Lee and Andre Jusuf, Dynamic Collaborative Business Process Formulation via Ontologised Hierarchical Task Network Planning, Proceedings of the **IEEE International Conference of Web Services (IEEE ICWS 2009)**, 6-10 July 2009, pp. 959-967, Los Angeles, California, IEEE Computer Society Press. (**Acceptance rate = 15.6%**)

- Ryan K. L. Ko, Stephen S. G. Lee and Andre Jusuf, Genesis Dynamic Collaborative Business Process Formulation Based on Business Goals and Criteria, 6-10 July 2009, Proceedings of the 2009 IEEE Congress on Services-I, pp. 123-129, Los Angeles, California, IEEE Computer Society Press.
- D. N. Le, N. S. Nguyen, Karel Mous, Ryan K. L. Ko and Angela E. S. Goh, Generating Request Web Services from Annotated BPEL, IEEE RIVF 2009 Conference, July 13-17, Da Nang City, Vietnam, IEEE Xplore Proceedings, 2009.
- K. Mous, Ryan K. L. Ko, S. G. Lee, P. S. Tan, and E. W. Lee, High-level Business Processes for Agile B2B Collaboration, Proceedings of the 35th International MATADOR Conference, pp. 169-172, 2007, Taipei, Taiwan, Springer-Verlag.
- **Book Chapter**
 - Ryan K L Ko, Stephen S G Lee, E W Lee, A Survey of Business Process Modeling Theory, Book Chapter submitted. Accepted 29 May 2010.
- **International Conference Poster Presentations**
 - Ryan K. L. Ko, E. W. Lee, S. G. Lee, Dynamic Formulation of Cross Enterprise Business Processes from User-Defined Business Goals and Constraints, Poster and Industry Demo Presentation, IEEE Asia Pacific Services Computing Conference (IEEE APSCC 2009), 9 Dec 2009, Biopolis, Singapore, 2009.
- **Research Seminars**
 - Ryan K. L. Ko, E. W. Lee, The Genesis+BOWL System for Dynamic Cross Enterprise Business Process Formulation, Research Presentation to Singapore Institute of Manufacturing Technology (SIMTech) Scientific Advisory Board, 26 Jan 2010.
 - Ryan K. L. Ko, Dynamic Formulation of Collaborative Business Processes From User-Defined Business Goals and Constraints, SIMTech PhD Student Sharing Session 2009, Singapore Institute of Manufacturing Technology, Agency for Science, Technology and Research, Singapore, 13 Nov 2009.
 - Ryan K. L. Ko, An Investigation into the Dynamic Formulation and Deployment of Collaborative Business Processes in B2B Collaborations, Seminar at School of Mechanical and Aerospace Engineering, College of Engineering, Nanyang Technological University, Singapore, July 2008.

Appendices

Supplementary material for thesis:

**A HYBRID HIERARCHICAL TASK NETWORK PLANNING AND ONTOLOGICAL
APPROACH FOR DYNAMIC CROSS-ENTERPRISE BUSINESS PROCESS
FORMULATION**

Ko Kok Leong Ryan

16 Feb 2011

Appendix A – Differentiating the BPM Terminologies

Appendix B – A Review of BPM Standards

Appendix C – Main Syntax of Pi-calculus Explained

Appendix D – Speech Acts Theory and Theory of Communicative Actions

Appendix E – Synthesised cBP Task Lists from RosettaNet, ebXML and OAGIS Reference Models

Appendix F – TQRDCEB Model for Supplier Evaluation

Appendix G – Industry Visit Questionnaire Enquiring High-Priority Vendor Selection Criteria

Appendix H – BOWL - Visio File Format

Appendix I – BOWL – Enhanced OWL/RDF Files Format

Due to the large size of the actual BOWL File, only the condensed version is printed.

The full version is available in the attached CD-ROM.

Appendix J – Compilation of Genesis Source Codes

The source codes of the GUI and the Algorithm for the Genesis System are attached.

Appendix K – BpelToDot.XSL File

Appendix L – Verification/ Path Testing Scenarios and Results of Prototype

Appendix M – Sample Verification Scenario in Visio Format

Appendix N – CollaGen Syntax in EBNF