

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

# IT Supported Business Process Negotiation, Reconciliation and Execution for Cross- Organisational E-Business Collaboration

by

Xi Chen

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of  
Doctor of Philosophy of Loughborough University

November 2008

© by Xi Chen 2008

## Abstract

In modern enterprises, workflow technology is commonly used for business process automation. Established business processes represent successful business practice and become a crucial part of corporate assets. In the Internet era, electronic business is chosen by more and more organisations as a preferred way of conducting business practice. In response to the increasing demands for cross-organisational business automation, especially those raised by the B2B electronic commerce community, the concept of collaboration between automated business processes, i.e. workflow collaboration, is emerging. Otherwise, automation would be confined within individual organisations and cross-organisational collaboration would still have to be carried out manually.

However, much of the previous research work overlooks the acquisition of the compatible workflows at build time and simply assumes that compatibility is achieved through face-to-face negotiation followed by a design from scratch approach that creates collaborative workflows based on the agreement resulted from the negotiation. The resource-intensive and error-prone approach can hardly keep up with the pace of today's marketplace with increasing transaction volume and complexity.

This thesis identifies the requirements for cross-organisational workflow collaboration (COWCO) through an integrated approach, proposes a comprehensive supporting framework, explains the key enabling techniques of the framework, and implements and evaluates them in the form of a prototype system – COWCO-Guru. With the support of such a framework, cross-organisational workflow collaboration can be managed and conducted with reduced human effort, which will further facilitate cross-organisational e-business, especially B2B e-commerce practices.

**Keywords:** Cross-Organisational Workflow Collaboration, Negotiation, Business Process Reconciliation, B2B E-Business

## **Acknowledgement**

I am indebted to my supervisor Professor Paul Chung for his guidance, challenging questions, encouragement, time and source of funding throughout the duration of the research.

I would like to give my sincere gratitude to my director of research Mr. Ray Dawson for his constructive suggestions and valuable advice given at each DoR meeting.

Special thanks to Loughborough University Innovative Manufacturing and Construction Research Centre (IMCRC), through which the project was funded.

Thanks to Judith, Larry, André, Yao Fang, Samra and Jeff for your administrative and technical support, my family members and friends who have supported and encouraged me on my way to the doctorate.

**To my parents**

# Table of Contents

<b>Abstract.....</b>	<b>ii</b>
<b>Acknowledgement .....</b>	<b>iii</b>
<b>Dedication .....</b>	<b>iv</b>
<b>List of Figures.....</b>	<b>xiv</b>
<b>List of Tables .....</b>	<b>xviii</b>

## Part I: Introduction and Background

<b>Chapter 1    Overview .....</b>	<b>1</b>
1.1    Introduction.....	1
1.2    Status Quo, Aim and Objectives .....	2
1.3    Approach.....	3
1.4    Contributions.....	5
1.5    Thesis Organisation .....	6
 <b>Chapter 2    Workflow Technology .....</b>	 <b>8</b>
2.1    Introduction.....	8
2.2    General Concepts .....	8
2.2.1    Workflow and Workflow Management System .....	8
2.2.2    Generic Workflow Models .....	10
2.2.2.1    Workflow Product Implementation Model .....	10
2.2.2.2    Workflow Reference Model .....	13
2.2.3    Activity Model .....	15
2.2.3.1    Activity .....	15
2.2.3.2    Activity-Based Workflow Model .....	16
2.2.4    Control Flow and Data Flow.....	17
2.3    Workflow Interoperability and Standardisation.....	18
2.3.1    Workflow Interoperability .....	18
2.3.2    Current State of Standardisation .....	19
2.4    WfMS Products.....	21
2.4.1    Categories of WfMS .....	21

2.4.2	WfMS Products.....	22
2.5	Summary and Conclusion .....	24
<b>Chapter 3</b>	<b>Business Collaboration and Cross-Organisational Workflows .....</b>	<b>25</b>
3.1	Introduction.....	25
3.2	From Business Collaboration to Workflow Collaboration .....	25
3.2.1	Business Collaboration .....	26
3.2.1.1	Partners' Relationships in Business Collaboration .....	26
3.2.1.2	Dynamic Environment .....	26
3.2.1.3	Forms of Business Collaboration.....	27
3.2.2	Workflow Collaboration .....	29
3.2.2.1	Between Two Workflows .....	29
3.2.2.2	Between Two "Stateful" Web Services .....	29
3.3	Patterns of Cross-Organisational Workflows .....	30
3.3.1	Hierarchical Workflow .....	30
3.3.2	Composite Workflow .....	32
3.3.3	Peer-to-Peer Workflow .....	33
3.4	An Example of Workflow Collaboration.....	34
3.5	Previous Build-Time Collaboration Approaches.....	36
3.5.1	Concrete Workflow Modelling Approach .....	36
3.5.2	Abstract Interaction Modelling Approach .....	37
3.5.2.1	Coalition Workflow .....	37
3.5.2.2	Coordination Dialogue.....	38
3.5.2.3	Interorganisational Workflow and Message Sequence Charts .....	39
3.5.2.4	Public B2B Collaboration Standards .....	39
3.5.3	Service Discovery Approach .....	40
3.5.4	Automated Reconciliation Approach.....	42
3.6	Previous Run-Time Collaboration Approaches .....	42
3.6.1	Centralised Workflow Enactment Approach.....	42
3.6.2	Sub-Flow Invocation Approach.....	43
3.6.3	Workflow Case Transfer Approach.....	44
3.6.4	Coordinated Data Exchange Approach.....	45
3.6.5	Messaging in a Nutshell.....	46

3.6.5.1	Message Content.....	46
3.6.5.2	Message Communication.....	47
3.7	Summary and Conclusion.....	48

## **Part II: Approach, Framework and Enabling Techniques**

<b>Chapter 4</b>	<b>An Integrated Approach .....</b>	<b>49</b>
4.1	Introduction.....	49
4.2	New Requirements and an Integrated Approach .....	49
4.2.1	Assumptions.....	50
4.2.2	New Requirements.....	50
4.2.2.1	Loose Coupling Principle .....	50
4.2.2.2	Negotiation Support .....	51
4.2.2.3	Decision Making Support .....	51
4.2.2.4	Process Change Support .....	52
4.2.2.5	Unified Collaboration Enactment Infrastructure .....	53
4.2.3	Integrated Approach.....	53
4.2.3.1	Workflow Dimension.....	53
4.2.3.2	Organisation Dimension .....	54
4.2.3.3	Operating Environment Dimension .....	55
4.3	Distribution vs Centralisation .....	55
4.3.1	Reasons for Distributed Coordination .....	55
4.3.2	From Centralisation to Distribution.....	56
4.3.2.1	Explicit Control Flow Specification .....	56
4.3.2.2	Implicit Control Flow Modelling.....	56
4.3.2.3	Modelling Distributed Workflow Coordination .....	57
4.3.3	Effectiveness of Distributed Workflow Coordination .....	58
4.3.3.1	Asynchronous Communication.....	58
4.3.3.2	Synchronous Communication.....	58
4.3.3.3	Hierarchical Workflow .....	59
4.3.3.4	Peer-to-Peer Workflows.....	60
4.3.3.5	Composite Workflow.....	61
4.4	Negotiation in a Nutshell .....	62
4.4.1	Definitions of Negotiation .....	62



4.4.2	Principles of Negotiation .....	64
4.4.3	Approaches of Negotiation .....	64
4.4.3.1	Concession-Convergence Approach.....	65
4.4.3.2	Mutual Gains Approach.....	65
4.4.3.3	Which to choose?.....	65
4.4.4	Operational Aspect of Collaboration Negotiation .....	65
4.5	Process Comparison Techniques .....	66
4.5.1	Comparison in General .....	66
4.5.2	Process Comparison.....	67
4.6	Summary and Conclusion .....	70
<b>Chapter 5</b>	<b>A Framework for Cross-Organisational Workflow Collaboration.....</b>	<b>71</b>
5.1	Introduction.....	71
5.2	Business View.....	71
5.2.1	Capturing Operational Aspect of Collaboration Negotiation with Workflow Technology .....	72
5.2.1.1	Capturing Interests .....	72
5.2.1.2	Capturing Communication.....	77
5.2.1.3	Capturing Options.....	82
5.2.1.4	Implication of Decisions .....	84
5.2.2	Agreement and Fulfilment .....	84
5.3	Architecture View .....	85
5.3.1	Logic Components .....	85
5.3.1.1	Collaboration Negotiation.....	86
5.3.1.2	Agreement Fulfilment.....	87
5.3.1.3	Collaboration Enactment Infrastructure.....	87
5.3.2	Service Stack.....	88
5.3.3	Implementation Architectures.....	90
5.3.3.1	Client/Server Architecture .....	90
5.3.3.2	Service-Oriented Architecture .....	94
5.4	Summary and Conclusion .....	95
<b>Chapter 6</b>	<b>Enabling Workflow Negotiation and Reconciliation.....</b>	<b>97</b>
6.1	Introduction.....	97

6.2	Interface Process Extraction .....	97
6.2.1	Design .....	98
6.2.2	Algorithm.....	99
6.3	Interface Process Comparison.....	100
6.3.1	Pre-treatment – Activity Bipartite Comparison .....	100
6.3.1.1	Unique Collaboration Message/Interface Activity .....	100
6.3.1.2	Estimated Effort of Design from Scratch.....	101
6.3.2	Process Behaviour Comparison .....	102
6.3.2.1	Conversion from Activity Diagram to State Transition System Diagram .....	102
6.3.2.2	State Transition System Comparison.....	104
6.3.3	Discrepancy Identification .....	106
6.4	Reconciliation Option .....	110
6.4.1	Destination Area .....	111
6.4.2	Target Activity .....	113
6.4.2.1	Candidate Removing Interface Activities and their Data Successors..	113
6.4.2.2	Candidate Moving Interface Activities and their Data Predecessors..	114
6.4.2.3	Candidate Inserting Interface Activities and their Data Generation Activities .....	118
6.4.3	Hypothetical Target Position .....	119
6.4.3.1	Sequence Perspective.....	119
6.4.3.2	Connectivity Perspective .....	122
6.4.3.3	Effective Adjustment with Minimum Impact .....	123
6.4.4	Information Prompting.....	123
6.4.4.1	Option at Activity Bipartite Comparison Stage .....	124
6.4.4.2	Reconciliation Option for CRIA.....	124
6.4.4.3	Reconciliation Option for CMA/CMIA.....	125
6.4.4.4	Reconciliation Option for CIIA .....	128
6.4.4.5	Sequence of Reconciliation Options .....	129
6.4.5	Option and Decision Dynamics .....	133
6.4.5.1	CRIA .....	133
6.4.5.2	CMA and CMIA .....	133
6.4.5.3	CIIA .....	137

6.4.6	Decision Making .....	137
6.4.6.1	Control Flow and Data Flow .....	138
6.4.6.2	Assessment and Confirmation .....	138
6.4.6.3	Manual and Automated Decision Making .....	138
6.5	Process Change .....	139
6.5.1	Conceded Target Activity .....	139
6.5.2	Control Flow Change Operations .....	140
6.5.3	Data Flow Change Operations .....	141
6.6	Summary and Conclusion .....	142

## **Part III: Design, Implementation and Evaluation**

<b>Chapter 7</b>	<b>System Design and Implementation .....</b>	<b>143</b>
7.1	Introduction .....	143
7.2	A Prototype System for Cross-Organisational Workflow Collaboration .....	143
7.2.1	Basic Facts Explained .....	143
7.2.2	System Architecture .....	145
7.2.2.1	Collaboration Interface .....	146
7.2.2.2	Interface Process Extractor .....	148
7.2.2.3	Compatibility Inspector .....	148
7.2.2.4	Process Tailor .....	148
7.2.2.5	Collaboration Gateway .....	149
7.2.2.6	Process Repository .....	149
7.2.2.7	Agreement Fulfilment Services .....	150
7.2.2.8	Collaboration Server .....	150
7.2.3	Implementation of the Negotiation Protocol .....	151
7.3	Workflow Collaboration Enactment Infrastructure .....	151
7.3.1	Collaboration Message .....	151
7.3.1.1	Types of Message Content .....	152
7.3.1.2	Message Format .....	153
7.3.1.3	Data Ontology .....	153
7.3.2	Message Sender/Requester .....	153
7.3.2.1	Message Sender .....	153
7.3.2.2	Message Requester .....	154

7.3.3	Collaboration Server .....	155
7.3.3.1	Collaboration Form.....	155
7.3.3.2	Form Filling .....	156
7.4	A Demonstration of the Prototype System at Build Time .....	158
7.4.1	On Partner A's Site .....	158
7.4.2	On Partner B's Site .....	162
7.5	Summary and Conclusion .....	164
<b>Chapter 8</b>	<b>Testing and Evaluation.....</b>	<b>166</b>
8.1	Introduction.....	166
8.2	Evaluation System .....	166
8.2.1	Evaluation Criteria at Collaboration Build Time.....	166
8.2.2	Evaluation Criteria at Collaboration Run Time .....	170
8.2.3	Testing Plan .....	170
8.3	Case Study 1: E-Commerce Case Study in Casual Trade.....	172
8.3.1	Example 1: Compatible Workflows (C1.E1).....	172
8.3.2	Example 2: Incompatible Workflows (C1.E2) .....	173
8.4	Case Study 2: Process Negotiation and Reconciliation .....	181
8.4.1	Examples.....	181
8.4.1.1	Example 1 (C2.E1).....	181
8.4.1.2	Example 2 (C2.E2).....	186
8.4.1.3	Example 3 (C2.E3).....	187
8.4.1.4	Example 4 (C2.E4).....	189
8.4.1.5	Example 5 (C2.E5).....	191
8.4.2	Control Flow Preservation .....	193
8.4.2.1	Effort in Reconciliation Option Preparation .....	193
8.4.2.2	Result after User Decision Making.....	194
8.4.3	Data Flow Preservation.....	194
8.4.3.1	Effort in Reconciliation Option Preparation .....	194
8.4.3.2	Result after User Decision Making.....	195
8.4.4	Considering Previous Decisions .....	195
8.4.5	User Flexibility .....	196
8.4.6	Implication Accompanied by User Flexibility.....	196

8.4.7	User Interface.....	199
8.4.8	Noise Reduction.....	201
8.4.8.1	Identification of CMIA .....	201
8.4.8.2	Non-Duplication Control Flow Decision Making .....	201
8.4.8.3	Sequence of Reconciliation Option .....	201
8.4.8.4	Automated Decision Making .....	202
8.4.9	Correctness of Change .....	202
8.4.10	Exceptional Circumstances .....	203
8.5	Case Study 3: Collaboration Enactment .....	204
8.5.1	Example 1 (C3.E1): Sequential, XOR Branching, Iterative and Parallel Structures .....	204
8.5.1.1	Example Description.....	204
8.5.1.2	Findings.....	207
8.5.2	Example 2 (C3.E2): Revisit Parallel Structures.....	207
8.6	Approach Comparison .....	209
8.6.1	openXchange Project Vs Guru.....	209
8.6.2	Workflow View Approach Vs Guru .....	209
8.6.3	InDiA Vs Guru.....	210
8.6.4	Automatic Process Negotiation Vs Guru.....	210
8.7	Summary and Conclusion .....	211
<b>Chapter 9</b>	<b>Conclusion and Future Work .....</b>	<b>212</b>
9.1	Introduction.....	212
9.2	Thesis Review .....	212
9.3	Summary of Contributions.....	215
9.4	Limitations and Future Work.....	216
9.4.1	Suitability in the Targeted Area .....	216
9.4.2	Fine Tuning Compatibility Definitions and Comparison Techniques .....	217
9.4.3	Automated Risk Assessment and Decision Making .....	217
9.4.4	Multi-Lateral Process Negotiation.....	217
9.4.5	Bring back XOR Structure.....	217
9.4.6	Evaluation for other Workflow Systems.....	218
9.4.7	Evaluation by Real Cases.....	218

9.4.8	Potential Industrial Partner and Knowledge Transfer.....	218
9.4.9	Potential Contribution to Standards .....	218
9.5	Overall Conclusion .....	218
<b>References.....</b>		<b>219</b>
<b>Appendices</b>		
<b>Appendix 1: List of Abbreviations .....</b>		<b>226</b>
<b>Appendix 2: List of Publications .....</b>		<b>228</b>

## List of Figures

Figure 2-1. Key workflow system features (adapted from WfMC, 1995).....	10
Figure 2-2. Workflow product implementation model (adapted from WfMC, 1999a).....	11
Figure 2-3. Workflow Reference Model – Components & Interfaces (WfMC, 1999a).....	13
Figure 2-4. Distribution within the workflow enactment service (adapted from WfMC, 1995).....	17
Figure 2-5. Currently available standards regarding process technology (adapted from Boyes-Schiller, 2003). ....	20
Figure 3-1. Dynamics from four aspects.....	27
Figure 3-2. Spectrum of business collaboration. ....	28
Figure 3-3. Hierarchical cross-organisational workflow. ....	30
Figure 3-4. Chained flow. (WfMC, 1999b) .....	31
Figure 3-5. Composite workflow.....	32
Figure 3-6. Peer-to-peer workflow. ....	33
Figure 3-7. An example of collaboration between workflows of Vendor and Customer_1..	35
Figure 3-8. Coalition workflow, workflow views and private workflows (adapted from Schulz and Orlowska, 2004). ....	38
Figure 3-9. An example of coordination dialogue (adapted from Biegus and Branki, 2004). .....	39
Figure 3-10. Centralised workflow enactment (adapted from Chen and Hsu, 2001) .....	43
Figure 3-11. Workflow case transfer approach (adapted from Chen and Hsu, 2001) .....	45
Figure 4-1. Asynchronous communication coordinated in a data dependency manner. ....	58
Figure 4-2. Synchronous communication coordinated in a data dependency manner. ....	59
Figure 4-3. Chained sub-flow represented in a data dependency manner. ....	59
Figure 4-4. Nested sub-flow represented in a data dependency manner. ....	60
Figure 4-5. Peer-to-peer workflows represented in a data dependency manner.....	61
Figure 4-6. Converting composite workflow into a peer-to-peer pattern. ....	62
Figure 4-7. How to tell the difference? .....	67
Figure 4-8. Conversion from an activity diagram to a state transition system diagram. ....	69
Figure 5-1. Business view of workflow collaboration.....	72
Figure 5-2. Capture negotiation interests from a workflow perspective. ....	73

Figure 5-3. Example of unnecessary delay. ....	75
Figure 5-4. Collaboration deadlock and unnecessary delay. ....	75
Figure 5-5. Interests representation.....	78
Figure 5-6. Overall negotiation process.....	80
Figure 5-7. Unilateral decision-making. ....	82
Figure 5-8. How much to adjust?.....	83
Figure 5-9. Logic components. ....	86
Figure 5-10. Architecture of the cross-organisational workflow collaboration framework. .	89
Figure 5-11. Implementation architecture: rich client C/S. ....	91
Figure 5-12. Collaborative messages and internal data by following rich client C/S architecture.....	92
Figure 5-13. Implementation architecture: thin client C/S. ....	93
Figure 5-14. Collaborative messages and internal data by following thin client C/S architecture.....	94
Figure 5-15. Implementation architecture: SOA. ....	95
Figure 6-1. Special situations in interface process extractions. ....	98
Figure 6-2. Algorithm for interface process extraction. ....	99
Figure 6-3. Algorithm for deciding reachability of join-type activity. ....	103
Figure 6-4. Algorithm for skipping dummy activity ....	103
Figure 6-5. Example of skipping dummy activity. ....	104
Figure 6-6. Algorithm for activity diagram comparison framework. ....	106
Figure 6-7. Discrepancy Scene in UML class diagram. ....	106
Figure 6-8. Candidate Adjusting Interface Activity Set in UML class diagram. ....	108
Figure 6-9. Example of discrepancy scene with CMIAS only. ....	109
Figure 6-10. Example of discrepancy scene with CMIAS and CIIAS. ....	110
Figure 6-11. Destination Area in UML class diagram.....	112
Figure 6-12. Destination area with continuous positions.....	112
Figure 6-13. Destination area with discontinuous positions.....	113
Figure 6-14. Identifying CMIAS with upstream independent rule.....	115
Figure 6-15. A process divided by destination area (DA) and candidate moving interface activities (CMIIAs). ....	116
Figure 6-16. Necessity of extending CMIAS. ....	117
Figure 6-17. Algorithm for capturing candidate moving activity set. ....	118



Figure 6-18. Algorithm for associating CMAs with their CMIA. ....	118
Figure 6-19. Can we still get a cup of tea? .....	119
Figure 6-20. Choice of hypothetical target position. ....	120
Figure 6-21. DBA Type 1 and 2 and their HTPs. ....	121
Figure 6-22. Connectivity perspective – CMA/CMIA and HTP(s).....	122
Figure 6-23. Sequence of reconciliation options: CRIA → CMIA. ....	130
Figure 6-24. Sequence of reconciliation options: CMIA → CIIA.....	132
Figure 6-25. Data flow implication between current CMA/CMIA and its data predecessor before DBA when not all options are rejected.....	135
Figure 6-26. Data flow implication between current CMA/CMIA and immediately preceding CMA that was provisionally accepted when not all options are rejected. ..	136
Figure 6-27. Data flow implication when all options are rejected.....	137
Figure 7-1. System architecture. ....	145
Figure 7-2. Primary interface.....	146
Figure 7-3. Control flow options dialogue.....	147
Figure 7-4. Shark Workflow’s administration interface. ....	147
Figure 7-5. Message exchanged in a rich client type of client/server architecture.....	152
Figure 7-6. Data Supply Message: Message Format (1) and (2) .....	154
Figure 7-7. Data Request Message: Message Format (3) .....	154
Figure 7-8. Data response message: Message Format (4) and (5).....	157
Figure 7-9. Run-time message exchange mechanism.....	158
Figure 7-10. Legends adopted in the demonstration.....	158
Figure 7-11. First round of comparison at Partner A’s site. ....	159
Figure 7-12. Reconciliation option for A3 in control flow.....	159
Figure 7-13. Reconciliation option for A.c[s] in control flow.....	160
Figure 7-14. Second round of comparison at Partner A’s site.....	160
Figure 7-15. Reconciliation option regarding A(1).d[d] and A6 in data flow. ....	161
Figure 7-16. Rejection of the reconciliation option regarding A(1).d[d] and A6 in data flow. .....	161
Figure 7-17. Reconciliation option regarding A6 and A(1).e[s] in data flow.....	162
Figure 7-18. Rejection of the reconciliation option regarding A6 and A(1).e[s] in data flow. .....	162
Figure 7-19. First round of comparison at Partner B’s site.....	163

Figure 7-20. Reconciliation option for B4 in control flow. ....	163
Figure 7-21. Reconciliation option for B.d[s] in control flow. ....	163
Figure 7-22. Compatible workflows B(1) and A(1). ....	164
Figure 8-1. Evaluation criteria at collaboration build time. ....	169
Figure 8-2. Interface processes extracted from Vendor and Customer_1's initial workflows. .....	172
Figure 8-3. STS comparison and result for Vendor and Customer_1. ....	173
Figure 8-4. Example 2: Workflows of Vendor and Customer_2. ....	175
Figure 8-5. Interface processes extracted from Vendor and Customer_2's initial workflows. .....	176
Figure 8-6. Process of Vendor(1). ....	177
Figure 8-7. 1st round STS comparison result between Vendor(1) and Customer_2. ....	178
Figure 8-8. Process Vendor(2). ....	180
Figure 8-9. Legends adopted in the examples. ....	181
Figure 8-10. Case 2 Example 1 – CMAs and Their HTPs. ....	182
Figure 8-11. Case 2 Example 2. ....	186
Figure 8-12. Case 2 Example 3. ....	188
Figure 8-13. Case 2 Example 4. ....	189
Figure 8-14. Case 2 Example 5. ....	191
Figure 8-15. Result of Example 5 (C2.E5). ....	203
Figure 8-16. Exceptional Circumstances. ....	203
Figure 8-17. Example 1 (C3.E1): Manufacturer and Retailer. ....	206
Figure 8-18. Example 2 (C3.E2): revisit parallel structures. ....	208

## List of Tables

Table 2-1. WfMS products comparison.....	24
Table 3-1. A collection of hierarchical workflows. ....	31
Table 3-2. A collection of composite workflows.....	32
Table 3-3. A collection of peer-to-peer workflows. ....	33
Table 3-4. Operations required for hierarchical workflow execution.....	43
Table 6-1. CMA/CMIA and their HTPs. ....	123
Table 6-2. Concession and operation for different target activities.....	140
Table 7-1. Headings of the collaboration form. ....	155
Table 7-2. An example of agreement fulfilment form. ....	156
Table 8-1. Testing plan. ....	170
Table 8-2. Decision Sequence 1 of Example 1 (C2.E1.DS1). ....	182
Table 8-3. Decision Sequence 2 of Example 1 (C2.E1.DS2). ....	184
Table 8-4. Decision Sequence 3 of Example 1 (C2.E1.DS3). ....	185
Table 8-5. Decision Sequence of Example 2 (C2.E2.DS1). ....	187
Table 8-6. Decision Sequence of Example 3 (C2.E3.DS1). ....	188
Table 8-7. Decision Sequence of Example 4 (C2.E4.DS1). ....	190
Table 8-8. Decision Sequence of Example 5 (C2.E5.DS1). ....	191
Table 8-9. Dialogue boxes. ....	199
Table 8-10. Scenarios and purposes of automated decision making. ....	202

# Part I

## Introduction and Background

*"Coming together is a beginning,*

*Staying together is progress,*

*Working together is success."*

*– Henry Ford*

# Chapter 1

## Overview

### 1.1 Introduction

In modern enterprises, workflow technology is commonly used for business process automation. Established business processes represent successful business practice and become a crucial part of corporate assets. In the Internet era, electronic business is chosen by more and more organisations as a preferred way of conducting business practice. In response to the increasing demands for cross-organisational business automation, especially those raised by the B2B electronic commerce community, the concept of collaboration between automated business processes, i.e. workflow collaboration, is emerging. Otherwise, automation would be confined within individual organisations and cross-organisational collaboration would still have to be carried out manually.

The purposes of this thesis are to identify the requirements for cross-organisational workflow collaboration, introduce a supporting framework, explain the key approaches and enabling techniques in the framework, and evaluate the prototype workflow collaboration supporting tool, COWCO-Guru, through various case studies. With the support of such a framework, cross-organisational workflow collaboration can be managed and conducted in a more cost-effective manner, which would further facilitate cross-organisational e-business, especially B2B e-commerce practices.

This chapter provides an overview of the whole thesis. §1.2 provides a brief description of the status quo of approaches to workflow collaboration and sets the aim and objectives. §1.3 describes the integrated approach adopted by the thesis. §1.4 highlights the contributions. §1.5 outlines the organisation of the whole thesis.

## **1.2 Status Quo, Aim and Objectives**

The complex and process-driven natures of business transactions as well as automation demands from business community put workflow technology as one of the effective tools for e-business. The flourish and success of B2B e-business together with the increasing popularity and adoption of workflow management system (WfMS) within organisations make workflow-based B2B e-business practically viable. However, much of the previous research work overlooks the acquisition of the compatible workflows at build time and simply assumes that compatibility is achieved through face-to-face negotiation followed by a design from scratch approach that creates collaborative workflows based on the agreement resulted from the negotiation. The approach is resource-intensive and error-prone. In fact, it is the manual negotiation and the approach of process creation from scratch that leave a gap in the chain of cross-organisational business automation. This approach is only acceptable for an organisation that has a handful of collaborative partners and once designed and implemented their processes do not change frequently. Unfortunately, neither of the assumptions reflects the true nature of today's marketplace. In a globalised economy, an organisation's client base and its variety can change rapidly. Partners' business processes are also modified to adapt to the ever-changing marketplace and regulatory requirements. Being faced with such demanding reality, manual negotiation followed by collaborative process creation from scratch can hardly keep up with the pace of the marketplace. Also, once created, existing workflow definitions are unlikely to be reused even if they have only minor difference to the desired compatible workflow, which represents a huge waste in terms of corporate assets. As a result, if the status quo is kept as it is, organisations will take growing risks of not responding to a potential collaboration in a cost-effective and timely manner. Other build-time approaches, e.g. passive services matching, fall short of the ability of process reconciliation once incompatible workflow definitions are encountered.

At run time, a workflow collaboration infrastructure is needed to extend workflow enactment services across organisational boundaries in a truly decentralised and distributed manner. Also, the infrastructure should easily accommodate the mainstream model of workflow management system.

To the best knowledge of the author, there is no other approach or system looks at the issue of cross-organisational workflow collaboration from a comprehensive view and provides an integrated framework. In line with the challenges, the aim of the thesis can be stated as:

*How to provide comprehensive IT support for business collaboration in the form of an integrated cross-organisational workflow collaboration supporting framework, which comprises support for compatible business process acquisition at build time and a loosely-coupled infrastructure for workflow collaboration enactment at run time.*

The objectives of this research are:

- To capture the operational aspect of business collaboration negotiation with workflow technology,
- To provide IT support to business collaboration negotiation and fulfillment through a comprehensive framework for cross-organisational workflow collaboration,
- To develop a prototype system that implements the key components of the framework for testing and evaluation purposes.

## 1.3 Approach

In order to meet these objectives and echo the comprehensive nature of business collaboration, an integrated approach is employed throughout this thesis, in which three dimensions are addressed:

- Workflow,
- Organisation,
- Operating Environment.

### Workflow Dimension

As the underpinning technology in this thesis, workflow is identified as one of the dimensions. Within the workflow dimension, two aspects can be further identified:

- the two-stage lifecycle – build time and run time stages,
- the two-layer flow model – control flow and data flow.

Build-time and run-time issues are distinguished and addressed accordingly throughout the thesis because the lifecycle of workflow technology spans across these two stages. By doing so, challenges and requirements can be thoroughly identified whilst design, implementation and evaluation of the proposed approaches and techniques can be logically organised. Also, it is more convenient to communicate the result of this research project with other workflow researchers and practitioners by following the stage division.

Control flow and data flow are two different but interrelated concepts in workflow technology. Data flow is attached on top of the control flow layer thus representing another layer of more stringent dependency. When a process changes, violations on both layers need to be identified.

### **Organisation Dimension**

As the owners of the participating business process collaboration, organisations have their main concerns reflected in three aspects of the organisation dimension. These include knowledge preservation, lean principle, and human-centric IT support to business users.

From an organisational perspective, knowledge preservation is a crucial task because knowledge is an important type of corporate asset. As corporate knowledge takes the form of business processes as well as their implementation – workflow definitions, knowledge preservation equals process preservation. Proper preservation enables knowledge reuse to the maximum level.

The lean principle – “add nothing but value” (Poppendieck, 2002) is gaining more and more importance over the years. Any waste or unnecessary operations during process negotiation should be eliminated. This principle forms the basis of some of the approaches and enabling techniques proposed in this thesis.

As one of the key objectives, in order to fully inform business users during their decision-making, user-centric IT support is needed. Before a user makes a decision, identification regarding all the information needed, as well as logical relationship among the information, is crucial to the effectiveness and quality of users’ decision-making procedure. After a



decision is made, the workflow should evolve according to users' intention with its consistency and correctness intact.

### **Operating Environment Dimension**

The operating environment dimension has two aspects, distribution and dynamism.

As cross-organisational business collaboration is, by nature, operating in a distributed environment, distribution should be taken into account and reflected in areas such as privacy protection, decentralised progress coordination, and acknowledgement of autonomy.

A changing and dynamic environment is always one of the important features of the real world. By acknowledging this aspect, this research finds its feet on a solid and realistic ground. Every proposed solution needs to take dynamism into account and be evaluated by it. Generally speaking, in order to address the dynamism found in business collaboration across different organisations, the loosely coupled principle needs to be observed. Also, flexibility would help adaptation stemmed from dynamic changes.

## **1.4 Contributions**

The contributions of the thesis can be summarised as:

- A new modelling approach that implicitly specifies cross-organisational workflow interaction. It provides an alternative to conventional explicit flow modelling and suits the need for loose coupling in a decentralised and distributed environment.
- Through the abstraction of interface process and comparison of different criteria adopted by existing process compatibility definitions, a new workflow collaboration compatibility is defined.
- Clarify and enhance the understanding of techniques applied in process comparison.
- Capture key operational elements (interests, communication and options) of a business negotiation with workflow technology.
- A novel integrated negotiation approach that implements lean thinking. It is able to bring about a cost-effective negotiation outcome.

- A novel approach that provides decision-making support to non-technical users during collaboration negotiation.
- A flexible cross-organisational workflow collaboration (COWCO) supporting framework that can be adapted to different implementation architectures.
- A new blackboard-based infrastructure provides distributed workflow enactment support at run time.
- Enhance the matching and reconciliation capability of web services where services are defined and described in the form of flow models.
- Preserve corporate intellectual asset in terms of business process/workflow as much as possible.
- Have the potential to facilitate B2B e-business in a dynamic market place from transaction negotiation to agreement fulfillment.

## 1.5 Thesis Organisation

The rest of this thesis is structured in three parts followed by a conclusion chapter.

### **Part I: Introduction and Background**

Chapter 2 introduces workflow technology and workflow management systems. With particular relevance to this thesis, the concept of workflow interoperability, current effort in workflow standardisation, and mainstream workflow management systems are also reviewed.

Chapter 3 looks at various aspects of business collaboration – the driving force behind workflow collaboration, categorises different patterns of cross-organisational workflows, and reviews previous build-time and run-time approaches to workflow collaboration. An example of workflow collaboration based on business transactions in trade community is depicted.

### **Part II: Approach, Framework and Enabling Techniques**

Chapter 4 identifies a number of requirements in response to the incapability exposed by existing approaches. Due to the wide spectrum of issues covered in these requirements, an integrated approach is adopted and relevant topics are introduced and reviewed, including distributed workflow coordination, negotiation and process comparison techniques.

Chapter 5 proposes a comprehensive framework in order to address the requirements and facilitate the integrated approach. The Framework is described from both business and architectural aspects. Three key operational elements of negotiation are captured from a workflow perspective.

Chapter 6 explains key enabling techniques for workflow negotiation and reconciliation that are at the heart of the implementation of the integrated approach and realisation of the comprehensive framework for cross-organisational workflow collaboration. These techniques include interface process extraction, process comparison, formulation, prompting and decision-making of reconciliation options, and process change operations.

### **Part III: Design, Implementation and Evaluation**

Chapter 7 describes the system architecture of the prototype system: COWCO-Guru. An overall negotiation process is implemented in the form of a pair of compatible workflows. Collaboration enactment infrastructure is described with various formats of collaboration messages defined.

Chapter 8 establishes an evaluation system comprising eight criteria. A testing plan is prepared specifying the criterion and example pairs. Three case studies are carried out to evaluate all the eight criteria together with the general principles. At last, the Guru approach is compared with four other relevant approaches to reveal differences as well as similarities.

Chapter 9 briefly reviews this thesis, summarises the achievements, identifies the limitations, and outlines the needs for future work in some areas.

# Chapter 2

## Workflow Technology

### 2.1 Introduction

Since workflow<sup>1</sup> technology was first introduced, nearly two decades have passed. Being an effective business process automation solution, workflow systems have been widely adopted within organisations. The prevalence of the Internet inevitably storms the workflow community and is pushing it to new frontiers, e.g. to support cross-organisational business collaboration and to enable workflow-driven web services.

Before discussing the exciting new developments, workflow – the old working horse – is introduced in this chapter. §2.2 describes some of the important concepts of workflow technology. §2.3 reviews the background of workflow interoperability and its current status. §2.4 categorises workflow management systems. Some of the mainstream workflow products are also mentioned. §2.5 summarises and concludes this chapter.

### 2.2 General Concepts

#### 2.2.1 Workflow and Workflow Management System

According to Workflow Management Coalition (WfMC)'s definition, a workflow is “the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules” (WfMC, 1999a).

---

<sup>1</sup> Although by definition, the term *workflow* is more technical prone and *business process* is more business oriented, in some cases it is difficult to draw a clear line between them. In this thesis, these two terms are used interchangeably.

In order to take full advantage of a workflow, workflow management system (WfMS) is introduced, which is “a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications” (WfMC, 1999a). Being mature and beneficial, workflow technology will remain important for many organisations (Smith, 2002).

WfMC (1995) identified that a WfMS should provide support in three functional areas:

- Build-time functions that are concerned with defining, and possibly modelling, the workflow process and its constituent activities;
- Run-time control functions that are concerned with managing the workflow processes in an operational environment and sequencing the various activities to be handled as part of each process;
- Run-time interaction functions that are concerned with processing the activity steps through human users or IT application tools.

Two stages, build time and run time, need to be differentiated in the lifecycle of any workflow application. According to WfMC (1999a):

**Build time** refers to “the time period when manual and/or automated (workflow) descriptions of a process are defined and/or modified electronically (WfMC, 1999a)”.

**Run time** refers to “the time period during which the process is operational, with process instances being created and managed”.

A standalone WfMS is illustrated in Figure 2-1.

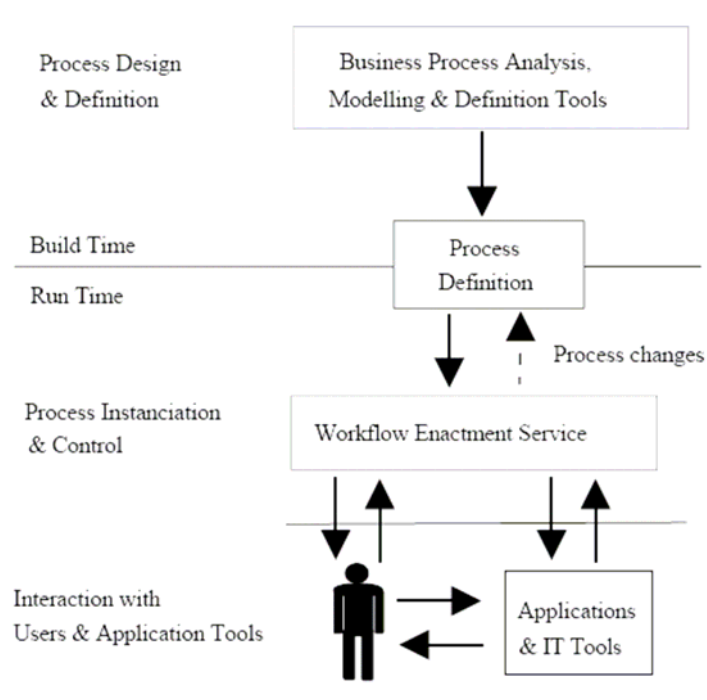


Figure 2-1. Key workflow system features (adapted from WfMC, 1995).

## 2.2.2 Generic Workflow Models

With the popularity of WfMS's adoption in organisations and the foreseeable future of workflow interoperation across organisational boundaries, Workflow Management Coalition (WfMC) initiated the first ever workflow standardisation effort. Two abstract models are generalised for different purposes, namely Product Implementation Model and Reference Model.

### 2.2.2.1 Workflow Product Implementation Model

Despite the variety in workflow products in the current market, WfMC (1999a) generalises an abstract implementation model for a workflow management system. The model captures the main functional components and the interfaces between them and can be matched to most products in the marketplace. The model is shown in Figure 2-2.

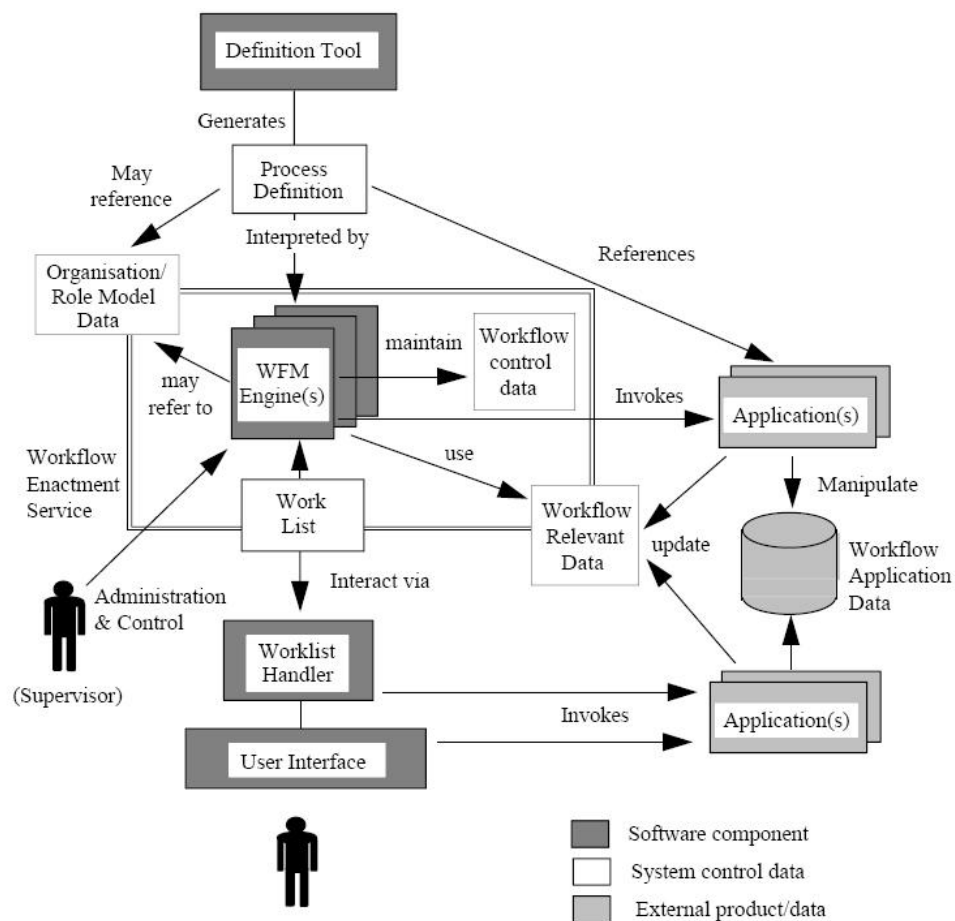


Figure 2-2. Workflow product implementation model (adapted from WfMC, 1999a).

The model comprises three types of components:

- software components that support various functions within the WfMS (shown in dark fill)
- various types of system definitions and control data used by one or more software components (shown in unfilled)
- applications and databases that are not part of the WfMS but may be invoked during enactment (shown in light fill).

Major functional components in the model are described below.

## Process Definition Tool

At build time, the process definition tool is used to create the process description in a computer executable form. The tool may provide a drawing environment or a text editor for

users to specify the process definition in the form of a formal process definition language, a script or a set of routing commands to transfer information between users.

It may be supplied as part of a workflow product or part of a business process analysis product.

### **Process Definition**

The process definition contains all information that is necessary for the execution of the process by the workflow enactment service. The definition includes information about:

- process starting and completion conditions,
- activity navigation rules,
- tasks for users to undertake,
- references to applications to be invoked,
- workflow relevant data defined by users etc.

### **Workflow Enactment Service**

At run time, the workflow enactment service is in charge of interpreting the process definition, navigating activities to follow sequential, branching or parallel executions, allocating work items to work lists of each user and invoking applications via application tools.

### **Workflow Data**

Three types of workflow data can be distinguished in the model, namely workflow control data, workflow relevant data and workflow application data. They are defined as follows (WfMC, 1999a).

- workflow control data: data that are managed by the WfMS that is internal to the WfMS and not normally accessible to applications.
- workflow relevant data: data that are used by a WfMS to determine the state transitions of a workflow instance, e.g. pre- and post-conditions, transition conditions or workflow participant assignment.
- workflow application data: data that are application specific and not accessible by the WfMS.



## Worklists

As an abstract concept, a worklist belongs to an individual user. It is used to temporarily hold work items assigned by a WfMS to the user for attention. The process of work item allocation may be either visible or invisible to participants for a particular WfMS.

## Worklist Handler and User Interface

The Worklist Handler together with the user interface act as a front end of a worklist and is in charge of prompting the content of a worklist to its owner. In some systems, it may be similar to a simple in-tray. In other systems, it might be much more sophisticated, e.g. providing facilities as load balancing and work reassignment.

### 2.2.2.2 Workflow Reference Model

To facilitate interoperability at different levels, WfMC (1999a) developed a common *Workflow Reference Model* from the generic Workflow Product Implementation Model, which put emphasis on a common set of workflow APIs and interchange formats in addition to workflow systems' common components. This enables specifications to be developed within the context of an overall common model for workflow systems. Figure 2-3 shows the Reference Model with its major components and interfaces.

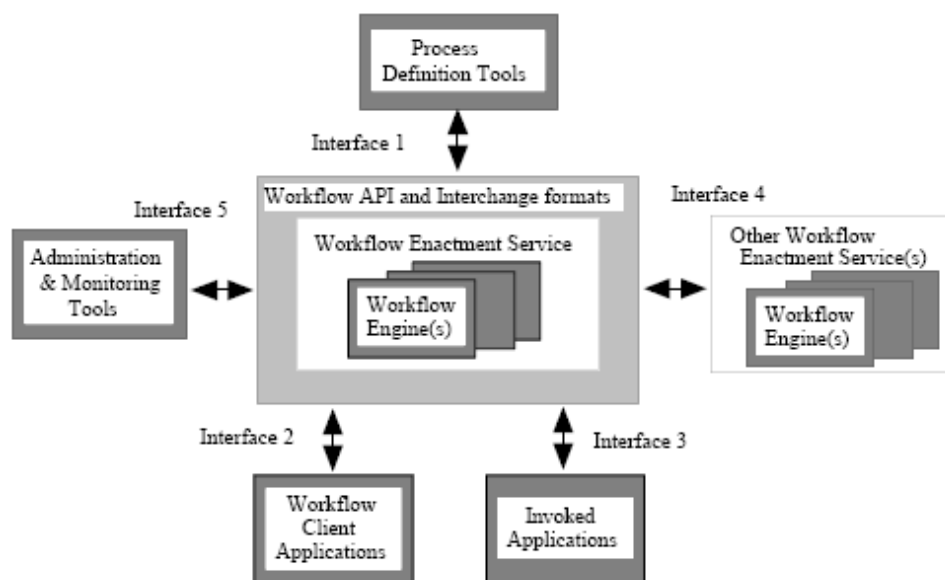


Figure 2-3. Workflow Reference Model – Components & Interfaces (WfMC, 1999a).

The five interfaces are described as follows.

### **Workflow Definition Interchange (Interface 1)**

This interface is defined as an interchange format and API calls that support the exchange of process definition information over a variety of interchange media.

### **Workflow Client Application Interface (Interface 2)**

The purpose of this interface is to contain the variety behind a standard set of APIs and provide a consistent mechanism for access from a workflow client application to the workflow engine and worklists.

### **Invoked Applications Interface (Interface 3)**

This interface is defined to allow the workflow enactment service to invoke required applications at certain point according to the process definition and transfer workflow relevant data to and from the invoked applications.

Due to the heterogeneous operating environments of today's applications, any particular WfMS implementation might not have sufficient logic to understand how to invoke all potential applications directly. Instead, *Tool Agents* are provided as specialised application drivers, which start up and terminate applications, transfer workflow relevant information to and from applications and control the applications' running status. The basic architecture of *Tool Agents* could be compared with a driver, e.g. ODBC (WfMC, 1998). By using *Tool Agents*, application invocation is conducted in a two-stage manner. Firstly, workflow enactment service communicates and exchanges workflow relevant data with a specific *Tool Agent* through standard API. Secondly, the *Tool Agent* uses its own domain knowledge, logic and interface to invoke and exchange data with the required application.

### **WAPI Interoperability Functions (Interface 4)**

As one of the key objectives of the Coalition, workflow interoperability can be realised on different levels. Although a more ambitious attempt is to enable workflow engines developed by different vendors to interpret a common process definition, share a common set of workflow control data and thus maintain a shared view of process states across different engines, a more realistic target set by the Coalition is to enable parts of a process to

be transferred to a different enactment service for run-time support. Interface 4 defines the common formats of workflow entities need to take in order to be exchanged as well as coordination WAPI calls.

### **Administration and Monitoring Interface (Interface 5)**

This interface specifies a common interface standard for administration and monitoring functions, which enables one vendor's management facility to work with another's workflow engine(s).

Among the five interfaces, Interfaces 1, 3 and 4 are particularly relevant to this thesis. Documented in WfMC (1998)'s *Workflow Management Application Programming Interface Specification*, Interface 3 Invoked Applications was amalgamated into Interface 2 Workflow Client Applications and became a new joint interface called *Workflow Client Application API*. As a result, the new interface includes a set of application programming interfaces (API) that is part of the Workflow Application Programming Interfaces (WAPI). This API set provides the functionalities of worklist handler, interactions with client applications and application invocation by the use of *Tool Agents*. However, for the purpose of clarity, Interface 2 and 3 are still kept separate in this thesis. Interface 4 aims to define standards that will allow workflow systems developed by different vendors to exchange work items seamlessly between one another. However, due to the co-existence of many workflow interoperability standards and the constant change nature of business environment, the effectiveness of Interface 4 is discounted. Further discussion regarding workflow interoperability and standardisation continues in §2.3.

## **2.2.3 Activity Model**

### **2.2.3.1 Activity**

Although being separated, the business logic and underlying applications are still linked in the form of *activity*. The term *activity* is widely used and adopted by organisations, such as WfMC, although its use is opposed by some academicians and practitioners, like Baeyens (2004), who claims that the terms *state* and *action* should be used instead because they provide clearer meanings. Aalst and Hee (2002) highlighted that the term *task* and the term *activity* are different as “an activity is the carrying out of an assigned task... (that) is related

to a specific case”. Putting the debate aside, what is important is the underlying activity-based model itself rather than the name that it is called. In this thesis, the term *activity* is used.

WfMC (1999a) defines an activity as “a description of a piece of work that forms one logical step within a process. An activity may be a manual activity, which does not support computer automation, or a workflow (automated) activity. A workflow activity requires human and/or machine resource(s) to support process execution; where human resource is required an activity is allocated to a workflow participant.”

An activity-based workflow is a workflow that is centred around a set of activities that someone (or something) has to do (Guillaume, 2003). Most commercial products and open source projects adopt this model, e.g., IBM’s WebSphere MQ Workflow and Enhydra’s Shark. The popularity of activity-based workflow is also reflected in the adoption of *activity* as a basic building block in the mainstream process definition languages, such as IBM’s WSFL, WfMC’s XPD and BPMI’s BPML (Shapiro, 2001).

#### **2.2.3.2 Activity-Based Workflow Model**

According to WfMC’s workflow model (1995), individual activities within a workflow process are typically subject to human operations, often realised in conjunction with the use of a particular IT tool (for example, filling in a web-based form), and automated operations requiring a particular software application to operate on some structured information (for example, updating a database storing purchase orders with a new record). Interaction with the process control software is necessary to transfer control between activities, to ascertain the operational status of a process, and to invoke application tools with the appropriate data, etc. There are several benefits in having a framework for supporting this type of interaction, including the use of a consistent interface to multiple workflow systems and the ability to develop common application tools to work with different workflow products.

The ability to distribute tasks and information between participants is a major distinguishing feature of workflow run-time infrastructure. The distribution function may operate at a variety of levels from different workgroups within an organisation to inter-organisation depending upon the scope of the workflows. It may use a variety of underlying

communication mechanisms (electronic mail, message passing, distributed object technology, etc). A top-level view of a workflow architecture that emphasises this distribution aspect is shown in Figure 2-4.

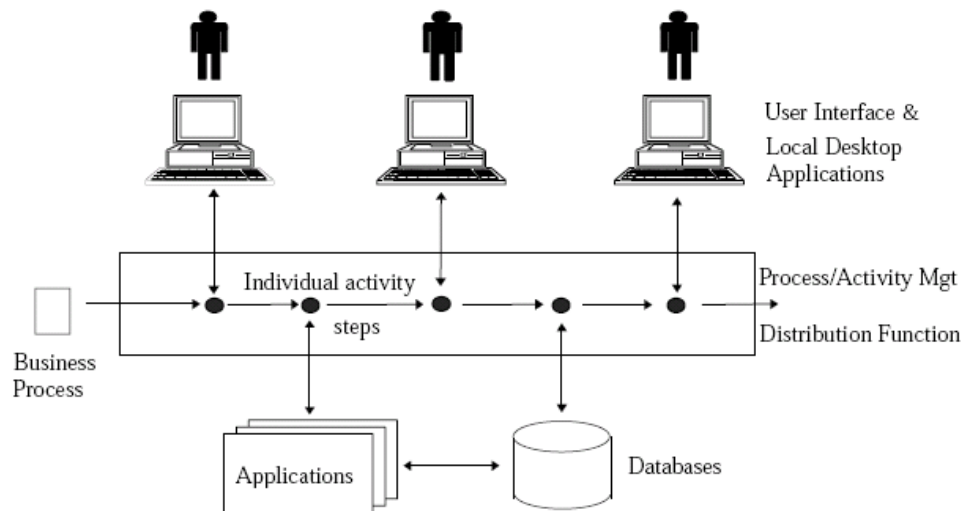


Figure 2-4. Distribution within the workflow enactment service (adapted from WfMC, 1995).

The workflow enactment service is shown as the core infrastructure function with interfaces to users and applications distributed across the workflow domain. Each of these interfaces is a potential point of integration between the workflow enactment service and other infrastructure or application components (WfMC, 1995).

Entity-based workflow is another type of workflow, in which a single entity, e.g., a document, is the main focus. The entity always has a state associated with it and a set of possible transitions to new states if certain conditions are met (Guillaume, 2003). In this thesis, it is included for the purpose of completeness only and will not be further discussed.

#### 2.2.4 Control Flow and Data Flow

According to Aalst (2003), a workflow definition can be viewed as a series of activities linked by control flow, on which data flow rests. Control flow perspective provides an essential insight into a workflow's effectiveness. It is specified as all the transitions between activities. At workflow build time, in order to correctly define control flow dependencies,

not only sequential logic between activities but also data dependencies need to be considered. By following such a design paradigm, the resulting process definition will be correct and at users' expectation in terms of activity sequence, i.e. every activity that succeeds a particular activity in data flow must succeed the same activity in control flow.

According to WfMC (1995), at run time, activities are instantiated and triggered by the workflow engine following the sequence specified in control flow. During the enactment, a set of internal control data (as shown in Figure 2-2) is maintained and managed by the workflow management system to identify the state of individual process or activity instances. The workflow control data is normally not accessible or interchangeable to applications via workflow application interface (WAPI). As to data flow, it is maintained by the WfMS in the form of transferring workflow relevant data (as shown in Figure 2-2) between activities. Not like the workflow control data, the workflow relevant data is accessible to applications and exchangeable between the WfMS and applications via WAPI.

## **2.3 Workflow Interoperability and Standardisation**

Being faced with ever increasing demands from the electronic business community, workflows are pushed to work across organisational boundaries. With various workflow products in the market, interoperability becomes a must-address issue. An obvious approach to workflow interoperability, standardisation is under consideration.

### **2.3.1 Workflow Interoperability**

Workflow interoperability is defined as “the ability of two or more workflow engines to communicate and interoperate in order to coordinate and execute workflow process instances across those engines” (WfMC, 1999b). From this definition, it can be seen that interoperability is a run-time feature of WfMS and there are three aspects that need to be addressed, namely, data flow, control flow and communication. By referencing the generic Workflow Product Implementation Model mentioned in §2.2.2.1, workflow interoperability at a low level can be interpreted as the ability of enabling selected internal workflow relevant data to flow in and out across workflow engines at the desired and/or necessary steps within the execution of workflow processes.

Faced with globalised economy and consolidation within and across industries, organisations no longer function in a standalone manner. Instead they collaborate with each other at different levels for the purpose of either short-term goals and/or long-term strategies. With growing popularity, B2B e-business is adopted to streamline the cross-organisational business transactions, which generates an increasing need for organisations' WfMSs to interoperate with each other. Whenever more than one WfMS are involved, there exists a potential need for workflow interoperability. With more than 150 software vendors providing business process management products and the market continuing to grow strongly in terms of both the user base and the number of vendors (Schurter, 2006), workflow interoperability is posing a demanding challenge.

### **2.3.2 Current State of Standardisation**

Standardisation is viewed as important or even necessary to achieve enterprise interoperability (Chen & Vernadat, 2002; Zelm and Kosanke, 2007). From a pure standardisation point of view, as long as every aspect is complied with a single standard, interoperability can be achieved, which makes standardisation an ideal approach to resolve any interoperability issue. However, the assumption is that there exists only one universal standard that fits the purpose. Unfortunately, standards in the real world are diverse. According to Boyes-Schiller's observation (2003), for the process technology industry, in 1995, there was only one standard group with the Workflow Reference model as the only one standard. In 2003, the number of standardisation groups jumped to ten and there were seven standards available in process modelling alone. The currently available standards are illustrated in Figure 2-5.

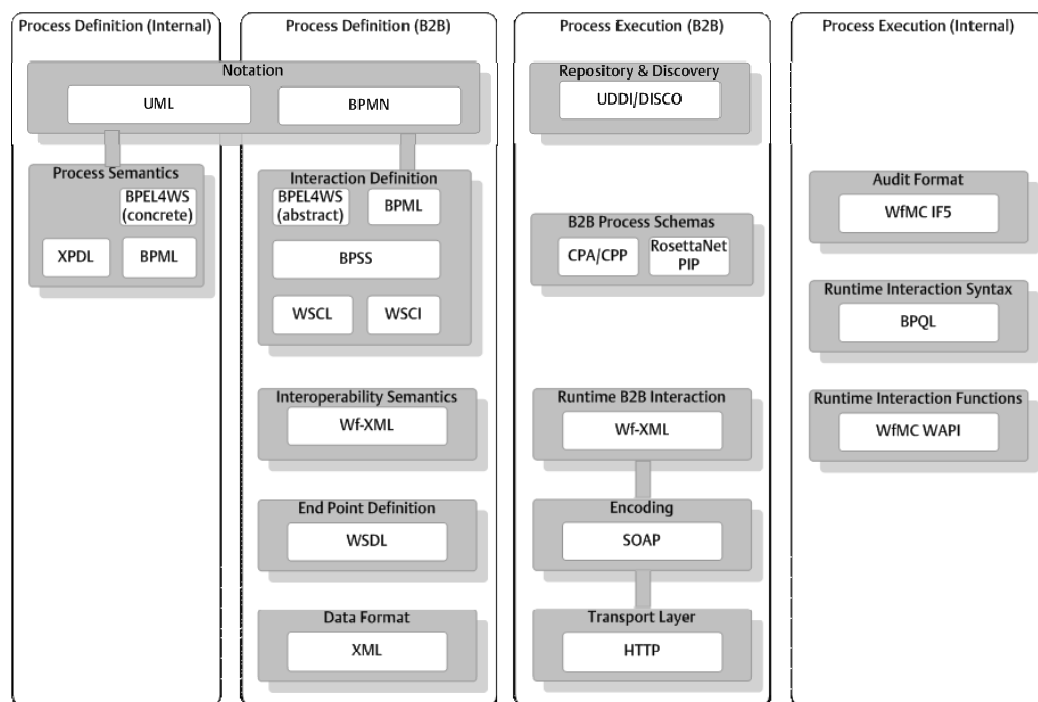


Figure 2-5. Currently available standards regarding process technology (adapted from Boyes-Schiller, 2003).

What is behind the scene of the standard plethora is the reluctant acceptance of a single set of standards. The underlying reason put forward by Boys-Schiller is that standardisation is mostly driven by vendors based on their own assumptions, such as:

- needing of a common platform,
- sharing processes and IP,
- running processes outside one organisation,
- restricting business needs to fit standards,
- paying for the standards by business users;

and with the ignorance of the real needs from business users, which are:

- no need for a “standard” platform,
- not using “standard” procedures,
- not sharing their intellectual property,
- exposing them as wide as possible,
- protecting investment with max agility,
- total control in terms of processes change depending on usage.



It can be observed that for both technical and commercial reasons, standards have not been standardised and therefore some other resorts need to be explored.

## **2.4 WfMS Products**

In a recent survey targeting 1,400 CIOs by Gartner Executive Programs (Gartner, 2006), the top business priority identified by CIOs has been ‘business process improvement’ for the second consecutive year. In order to response such a signal in the market, vendors and developers are keen to provide a variety of workflow/business process management tools. In this section, categories of WfMS are introduced and several WfMS products developed by some major players are described.

### **2.4.1 Categories of WfMS**

According to Ader (1997), four categories are identified in his Workflow Classification Scheme, namely production, administrative, collaborative and ad-hoc.

#### **Production**

Production WfMSs are used to process large number of similar tasks and to improve productivity. Highly repetitive and complex activities are automated, usually in a non-stop manner to achieve Straight-Through-Processing. Human interaction is only required for exceptions handling.

#### **Administrative**

Administrative WfMSs are used to automate administrative tasks featured by a series of form filling. It is required that processes should be easily defined. Once created, the process definition is unlikely changed throughout the execution. Other features include concurrent execution and constant human involvements. Flexibility is viewed much more important than productivity for this category of WfMS.

#### **Collaborative**

Collaborative WfMSs concentrate on less structured business-critical processes where emphasis is on collaboration and contribution from teams of different size. Throughput is no longer a critical issue rather process definitions are subject to frequent changes. The ability

to integrate the WfMS with Internet-based team communications is a key success factor for collaborative WfMSs.

### **Ad-hoc**

Ad-hoc WfMSs are applied to support routine work that is based on unstructured information. Process definitions need to be created quickly and amended on the fly to constantly adapt to new situations. Flexibility and adaptability are the most important features of ad-hoc WfMSs. Also, users own the processes and are allowed to make changes.

## **2.4.2 WfMS Products**

With more than 150 software vendors providing products with regard to business process/workflow management, the marketplace is highly competitive. Different BPM/WfMS products adopt different models and focus on different aspects. Four WfMS from major vendors and one from the open source community are introduced as follows.

### **IBM WebSphere MQ Workflow 3.4**

WebSphere MQ Workflow (formerly known as MQSeries Workflow) is a business process workflow engine built on top of the message queuing facility of WebSphere MQ following the client/server architecture. With a true object-oriented design, WebSphere MQ Workflow offers a high level of re-usability. It supports process management and organisation modelling. Activities can be defined or implemented with the assistance of ActiveX objects and Java APIs. Web Client technology is adopted for rapid application development and JSP files can be automatically generated based on process definitions. With IBM WebSphere Business Integration Modeller and Monitor (former Holosofx BPM tool), WebSphere workflow integrates business process analysis, simulation and development tool together with a comprehensive monitoring environment. It interacts with users through WebSphere features such as forms and portlets.

### **FileNET P8 BPM Suite**

FileNET P8 BPM Suite is the J2EE evolution of the first workflow ever developed – FileNET. It leverages the scalability of the P8 platforms' distributed architecture, its EAI capabilities through CrossWorlds, and a Java/COM API for tailored developments and integration. P8 BPM has a "production capable" process model, with an "ad-hoc capable"

process definition tool running from a Web-browser. P8 BPM Suite natively cooperates with its companion products Content Manager, and Web Content Manager.

### **Staffware Process Suite**

Staffware Process Suite (SPS) offers a balanced solution between production and administrative workflow requirements while delivering high production and path through throughput. Interactive activities implementation uses form definition and a scripting language. Automatic activities can use EAI adapters for scripts, SQL database access, and Tuxedo transactions. SPS provides an EAI adapter to BEA and Actional Control Broker. SPS supports distributed configurations and delivers activities in Lotus Notes and Microsoft Exchange environments. A Process Monitoring tool covers both operation and management needs.

### **TIBCO InConcert**

InConcert integrates object-oriented technology, document management, and a process model enabling users to rapidly build and tailor workflow. Users can modify process instances on-the-fly as exceptions and changes might require. It offers easy definition of procedures that can be deployed through client/server, Java and HTML-based clients. With object-oriented API offered in C, C++, OLE and Java, InConcert works with TIBCO IntegrationManager orchestration engine and TIBCO RendezVous messaging for applications integration. TIBCO BPM Designer supports both InConcert process and IntegrationManager orchestration definitions.

### **Enhydra Shark Workflow**

Among several open source workflow engines, Enhydra's Shark Workflow has been a successful one. It uses an extensible Java/XML workflow engine framework, which includes the implementation that is completely complied with WfMC's specifications. XML Process Definition Language (XPDL) is adopted as its native workflow process definition format and the WfMC *Tool Agents* API for server-side execution of system activities. An XPDL compliance Java based process editor JaWE can be used to specify workflows at build time. A graphic interface-based administration tool is also included in Shark 1.1 to facilitate workflow enactment management at run time. Other popular open source

workflow engines include Icube's Openflow, JBoss jBPM and ECOO's Bonita, to name only a few. Table 2-1 highlights the key features of five WfMSs.

Table 2-1. WfMS products comparison.

<b>WfMSs Features</b>	<b>IBM WebSphere MQ Workflow 3.4</b>	<b>FileNET P8 BPM Suite</b>	<b>Staffware Process Suite</b>	<b>TIBCO InConcert</b>	<b>Enhydra Shark Workflow</b>
<b>Process Design Tool</b>	√	√	√	√	√
<b>Process Management</b>	√	√	√	√	√
<b>Web Client</b>	√	√	√	√	x
<b>EAI Capability</b>	√	√	√	√	x
<b>Programming API</b>	Active X, Java	Java, COM	a scripting language	C, C++, OLE, Java	Java
<b>Ownership</b>	proprietary	proprietary	proprietary	proprietary	open source

## 2.5 Summary and Conclusion

This chapter reviewed key concepts of workflow technology with the emphasis on the generic Workflow Product Implementation Model, the Workflow Reference Model, and the popular underpinning activity model. Also, workflow control and data flows were discussed as two important aspects of a workflow definition. As response to B2B e-business, the issue of workflow interoperability was raised. As an approach to tackle interoperability, the current status of workflow standardisation was revealed. Finally, workflow products were categorised and mainstream ones were introduced.

It is identified that despite the variety of workflow products in the market today, they more or less comply with Workflow Management Coalition (WfMC)'s generic Workflow Product Implementation Model and are underpinned by the activity-based model. As to standardisation in workflow interoperability, due to commercial and technical reasons, a unified interoperability standard has not been achieved. Therefore, in the near future (if not forever), neither a standard process definition language (PDL) nor a set compatible interoperability interfaces can be relied on to achieve workflow interoperability.

# Chapter 3

## **Business Collaboration and Cross-Organisational Workflows**

### **3.1 Introduction**

Business collaboration has long existed for organisations to jointly achieve their goals. With the flourish of information technology and the prevalence of the Internet, the way of business collaboration is transforming. Strongly driven by the business-to-business electronic commerce community, the demand for business collaboration automation is on the rise. With workflow proven as an effective and popular tool for business process automation within organisations, cross-organisational workflow collaboration has become a hot research area for some time, during which various approaches and techniques have been explored. In this chapter, §3.2 depicts the landscape of business collaboration and the application areas of cross-organisational workflow. §3.3 identifies three patterns of cross-organisational workflow, namely hierarchical, composite and peer-to-peer. §3.4 describes an example of workflow collaboration. §3.5 and §3.6 review previous approaches to workflow collaboration at build time and run time respectively. Also in §3.6, fundamentals of message communication are introduced. §3.7 summarises this chapter.

### **3.2 From Business Collaboration to Workflow Collaboration**

Business collaboration is a long existing activity between organisations and has evolved into different forms over the years. In modern enterprises, workflow technology is commonly used for business process automation. Established workflows represent successful business practice and become a crucial part of corporate assets. In the Internet era, electronic business

is chosen by more and more organisations as the preferred way of doing business. It is transforming the way business collaboration is conducted. In response to the increasing automation requirements for e-business collaboration, the concept of workflow collaboration, i.e. collaboration between automated business processes, is emerging.

### **3.2.1 Business Collaboration**

Business collaboration refers to multiple enterprises working together to achieve a business goal (Orriëns and Yang, 2005). Such a goal can be opportunistic in the short term or strategic in the long run. The lifecycle of collaboration can be broken down into several stages, namely negotiating for a deal, reaching an agreement and fulfilling the agreement.

#### **3.2.1.1 Partners' Relationships in Business Collaboration**

Partners involved in business collaborations are in different positions and have various relationships with each other. Generally speaking, there are two types of relationships between business-to-business partners as identified by Tagg (2001) in the context of virtual enterprise (VE): domination and equal partnership. During the formation stage of a domination type VE, a dominant player identifies potential partners. Business (including business processes) of the VE is centrally developed and coordinated by the dominator. Examples include aerospace/automobile manufacturer and parts suppliers, building contractor and subcontractors, and leading retailers and wholesalers. In contrast, an equal partnership-based VE is established through pairwise negotiation and their business processes are kept autonomous. Examples include trade association, research collaboration and unplanned network of continuing contracts.

#### **3.2.1.2 Dynamic Environment**

Business collaboration extends individual business processes beyond organisational boundaries and brings about the concept of cross-organisational business processes. The dynamic nature of the marketplace together with autonomy within individual organisations makes cross-organisational business processes dynamic and unpredictable in the form of external influence. Figure 3-1 shows that among the four identified sources of changes, both internal and external business processes are key to bringing dynamic and unexpected changes (Goranson, 1999), where previous internal business processes of partners are increasingly exposed to the external world. Therefore, for each collaborating organisation,

impacts brought by the changes of external processes need to be effectively addressed when managing cross-organisational business process.

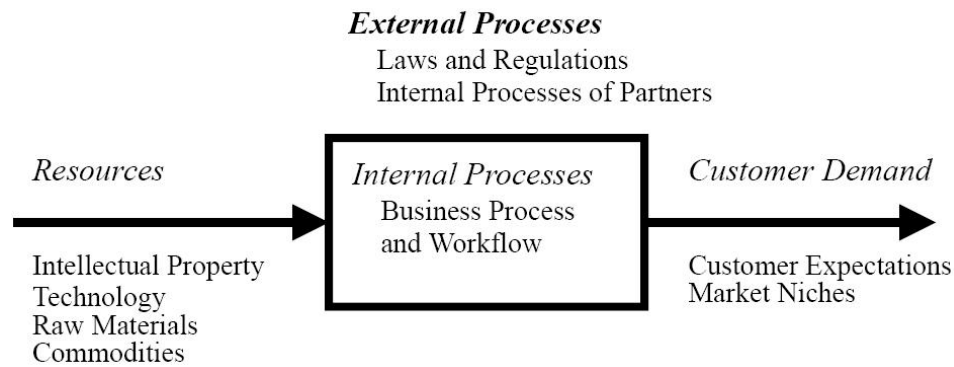


Figure 3-1. Dynamics from four aspects.

### 3.2.1.3 Forms of Business Collaboration

According to different motivations, business collaboration can be categorised as post-completion of merger and acquisition integration, virtual enterprise, and casual trading.

#### Post-Completion of Merger and Acquisition Integration

In today's business world, organisations are constantly seeking opportunities in merger and acquisition (M&A) for the purpose of competition, expansion and survival. According to statistics (Australian Taxation Office, 2008), M&A deals exceeded US\$4.5 trillion worldwide in 2007. Commonly identified as a challenging area, technology integration associated with post-merger transaction draws significant attention of IT professionals. However, post-completion M&A represents an extreme type of collaboration and is located at one end of the business collaboration spectrum as shown in Figure 3-2 as participating partners are literally from one business entity and collaboration becomes internal business operations. Post-completion of M&A process integration normally requires the attention of appropriate integration approach such as enterprise application integration (EAI) to realign the business operations. This is beyond the scope of this thesis and is included only for the purpose of completeness.

#### Virtual Enterprise

By definition, a virtual enterprise (VE) is a temporary aggregation of core competencies and associated resources collaborating to address a specific situation in a business context

(Goranson, 1999). Goranson grouped VE into four types based on their goals, namely opportunity-driven, capability-driven, supplier chain and bidding consortium. Operation style varies according to different relationships among partners. In the case where there is a dominant player, a virtual enterprise behaves and operates more like a traditional organisation. Otherwise, equal partnership reigns. In any type of the VE, business-to-business operational supports are needed for the VE to function effectively and achieve the partners' goals.

### Casual Trade

Collaboration involved in a casual trade transaction is purely opportunistic and the duration is very short. In a *vendor/customer* type collaboration, the most common in the business world, two otherwise independent organisations come together to cash in on a deal without any interest of long-term commitment (Boivie, 2007). The emphasis is on how to seize the opportunity quickly by making a deal after a potential trading partner is located. Due to the uniqueness of the opportunity, each time the partner could be a new one and there is a good chance that no other suitable partners exist. Once a potential partner is located, all the focus should be on how to work out a commonly agreed deal and execute the agreement once it is reached.

From the above, the three forms of business collaboration are included in a spectrum of business collaboration. As shown in Figure 3-2, towards the end of *causal trade*, increasing flexibility and response speed are required. In contrast, the end of *integration for post-completion of M&A* represents collaboration with longer time span and higher degree of coupling.

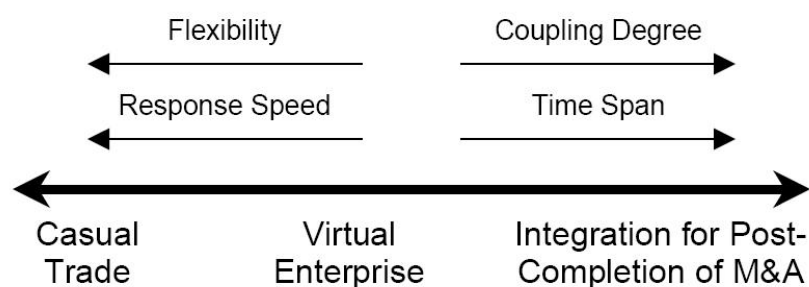


Figure 3-2. Spectrum of business collaboration.



### **3.2.2 Workflow Collaboration**

As an emerging concept, workflow collaboration aims to extend business process automation beyond the organisational boundaries. It can be carried out between conventional workflows. Also, with increasing popularity, web service providers are increasingly asked to expose underlying flow models of their services in order to meet service requestors' more detailed requirements for matching and selecting more complicated services with quality.

#### **3.2.2.1 Between Two Workflows**

In response to the increasing business automation demands, the concept of workflow collaboration, i.e. collaboration between automated business processes, is emerging to extend process automation beyond organisational boundaries. Due to the complementary roles found in business transactions, most business collaborations between different organisations are carried out between their relevant workflows. For example, in order to achieve the goal of purchasing, a buyer should collaborate with a seller by exchanging relevant information (e.g. purchase order, commercial invoice etc.) between their purchase and sales workflows.

#### **3.2.2.2 Between Two “Stateful” Web Services**

Web services have been around for some time. Thanks to the prevalence of the Internet, they become more and more popular. However, due to the increasing complexity of business transactions as well as more detailed requirements in terms of quality of services (QoS) demanded by service requestors, it is unlikely that business transactions wrapped in the form of web services can still be executed in a stateless manner with a single round of message exchange (Fancey, 2005). Web service providers are increasingly asked to expose underlying flow models of their services, which transform web services from stateless to *stateful*. As a result, service requestors can specify their requirements with more details (e.g. by expressing their requirement in flow models), which enables them to match and select more complicated services with quality.

Collaboration between both workflows and stateful web services has the potential to push further business automation across organisational boundaries. However, it also poses

challenges to previous approaches in areas of flow model-based matchmaking, process reconciliation and collaboration execution.

### 3.3 Patterns of Cross-Organisational Workflows

Workflows work together in different manners, which results in different collaboration modelling approaches, communication styles and supporting infrastructures. It is identified that collaboration between workflows can follow a traditional hierarchical structure, a composite style and a peer-to-peer manner according to partners' business relationships. In this section, three patterns of workflow collaboration are described together with the explanation of corresponding business relationships to which these patterns are applied.

#### 3.3.1 Hierarchical Workflow

Hierarchical interactions are often found in stable and long-term relationships between partners. Especially when there is a partner dominating a collaboration, formal and hierarchical collaboration models are likely to be centrally constructed and hosted by this partner. In a hierarchical model, collaboration between business partners is expressed as a single top-level (root) workflow owned by a dominant partner. In the top-level workflow, where necessary, activities can be defined as sub-flows. The hierarchical relationship may be continued across several levels, forming a set of nested sub-flows. This workflow collaboration pattern follows the conventional workflow design paradigm, except that the execution domains of sub-flows belong to different owners.

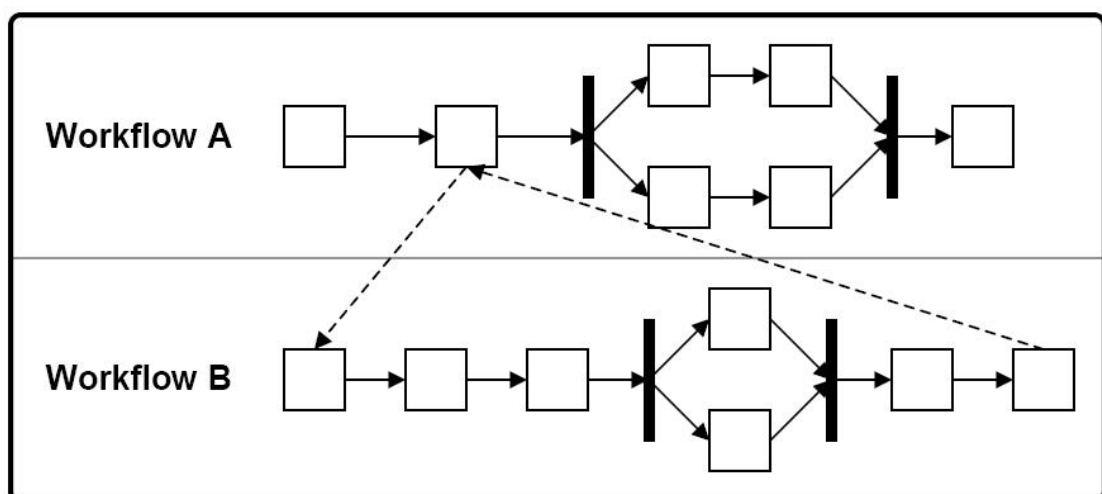


Figure 3-3. Hierarchical cross-organisational workflow.

Figure 3-3 assumes that a workflow instance being enacted on Workflow System A (owned by Partner A) triggers the enactment of a sub-flow instance on Workflow System B (owned by Partner B). The activity on the invoking workflow engine remains suspended until the sub-flow completes all its activities and notifies the parent workflow of changes in either the values of designated parameters or its current state of completion. Such a pattern of workflow collaboration is identified by a number of researchers and organisations as shown in Table 3-1.

Table 3-1. A collection of hierarchical workflows.

Different Names Adopted	Source
nested sub-process	WfMC, 1999b
sub-contracting interorganizational workflow	Aalst, 1999
hierarchical interaction between business partners	Leymann et al., 2002
outsourced collaborative workflow	Schulz, 2004

Another pattern of workflow collaboration described by WfMC (1999b) is *chained processes*. Shown in Figure 3-4, the process instance being enacted on Workflow System A triggers the enactment of another process instance on Workflow System B. Once enactment of the process instance has begun on Workflow System B, Workflow System A may terminate or continue with the enactment of its own process instance. It takes no further interest in the newly created process instance. Since the chained processes pattern is a relaxed version of the hierarchical workflow collaboration by removing the synchronisation mechanism, it is also categorised under this pattern in this thesis.

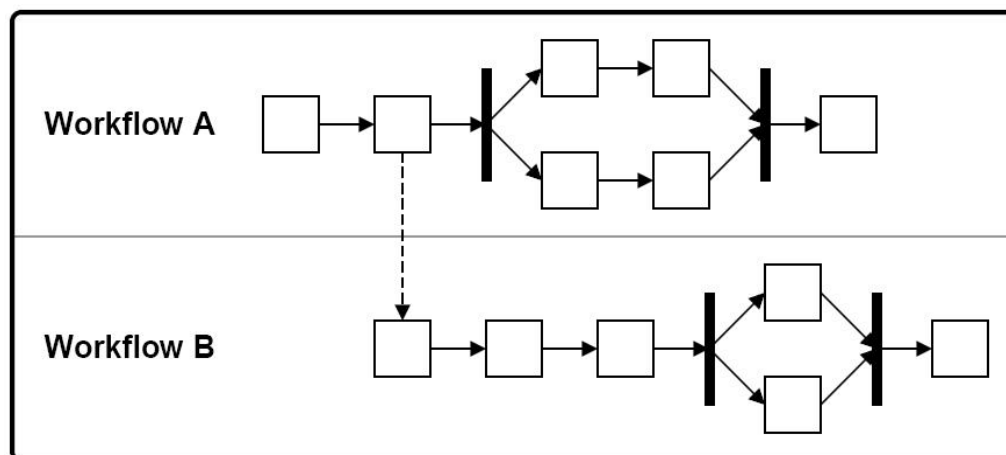


Figure 3-4. Chained flow. (WfMC, 1999b)

### 3.3.2 Composite Workflow

Similar to a hierarchical workflow, a composite workflow is often found in long-term strategic relationships between partners. A composite workflow comprises activities that need to be executed across multiple workflow systems owned by different organisations. As illustrated in Figure 3-5, in process C, activities C1, C2 and C5 need to be executed by Workflow System A of Organisation A whilst C3, C4 and C6 need to be executed by Workflow System B of Organisation B. Such a composite workflow can be designed either solely by a dominant partner or based on the result of a bilateral negotiation between two business partners of equal partnership. This workflow collaboration pattern is identified by a number of researchers and organisations in their projects as summarised in Table 3-2. They are similar but can still be differentiated by some unique features. More discussion can be found in §3.5 and §3.6.

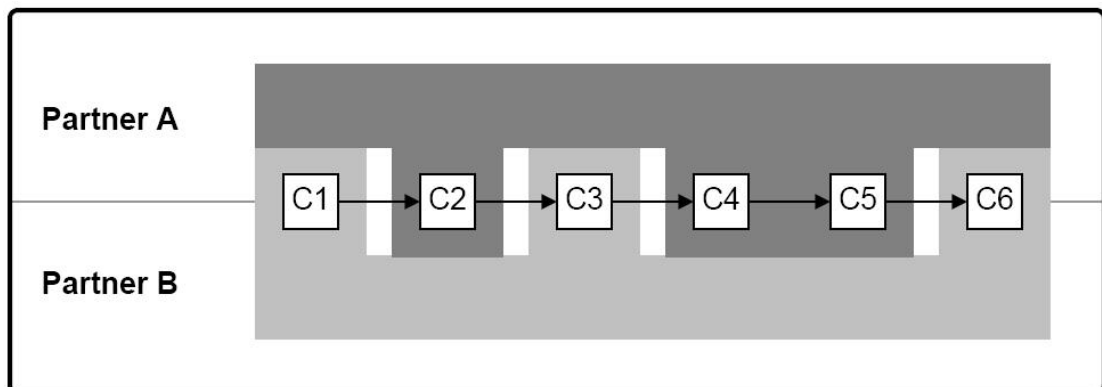


Figure 3-5. Composite workflow.

Table 3-2. A collection of composite workflows.

Different Names Adopted	Source
connected indiscrete workflow	WfMC, 1995
case transfer workflow	Aalst, 1999
distributed workflow	Schulz, 2004

### 3.3.3 Peer-to-Peer Workflow

Peer-to-peer workflow (Figure 3-6) reflects relationships that are often established dynamically on a per-instance basis. It is the most general pattern of workflow collaboration in terms of the interaction structure, in which the composite workflow (as described in §3.3.2) is implicitly specified by the flow instances of the peer workflows on both sides. Business partners of equal partnership are most likely to follow this workflow collaboration pattern although it is also possible that a dominant partner insists other subordinate partners to use this pattern due to business transactions' needs.

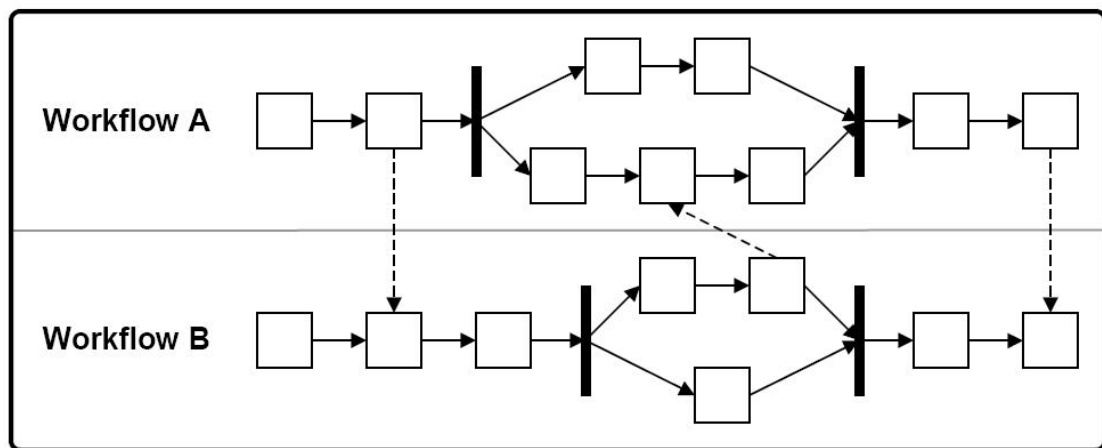


Figure 3-6. Peer-to-peer workflow.

This pattern of workflow collaboration is identified by a number of researchers and organisations as shown in Table 3-3.

Table 3-3. A collection of peer-to-peer workflows.

Different Names Adopted	Source
event synchronised sub-process	WfMC, 1999b
loosely coupled interorganisational workflow	Aalst, 1999
peer-to-peer interaction	Leymann et al., 2002

### 3.4 An Example of Workflow Collaboration

To illustrate the way that workflows collaborate with each other, a fictitious example based on business transactions of the trade community is introduced, which comprises a vendor *Vendor* and a customer *Customer\_1*. Each of them has an existing workflow as shown in Figure 3-7. Solid and dashed arrows represent control and data flow respectively.

*Vendor* in Figure 3-7 represents a manufacturer who produces and exports its products to overseas markets. After it receives an advance payment from a customer, it begins the manufacturing process, which is followed by issuing a *Commercial Invoice*, shown as *Invoice*. Factory inspection is conducted as a standard procedure and an *Inspection Certification*, shown as *Insp Cert*, is produced. The *Inspection Certificate* is sent to the customer and *Vendor* then waits for a *Shipping Advice* as a signal to start shipping the goods. Since *Vendor* normally uses CIF (Cost, Insurance and Freight) as its trade term, it needs to arrange shipment as well as insure the goods. After an *Insurance Policy*, shown as *Ins Policy*, and a *Bill of Lading*, shown as *B/L*, are obtained, they are sent to the customer. Simultaneously, *Commercial Invoice* is also sent out. After applying for a *Certificate of Origin*, shown as *Cert of Origin*, from *Vendor's* local authority and sending it to the customer, *Vendor* is waiting for the final *Invoice Payment*, shown as *Inv Pay*. As business requirements, *Commercial Invoice* is needed when *Vendor* arranges shipment and cargo insurance, and applies for the *Certificate of Origin*, which are reflected in data flow.

*Customer\_1* is an overseas importer. After effecting the advance payment to a desired vendor, it needs to review the *Inspection Certificate* issued by the vendor as a proof of the quality of the goods. Satisfied with the pre-shipment inspection, *Customer\_1* issues a *Shipping Advice* to inform the vendor to ship the goods. It then needs to use the *Commercial Invoice* received from the vendor to declare the goods at its Customs and waits for the *Bill of Lading* to get the goods from the shipper. Since *Customer\_1* also applies CIF terms, the original copy of the *Insurance Policy* is expected to arrive too. After the goods are delivered, *Customer\_1's* own inspection will be carried out. At last, *Customer\_1* needs to have the *Certificate of Original* as an official proof of the goods' country of origin before approving and effecting payment for the rest of the invoice amount to the vendor.

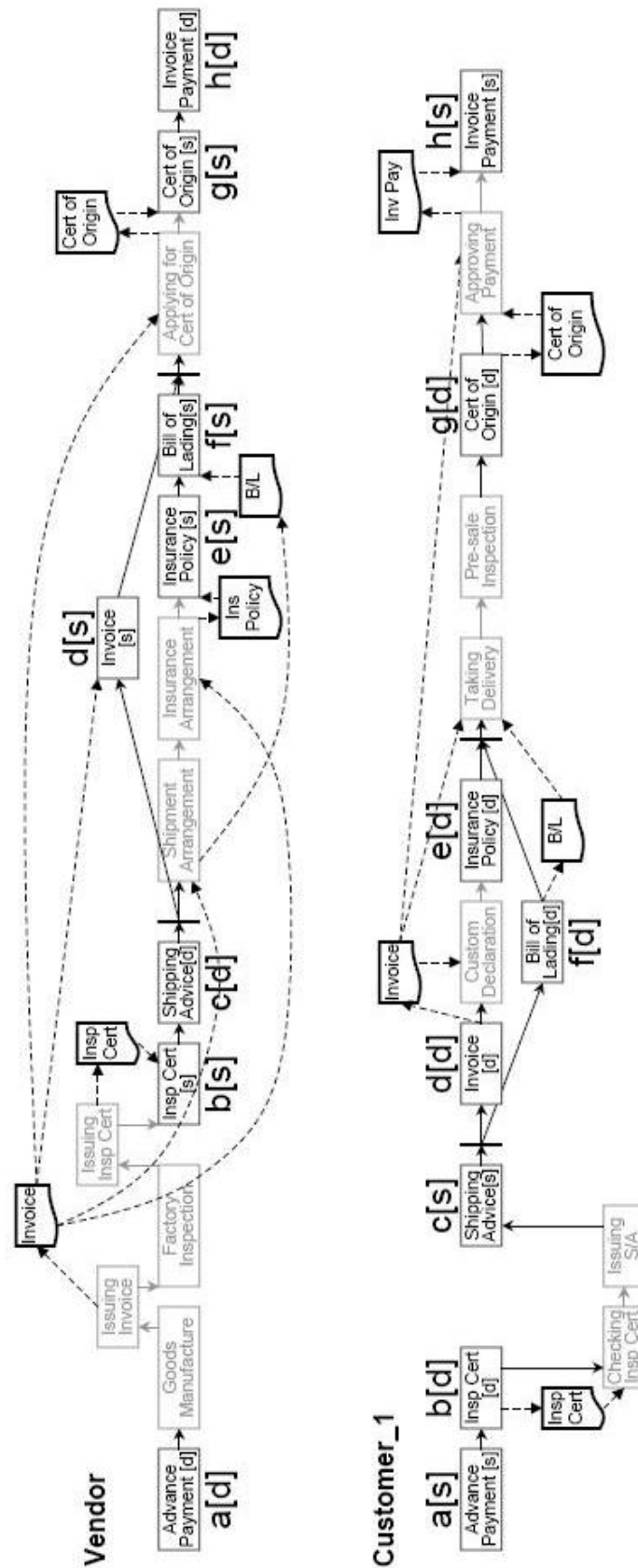


Figure 3-7. An example of collaboration between workflows of Vendor and Customer\_1.

During the remainder of this thesis, the case depicted in Figure 3-7 and its variations are referred to in order to explain relevant concepts and approaches.

### **3.5 Previous Build-Time Collaboration Approaches**

Previous workflow collaboration approaches at build time can be categorised according to the effort involved during the course of bringing about executable cross-organisational workflow definitions. Four types of approaches are identified in this thesis, namely concrete process modelling, abstract interaction modelling, service discovery and automated reconciliation. For the first two approaches, the starting point is the collaboration agreement as a result of the bilateral negotiation between the two partners, upon which either a concrete workflow definition or an abstract cross-organisational workflow model needs to be manually developed from scratch. Both service discovery and automated reconciliation directly gets into the procedures of comparison and matching between participating workflows. The difference is the service discovery approach terminates when any discrepancy between the two workflows is encountered while the automated reconciliation approach has the ability to reconcile some minor differences.

#### **3.5.1 Concrete Workflow Modelling Approach**

The most effort-consuming collaboration approach is concrete workflow modelling, in which the whole collaboration between business partners together with each partner's private workflow is treated as a single new workflow spanning across organisational boundaries. The modelling of the overall workflow is based on the common agreement produced by the collaboration negotiation. Therefore, to some extends, principles and techniques of modelling a concrete cross-organisational workflow are very similar to those of modelling a private workflow and thus can still be applied. However, it does impose certain requirements for current workflow systems. This is because a cross-organisational workflow can be enacted either centrally or in a distribution manner. For centralised enactment, a centralised workflow engine is needed. For enactment in a distributed manner, the workflow definition language needs to support interoperability in terms of remote activity invocation and sub-flow calls. Both of these are run-time features and will be addressed in §3.6.



Also, the close coupling between partners implied by the concrete process modelling approach leaves very little space for privacy in terms of details of workflows. This is only acceptable for some collaborations within certain virtual enterprise but in general it is not the case. In contrast, as corporate assets, business processes and their executable forms – workflows – represent sensitive and private knowledge. Even in business collaboration, only necessary information is exposed to partners in a controlled manner.

### **3.5.2 Abstract Interaction Modelling Approach**

A number of research projects and public standards have been trying to address cross-organisational workflow without defining an overall executable workflow. Instead, centralised coordination mechanism is modelled in the form of an abstract flow model, which is then split among partners for implementation as local collaborative workflows. Due to the number of research projects adopting the approach, it can be described as one of the most popular approaches to cross-organisational workflow modelling so far. Some of them are described as follows.

#### **3.5.2.1 Coalition Workflow**

Schulz and Orlowska (2004) approach cross-organisational workflow from the aspect of workflow view, around which a three-tier cross-organisational workflow model is proposed, i.e. coalition workflow, workflow view and private workflow. The mechanism of the approach is illustrated in Figure 3-8 and explained as follows:

- **defining coalition workflow** – a coalition workflow containing a series of abstract activities is constructed based on the agreement reached between business partners,
- **forming workflow views** – partners choose the tasks in the coalition workflow that they want to implement privately and obtain the required relationship of these tasks in the context of the coalition,
- **synchronising workflow views** – add artificial route activities (AND-splits and AND-joins) to workflow views according to the coalition workflow definition,
- **connecting workflow views to private workflows** – each partner then either develops new private workflows or re-uses existing private workflows, and connects them with their workflow views through state dependencies.

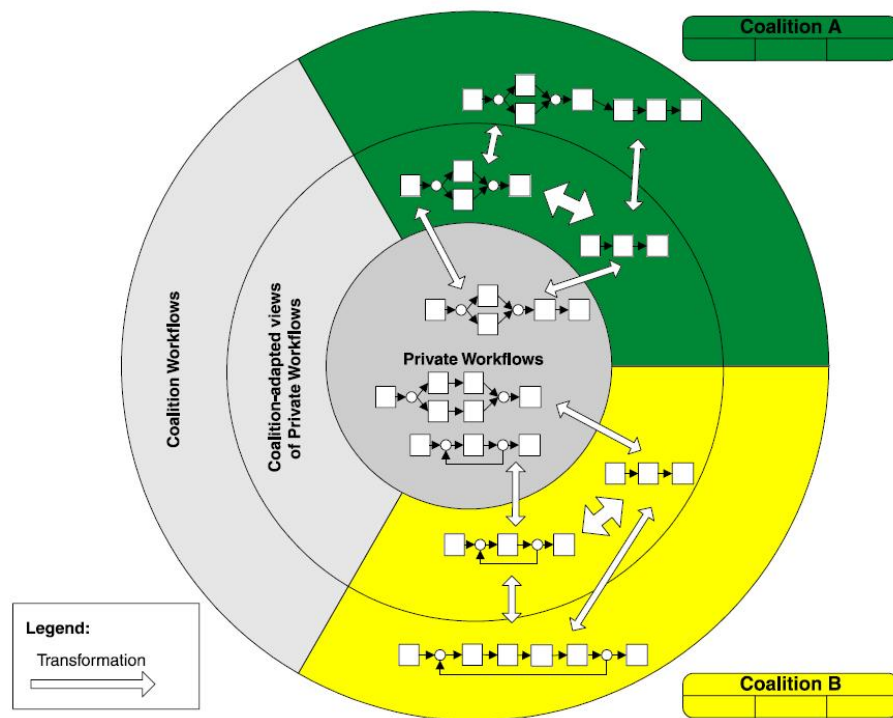


Figure 3-8. Coalition workflow, workflow views and private workflows (adapted from Schulz and Orłowska, 2004).

### 3.5.2.2 Coordination Dialogue

Biegus and Branki (2004) propose a dialogue approach to coordinate the interaction between two workflows. The two participating workflows must implement the two roles defined in a dialogue. A dialogue specifies messages in request-response pairs. Two types of messages are identified in a dialogue: final (terminate the dialogue) and non-final (requires a response). Figure 3-9 shows an example of a dialogue definition in the form of bipartite graph. In this example, *C* is a customer and *V* is a vendor. *C* makes an *Enquiry* to *V*. *V* needs to decide whether to prepare a *Quotation* or to reject the enquiry (*Rejection*). If *V* prepares a quotation for *C*, *C* needs to decide whether to accept it (*Acceptance*), reject it (*Rejection*) or request amendment to the quotation (*RAQ*). If *C* requests an amendment, *V* will decide whether to amend the quotation (*Quotation*) or reject the amendment request (*Rejection*). In this example, *Acceptance* and *Rejection* are final messages whilst *Enquiry*, *Quotation* and request for amendments (*RAQ*) are non-final.

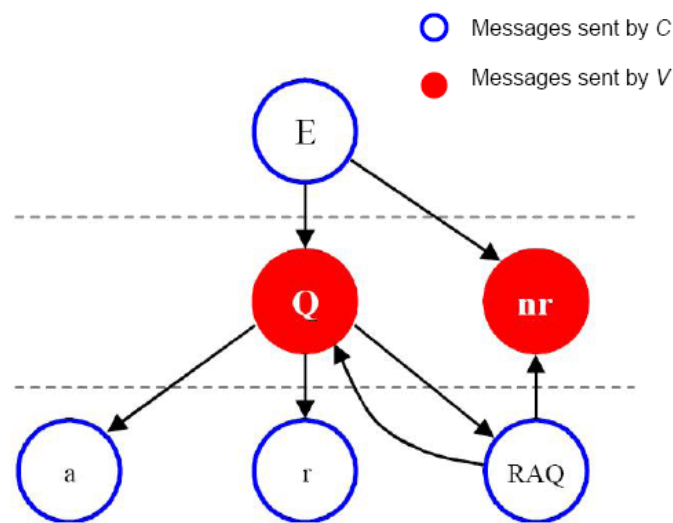


Figure 3-9. An example of coordination dialogue (adapted from Biegus and Branki, 2004).

Two types of activities distinguished as *Send* and *Wait* need to be added to the process definition language, which, when reached, reference the dialogue definition to verify whether the occurrence of the current message is consistent with the dialogue.

### 3.5.2.3 Interorganisational Workflow and Message Sequence Charts

According to Aalst (2001), organisations involved in a business collaboration should agree on a common public workflow. Such a public workflow needs to be partitioned among partners and each partner then autonomously designs a private workflow within the constraint of its part of the public workflow. As the starting point for the design of a complex interorganisational workflow, *Message Sequence Charts* (MSC) is used to capture the communication structure (Aalst, 1999). At the end of the design phase, MSC can also be used to decide whether the newly constructed interorganisational workflow meets the communication specification captured in the MSC.

### 3.5.2.4 Public B2B Collaboration Standards

A couple of public B2B collaboration protocols/standards for e-business in different industries have been developed to coordinate collaboration and streamline operations. Partners need to agree on which public protocols they want to adopt and then implement the activities corresponding to the role they choose.

### **Financial Supply Chain Solution by Bolero.net**

Bolero.net specialises in streamlining financial supply chain among organisations involved in international trade. Financial supply chain refers to the end-to-end trade processes and information that drive a company's cash, accounts, and working capital. The Bolero Trusted Trade Platform is the core infrastructure underpinning all Bolero services. It works as a hub connecting trade partners, banks, insurers, carriers, and freight forwarders that participate in a trade cycle, upon which Documentary Credit and Open Account automation and optimisation can be achieved by adopting relevant flow model-driven application suites (Bolero.net, 2008).

### **Partner Interface Processes of RosettaNet**

RosettaNet is a consortium of major computer and consumer electronics, electronic components, semiconductor manufacturing, telecommunications and logistics companies. Since 1998, it has been working on defining industry-wide, open e-business process standards. As one of the consortium's most important standards, the Partner Interface Processes (PIPs) define the processes and data elements necessary for a broad set of supply chain scenarios. In PIPs, common interface tasks for supply chain collaboration are defined. It enables partners to plug their internal processes to the interface processes for execution (RosettaNet, 2008).

Compared with the concrete workflow modelling approach, the abstract interaction modelling approach starts decoupling the close ties embedded in the single overall workflow definition. Activities are distributed between both partners. However, dependencies between the abstract flow model and the private workflows need to be explicitly specified in the abstract flow model, which makes the coupling still tight.

### **3.5.3 Service Discovery Approach**

Having realised the expensive cost of negotiation associated with the explicit modelling approach, researchers started exploring new approaches that recognise the importance of existing workflows. The new approaches are based on comparing and matching potential business partners' existing workflows to decide whether they can collaborate with each other.

Driven by the idea of streamlining B2B e-business, in openXchange project, Krukkert (2003) found a solution that takes two activity diagrams as input and compares them to discover all common execution sequences. If there is a successful match, a common process represented in the form of an activity diagram is constructed based on the discovered common sequences for both partners for collaboration. Although Krukkert does not discuss enactment based on the discovered common process, it can be deduced that the next possible move could be either to use the resulting common process as a centralised workflow definition or split it into a symmetric pair of workflow definitions, distribute (and/or implement) them on both sides and enact the workflows in a distributed manner.

In order to match two activity diagrams with graph theory methods, parallel structures in activity diagrams need to be eliminated. With the assumption that all the involving activities are atomic (Krukkert, 2003), parallel structures are represented in the form of branching time parallelism (Pratt, 1991), which enables activity diagrams to be converted to standard state transition systems. Further details regarding activity diagram conversion can be found in Chapter 4 and Chapter 6.

From a negotiation aspect, Krukkert's approach can be grouped as following a mutual gain approach (details of the mutual gain approach will be discussed in Chapter 4). Despite Krukkert's contribution to process comparison methodology, two questions still need to be asked regarding his approach:

- Is it necessary for the two initial business processes to be replaced by the common process?
- What if no common sequence is found?

The answer to the first question depends on the collaboration compatibility definition and the enactment infrastructure. However, it is preferable that the original business processes are kept when a match is found according to the organisational knowledge preservation principle. In §4.3, a collaboration modelling approach will be introduced, which enables collaboration to function even when the participating processes are not the same. The second question is common to all the techniques that follow the service discovery approach.

Krukkert's approach is not able to deal with situations where no common sequence is found. To tackle such situations, conflict reconciliation mechanism needs to be introduced.

### **3.5.4 Automated Reconciliation Approach**

Byde, Piccinelli and Lamersdorf (2002) propose a negotiation framework, which they claim can conduct automated negotiation over B2B processes. The idea is to get a unified process out of a combination of both participating processes, compare the unified process with both of the initial input processes to reveal the differences and indicate the differences together with their cost to the user for decision making. However, a closer investigation reveals that the automated reconciliation mechanism is barely an extension of the transformation model proposed by Aalst and Anyanwu (1999), which can only tackle differences caused by different activity content through methods called blocking and hiding. All differences with regard to activity sequences are deemed as irreconcilable. Secondly, it is not clear which process comparison technique it adopts to reveal the difference between the initial process and the automatically acquired unified process.

## **3.6 Previous Run-Time Collaboration Approaches**

In this section, one centralised and three distributed run-time approaches (sub-flow invocation, case transfer and coordinated data exchange) are introduced with corresponding examples.

### **3.6.1 Centralised Workflow Enactment Approach**

By following a centralised workflow enactment approach, the standalone workflow enactment service is expanded across organisational boundaries. Business partners are treated as workflow participants and are required to share private data, a common process definition and a centralised workflow engine as shown in Figure 3-10. Due to the high level of private information exposure and the centralised operation style, it is only suitable for collaboration based on very close business relationship, e.g. different operational arms within one organisation or virtual enterprise with long-term commitment. It will be too expensive and rigid for business partners with equal partnership as well as business collaborations taking the form of casual trade to adopt such an approach.

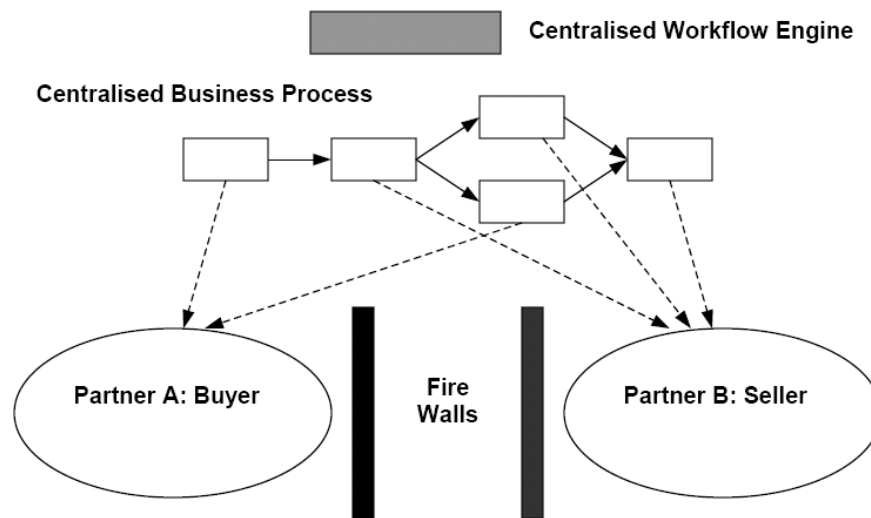


Figure 3-10. Centralised workflow enactment (adapted from Chen and Hsu, 2001)

### 3.6.2 Sub-Flow Invocation Approach

In order to enact cross-organisational workflow of the hierarchical pattern (mentioned in §3.3.1) in a distributed environment, the process definition language (PDL) and workflow engines must support sub-flow invocation mechanism through a commonly agreed interface. WfMC (1999b) drafts abstract interoperability specifications for this purpose, in which a series of operations are proposed and form part of the Interface 4 as depicted in the workflow reference model in Chapter 2. Operations required to start a chained process and those required to complete a nested sub-flow are summarised in Table 3-4. A collaboration message acts exactly as an operation call message in UML 2.1, which contains two elements – name of the operation (workflow control data) and parameters passed to the operation (workflow relevant data).

Table 3-4. Operations required for hierarchical workflow execution.

Start a Chained Process		Start and Complete a Nested Sub-Flow	
Wf Engine	Operations	Wf Engine	Operations
A	Create Process Instance	A	Create Process Instance
B	Response	B	Response
A	Set Process Instance Attributes	A	Set Process Instance Attributes
B	Response	B	Response

B	Get Process Instance Attributes	B	Get Process Instance Attributes
A	Response	A	Response
A	Change Process Instance State	A	Change Process Instance State
B	Response	B	Response
A	Relinquish Process Instance	B	Process Instance Attribute Changed
B	Response	A	Response
		A	Get Process Instance Attributes
		B	Response
		A	Relinquish Process Instance
		B	Response

However, this requires interoperability at PDL level. Given the current PDL standardisation reality as mentioned in §2.3.2, this approach is technically feasible but not practically viable.

### 3.6.3 Workflow Case Transfer Approach

By applying the approach of workflow case transfer, all partners share one definition of a cross-organisational workflow. At run time, a copy of the workflow definition is distributed to and enacted by each partner's local workflow engine. Partners recognise their own share of the activities based on role-matching and is only responsible for these activities. Whenever a partner finishes its own share of activities, it needs to inform others partners about the current progress in order for them to synchronise their workflow instance and be prepared for the subsequent activities. Workflow control and relevant data are enclosed in collaboration messages and exchanged between workflow engines. As illustrated in Figure 3-11, at build time it is decided that Partner A is in charge of *C1* and *C3* whilst Partner B takes care of *C2* and *C4*. At run time, execution is first active in Workflow System A. After *C1* is completed, Workflow System A sends a collaboration message to inform Workflow System B about the latest enactment status in terms of workflow control and relevant data, which leads to the activation of *C2* in Workflow System B. Upon completion of *C2*, the case is returned to Workflow System A and so on. As a real-world example, Chen and Hsu (2001)'s Collaborative Process Manager (CPM) follows such an approach.



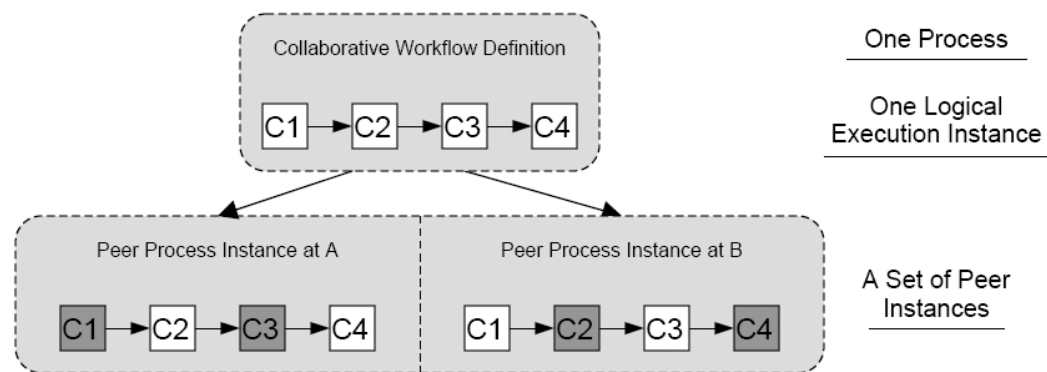


Figure 3-11. Workflow case transfer approach (adapted from Chen and Hsu, 2001)

The approach is targeting the distributed run-time environment. Also, benefiting from a single global view, workflow collaboration management is straightforward. However, the approach assumes either the deployment of homogeneous workflow engines among all the business partners or a satisfactory level of standardisation among process definition languages where different PDLs and WfMSs are adopted. Furthermore, in order to support case transfer and specify collaborative process, mainstream process definition languages need to be extended as described by Chen and Hsu (2001) with regard to their extended Collaborative Process Definition Language (CPDL).

### 3.6.4 Coordinated Data Exchange Approach

The approach is named as coordinated data exchange because only workflow relevant data is exchanged between collaborating workflow engines. The coordinating control flow that has been explicitly specified can be either centrally positioned or distributed among partners. Apart from being specified in the form of a workflow, a control flow can also take other abstract forms, e.g. a Message Sequence Chart, a coordination dialogue definition and some public B2B collaboration standards as mentioned in §3.5.2 Abstract Interaction Modelling Approach.

At run time, whenever an interaction point is reached in a local workflow, as a common feature, the local workflow system needs to reference the abstract coordinating flow model to ensure the legitimacy of the current interaction in terms of its sequence. Routing information is obtained from the abstract model as well. For example, in the workflow view

approach proposed by Schulz and Orlowska (2004), a coalition workflow is centrally specified and positioned. At run time, the coalition workflow can be used actively to drive the collaboration; or it can be referenced passively by activities in the workflow views that conduct conversation with their peers. Technically, the conversation is supported by Simple Object Access Protocol (SOAP) and higher-level protocols such as WS-Coordination and WS-Transaction. In Biegus and Branki (2004)'s InDiA, a coordination dialogue is centrally defined and distributed among the partners in the form of an extended process definition language. When an interaction point is encountered, the coordination dialogue is referenced locally. Once the interaction is found complying with the dialogue, collaboration messages are exchanged. Since InDiA adopts an agent-enhanced approach, these tasks are conducted by various agents.

### **3.6.5 Messaging in a Nutshell**

In a distributed environment, partners need to exchange information, e.g. request, response and data, to proceed to the collaboration. The directed information exchange between computer systems is often referred to as messaging (Schulz, 2002). Two important issues in messaging, message content and message passing, are discussed.

#### **3.6.5.1 Message Content**

Information carried by a message is organised in a common syntactic format in order for the recipient to interpret what the sender includes in the message. However, to achieve an unambiguous understanding of the semantics of content between partners, an effective ontology is needed to map apparently different terms. A number of researchers and organisations are working on the research and application domains of ontology. A selection of several widely acknowledged organisations specialised in business and financial domain ontology is introduced as follows.

**RosettaNet Business Dictionary (RNBD)** designates the properties used in basic business activities to support Partner Interface Processes (PIP) – a collection of predefined process templates for partners to adopt and implement. RNBD as well as PIP are developed by RosettaNet, a globally supported standards organisation with the endorsement from more than 500 companies around the world (RosettaNet, 2002).

**SWIFTStandards** is part of the services provided by the Society for Worldwide Inter-bank Financial Telecommunication (SWIFT), a member-owned cooperative providing proprietary communications platform, products and services for over 8300 institutional clients in more than 208 countries. SWIFTStandards develops business standards to support transactions in the financial markets for commercial and inter-bank payments, securities, trade services, treasury and over-the-counter derivatives. Complemented by new XML-based (MX) messages (to the traditional MT messages), transfer of richer data for more complex business transactions becomes possible (SWIFT, 2008).

**International Financial Reporting Standards (IFRS)** are standards and interpretations adopted by the International Accounting Standards Board (IASB). The latest IFRS Taxonomy Guide 1.00 was published in August 2008 (IASB, 2008), which is specified by eXtensible Business Reporting Language (XBRL).

Ontology is an important and challenging issue for any collaboration related topic. However, it is not the key concern of this thesis. It is assumed that all the business terms of both partners can be properly mapped onto a well established ontology, e.g. the RosettaNet Business Dictionary.

### **3.6.5.2 Message Communication**

In general, there are two ways for a message to be passed from the source to the destination, namely peer-to-peer and mediated.

#### **Peer-to-Peer Communication**

In peer-to-peer message communication, all partners directly contact each other to exchange messages. It is a simple method but since the approach must be based on their explicit knowledge about each other, it implies tight coupling between partners.

#### **Mediated Communication**

The main difference between a mediated communication and a peer-to-peer one is that in a mediated environment, a third entity (or mediator) is required to route information between communication partners, who may not know and do not have to know each other. Based on whether the information passed through the mediator is logged, a mediated communication

can be further differentiated into stateless and stateful. A stateless mediation requires less in terms of message storage facility while the information logged by a mediator can be used for collaboration progress monitoring and error handling.

As to the question of ‘Which to choose?’, it depends on the purpose. Where clear information regarding message senders/recipients is available, peer-to-peer communication might be more appreciate due to direct and simple message passing. However, when the degree of coupling is the major concern, a mediated communication can significantly bring down the coupling level.

### **3.7 Summary and Conclusion**

This chapter studies key aspects of business collaboration and in particular workflow collaboration so as to automate business process collaboration. It categorises cross-organisational workflows as three patterns. An example of workflow collaboration is described. Previous approaches for cross-organisational workflows at both build time and run time are reviewed.

It is discovered that by following the existing build-time approaches, the negotiation process is left with little IT support, which is expensive, inflexible and error-prone for cross-organisational business collaboration, the casual trade type in particular. Also, all the identified run-time approaches are tied up with the conventional build-time philosophies. Therefore, a novel approach needs to be found.

# Part II

## Approach, Framework and Enabling Techniques

*"Strive for perfection in everything you do.  
Take the best that exists and make it better.  
When it does not exist, design it. "*

*– Sir Henry Royce*

# Chapter 4

## An Integrated Approach

### 4.1 Introduction

In order to address cross-organisational workflow collaboration more effectively, a set of new requirements is identified in response to the limitations of previous approaches. Due to the wide spectrum of issues covered in these requirements, an integrated approach is adopted and relevant topics are introduced and reviewed.

In this chapter, §4.2 identifies new requirements essential to the realisation of the aim of the thesis and describes three integrated dimensions of the approach. §4.3 introduces a new distributed workflow coordination mechanism §4.4 reviews key aspects of the negotiation activity. §4.5 discusses matching techniques for processes. §4.6 summarises the chapter.

### 4.2 New Requirements and an Integrated Approach

With the recognition of the limitations of the centralised cross-organisational workflow modelling approach, all the unanswered questions left by the service discovery approach and the incapability suffered by the automated adjustment approach, a novel solution is needed to satisfy the demanding goals set by cross-organisational workflow collaboration. Based on the assumptions that define the starting point and the scope of the thesis, an integrated approach is adopted to address the target problem from three dimensions. A number of requirements are identified and discussed.

### 4.2.1 Assumptions

To keep this thesis focused on the key issues, the following assumptions are made. They define the starting point, mark the scope and provide a clear context for the work reported in the thesis.

**Bilateral Collaboration:** Collaboration is carried out between two partners but not any particular two.

**Workflow-Enabled Environment:** Both partners have established workflow solutions for their business processes, which participate in the collaboration.

**Non-Split Process:** At process build time for collaboration, it is assumed that all the processes under discussion have already been transformed into a non-split form, i.e. they only have sequential and/or parallel structures. The reasons for excluding exclusive branch are: firstly, the navigation criteria for different exclusive OR branches could be sensitive and should not be directly exposed to external partners during collaboration; secondly, a process with exclusive OR structures can always be broken down into several non-split sub-processes by using the approach proposed by Juan and Ou-Yang (2005).

**Readiness for Participation:** Workflows intending for collaboration already have corresponding interface activities inserted (*data supply activity* and *data demand activity*) at the desired positions. Each interface activity is associated with a semantic identifier with regard to the collaboration message that the activity is going to exchange with its counterpart.

### 4.2.2 New Requirements

A number of requirements are identified to target the problems of existing approaches. It is expected to bridge the automation gap in business collaboration with adequate IT supports.

#### 4.2.2.1 Loose Coupling Principle

It is crucial to be aware that no matter which type of business relationship is between the partners, business collaboration takes place in a distributed environment. Participants are independent business entities and have autonomous control over their own business

processes involved in collaboration, although the level of control varies. As a golden rule in business-to-business integration, loose coupling, i.e. to have less detailed knowledge about trading partners' private business processes, enables organisations to evolve their processes without affecting their partners' processes (Keen et al., 2006). This is particularly suited to the distributed and autonomous environment and therefore needs to be applied to the proposed solution.

#### **4.2.2.2 Negotiation Support**

As mentioned in Chapter 3, by following a cross-organisational workflow modelling approach, an automation gap exists between the enactment of private workflows and workflow collaboration. This is because the conventional negotiation approach followed by manual process design is applied. Even the service discovery approach that starts from existing executable workflows, there are still many questions left unanswered. For example, what if existing workflows from the two desired business partners do not match?

A simple answer is that the two partners should carry out negotiation to see whether an agreement can be reached. Therefore, it can be seen that negotiation support becomes the focal point. It is desirable that the proposed solution is able to integrate the task of negotiation with the aid of information technology for the purpose of bridging the automation gap as much as possible. In order to do so, negotiation should be properly understood and captured in association with workflow technology.

#### **4.2.2.3 Decision Making Support**

At the end of each conflict reconciliation cycle in an overall negotiation process, human users must make the decisions to accept or reject the options associated with certain process adjustments. Due to the complexity of the concessions implicated by each decision, in order for a user to make informed decisions, information and the way it is presented are two crucial aspects.

#### **Information Required**

The first piece of information required is the adjustment suggestions on both the control flow and/or data flow that contribute directly to the reconciliation of any discrepancy. With



this information, the user will know what to do when being faced with a discrepancy situation.

Secondly, due to the causal relationship among process activities, the user needs to know all implications on control flow and data flow if a particular option is taken. Such information enables the user to evaluate their choices.

### **Information Presentation**

Due to the importance of user involvement and interaction in the process reconciliation stage, information should be presented in an interactive and human understandable way. Also, fully informed decision making means that relevant information should be presented to the user before their final decision is made.

#### **4.2.2.4 Process Change Support**

After the user has made a decision the next task is to adjust the process and the workflow definition accordingly. However, due to the inherent technical nature as well as the associated workload, this task poses several challenges to business professionals, such as ensuring completeness and maintaining integrity of a process.

### **Ensuring Completeness**

A process should be changed to reflect users' latest decisions regarding the reconciliation options. IT support is needed in seamlessly associating process change with decision-making to ease the workload for the user and to ensure completeness of the process.

### **Maintaining Integrity**

Every change operation must maintain the integrity of the digraph representation of the process. Process digraph integrity means that there is no disconnected vertex in the graph, i.e. every vertex (except the Start and End vertex) should have at least one incoming and one outgoing edges. Maintaining integrity manually is resource-intensive and error-prone because of the inherent complexity of a process.

### **Achieving Technical Independency**

The goal of process change is to generate updated versions of executable workflow definitions. Such a task could be technically demanding to non-technical business

professionals due to the technical details required in executable workflows. At the Gartner BPM Summit 2006, Michael Melenovsky expressed his view about the purpose of workflow as (Swenson, 2007):

*“The ultimate goal of workflow is to place in the hands of business professionals the ability to modify their processes, with no involvement from the IT organization.”*

In achieving his goal, sufficient workflow change support should be provided to non-technical staff following the decision-making stage.

#### **4.2.2.5 Unified Collaboration Enactment Infrastructure**

In addition to the above-mentioned desirables at build time, a unified collaboration enactment infrastructure is needed to effect the enactment of workflow collaboration in a distributed manner. The infrastructure should be designed in line with the build-time principles and requirements highlighted earlier. Also, due to the popularity of workflow management systems (WfMS), it is preferred that the infrastructure should not impose any substantial extension to current mainstream WfMSs and be able to fit in with them seamlessly.

### **4.2.3 Integrated Approach**

Due to a wide range of requirements in relation to cross-organisational workflow collaboration, an integrated approach is needed to address each individual aspect within the context of others. For this regard, at least three dimensions need to be covered, namely workflow, organisation and operating environment.

#### **4.2.3.1 Workflow Dimension**

As the underpinning technology in this thesis, workflow is identified as one of the dimensions. Within the workflow dimension, two aspects can be further identified: the two-stage lifecycle and the two-layer flow model.

##### **Two-Stage Lifecycle**

The two stages refer to build time and run time corresponding to the ones found in the lifecycle of workflow-based solutions. Issues involved in the two stages are distinguished

and addressed separately throughout the thesis. With such a stage division, challenges and requirements can be thoroughly identified whilst design, implementation and evaluation of the proposed approaches and techniques can be logically organised. Furthermore, it is easier to communicate the result of this research.

### **Two-Layer Flow Model**

Control flow and data flow are different but related concepts in workflow technology. They can be found in each workflow definition. Data flow is attached on top of control flow and thus imposes a more stringent dependency. When a process changes, violations of both layers need to be identified.

#### **4.2.3.2 Organisation Dimension**

As the owners of the participating business process collaboration, organisations are concerned with three aspects of the organisation dimension. These include knowledge preservation, lean principle, and human-centric IT support for business users.

### **Knowledge Preservation**

From an organisational perspective, knowledge preservation is crucial because knowledge is an important type of corporate asset. As organisational knowledge takes the form of business processes, i.e. workflow definitions, knowledge preservation equals process preservation. Preservation enables reuse of knowledge during process negotiation.

### **Lean Principle**

Lean thinking is gaining more and more importance over the years. “Add nothing but value” is one of the principles (Poppendieck, 2002). Any waste in terms of unnecessary operations during process negotiation should be eliminated. Where possible, this principle is applied in the approaches and underpinning the enabling techniques proposed in this thesis.

### **Human-Centric IT Support**

As one of the key objectives, in order to fully inform business users during their decision-making, human-centric IT supports are needed. Before users make decisions, identifying and providing decision-relevant information in a logical manner is crucial to the effectiveness and quality of the decisions that users are going to make. Where possible, decisions are made automatically to reduce user workload. After decisions are committed,

the workflow should be modified according to the decisions with the resulting workflow's consistency and correctness intact.

#### **4.2.3.3 Operating Environment Dimension**

The operating environment dimension has two aspects, distribution and dynamism.

##### **Distribution**

As cross-organisational business collaboration operates in a distributed environment, distribution should be taken into account and reflected in areas such as privacy protection, decentralised progress coordination, and acknowledgement of autonomy.

##### **Dynamism**

A changing and dynamic environment is always one of the important features of the real world. By acknowledging this aspect, this research finds its feet on a solid and realistic ground. Every proposed solution needs to take dynamism into account and be evaluated against it. Generally speaking, in order to address the dynamism found in business collaboration across different organisations, the loosely coupled principle needs to be observed. Also, flexibility would help adaptation stemmed from dynamic changes.

### **4.3 Distribution vs Centralisation**

In general, two coordination mechanisms exist: centralised and distributed. A distribution-based approach is more suitable than its centralisation counterpart for cross-organisational workflow collaboration. The effectiveness of the distributed coordination mechanism will have direct impact on the overall performance of the proposed approach. In this section, the reasons are explained, the distributed mechanism is designed and its effectiveness is examined.

#### **4.3.1 Reasons for Distributed Coordination**

The centralised coordination mechanism is widely adopted in modelling workflow collaboration (Aalst, 2001; Schulz, 2004; Biegus and Branki, 2004) due to its simplicity and effectiveness in specifying the desired interaction pattern. However, it has several drawbacks. Firstly, by following such a mechanism, the distributed and autonomous reality of business partners as well as their private workflows is ignored, which makes it only

suitable for a small part of the identified scenarios of business collaboration. Secondly, centralised coordination always imposes a tightly coupled structure, which contradicts the golden rule of loose coupling in modelling and implementing inter-organisational collaboration. Thirdly, the design process of centralised coordination deprives the opportunity of IT-supported negotiation because it is assumed that a coordination blueprint has already been reached between partners. Lastly, due to the mixed picture in workflow standardisation, there is little common ground to facilitate the implementation of centralised coordination designs apart from adopting a centralised workflow engine. Therefore, distributed coordination is preferred given the research context of the thesis. The challenge is how to convert centralised coordination into effective distributed coordination.

### **4.3.2 From Centralisation to Distribution**

In order to achieve equal effect found in centralised coordination through a distributed manner, the underlying control flow model of centralised coordination needs to be investigated.

#### **4.3.2.1 Explicit Control Flow Specification**

In centralised workflow coordination, collaboration is explicitly modelled as control flow. It can be either a transition that connects two activities or a sub-process call that has detailed knowledge of the target process. This inevitably brings about tight coupling because detailed information regarding private workflows needs to be revealed and shared. For example, the collaboration initiator must know the identification of the target activity or sub-process. Therefore, whenever the target side changes its processes, relevant details needs to be propagated to its associated trading partners in order to update key coordination information. To bypass the explicit knowledge constraint, the key question is whether there is an alternative approach to collaboration modelling that can implicitly represent the explicit control flow and thus remove tight coupling. The solution is through implicit control flow modelling driven by collaboration data flow.

#### **4.3.2.2 Implicit Control Flow Modelling**

Data flow between workflows is in fact the whole sequence in which collaboration messages are exchanged between partners. In a private workflow or cross-organisational workflow following centralised coordination paradigm, data flow is attached upon control

flow. With correct coordination of control flow, data flow conflicts will not occur. However, the question is whether the correctness of data flow can still be assured without the coordination of explicit control flow. The answer is yes as it is found that as far as workflow collaboration is concerned, the centralised coordination mechanism can be converted into distributed coordination with the relevant data flow dependency if the pair of participating private workflows conform to the following:

- whenever one workflow wants to supply information to its partner, it has the message containing the information sent and moves on;
- whenever one workflow requires a particular piece of information from its partner, it waits until such a message containing the information comes.

#### 4.3.2.3 Modelling Distributed Workflow Coordination

Since interaction is a necessity in all patterns of workflow collaboration, interaction points need to be modelled to cope with the absence of centralised mechanism. For this purpose, the concept of *interface activity* is introduced. Interface activities are generic workflow activities and have two types, namely *data supply activity (DSA)* and *data demand activity (DDA)*. Each of the interface activity is associated with the data it is dealing with. When triggered,

- a *DSA* has the associated data sent and then allows the hosting process to move on to the next activity;
- a *DDA* blocks itself until the desired data comes from its partner.

The purposes of interface activities are similar to those of *synchronisation activities* (including *send-type* and *wait-type* activities) adopted by Biegus and Branki (2004). However the fundamental difference lies in the operation style between a *data supply activity* and *send-type* activity as the *send-type* activity blocks itself after having the data sent.

As a result, tightly-coupled control flow can be replaced by loosely-coupled data dependency, which provides the foundation for the desired distributed approach. However, the effectiveness of the distributed coordination mechanism still needs to be examined.

### 4.3.3 Effectiveness of Distributed Workflow Coordination

In this subsection, two communication styles (asynchronous and synchronous) and three collaboration patterns (hierarchical, composite and peer-to-peer) are considered to demonstrate how workflow collaboration could still be modelled and equally carried out by following the proposed data dependency-based distributed coordination with a message relay facility.

#### 4.3.3.1 Asynchronous Communication

Asynchronous communication refers to the exchange of messages between private workflows (Aalst, 1999). Based on definitions of interface activities, asynchronous communication can be modelled as a message sent from a *data supply activity* to a *data demand activity* as shown in Figure 4-1.

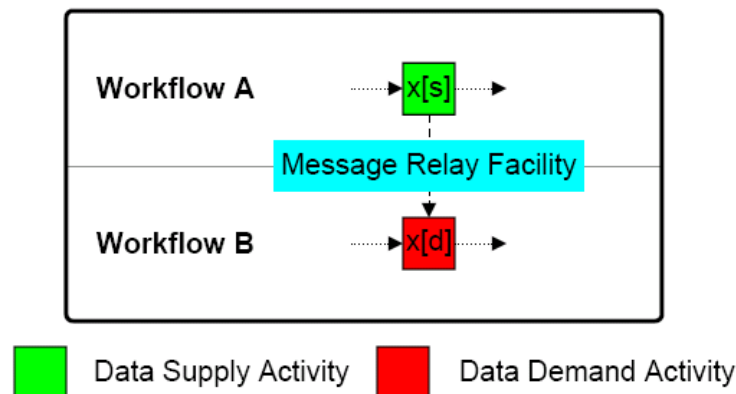


Figure 4-1. Asynchronous communication coordinated in a data dependency manner.

#### 4.3.3.2 Synchronous Communication

Synchronous communication mandates private workflows to execute specific activities at the same time (Aalst, 1999). A synchronisation point is needed in this type of communication. It is modelled as two consecutive asynchronous communications in opposite positions as shown in Figure 4-2.

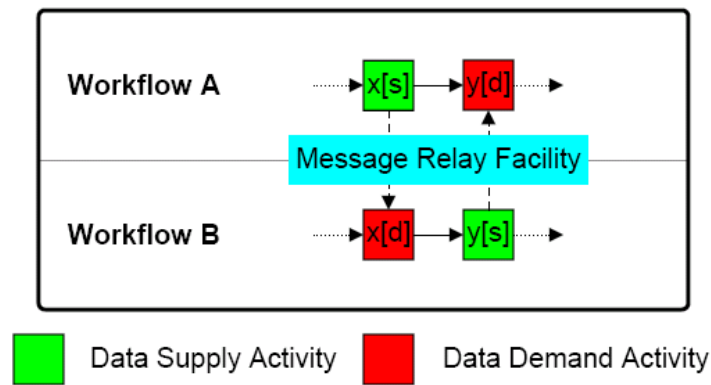


Figure 4-2. Synchronous communication coordinated in a data dependency manner.

For the purpose of simplicity, the following workflow collaboration patterns are discussed on the assumption that they follow asynchronous communication style.

#### 4.3.3.3 Hierarchical Workflow

##### Chained Sub-Flow

Chained sub-flow collaboration is represented as a *data supply activity* in the superior workflow and a *data demand activity* waiting for the same data as the first activity in the sub-flow. It is shown in Figure 4-3.

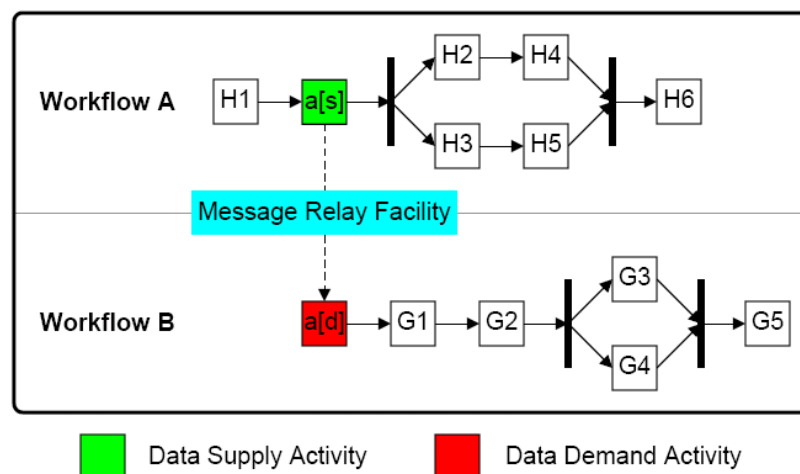


Figure 4-3. Chained sub-flow represented in a data dependency manner.



### Nested Sub-Flow

For nested sub-flow collaboration, since a superior workflow needs to wait for the result returned by the sub-flow, synchronisation on the superior workflow is represented as a *data supply activity* immediately followed by a *data demand activity*. On the sub-workflow side, a *data demand activity* waiting for the invocation message is the first activity whilst a *data supply activity* returning the result back to the superior workflow is located at the end of the sub-flow. It is shown in Figure 4-4.

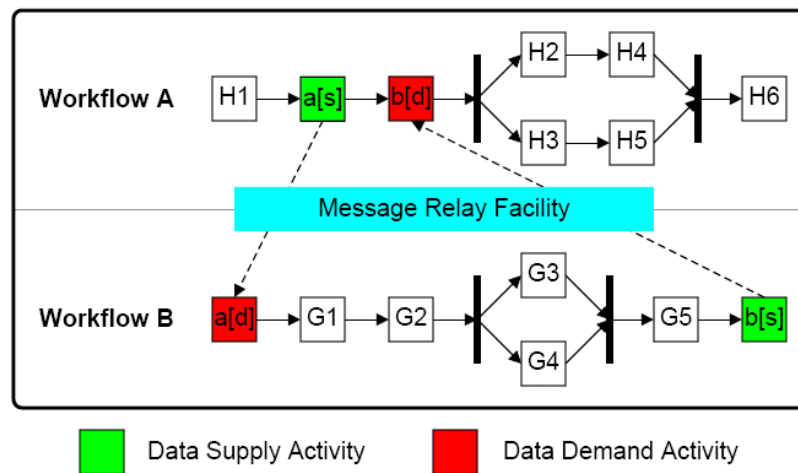


Figure 4-4. Nested sub-flow represented in a data dependency manner.

#### 4.3.3.4 Peer-to-Peer Workflows

Similar to chained and nested sub-flows, in peer-to-peer workflows, interaction points are modelled in the form of pairs of interface activities that are associated with the same type of data.

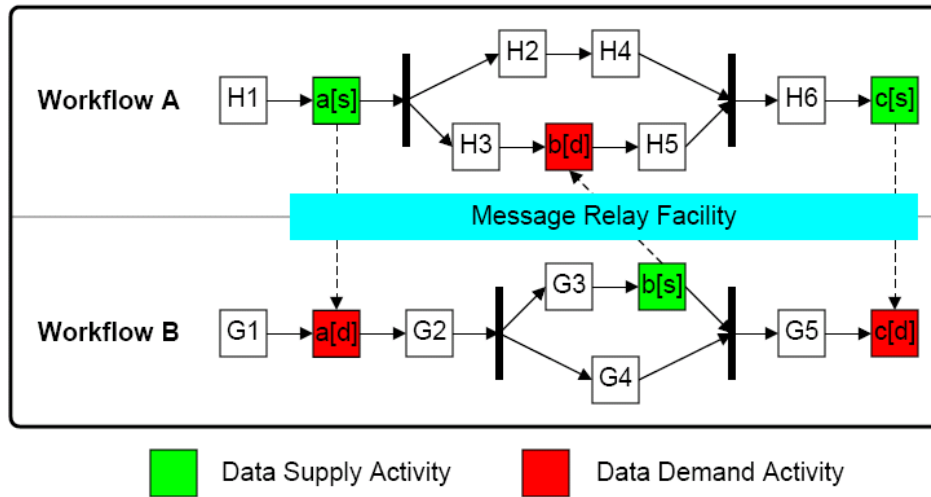


Figure 4-5. Peer-to-peer workflows represented in a data dependency manner.

As shown in Figure 4-5, between peer-to-peer workflows, there are normally several rounds of interaction rather than one. As a result, the sequence of the interface activities is of great importance as improper sequence could bring deadlock to collaboration. As a brief introduction, two distributed workflows can successfully interact with each other without explicit control flow modelling if all the following four conditions are met:

- each interface activity follows a self-disciplinary operation manner according to its role (DSA or DDA),
- interface activities regarding each message are in pairs,
- messages can be delivered through a message relay facility,
- the sequences of interface activities that appear in both processes will not cause message deadlock.

More discussion on sequences of interface activities can be found in Chapter 5.

#### 4.3.3.5 Composite Workflow

A composite workflow collaboration pattern can be converted into a peer-to-peer pattern after all the required interface activities are inserted. An example is shown in Figure 4-6. In the initial composite workflow, interaction between Partner A and Partner B happens four times at  $C1 \rightarrow C2$ ,  $C2 \rightarrow C3$ ,  $C3 \rightarrow C4$ , and  $C5 \rightarrow C6$ , so four pairs of interface activities are inserted into the two processes. Once inserted, the composite workflow is transformed into a

pair of peer-to-peer workflows, which can be enacted separately with the interface activities taking care of the coordination through a message relay facility.

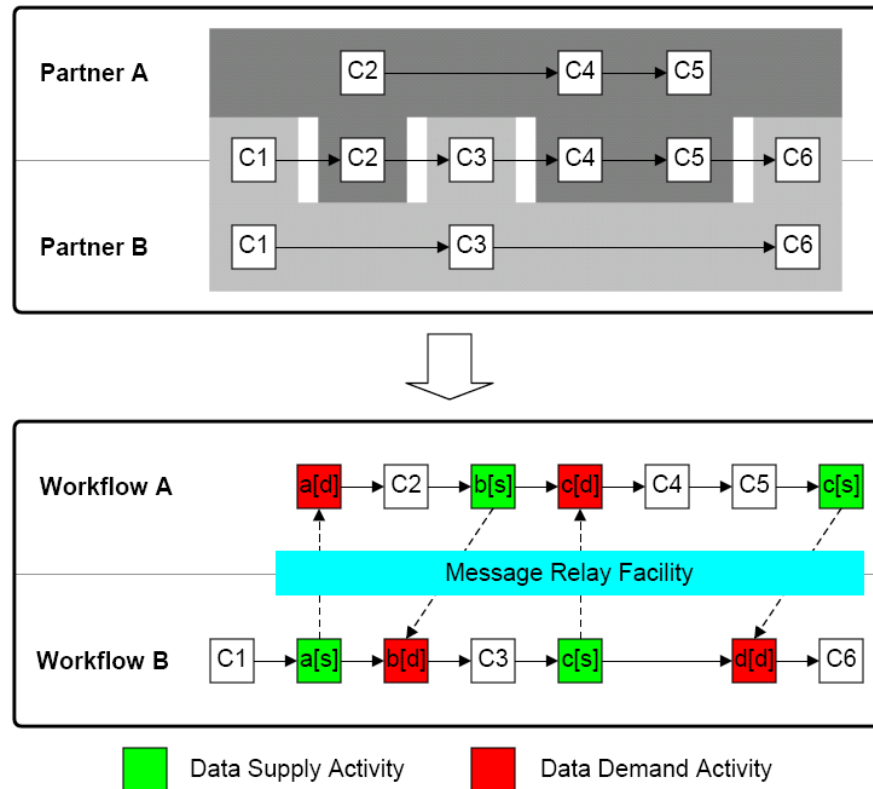


Figure 4-6. Converting composite workflow into a peer-to-peer pattern.

It is safe to say that data dependency-based distributed coordination supported by a message relay facility can achieve the same coordination result as its centralised counterpart.

## 4.4 Negotiation in a Nutshell

Mentioned in two of the objectives of this thesis, business collaboration negotiation needs to be captured by workflow technology and IT support should be provided as much as possible during business process negotiation and reconciliation. In order to gain essential understanding of negotiation, this section reviews the key concepts.

### 4.4.1 Definitions of Negotiation

The word *negotiation* is derived from the Latin word *negociare* – to conduct business. Negotiation was first understood in the context of business transactions although it is used in

a wider context for quite sometime. As an alternative dispute resolution, negotiation has been studied and defined in many ways.

Negotiations are formal discussions between people who have different aims or intentions, especially in business or politics, during which they try to reach an agreement (Collins Cobuild English Dictionary, 1995).

Negotiations are official discussions between the representatives of opposing groups who are trying to reach an agreement, especially in business or politics (Longman Dictionary of Contemporary English, 2002).

Negotiation is a joint decision process between two or several parties or their representatives. Negotiation tends to be a matter of finding a formula encompassing the optimum combination of interests of both parties and then of working out the details that implement these principles and affect the agreement. Negotiation is a dynamic and on-going process, involving moves and countermoves (Zartman, 1977).

Negotiation is the interaction that occurs when two or more persons attempt to agree on a mutually acceptable outcome in a situation where their orders of preference for possible outcomes are negatively correlated (Hammer and Yukl, 1977).

Negotiation is one kind of problem-solving process, in which people attempt to reach a joint decision on matters of common concern in situations where they are in disagreement and conflict (Gulliver, 1979).

Negotiation is a form of decision-making process where two or more parties jointly search a space of possible solutions with the goal of reaching a consensus (deal) (Rosenschein and Zlotkin, 1994).

The purpose of listing the above definitions is to extract a common understanding of negotiation but not to provide an exhaustive list of negotiation definitions. It is widely agreed that negotiation is a process that normally includes the following stages:

- Two partners each with its own interests come together with the intention of reaching some agreement,
- Through communication, if there are any conflicts between their individual interests, they are identified,
- Partners seek possible options to reconcile the differences through a range of strategies and approaches, such as concession making, contending, problem solving, inaction, withdrawal (Pruitt and Carnevale, 1993), or exploration of mutual gains (Follett, 1942),
- If successful, an agreement is reached be it only in favour of one partner's interests or a win-win result.

The outcomes of negotiation heavily depends on a number of factors related to complex human interactions, such as power of partners, negotiators' personal capabilities, negotiation strategies, time constraints etc.

#### **4.4.2 Principles of Negotiation**

Throughout a negotiation process, two core principles should be observed (Hiltrop and Udall, 1995):

- Negotiation is a voluntary activity, either party can break away from a discussion at any time.
- A successful outcome of a negotiation is to get what both sides want rather than to win at any cost.

These two principles will act as guidelines for later discussion on mutual gains discovery, concession giving and decision-making activities during a negotiation process.

#### **4.4.3 Approaches of Negotiation**

Two approaches that are often found in negotiation practice are concession-convergence and mutual gains.

#### **4.4.3.1 Concession-Convergence Approach**

The concession-convergence approach earns its name by following such a paradigm: two parties start with standing on opposite sides and approach each other by giving something up with the aim of making a deal (Rubin, 1994). For example, a seller begins by asking for more than it expects and the buyer begins by offering less than it is willing to pay. Through a series of concessions, the two sides converge to a point that each finds acceptable. Also known as ‘distributive bargaining’ (Walton and McKersie, 1965), early negotiation research was almost exclusively centred around the concession-convergence approach. However, this approach was considered primitive, competitive or even mindless by some scholars and began to fall into disfavour in the wake of the mutual gains approach (Rubin, 1994).

#### **4.4.3.2 Mutual Gains Approach**

Deemed as the founder of the discipline of organisational behaviour, Follett (1942) advocates the mutual gains approach to negotiation and was considered one of the earliest advocates of the approach (Rubin, 1994). Different from concession-convergence that assumes concealment, inflated initial demands and zero-sum, the mutual gains approach tries to redefine negotiation as a shared problem to be resolved. Knowledge and resources are pooled and maximum mutual gains are sought after in order to yield greater payoffs to all parties. Walton and McKersie (1965) have used the term “integrative bargaining” and Lax and Sebenius (1986) have coined the term “creating value” to capture the same idea.

#### **4.4.3.3 Which to choose?**

As to the choice between the two approaches, neither is necessarily better than the other. This point has been conveyed clearly by Walton and McKersie (1965), Tracy and Peterson (1985), and Lax and Sebenius (1986). The effectiveness of each approach depends on specific application domain. These researchers also encourage an integrative rather than antagonistic relationship between the two approaches. This thesis adopts a combined negotiation approach and more details will be given in §5.2.1.

#### **4.4.4 Operational Aspect of Collaboration Negotiation**

As a very complicated human centric behaviour, negotiation covers strategic, behavioural and operational aspects. With the concern of this thesis in the operational aspect only, three elements are identified based on Fisher (2003)’s Seven Elements Framework. The terms are

kept the same, namely interests, communication and options although the meanings have been extended and adapted to fit in the context of this thesis.

- Interests – individual needs, concerns, goals, hopes and fears that motivate both partners. Commonality and/or conflict can be observed when two sets of individual interests are compared.
- Communication – the transfer of messages by speech, writing or other means for the purpose of comparing individuals' interests, which also includes negotiation protocol, i.e. the manner of message exchange.
- Options – ideas about how the parties might meet their interests together, which includes all the necessary alterations to individuals' initial interests as well as any associated concession in order to reconcile any encountered conflicts.

## 4.5 Process Comparison Techniques

Process comparison (or process match-making) is an independent research area. Techniques developed are key enabling factors to service discovery and automated reconciliation approaches mentioned in Chapter 3. It is also used in the field of workflow change management (Yeoh et al, 2004). This section introduces comparison in general terms then goes on to give more details about process comparison at structural and behavioural levels.

### 4.5.1 Comparison in General

Comparison is a common action we practise on a daily basis. Through comparison, some objects are grouped together and some others are differentiated. By definition, compare means “to estimate, measure, or note the similarity or dissimilarity<sup>1</sup> between” (The Oxford Compact English Dictionary, 2000). In order to measure similarity (or reveal dissimilarity), a set of matching characters needs to be identified. This is illustrated by two simple examples. As shown in Figure 4-7 (a), comparisons between the two numbers 1979 and 2008 are carried out at the levels of thousand ( $10^3$ ), hundred ( $10^2$ ), ten ( $10^1$ ) and one ( $10^0$ ) with differences found as 1000 vs. 2000, 900 vs. 0, 70 vs. 0, and 9 vs. 8. The  $10^n$  are the set of matching characters used in this case. As in Figure 4-7 (b), in order to compare the two designs of coat of arms to see how similar or dissimilar they are, features such as the designs of crest, helm, supporter, shield and motto need to be compared.

---

<sup>1</sup> As far as (dis)similarity is concerned in this thesis, it refers to exact (dis)similarity rather than approximation.

In this case, all these constituent components form the set of matching characters, on which the (dis)similarity can be claimed.

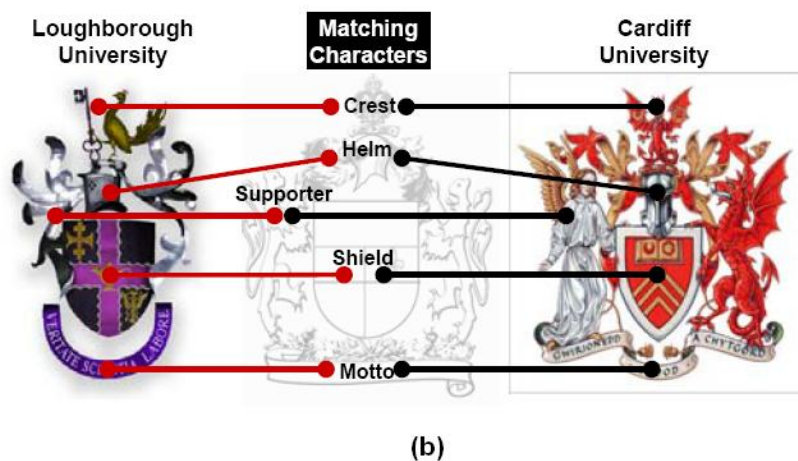
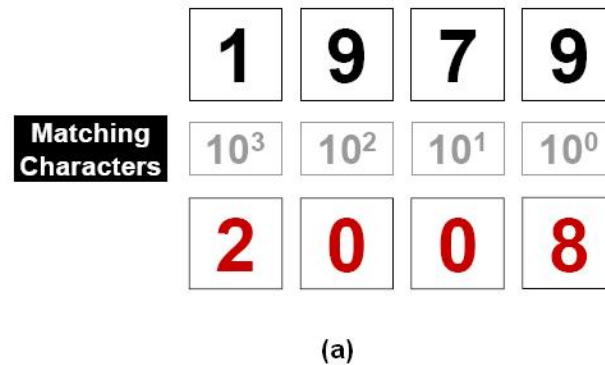


Figure 4-7. How to tell the difference?

## 4.5.2 Process Comparison

As far as processes are concerned, generally speaking, comparisons can be conducted on two dimensions, structure and behaviour.

### Structure Comparison

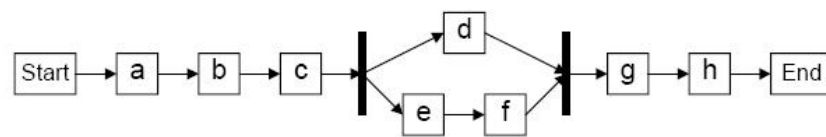
For a structure comparison, the set of matching characters comprises the building blocks of an activity diagram, namely vertices and edges. In practice, structure comparison is carried out by comparing the two adjacent matrices of the two activity diagrams. Each difference, in terms of vertices or edges, will be picked up and all the different structures will be captured in one go. The purpose of structure comparison is to reveal a full picture of (dis)similarity between the two processes' activity diagrams. Yeoh et al (2004) propose and implement a



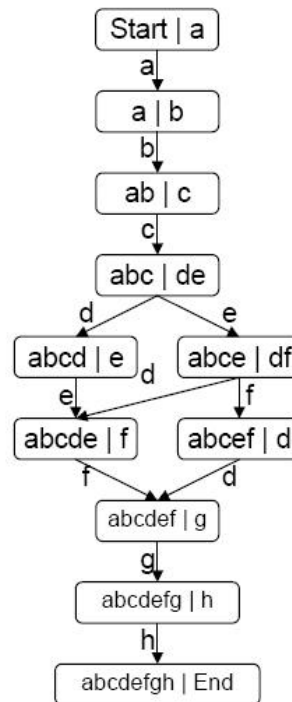
comparison technique based on structure comparison for a research project in the area of workflow change management.

### **Behaviour Comparison**

Behaviour of a process refers to all the possible activity execution sequences that the process can experience. In order to get hold of the behaviour of a process, from start to end, each different possible step for each activity in each process needs to be captured and recorded. Process behaviour can therefore be expressed as a series of states at certain step(s) away from the beginning of the process. Each state comprises activities that both have been completed and are immediately reachable based on the state's unique activity completion status. States having the same step(s) away from the beginning of their processes are chosen as the matching characters. As a result, a comparison between two processes' behaviour is able to reveal the (dis)similarity between two comparable states, or more specifically between the completed and the reachable activities belonging to the states. However, since processes are not normally defined in such a state transition manner, a conversion from an activity diagram into a state transition system is needed. Krukkert (2003) proposes such a conversion method. In his project – openXchange, with the assumption that each activity is atomic, true parallelism can be considered as branching time parallelism, which makes it possible to convert parallel structures into sequential ones and thus eliminate parallelism found in an activity diagram. This makes it possible to convert an activity diagram into a standard state transition system (STS) containing no parallelism. In an STS, each possible state of the system is represented by exactly one node, which, as a result, makes the states from both sides comparable.



(a) Activity Diagram



(b) State Transition System Diagram

Figure 4-8. Conversion from an activity diagram to a state transition system diagram.

Figure 4-8 (b) illustrated the resulting STS diagram that is converted from the corresponding activity diagram as in Figure 4-8 (a). Each round rectangular represents a state at a certain number of step(s) away from the beginning of each process. The vertical bar in each state acts as a delimiter sitting between the activities having been completed (to the left) and those being reachable (to the right). A guarded edge explicitly indicates through which activity in transit one states reaches another. The *Start* point is always by default deemed as the completed activity in the first state. In this example, since the only activity reachable so far is *a*, *a* appears to the right hand side of the bar. After *a*'s execution and completion (indicated by the edge guarded by *a*), the next state is reached. Activity *a* becomes a completed activity and thus appears to the left hand side of the bar whilst *b* becomes the next reachable activity and appears to the right of the bar, etc. For the purpose of clarity,

*Start* is omitted in all the state boxes after the first one. When the state marked as '*abc / de*' is reached, since both *d* and *e* are reachable, there are two outgoing edges from this state following activity *d* and *e* respectively. Note, such a branch is of type exclusive OR. By repeating the same procedure till the end of the process, an activity diagram can be converted into a state transition system diagram, which captures all the possible activity sequences of a given process. As a result, comparison between two such STS diagrams is able to reveal the (dis)similarity between the behaviour of the two processes.

## 4.6 Summary and Conclusion

In this chapter, an integrated approach is formed across three dimensions, namely workflow, organisation, and operating environment. The new approach is targeting short-term workflow-based business collaboration, casual trade in particular, between equal partners. Due to the distributed and autonomous operating environment, this approach needs to be based on the loosely coupled paradigm. It is required to guide users through the collaboration negotiation, support decision-making and effect changes to workflow definitions. A unified collaboration enactment infrastructure is needed to enable the collaboration message exchange at both build time and run time.

In relation to the wide spectrum of issues covered by the approach, several key topics are addressed. Firstly, a new distributed workflow coordination mechanism based on loosely coupled data dependency is designed, evaluated and found capable of dealing various communication styles and workflow collaboration patterns. Secondly, key issues in negotiation activity are reviewed and three elements in the operational aspect are identified. Lastly, process matching techniques are compared and the behaviour-oriented approach is discovered as in line with the need of this thesis.

# Chapter 5

## A Framework for Cross-Organisational Workflow Collaboration

### 5.1 Introduction

In order to fulfil the requirements for the integrated approach mentioned in Chapter 4, a novel cross-organisational workflow collaboration (COWCO) supporting framework (hereinafter referred as the Framework) is constructed to capture and manage all the constituent elements.

In this chapter, §5.2 introduces the Framework from a business view, within which operational aspect of collaboration negotiation is captured from a workflow perspective. §5.3 unveils the Framework from an architectural angle, which includes descriptions of the logic components, the design of a generic architecture in the form of a service stack, and its adaptation to possible implementation architectures. §5.4 summarises the chapter.

The Framework involves two partners with equal importance. In the remainder of the thesis, they are referred to as Partner A and B when viewed from a neutral point. When the emphasis is put on one side, it is referred to as the Host whilst its partner as the Guest.

### 5.2 Business View

From a business view, a typical business collaboration comprises deal negotiation, agreement reaching and fulfilment although it is not guaranteed that an agreement could be reached each time due to irreconcilable divarication between the two partners. The business

collaboration steps are shown in Figure 5-1. In this section, the way to capture each of the steps by workflow technology is explained.

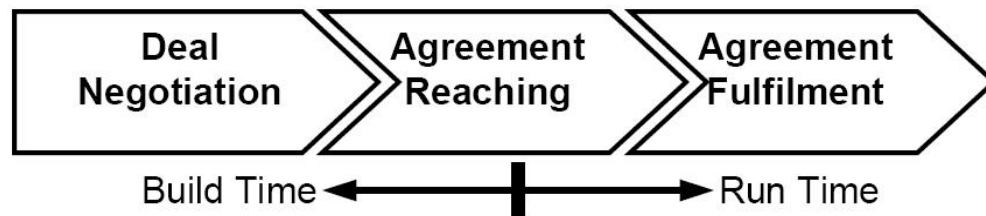


Figure 5-1. Business view of workflow collaboration.

## **5.2.1 Capturing Operational Aspect of Collaboration Negotiation with Workflow Technology**

In order to provide IT support to workflow-based negotiation, negotiation should be understood and captured within the context of workflow technology. As the focus of this thesis is on operational aspect of collaboration negotiation, IT is aimed to provide support in this aspect only. The three elements identified in §4.4.4 – namely interests, communication and options – will be discussed as follows.

### **5.2.1.1 Capturing Interests**

Interests are the issues that one is concerned about. In the negotiation context, they are categorised into three types: individual, common and conflicting interests.

#### **Individual Interests**

When workflow technology is used to automate business processes, relevant business rules and preference have been embedded into workflow process definitions. Each individual's interests are expressed in the form of the functions of workflow activities and their control sequences and data dependencies in a process definition. In the collaboration context, for each partner, such interests are expressed as desired collaborative messages to be sent and received following particular orders by individual workflows, which can be collectively represented as collaborative message flows and equally reflected as the corresponding interface activities and their control flow dependencies (or interface process) within a partner's workflow as illustrated in Figure 5-2.

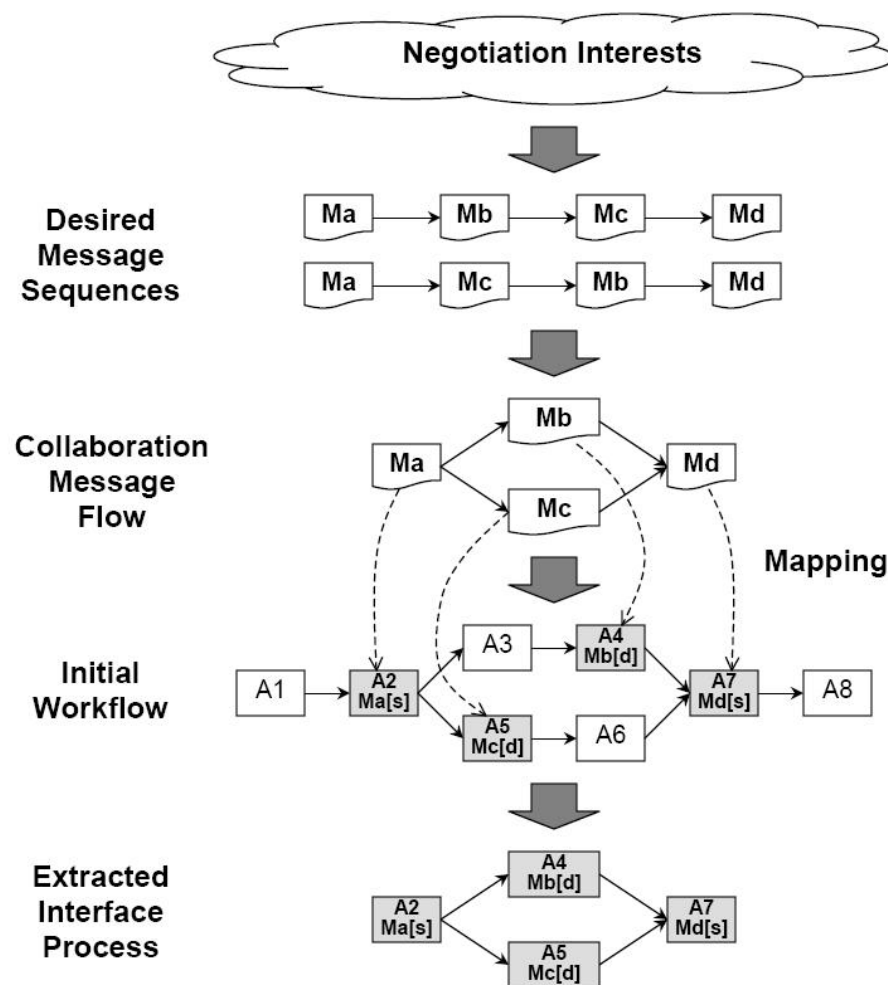


Figure 5-2. Capture negotiation interests from a workflow perspective.

### Common Interests

Between two workflows, common interests lie with the compatibility of the collaborative data and the control flow of the activities that handle collaborative data. This defines a set of process matching criteria. However, whether a particular set of criteria can truly reflect the commonality between two workflows is a challenging question. For this reason, the mutual gains approach is adopted in searching for such matching criteria.

Apparently, if the control flows of two processes are exactly the same and the interaction points are properly modelled and matched on both sides, the two processes can certainly collaborate with each other given a proper message relay facility. It can be formally defined as:

**Definition 5-1 Absolute Compatibility:** Two collaborative workflows are of absolute compatibility if:

- interaction points are modelled as *interface activities*,
- their interface activities regarding each message are in pairs,
- messages can be delivered through a message relay facility,
- the sequence that interface activities appear in both workflows are exactly the same.

However, the cost of achieving compatibility at this level will be very high under the assumption that the two interface processes must be transformed into exactly the same in order for the collaboration to proceed. A set of less strict and effective matching criteria is desired.

In order to achieve successful collaboration, at least, there should be no deadlock when partners' workflows are enacted at run time. As mentioned briefly in Chapter 4, enact-able compatibility can be formally defined as:

**Definition 5-2 Enact-able Compatibility:** Two collaborative workflows are of enact-able compatibility if:

- the first three conditions of Absolute Compatibility are met and,
- the sequence that interface activities appear in both workflows will not cause message deadlock.

Since the enact-able compatibility exists as the minimum requirement for successful collaboration between two workflows, its effectiveness needs to be examined to ensure that it is not too relaxed. Let us revisit the enact-able compatibility through the example shown in Figure 5-3. Partners  $P_A$  and  $P_B$  are the owners of interface processes  $A$  and  $B$  respectively shown in Figure 5-3. Although  $A$  and  $B$  comply with enact-able compatibility, a satisfactory collaboration cannot be safely guaranteed because activity  $A.g[d]$  has to wait for message  $g$  until after activity  $B.g[s]$  is completed. The wait for  $P_A$  could last for days or weeks and unnecessarily delay the completion of  $A.g[d]$  and thus the whole process  $A$ , which, as a result, may not be acceptable for  $P_A$  from a business perspective despite the fact that no execution deadlock will occur.

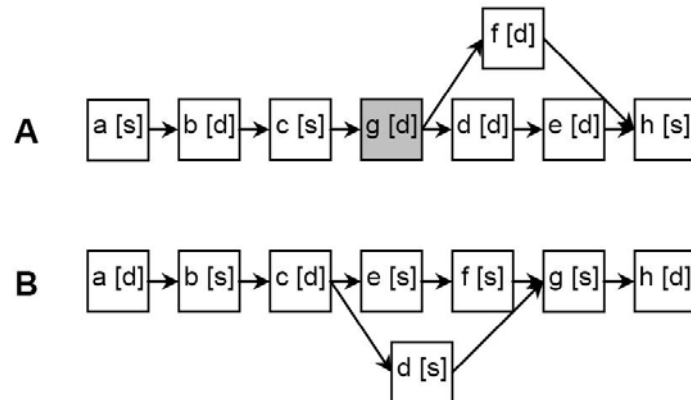


Figure 5-3. Example of unnecessary delay.

Therefore, apart from the execution deadlock situation as illustrated in Figure 5-4 (a), unnecessary delay is identified as another issue that needs to be excluded by an effective set of matching criteria. By unnecessary delay, it means a situation where a data demand activity is reached in one partner's process, before executing the desired data supply activity, the process of the other partner still need to execute other data supply activity(ies). Figure 5-4 (b) shows a scenario of an unnecessary delay, where at a certain stage of a collaboration, A reaches  $A.x[d]$  and starts waiting whilst  $B.x[s]$  cannot be executed until after  $B.y[s]$  is cleared. The wait for the clearance of  $B.y[s]$  is the unnecessary delay for A.

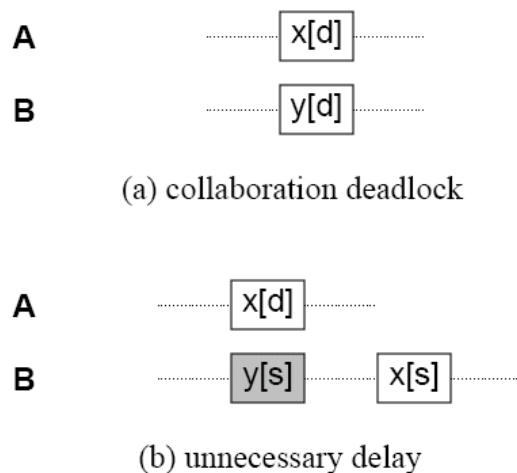


Figure 5-4. Collaboration deadlock and unnecessary delay.



As a result, before any two business partners can proceed in conducting B2B e-business transactions, their workflows that will be involved in the transactions should be compatible with each other at the business level (Yang and Papazoglou, 2000), i.e. they have an agreed sequence(s) of collaborative messages. Wombacher (2005) approaches bilateral process compatibility through modelling of message sequence and concludes that bilateral collaboration consistency requires trading partners share at least a common message sequence between their message sequence models. However, Wombacher does not state why such a requirement is needed. With the identification of the requirements of avoiding deadlock and unnecessary delay, the researchers (Krukkert, 2003; Wombacher 2005)'s intuition now has a firm stand. Since the message sequence can be equally represented by the sequence of each partner's interface activities extracted from an initial workflow, the business collaborative compatibility can therefore be defined as:

**Definition 5-3 Business Collaborative Compatibility:** Two collaborative workflows are of business collaborative compatibility if:

- the first 3 conditions of Absolute Compatibility are met and,
- the two workflows have at least one common sequence of corresponding interface activities.

The justification of the second condition is that corresponding interface activities on a common trace can always be reached by both interface processes in a timely manner, which will leave no collaborative message unattended and thus can guarantee a successful collaboration without any deadlock or unnecessary delay.

Acting as process matching criteria, Definition 5-3 implements the mutual gains approach and highlights the behavioural aspect of processes rather than the structural one between the digraph representations of control flow. However, when two processes do not satisfy Business Collaborative Compatibility, conflicting interest emerges.

### **Conflicting Interests**

Conflicts occurred in negotiations are caused by differences between two partners' interests. In workflow collaboration context, they refer to the differences between two interface processes composed of interface activities and their control flow sequences.

As far as two processes are concerned, differences can be found at two levels: structural and behavioural. In this thesis they are named as structural dissimilarities and behavioural discrepancies respectively. Structural dissimilarities refer to the differences between the two digraph representations of corresponding interface processes, i.e. any differences from vertices to edges. Behavioural discrepancies capture process difference by following the control flow and traversing every possible flow trace. According to the common interests matching criteria stated in Definition 5-3, if there is any differences encountered, it will be a behavioural discrepancy rather than a structural dissimilarity. That is to say Business Collaborative Compatibility tolerates structural dissimilarities to a certain extent as long as they do not cause behavioural discrepancies, which effectively reflects the desired results that the mutual gains approach should bring about.

#### **5.2.1.2 Capturing Communication**

The flexibility of the loosely coupled principle lies with a wider tolerance in autonomous changes of distributed partners. However, it is the inherent uncertainty associated with partners' autonomy that presents the disadvantage of loose coupling. Therefore, before each deal moves on into the execution stage, the deal's two participating workflows need to be properly expressed, exchanged and compared to decide whether they are a matched pair according to the matching criteria described in Definition 5-3. If not, conflicts need to be captured and revealed. Also, in order to coordinate the communication, a negotiation protocol should be selected for both partners to follow.

#### **Interests Representation**

As discussed in §5.2.1.1, since collaborative interests are essentially reflected by the sequences of collaborative messages and collaborative messages are exchanged through interface activities, comparison between the sequences of interface activities can fulfil the task of negotiation interests comparison. However, it is the initial workflow definitions of both partners with interface activities embedded that are available. For each partner, a gap

lies in between the workflow definition and the sequence of interface activities as shown in Figure 5-5. In order to bridge the gap, firstly, interface activities should be extracted from their hosting workflow definition with the initial causal relationship attached. The result is also known as an *interface process*. Secondly, based on the interface process, all the possible sequences of relevant interface activities (i.e. the behaviour of the interface process) need to be worked out and represented in a proper form ready for comparison.

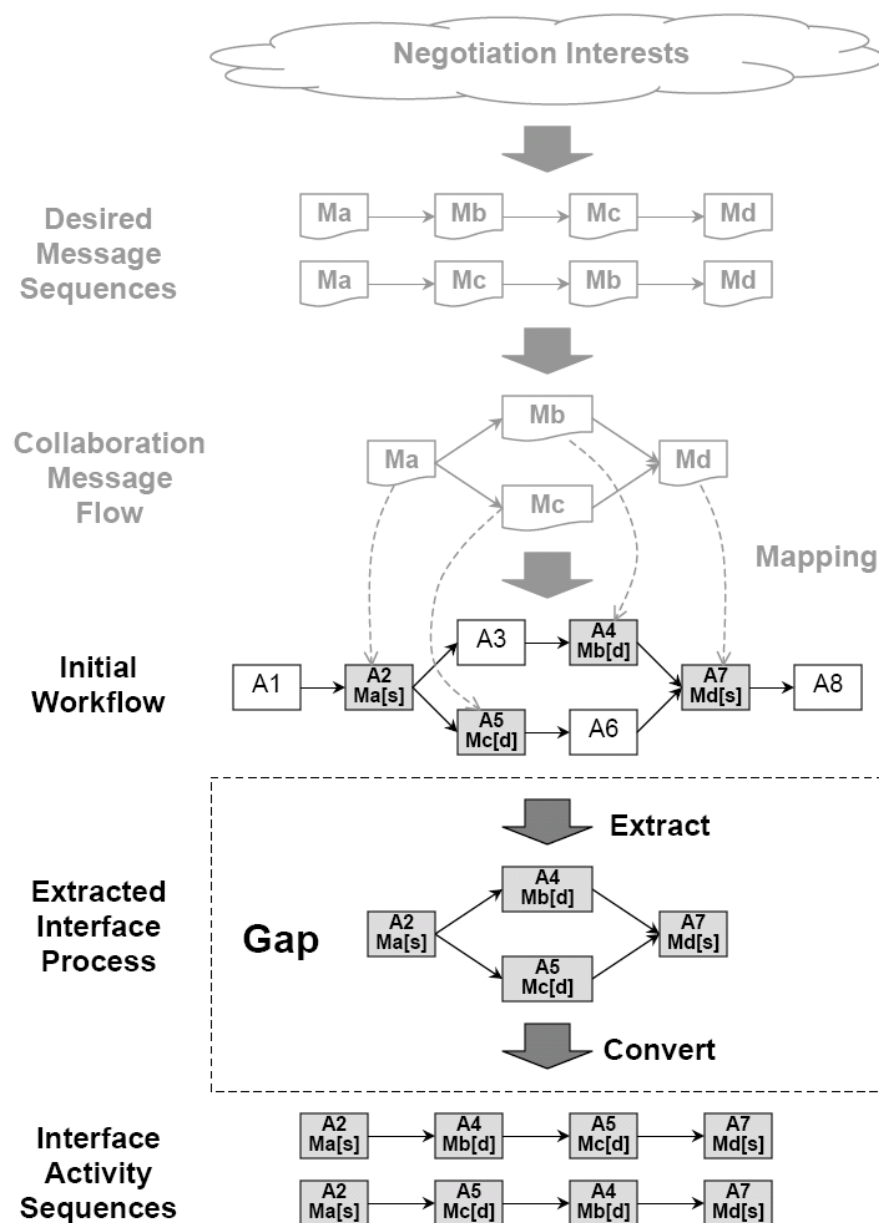


Figure 5-5. Interests representation.

More details of both of the enabling techniques adopted to extract an interface process from a hosting workflow definition and to reveal the behaviour of the extracted interface process will be introduced in Chapter 6.

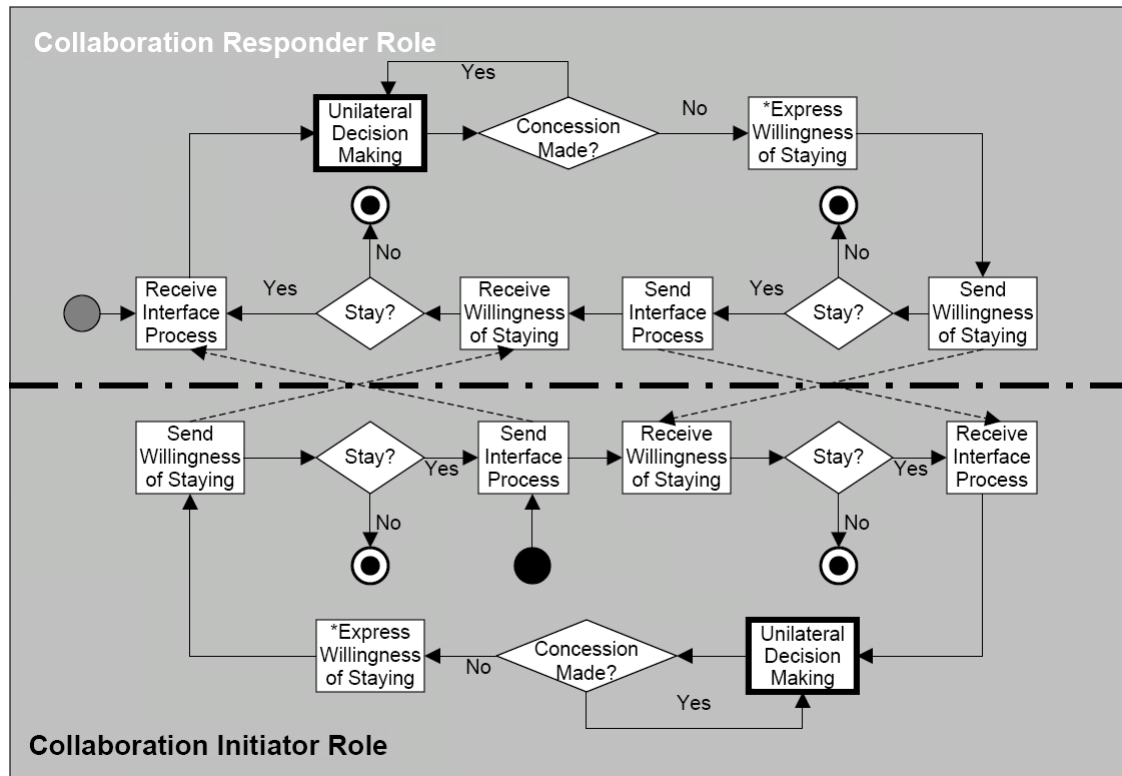
### **Negotiation Protocol**

According to Jennings et al (2001), a negotiation protocol is a set of rules that governs the interaction during negotiation. This covers the permissible types of participants (e.g. the negotiators and any relevant third parties), the negotiation states (e.g. accepting bids, negotiation closed), the events that cause negotiation states to change (e.g. no more bidders, bid accepted) and the valid actions of the participants in particular states (e.g. which messages can be sent by whom, to whom, at what stage).

The negotiation protocol to be applied depends on the underlying negotiation approach adopted and the negotiation power of the two partners. In this thesis, in order to make the result more generic, it is assumed that the two partners have equal negotiation power. Also, for the purposes of maximum resource preservation as well as process reconciliation, a combined negotiation approach is adopted, in which the mutual gains approach is applied during process comparison to absorb structural dissimilarities and discover hidden common behaviour; if any conflict in the form of behavioural discrepancy is encountered, it is coped with by the concession-convergence approach. In time of conflict, due to the decision towards a concession should be made at real time and based on the result of assessment between the risk associated with the concession and the risk as a result of the conflict (Hicks, 1932), the process of successive decision-making should be orchestrated in a manner that at real time only when one partner has no further concession to make with regard to the currently identified conflict, a counteroffer is constructed and passed on to the other partner for consideration. Therefore, the negotiation protocol is named as *real-time sequential protocol*.

An overall negotiation process is defined to capture the essence of, implement and enforce this negotiation protocol, which comprises two collaborative processes corresponding to two interrelated roles: *Collaboration Initiator* and *Collaboration Responder*. Each partner chooses or is assigned to its role to play. The two processes synchronise at certain points, where relevant information is exchanged, e.g. *willingness of staying* and the most updated

version of the interface process. The overall process is illustrated in Figure 5-6 as a pair of collaborating workflows.



\*If the reason for no concession is discrepancy-free, willingness of staying will be set as 'No'.

Figure 5-6. Overall negotiation process.

It can be seen that the processes on both sides are exactly the same except for the starting points. The overall negotiation process starts from *Collaboration Initiator* (CI)'s *Sending Interface Process* (IP) activity. When the *Collaboration Responder* (CR) receives CI's IP, the *Unilateral Decision Making* activity is triggered on CR's side. Whenever there is any concession having been made by CR, the operation stays on CR's side until no further concession has ever been made by CR. Then, CR needs to decide whether to stay in the collaboration negotiation. No matter what decision has been made by CR, the decision regarding the willingness of staying is sent back to CI for CI to choose an appropriate option in line with the decision. If CR decides to stay, it needs to send its most updated interface process to CI. After CI receives CR's IP, the same procedure repeats on CI's side.

Otherwise, if CR decides to leave the negotiation, the negotiation process terminates on both sides.

As the core task in the overall negotiation process, the unilateral decision making activity is itself a sub-process. It is designed to accommodate a series of activities, e.g. interface process extraction, process comparison, discrepancy identification, option prompt, user decision-making support and process change. For the convenience of discussion, the Host represents the partner who is actively engaged in a unilateral decision making sub-process whilst the Guest represents the partner who is suspended and waiting for the Host's decisions.

As shown in Figure 5-7, the Host's interface process is extracted from the Host's initial workflow definition. Then, both the Host's and the Guest's interface processes are compared, during which the first encountered discrepancy will be singled out. By consulting the Host's initial workflow definition, control flow reconciliation options are formulated and then prompted to users (Business Administrator). Supported by the real-time user decision assessment services, impact of users' decisions is evaluated by consulting the Host's initial workflow definition. Any implicated new option will be brought to users' attention. After all the reconciliation options have been prompted and all relevant decisions have been made, the Host's initial workflow is updated with all the committed adjustment and the second round of unilateral decision making cycle begins given users' consent.

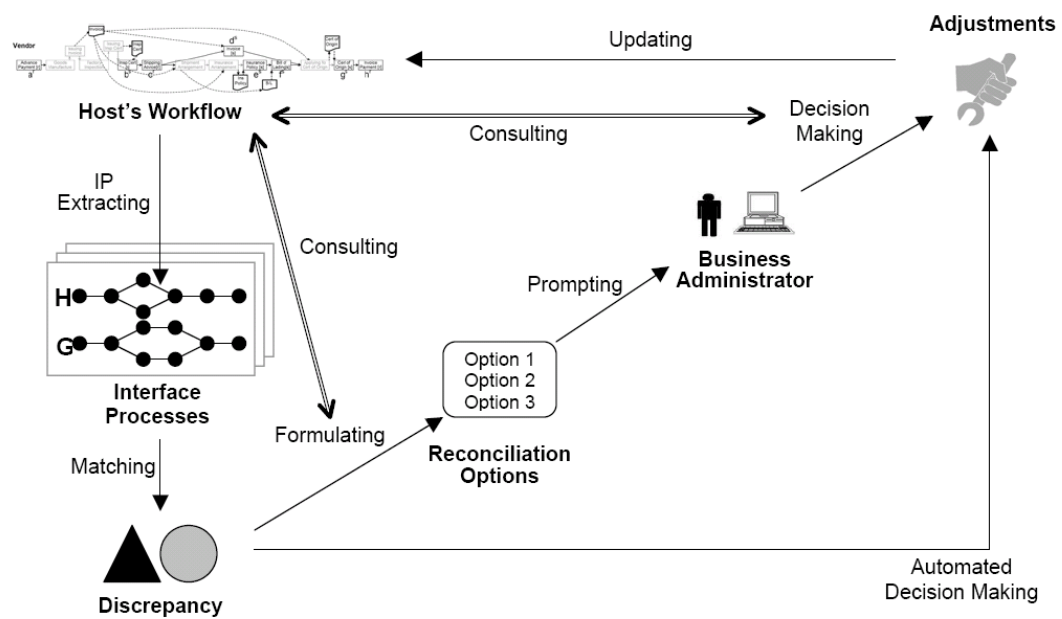


Figure 5-7. Unilateral decision-making.

### 5.2.1.3 Capturing Options

Due to the dynamic and changing business environment, there is a good chance that communication could reveal conflicting interests between two business partners. According to the negotiation principles mentioned in §4.4.2, when faced with a conflict, partners should try to figure out how to adjust initial individual interests in order to reconcile the conflict and make a deal. Such adjustments are included in the reconciliation options. For collaborative workflows, since the interests take the form of interface activities and the associated control flow and data flow, reconciliation options are expressed as the adjustments associated with these flow models.

### Control Flow Driven Adjustment

However, a practical question is raised as “To what extent should a workflow be adjusted?” On the one hand, as a general principle, adjustments must contribute to the reconciliation of the conflict. In workflow collaboration context, behavioural conflicts are caused by differences in control flow. Due to the independent relationship between two parties’ interface processes, any adjustment carried out by one partner that does not eradicate the conflict will not have any contributing effect on later options available to the other partner. It is explained in the following example shown in Figure 5-8.

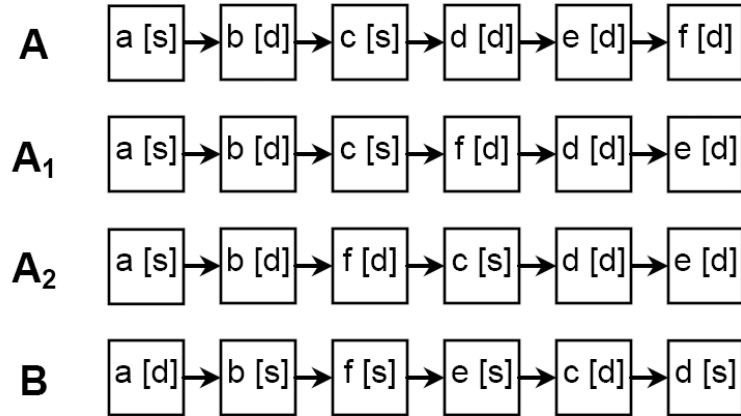


Figure 5-8. How much to adjust?

Partners  $P_A$  and  $P_B$  are the owners of interface processes  $A$  and  $B$  respectively. If  $P_A$  takes the option of ‘move activity  $A.f[d]$  before  $A.d[d]$ ’ and process  $A$  becomes  $A_1$ , there will still be conflict between  $A_1$  and  $B$  and the option available to  $P_B$  to reconcile the discrepancy will still be the same, i.e. ‘move activity  $B.c[d]$  before  $B.f[s]$ ’. That is to say the option of adjustment ( $A \rightarrow A_1$ ) is not a contribution to resolving the conflict. Therefore, if  $P_A$  decides to concede, it must fully concede with respect to the current conflict, i.e. ‘move  $A.f[d]$  before  $A.c[s]$ ’, which would result in process  $A_2$ .

On the other hand, given the causal relationships between activities, the adjustment should not cause any negative side effect on the hosting process, which could potentially lead to some other conflicts. This means any adjustment shall not only target where the discrepancy occurs, but also where the common trace ends. For processes with only sequential structures, these two types of target position are structurally continuous; but for processes with parallel structures, they may turn out to be structurally discontinuous. More examples are given in §6.4.1 but for now a simple example, as shown in Figure 5-8, is used to briefly explain the two-tailed nature of control flow adjustment. Given the two interface processes  $A$  and  $B$ , in order for  $P_A$ ’s reconciliation option to satisfy control flow compatibility requirement, activity  $A.f[d]$  should be brought forward but not before the last common activity  $A.b[d]$ . Otherwise, a new conflict would occur due to  $A.f[d]$ ’s overshooting.



### **Control Flow Relevance Testing**

When the two-tailed control flow adjustment is structurally discontinuous, a safe option is to join all the discontinuous candidate target positions. However it imposes more stringent changes to the initial process model. In order to work out a more accurate reconciliation option, relevant control flow connectivity needs to be examined. That is to say if no control flow dependency is imposed from a candidate target position, there is no need to include such a position for the purpose of process adjustment, which, as a result, also preserves the initial workflow semantics. More detailed explanation can be found in §6.4.3.2 Connectivity Perspective.

#### **5.2.1.4 Implication of Decisions**

A decision made on an adjustment option can be acceptance or rejection. Acceptance implicates certain concessions whilst rejection means no concession has been made. However, for multiple simultaneous options, things can become much more complicated because various combination of decisions may lead to different implications, e.g. concessions and/or new options for users to decide. Therefore, by following the concession-convergent approach, it is important that the implication of decisions are fully captured and prompted to human users so that they can thoroughly understand the dynamics that their decisions can bring about and make informed decisions. As a result of acceptance decision on adjustment options, concessions emerge, again, on two layers – control flow and data flow. For any control flow reconciliation adjustment, control flow concessions are inevitable. However, whether there will be any implied data flow concession depends on whether the control flow adjustment will cause any data flow conflict. Should any data flow conflict occur, it must be brought to users' attention before relevant decisions are made. More details with regard to the dynamics of option and decision can be found in §6.4.5 Option and Decision Dynamics.

#### **5.2.2 Agreement and Fulfilment**

After two partners have successfully reconciled all the conflicting interests, an agreement can be reached, which represents the common interests between the two partners, in the form of a commonly agreed sequence of exchanging collaborative messages. However, different from a conventional agreement, the facts agreed by both parties do not have to be

explicitly written or recorded somewhere but be embedded in both partners' workflow definitions, i.e. in the form of a pair of compatible workflows.

Once a collaboration agreement is reached, it needs to be fulfilled by both partners. Being captured and represented as a pair of compatible workflows, the agreement can be fulfilled through run-time enactment of the pair of workflows connected by a message relay facility.

## 5.3 Architecture View

This section looks at the Framework from an architecture view. Logic components are introduced followed by the arrangement of the architecture in the form of a service stack.

### 5.3.1 Logic Components

Based on the discussion of the business view, it is identified that apart from the existing workflow systems, three logic components need to be provided by the Framework covering both collaboration build time and run time, they are:

- *Collaboration Negotiation Block*, which comprises *Process Matching*, *User Interaction* and *Process Change*,
- *Agreement Fulfilment Block* (if an agreement is reached),
- *Collaboration Enactment Infrastructure*.

Their relationship is shown in Figure 5-9. Due to the nature of the concession-convergent approach, it may take several rounds for partners to completely compare their interests, plough through available options, assess concessions, make decisions and effect changes at collaboration build time. This is reflected as the cyclical operation manner among the build-time logic components before entering the run-time fulfilment stage if an agreement has been achieved. At both stages, message relay facilities are provided by the collaboration enactment infrastructure to facilitate information exchange.

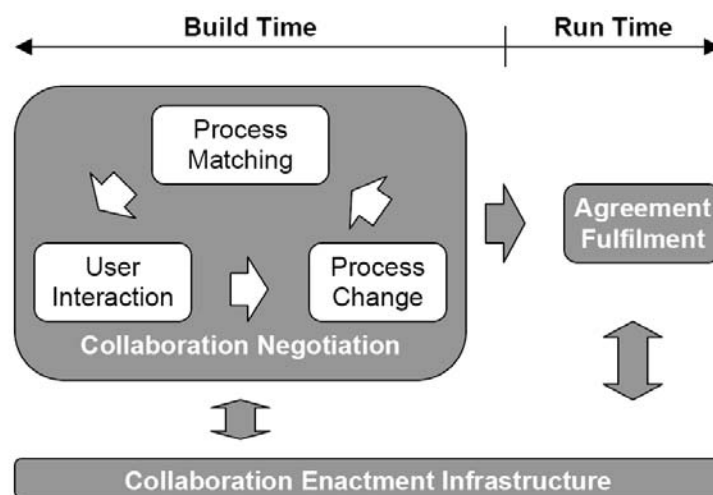


Figure 5-9. Logic components.

### 5.3.1.1 Collaboration Negotiation

*Collaboration Negotiation* engages both partners in the process of collaboration negotiation and drives the way to the formulation of a potential agreement. It coordinates the running of all the key operational elements of negotiation on each side and makes sure both partners exchange information according to the adopted negotiation protocol the communication support provided by the collaboration enactment infrastructure. Within this logic component, three sub-components are identified, namely *Process Matching*, *User Interaction* and *Process Change*.

#### Process Matching

*Process Matching* is the core component of *Collaboration Negotiation*. It is in charge of interface process extraction followed by process comparison and discrepancy identification. The output, either a successful match or an encountered discrepancy, will be forwarded to the *Users Interaction* component for further processing. Exceptions to this are a number of situations that have been identified to qualify automated decision-making, which can save users' effort and reduce unnecessary confusion. These situations will be introduced in §6.4.6.3 Manual and Automated Decision Making.

#### Users Interaction

When discrepancy is encountered, users must be consulted with how and whether to reconcile the discrepancy. Through user interface, reconciliation options together with all

relevant information that can help the decision-making are immediately prompted to users in a comprehensive, accurate and human understandable manner. It is also desired that the user interface can provide users with a test bed so that different decision combinations to be made on the same set of reconciliation options can be tried, compared, evaluated and decided.

### **Process Change**

In order to meet the requirements for ensuring completeness, maintaining integrity and achieving independency from technical personnel during process changes, IT support is provided to materialise all the accepted adjustments following the decision-making by users. After users confirm their decisions on the currently available reconciliation options, relevant workflow definitions in the process repository are accessed and adjusted based on each of users' decisions of acceptance with integrity of process digraph closely observed. This provides the non-technical business professionals with the ability in fully controlling the procedure of process negotiation and reduces their workload and the possibility of potential human error.

#### **5.3.1.2 Agreement Fulfilment**

*Agreement Fulfilment* in workflow collaboration context refers to the enactment of a compatible pair of collaborative workflow definitions following a successful collaboration negotiation. This task is carried out by workflow enactment services of each partner's WfMS with the communication support provided by the collaboration enactment infrastructure.

#### **5.3.1.3 Collaboration Enactment Infrastructure**

Due to the distributed nature featured by cross-organisational collaboration, partners' corresponding services at both of the build-time process negotiation and the run-time agreement fulfilment stages need to be connected and linked together. For this reason, a collaboration enactment infrastructure underpinned by a message relay facility is recognised as a constituent component of the Framework.

As a requirement mentioned in §4.2.2.5, due to the popularity of currently available workflow management systems, the design of the enactment infrastructure should be in line

with the mainstream paradigm of existing WfMSs without imposing any substantial extension. Therefore, at build time, the infrastructure should be able to access process repository to get interface processes that need to be exchanged with partners. At run time, the infrastructure should be attached to the workflow engine to access and pass on workflow relevant data between partners' corresponding workflows.

The loose coupling principle being adopted throughout the thesis requires the message relay facility to take one more role – collaboration mediator. As mentioned in §3.6.5.2, mediated message passing can make communicating partners exchange messages without the exact knowledge of the sender and recipient's technical details. Messages need to be sent to, matched within, and dispatched from the message relay facility. Matchmaking is through the comparison between data semantic identifiers that are carried by messages and mapped onto a commonly agreed business domain ontology. By recording the messages in transit in the facility, the progress of both the negotiation and agreement fulfilment can be monitored closely as well.

### **5.3.2 Service Stack**

All the logic components identified in §5.3.1 can be wrapped as services and arranged in the form of a layered service stack. Together with the existing workflow management systems, they compose the overall Framework. Figure 5-10 shows the structure of the Framework.

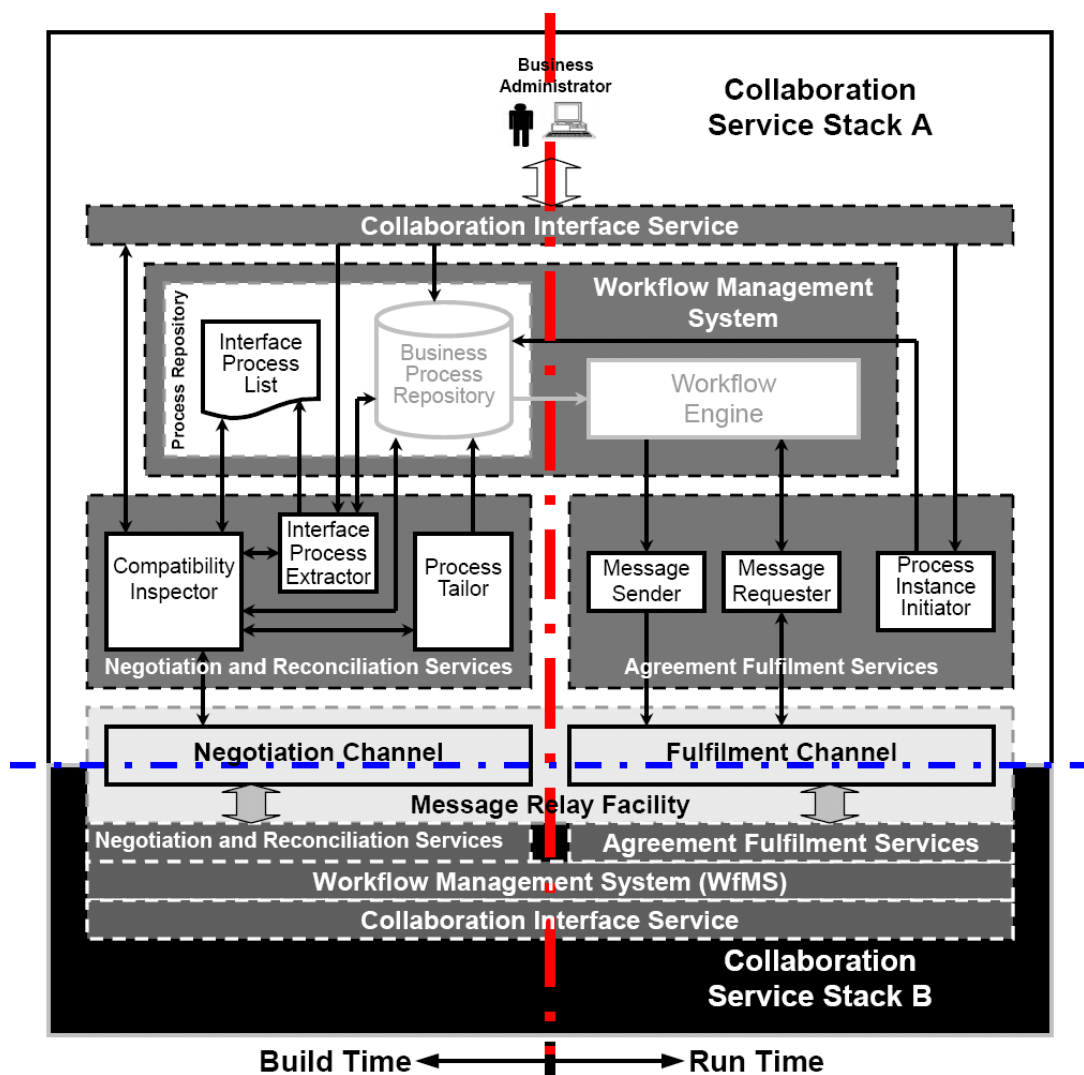


Figure 5-10. Architecture of the cross-organisational workflow collaboration framework.

For each partner, relevant services include the *Collaboration Interface Service*, the existing *Workflow Management System*, the *Negotiation and Reconciliation Services* and the *Agreement Fulfilment Services*. The *Message Relay Facility* is physically linking the two service stacks that represent both partners respectively.

The services associated with collaboration negotiation can be seen on the left hand side, i.e. the build-time half of Figure 5-10 whilst the agreement fulfilment services are on the right hand side or the run-time half. In the figure, objects with solid grey borders denote existing conventional WfMS components. The *Collaboration Interface* serves as user interfaces

between human users and other services. It also coordinates the overall progress from build time through to run time. At build time, the *Interface Process Extractor* extracts interface processes from initial business processes. The *Negotiation and Reconciliation Services* provides services for interface process comparison, compatibility checking, discrepancy detection, reconciliation formulation and process tailoring. The *Negotiation Channel*, acting as a mediator, is used for exchanging intermediate interface processes as negotiation offer/counteroffers. The *Message Sender* and *Requester* of the *Agreement Fulfilment Services* are attached to a conventional workflow enactment service. They pass collaboration messages back and forth through the *Fulfilment Channel*.

### 5.3.3 Implementation Architectures

With clear boundary divisions and the support of public interfaces, the Framework has the flexibility to be adapted to different implementation architectures, e.g. the client/server and service-oriented architectures.

#### 5.3.3.1 Client/Server Architecture

The COWCO Framework can be adapted to the classic client/server architecture. In general, two types of client/server architectures exist, namely rich client and thin client.

Figure 5-11 illustrates the architectural arrangement of the rich client type for the Framework. The collaboration interface, the negotiation and reconciliation services, and the agreement fulfilment services compose parts of the client application. They are all connected to the workflow management system currently used by a partner through public interfaces defined by the Workflow Management Coalition (WfMC). The message relay facility is implemented on the server side, which is in charge of message matching and relaying.

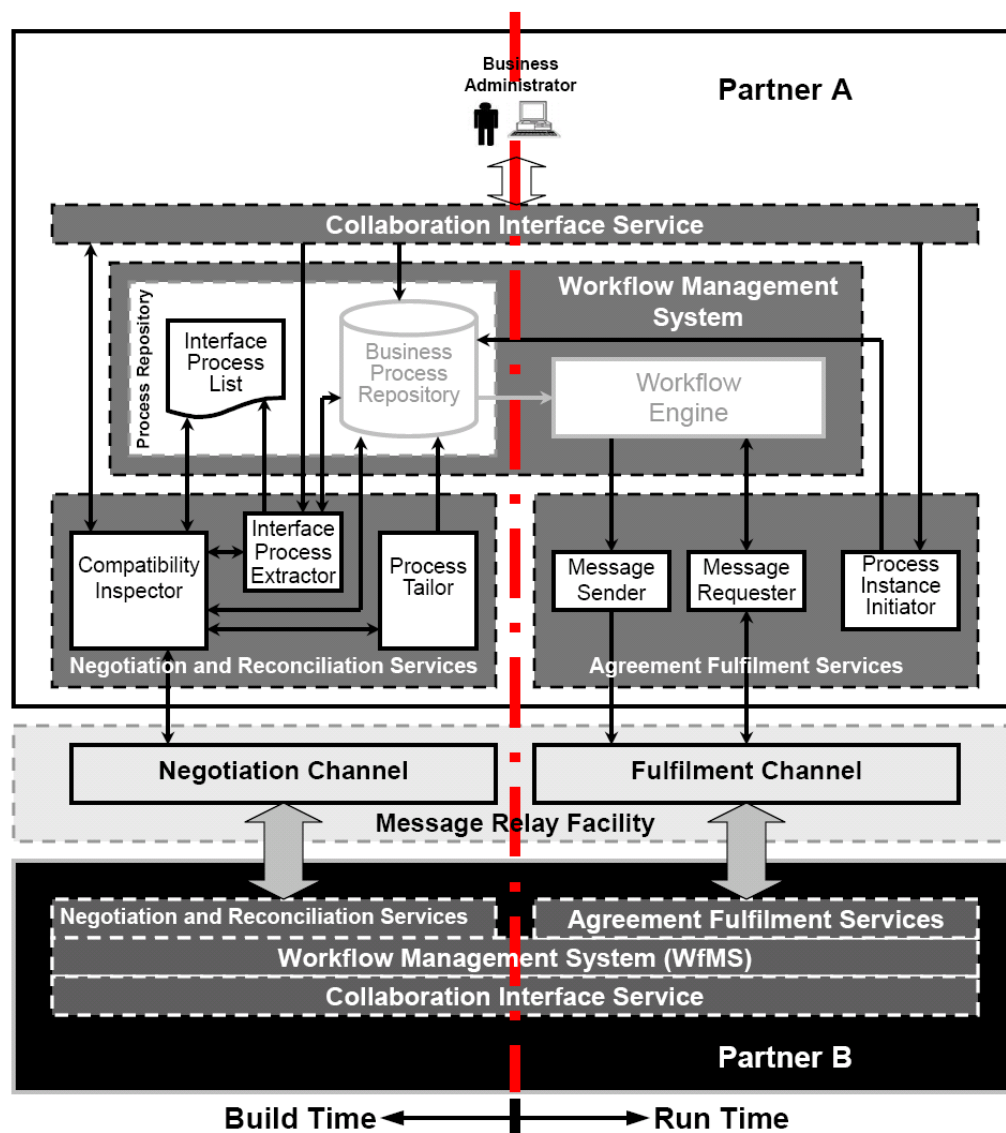


Figure 5-11. Implementation architecture: rich client C/S.

Associated with the discussion regarding the overall negotiation process (mentioned in §5.2.1.2 Capturing Communication), Figure 5-12 illustrates the movements of collaborative messages in the rich client C/S architecture that are exchanged between clients and server together with key data that is generated and used within the scope of a single logic component.



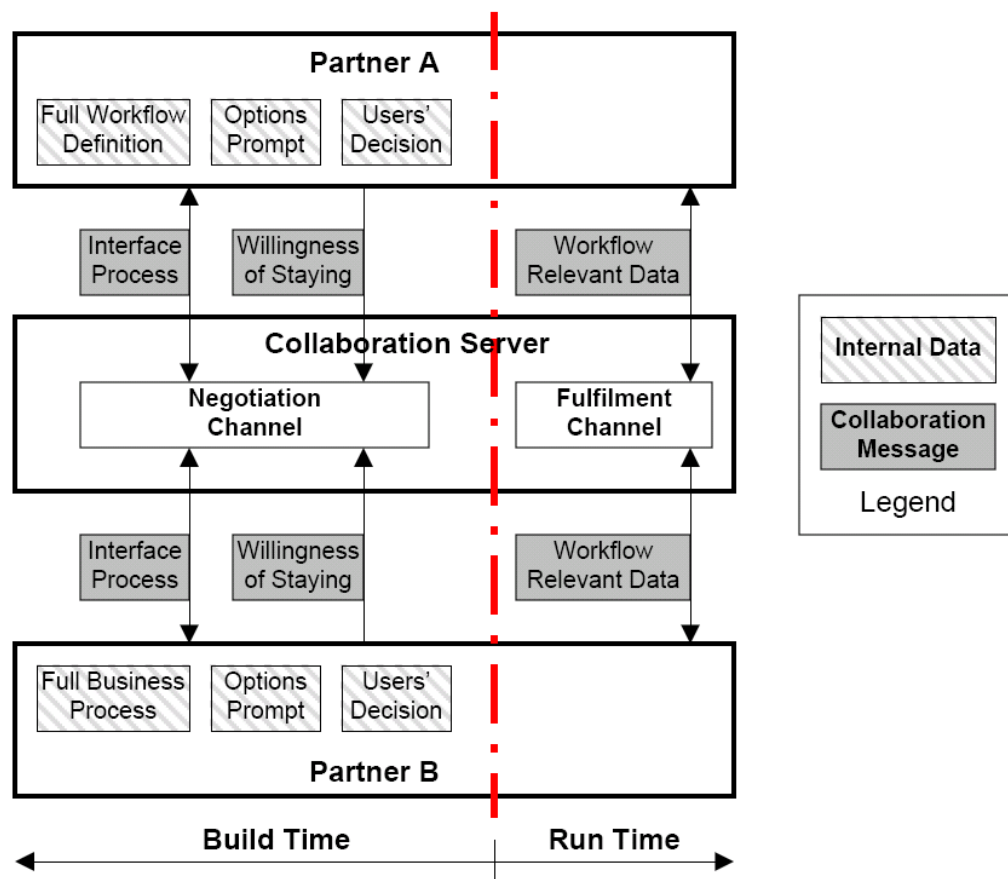


Figure 5-12. Collaborative messages and internal data by following rich client C/S architecture.

To make the client side thinner, the negotiation and reconciliation services of both sides can be chopped off from clients and relocated onto the server side. Each partner's WfMS remains on clients' sides together with the collaboration interface. Due to the close relationship between the agreement fulfilment services and the workflow enactment service of each partner's WfMS, the agreement fulfilment services is preferred to stay together with the WfMS on both clients' sides. The thin client C/S architecture is shown in Figure 5-13.

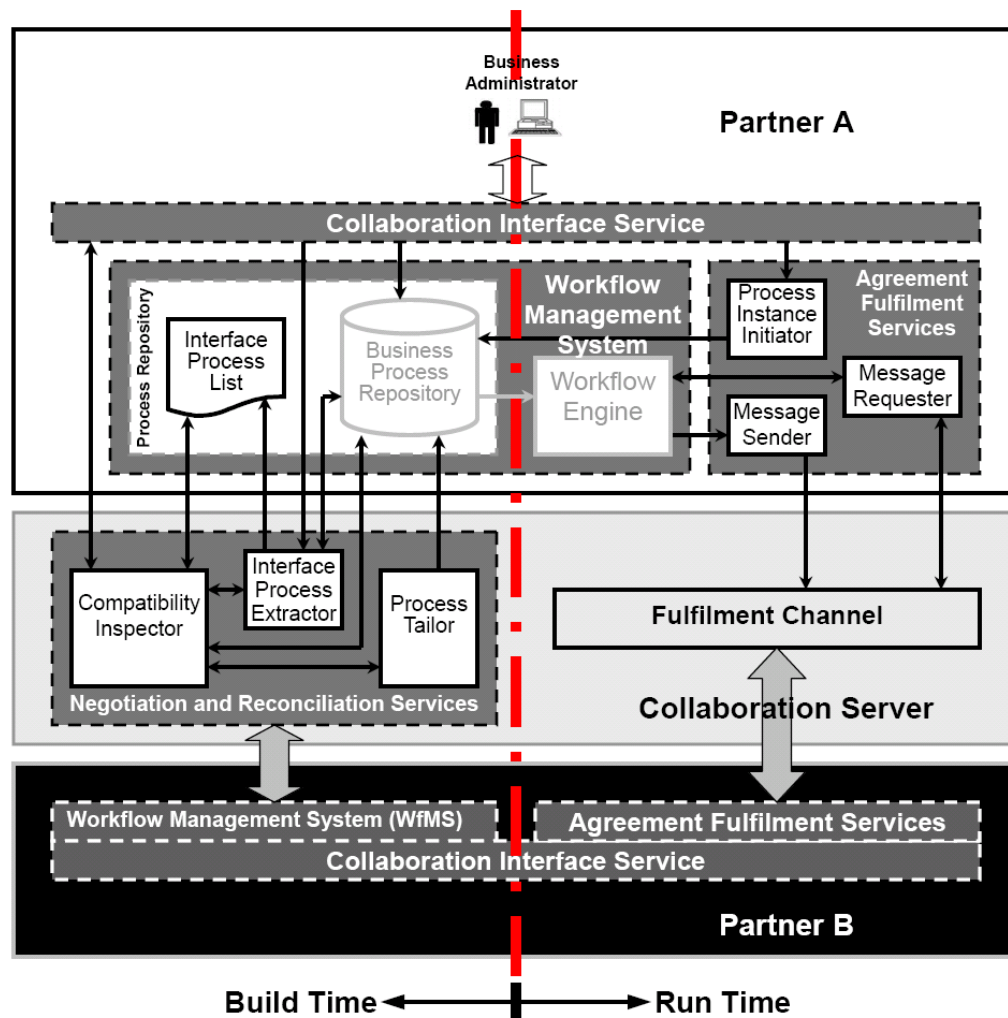


Figure 5-13. Implementation architecture: thin client C/S.

As a result, collaborative messages need to expand to include full workflow definitions and reconciliation options. Since interface extraction and comparison are now based on the server, interface processes no longer need to be exchanged between clients and server and become internal data of collaboration negotiation and reconciliation services. These changes are reflected in Figure 5-14.

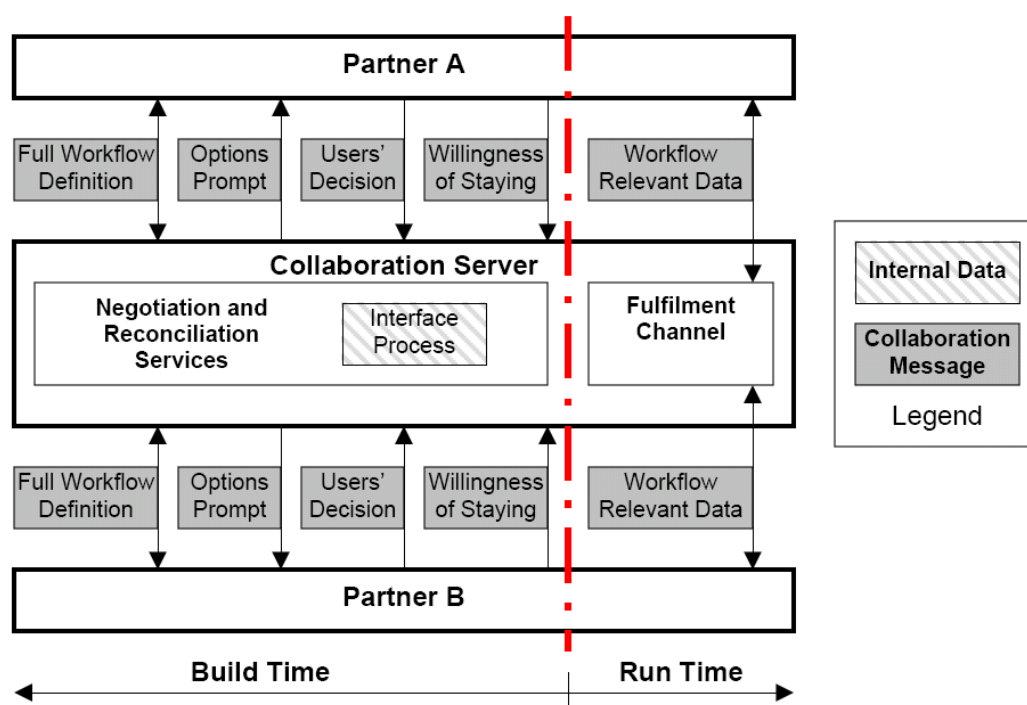


Figure 5-14. Collaborative messages and internal data by following thin client C/S architecture.

### 5.3.3.2 Service-Oriented Architecture

The most distinctive feature of the popular service-oriented architecture (SOA) is the dynamic service matching and binding. In order to adapt the Framework into an SOA, a *preliminary directory server* is needed for business partners to publish descriptions of their collaborative processes by using technologies like Universal Description Discovery and Integration (UDDI). Whenever a service requester initiates a collaboration request to the preliminary directory server, a coarse<sup>1</sup> granule service matching is conducted. Following a successful match, information regarding collaboration server will be sent to both partners as binding notices and relevant services (e.g. the collaboration interfaces, WfMS and agreement fulfilment services) of both partners will be bound to a *collaboration server* where negotiation and reconciliation services are hosted. From the moment of successful bindings, collaboration services can be invoked and executed by using the same architecture

<sup>1</sup> It is called 'course' because at this stage only the service description is matched against the request query compared with a fine granule process matching during collaboration negotiation at a later stage.

as the thin client client/server. The procedure of service discovery and binding is shown in Figure 5-15.

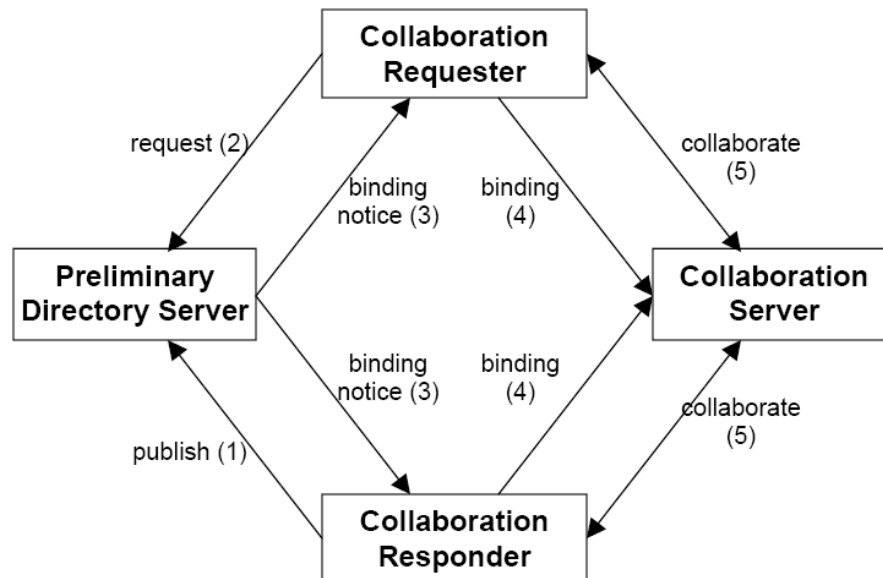


Figure 5-15. Implementation architecture: SOA.

## 5.4 Summary and Conclusion

In this chapter, a novel cross-organisational workflow collaboration supporting framework is proposed as response to the requirements and facilitate the integrated approach mentioned in Chapter 4. The Framework captures the three elements (interests, communication and options) in negotiation operation from workflow technology perspective. Logic components that compose the Framework are identified and described. The architecture of the Framework is designed as a service stack. Finally, the ways, by which the Framework is adapted to three implementation architectures, are explained.

The Framework covers workflow collaboration from build time to run time corresponding to the cycle of negotiation through agreement to fulfilment found in conventional business collaboration. A new criterion of process compatibility is discovered as Business Collaborative Compatibility addressing issues of enactment deadlock and unnecessary delay. The build time services are organised by an overall negotiation process, an

implementation and enforcement of the real-time sequential negotiation protocol. The mutual gain and concession convergence negotiation approaches are combined and employed by the Framework in discovering the behavioural discrepancies. Effective user interaction is key for such a human-centric approach. A workflow collaboration enactment infrastructure is needed to facilitate message exchange at both build time and run time.

# Chapter 6

## Enabling Workflow Negotiation and Reconciliation

### 6.1 Introduction

Following the adoption of an integrated approach and the proposed design of a comprehensive framework, key enabling techniques for workflow negotiation and reconciliation are explained in details in this chapter.

§6.2 introduces the technique of interface process extraction, which extracts an interface processes from its hosting workflow definition. §6.3 describes the procedure of comparing two processes, including interface activity bipartite comparison and process behaviour comparison with descriptions of a discrepancy scene if discrepancies are encountered. §6.4 explains the details of formulation, prompting and decision-making with regard to reconciliation options. §6.5 describes measures that ensure the correctness of process change operations. §6.6 summaries the chapter.

### 6.2 Interface Process Extraction

Due to the privacy requirement, no other information apart from collaboration critical information is allowed to be exposed to the collaborating partners. Apart from being used to model interaction points, interface activities are introduced for such access control purpose as well. Collaboration messages are sent or received through interface activities, which represent collaboration interests. However, an interface activity only represents a single isolated point of interest in a collaboration. It is the combination of all the interface activities and their causal relationship that reflect the overall interests of the hosting process, which

leads to the concept of an interface process as described in §5.2.1.2 Capturing Communication. In this section, the mechanism of extracting an interface process from its hosting workflow definition is discussed.

### 6.2.1 Design

The task of interface process extraction is approached by identifying and removing every activity that is not an interface activity and rejoining its control flow predecessor(s) and successor(s). After all the non-interface activities are removed, what is left is the desired interface process. When a non-interface activity is removed and its predecessor(s) and successor(s) are rejoined, three special situations need to be considered and handled differently. If the non-interface activity:

- i. is the first activity and it has multiple immediate successors,
- ii. is the last activity and it has multiple immediate predecessors,
- iii. has both multiple immediate predecessors and successors.

In any of these circumstances, a dummy activity for control flow needs to replace the non-interface activity to keep all the intermediate and the final interface process definitions syntactically and semantically correct. Otherwise, in the first two situations, multiple edges would connect directly to the start and end point, which are syntactical violations of a workflow definition. In the last situation, the workflow semantics would be changed. This is because in the initial workflow definition all the immediate predecessors need to be completed before any of the immediate successors can be activated; but in the adjusted version, without the dummy activity, as soon as any of the immediate predecessors is completed then all the successors will be activated. Figure 6-1 illustrates the three situations.

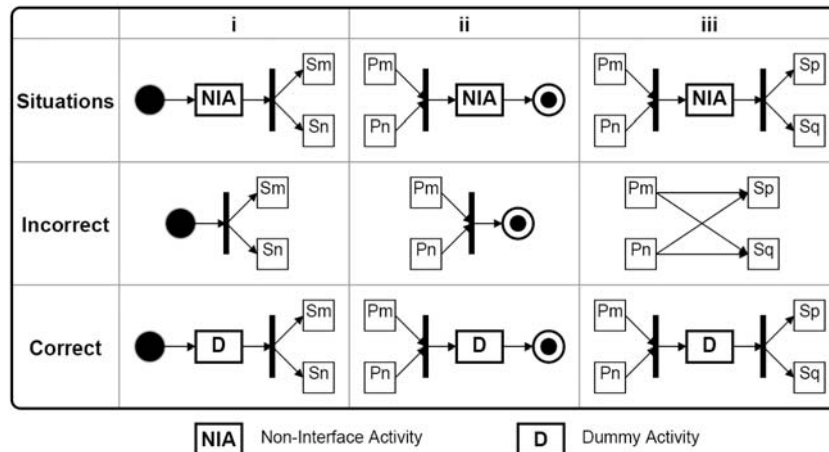


Figure 6-1. Special situations in interface process extractions.

### 6.2.2 Algorithm

The algorithm is shown in Figure 6-2.

```
getInterfaceProcess(AD_Wf): AD_IP
AD_IP = copy(AD_Wf)
FOR each v in AD_IP
    IF v is a non-interface activity
        pred = getPred(v, AD_IP)
        succ = getSucc(v, AD_IP)
        reconnect(pred, succ, AD_IP) //add dummy activity when necessary
        remove(v, AD_IP)
    ELSE
        IF v isSupplyType(v)
            v.role = 1
        IF v isDemandType(v)
            v.role = 2

Function:
void reconnect(pred, succ, AD_IP)
IF pred = startVertex AND succ.size > 1
    OR
    succ = endVertex AND pred.size > 1
    OR
    pred.size > 1 AND succ.size > 1
    dummyVertex = createVertex()
    connect(pred, dummyVertex, AD_IP)
    connect(dummyVertex, succ, AD_IP)
ELSE
    connect(pred, succ, AD_IP)
```

Figure 6-2. Algorithm for interface process extraction.



Implementation details of the algorithm vary due to the difference among the target process definition languages (PDL). For this project, the target PDL is chosen as WfMC's XML Process Definition Language (XPDL) and the implementation is specially tailored for it.

## 6.3 Interface Process Comparison

After the extraction of the interface processes from collaboration partners, they need to be compared to reveal whether there is any discrepancy between them.

### 6.3.1 Pre-treatment – Activity Bipartite Comparison

Before any process comparison procedure takes place, one type of discrepancy can be immediately identified. It is caused by the uniqueness of collaboration message semantics and can be revealed through interface activity bipartite comparison.

#### 6.3.1.1 Unique Collaboration Message/Interface Activity

Since in this thesis the semantic identifier of a collaboration message is used as the identifier of its portal interface activity, the identification of a unique collaboration message equals the identification of the corresponding unique interface activity. In practice, if a semantic identifier only appears in the interface activities of one partner but not of the other, the semantic identifier is deemed to be unique.

As interface activity has two types – data supply and data demand, an identified unique interface activity also can be grouped as *unique data supply* as well as *unique data demand* interface activities, abbreviated as UDSIA and UDDIA respectively.

On the one hand, as a data supplier, a UDSIA supplies data that a partner does not demand, which makes the handling of such a discrepancy a waste of resource. Therefore, it is safe to discard such an interface activity automatically, be it either on the Host side or the Guest side. However, whether the data generating activity – the UDSIA's immediate data predecessor – is to be removed or retained still needs to be decided by the user. When the user is prompted with relevant information, the data generating activity together with all its immediate data successors (if any) apart from the UDSIA should all be included for assessment. An example can be found in §8.3.2 Example 2: Incompatible Workflows

(C1.E2). Automated adjustments will be collectively addressed in §6.4.6.3 Manual and Automated Decision Making.

On the other hand, as a data consumer, a UDDIA expects a partner to generate and supply certain data. For the convenience of the following discussion, it is assumed that the UDDIA is located in the Guest's process. Two different situations can be observed in the Host's process that does not have the corresponding data supply interface activity. Firstly, if the Host does not have such data available but still wants to meet the Guest's requirement, it needs to design and insert the data generation activity(ies). Secondly, if the Host has already got the required data generated by some other activity in its process but only for internal use, a corresponding data supply interface activity will be inserted immediately after the data generation activity in the Host's process given users' consent, which, as a result, can effectively eliminate the uniqueness of the UDDIA in the Guest's process. This procedure is called UDDIA purification in this thesis.

#### **6.3.1.2 Estimated Effort of Design from Scratch**

Before conducting process negotiation and reconciliation, a question must be asked "Is it worth doing?" The answer is that if it requires a lot of effort in creating and inserting new activities and data (messages) into an existing workflow definition during the reconciliation process then it may be more cost-effective to design the collaborative workflow from scratch than to reconcile the differences between existing workflows. Therefore, an estimation of the *effort of design from scratch* needs to be studied.

Based on the discussion in §6.3.1.1, it can be seen that the number of new data generation activities in the Host's workflow is positively correlated with the number of UDDIAs in the Guest's workflow following the procedure of UDDIA purification. Therefore, the number of UDDIAs in the Guest's workflow is used to estimate the number of new data generation activities that need to be created and inserted in the Host's workflow. Based on this relationship, the *estimated effort of design from scratch* for a Host's workflow can be formulated as:

Equation 1. Effort of design from scratch.

$$\text{estimated effort of design from scratch (EEDS) for a Host's workflow} = \frac{\text{number of Guest's UDDIAs}}{\text{number of common interface activities between the Host and the Guest}}$$

A threshold can be defined as the maximum level of acceptance for conducting process negotiation and reconciliation rather than design from scratch. For multiple workflows whose EEDS are all below the threshold, the one with the lowest EEDS is the most preferred workflow for the Host to start with in the collaboration negotiation.

### 6.3.2 Process Behaviour Comparison

Given the target problem area in this thesis, processes need to be compared from a behavioural perspective. Measures are developed to conduct the process comparison task in two steps: activity diagram conversion and state transition system comparison.

#### 6.3.2.1 Conversion from Activity Diagram to State Transition System Diagram

The required input and the expected output for activity diagram conversion are largely the same as Krukkert's (2003) approach as introduced in §4.5.2 Process Comparison. The differences are described below.

#### Dealing with Reachability of a Join-Type Activity

If at any given point of time, an activity is ready to be triggered, it is described as a reachable activity. Without adopting the 'wait state' notion in the process representation, the task of deciding whether a join-type activity is reachable appears different but more straightforward. The reachability of a join-type activity depends on the completion status of all its immediate predecessor(s). When all the immediate predecessor(s) are completed, the join-type activity becomes reachable. The algorithm is shown in Figure 6-3.

```
isReachable(currVertex, AD_IP): {true | false}
IF isJoinType(currVertex)
    immePredList = getImmePredList(currVertex, AD_IP)
```

```
IF allCompleted(immePredList, currVertex)
    RETURN true
ELSE
    RETURN false
```

Figure 6-3. Algorithm for deciding reachability of join-type activity.

### **Skipping Dummy Activity**

During interface process extraction, control flow dummy activities might be introduced to retain the syntactic and semantic integrity of the initial activity diagram. However, since a dummy activity is merely a placeholder and bears no business-related meaning, it should not appear in the resulting STS diagram. The algorithm with regard to skipping dummy activities is shown in Figure 6-4.

```
getNonDummyReachable(currVertex, AD_IP): {act : act ∈ vertexList_IP}
FOR each v in vertexList_IP
    IF isImmeSucc(v, currVertex, AD_IP)
        IF isReachable(v, AD_IP)
            IF isDummy(v)
                tempResultList = getNonDummyReachable(v, AD_IP)
            ELSE
                tempResultList.append(v)
    IF countElement(tempResultList) >= 1
        currResultList.append(tempResultList)
RETURN currResultList
```

Figure 6-4. Algorithm for skipping dummy activity

Consider Figure 6-5 and assume that at this state activity *j* and *k* have been completed. Activity *dummy* is the next reachable activity. In order to get the desired state, *dummy* needs to be skipped by introducing its immediate successors (*l* and *m*) as the reachable activities.

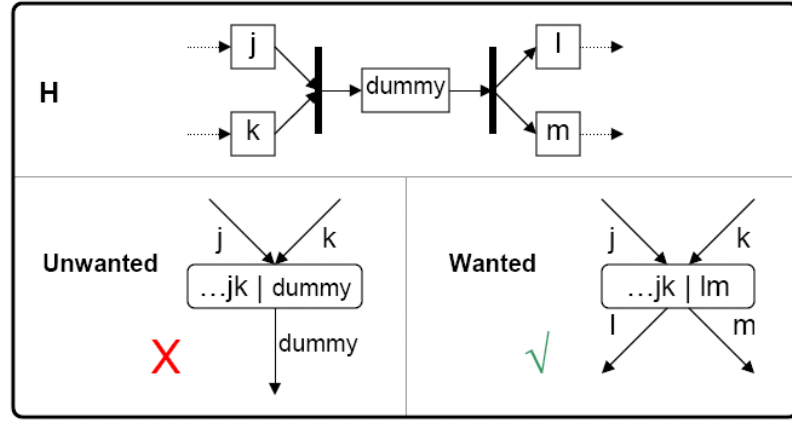


Figure 6-5. Example of skipping dummy activity.

### 6.3.2.2 State Transition System Comparison

In this thesis, the expected outcome from activity diagram comparison is substantially different from that in Krukkert's approach. Krukkert aims to have all the common activity sequences recorded and represented in the form of a common STS diagram, which can then be converted back into a common activity diagram. It is discovered that by following the collaboration modelling and design principles underpinning the workflow collaboration enactment infrastructure proposed in this thesis, there is no need to convert any two compatible workflows back into the form of a common workflow definition. For this reason the common trace recording procedure is eliminated in the STS comparison module in this thesis.

Due to the number of activities of both participating interface processes might be different, four situations are identified along the way of STS comparison, which are highlighted in the algorithm shown in Figure 6-6.

```
compareStateModels(STSD_H, STSD_G, FlowModel_H)
explorationResult: {commonTrace{true | false}, discrepancyScene{CCPIAS_H,
DIAS_H, DIAS_G} }
comparisonResult: {compatible | incompatible_concession | incompatible_rejection}
decision: {concession | rejection}
```

```
//Both interface processes(IPs) have not reached their ends
//Will result in either compatibility or discrepancy
WHILE (!isEnd(STSD_H) AND !isEnd(STSD_G))
    explorationResult = exploreCommonTrace(STSD_H, STSD_G)
    IF explorationResult.commonTrace
        comparisonResult = compatible
        RETURN comparisonResult
    ELSE
        decision = handleDiscrepancy(explorationResult.discrepancyScene,
                                      FlowModel_H)
        IF decision = concession
            comparisonResult = incompatible_concession
        ELSE
            comparisonResult = incompatible_rejection
        RETURN comparisonResult
//Host's IP has reached its end but Guest's has not
//Will certainly result in discrepancy
IF (isEnd(STSD_H) AND !isEnd(STSD_G))
    explorationResult = exploreCommonTrace(STSD_H, STSD_G)
    decision = handleDiscrepancy(explorationResult.discrepancyScene,
                                FlowModel_H)
    IF decision = concession
        comparisonResult = incompatible_concession
    ELSE
        comparisonResult = incompatible_rejection
    RETURN comparisonResult
//Host's IP has not reached its end but Guest's has
//Will certainly result in discrepancy
IF (!isEnd(STSD_H) AND isEnd(STSD_G))
    explorationResult = exploreCommonTrace(STSD_H, STSD_G)
    decision = handleDiscrepancy(explorationResult.discrepancyScene,
                                FlowModel_H)
```

```

IF decision = concession
    comparisonResult = incompatible_concession
ELSE
    comparisonResult = incompatible_rejection
RETURN comparisonResult
//Both IPs have reached their ends
//Will certainly result in compatibility
IF (isEnd(STSD_H) AND isEnd(STSD_G))
    comparisonResult = compatible
RETURN comparisonResult

```

Figure 6-6. Algorithm for activity diagram comparison framework.

### 6.3.3 Discrepancy Identification

Discrepancies occur when no single common trace is found. At the discrepancy scene, for each partner, two elements can be identified, namely *current common preceding interface activity set* (CCPIAS) and *discrepant interface activity set* (DIAS). A third element can be derived from both partners' DIASs, which is *candidate adjusting interface activity set* (CAIAS). In state transition system diagrams, given the two states in each STSD where the current common trace terminates, the following terms are defined. Their relationships are shown in Figure 6-7.

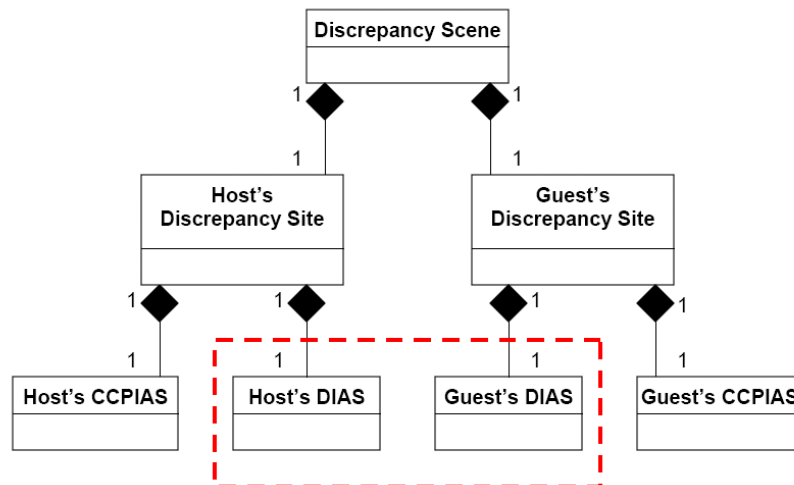


Figure 6-7. Discrepancy Scene in UML class diagram.

Definition 6-1: A *discrepancy scene* includes both discrepancy sites of the partners. For one partner, a discrepancy site refers to the state together with its incoming edges where common trace between the two state transition systems ends.

Definition 6-2: A *current common preceding interface activity set* (CCPIAS) contains all the common interface activities that lead to the state where common trace terminates. It exists on both sides.

Definition 6-3: A *discrepant interface activity set* (DIAS) contains all the reachable activities of the state where common trace terminates. It exists on both sides.

Definition 6-4: A *candidate adjusting interface activity set* (CAIAS) is the union set of all the interface activities that need to undertake certain form of adjustment that contributes to the reconciliation of the currently identified discrepancy.

Due to different operations carried out on the interface activities within this set, the CAIAS can be further differentiated into three subsets: *candidate removing interface activity set* (CRIAS), *candidate moving interface activity set* (CMIAS) and *candidate inserting interface activity set* (CIAS). All the potential adjustment operations involved are named from the Host's perspective. Figure 6-8 illustrates the concept model of CAIAS.



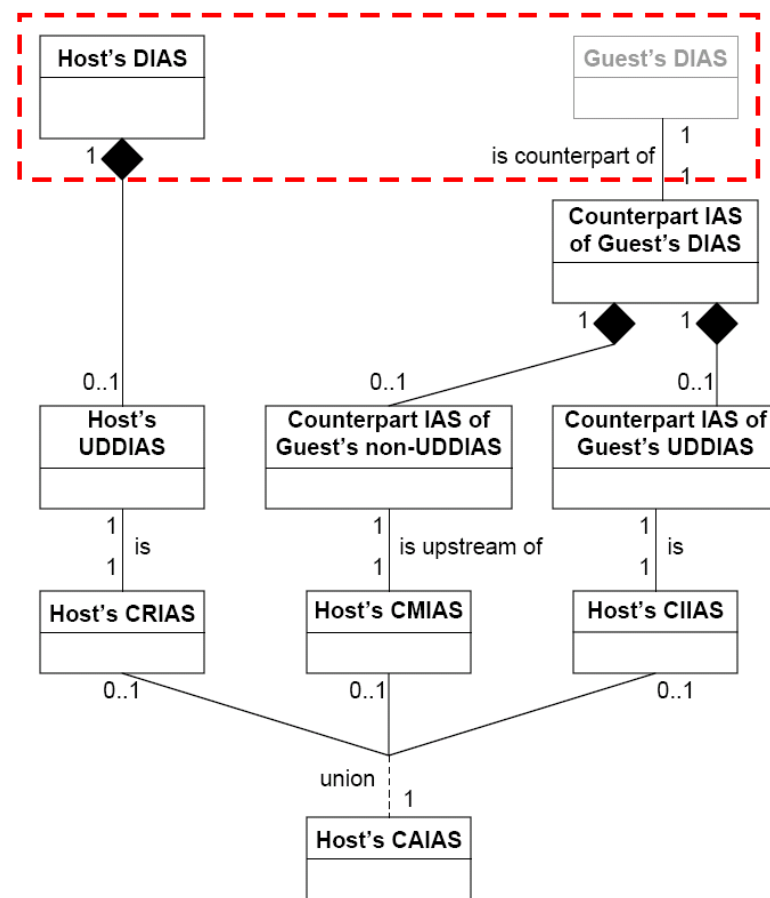


Figure 6-8. Candidate Adjusting Interface Activity Set in UML class diagram.

Definition 6-5: A *candidate removing interface activity set* (CRIAS) contains all the unique data demand interface activities in the Host's DIAS. In order to reconcile the discrepancies caused by activities in this set, the activities need to be removed from the Host's workflow definition. A more detailed discussion can be found in §6.4.2.1 Candidate Removing Interface Activity and their Data Successors.

Definition 6-6: A *candidate moving interface activity set* (CMIAS) contains all the most upstream<sup>1</sup> positioned common interface activities in the Host's workflow, whose counterpart<sup>2</sup> activities can be found in the Guest's DIAS. In order to reconcile the

<sup>1</sup> The direction towards the start of a process is denoted as upstream and that towards the end is downstream.

<sup>2</sup> A counterpart activity refers to an interface activity with the same semantic identifier but opposite message delivery direction, e.g. *H.b[s]* is the counterpart activity of *G.b[d]*.

discrepancies caused by activities in this set, the activities need to be moved to appropriate destination positions. A more detailed discussion can be found in §6.4.2.2 Candidate Moving Interface Activity and their Data Predecessors.

**Definition 6-7:** A *candidate inserting interface activity set* (CIIAS) comprises all the interface activities that need to be inserted into the Host's process. Activities in this set need to be created on the Host side as data supply interface activities, which are counterpart activities of all the unique data demand interface activities in the Guest's DIAS. In order to reconcile the discrepancies caused by activities in this set, the activities need to be inserted into the Host's workflow definition at their desired destination positions together with relevant data generation activities. A more detailed discussion can be found in §6.4.2.3 Candidate Insertion Interface Activity and their Data Generation Activities.

For example, as shown in Figure 6-9, by following the *current common preceding interface activity set*, process *H* reaches the state  $H.[ab \mid c]$  and *G* reaches  $G.[ab \mid d]$ , where no more common reachable activity can be found and thus a discrepancy occurs. For process *H*, CCPIAS is  $\{H.b\}$ , DIAS is  $\{H.c\}$  and CMIAS is  $\{H.d[s]\}$  (the most upstream positioned counterpart activity of *G*'s DIAS –  $\{G.d[d]\}$  – in *H*).

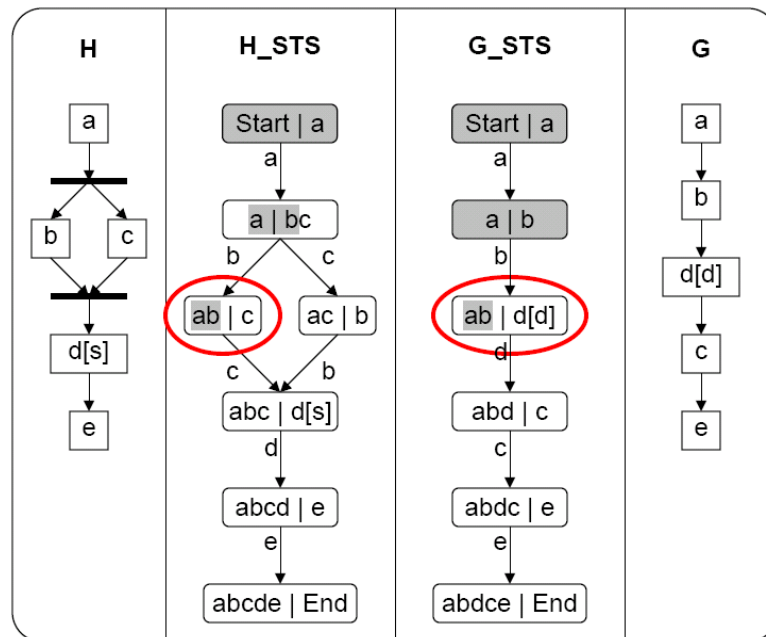


Figure 6-9. Example of discrepancy scene with CMIAS only.

Figure 6-10 illustrates a scenario with a unique interface activity ( $G.f$ ). For process  $H$ , CCPIAS is  $\{H.b\}$ , DIAS is  $\{H.c\}$ , CMIAS is  $\{H.e[s]\}$  (the most upstream positioned counterpart activity of  $G$ 's DIAS –  $\{G.e[d]\}$  in  $H$ ) and CIAS is  $\{H.f\}$  (since  $H.f$  is a unique interface activity appeared in  $G$ 's DIAS).

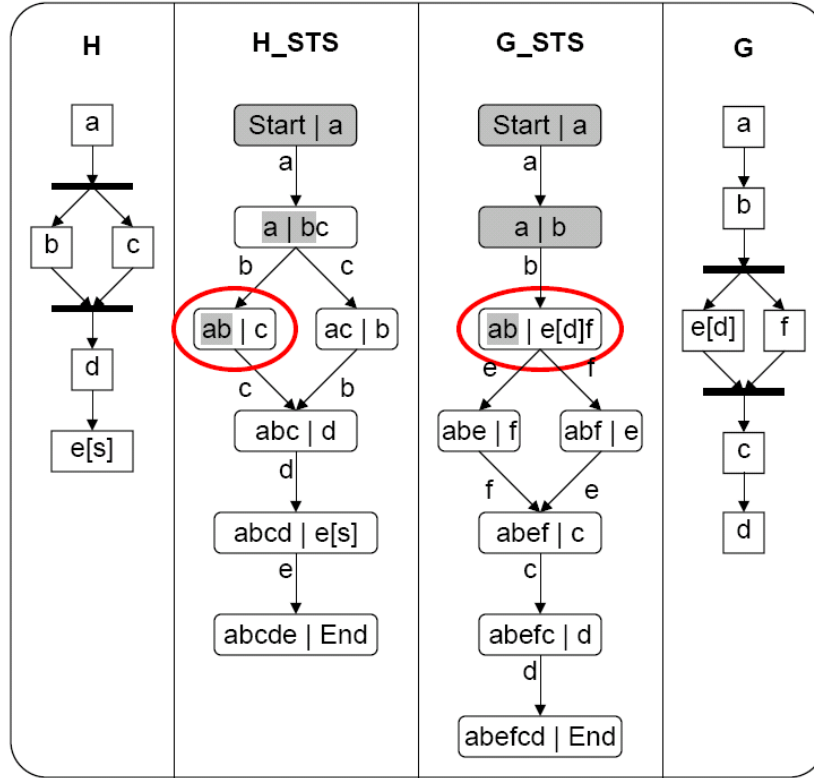


Figure 6-10. Example of discrepancy scene with CMIAS and CIAS.

The elements identified at a discrepancy scene will also be used to determine corresponding *destination area* in §6.4.1.

## 6.4 Reconciliation Option

In order to formulate reconciliation suggestions to reconcile any discrepancy, two factors need to be identified, namely *destination area* and *target activity*. Both of them are discussed in detail in this section before the explanation is given on how to formulate reconciliation options by combining the two.

### 6.4.1 Destination Area

Having known the commonality and discrepancy between two processes, the next step is to reconcile the discrepancy on the ground of the commonality. Due to the fact that behavioural discrepancies have their roots in control flow, the reconciliation should first target control flow aspect. For the convenience of further discussion, *destination area* is defined as follows.

Definition 6-8: In a given process, a destination area (DA) is the collection of all positions to which target activities of certain types are to be placed in order to reconcile the currently identified discrepancy. By position, it means a place between any two connected activities in a valid process diagram.

For the purpose of reconciling the discrepancy introduced in §6.3.3 Discrepancy Identification, the two-tailed reconciliation feature (explained in §5.2.1.3 Capturing Options) requires that target activities of certain types should be placed to neither before current common interface activity(ies) nor after the identified discrepant interface activity(ies). This requirement suggests the way of specifying a destination area, which involves two types of boundary activities in a given process diagram, namely upstream boundary activity (UBA) and downstream boundary activity (DBA).

In a process diagram, a UBA can be either a *current common proceeding interface activity* or one of the most upstream positioned activities that are in parallel with all current common proceeding interface activities of a discrepancy scene.

As to a DBA, similarly, it refers to either a *discrepant interface activity* or one of the most downstream positioned activities that are in parallel with all discrepant activities of a discrepancy scene, which are named as *downstream boundary activity type 1* (DBA Type 1) and *downstream boundary activity type 2* (DBA Type 2) respectively.

As a summary, Figure 6-11 depicts the concept of destination area in a UML class diagram, in which ‘<||CCPIAS’ means the most upstream positioned activity set that is in parallel

with all the activities in CCPIAS; ‘DIAS//>’ means the most downstream positioned activity set that is in parallel with all the activities in DIAS.

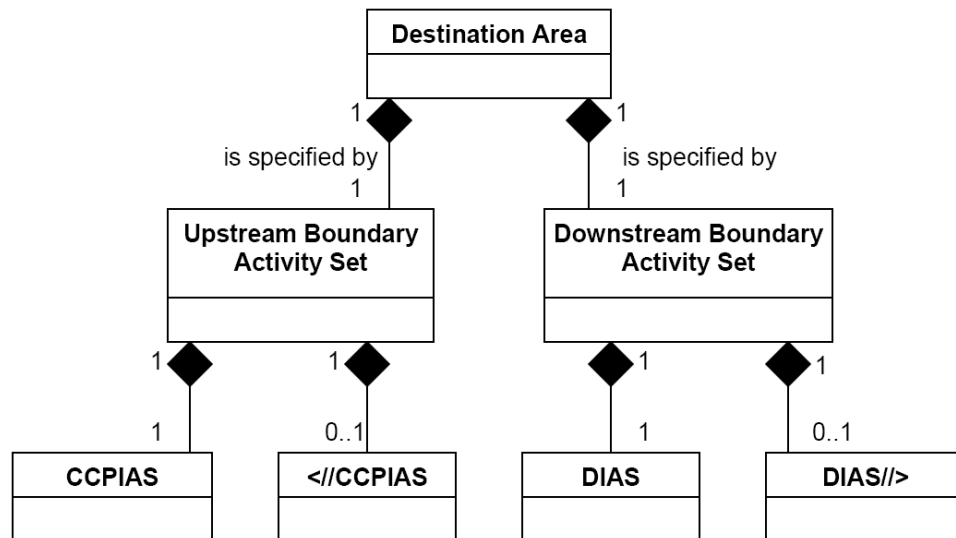


Figure 6-11. Destination Area in UML class diagram.

Positions in a destination area can be structurally continuous as well as discontinuous as illustrated in Figure 6-12 and Figure 6-13 respectively.

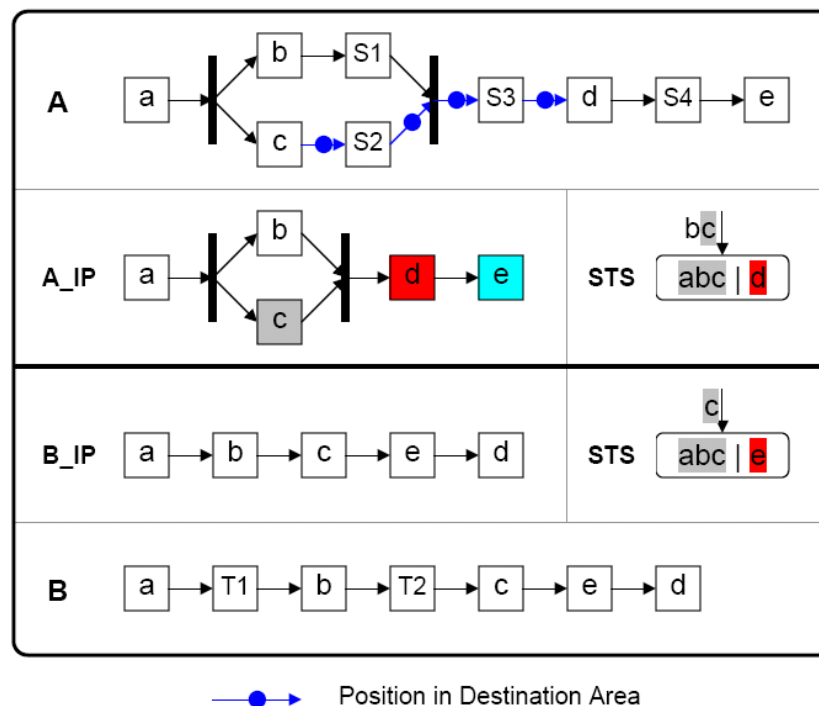


Figure 6-12. Destination area with continuous positions.

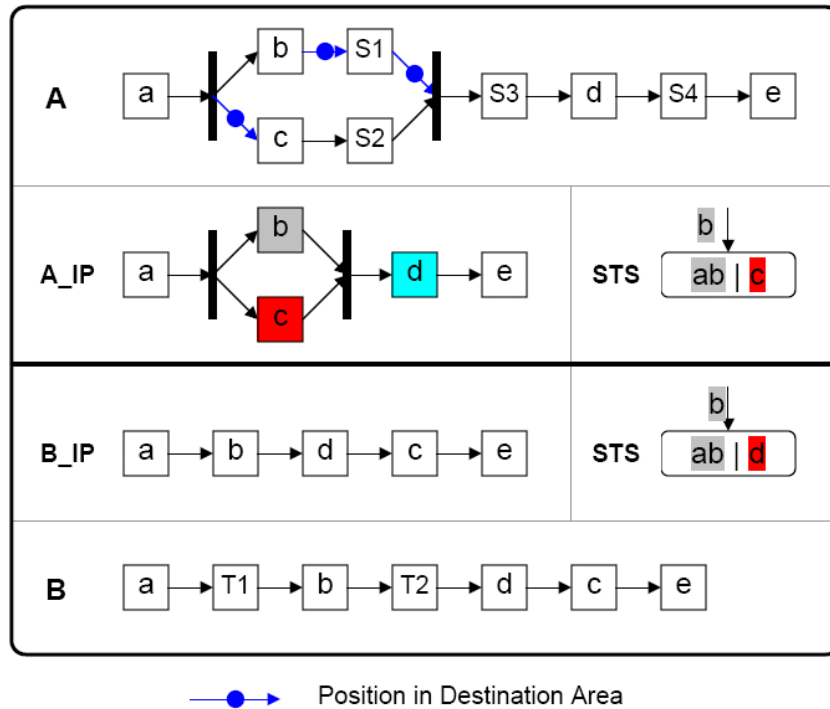


Figure 6-13. Destination area with discontinuous positions.

## 6.4.2 Target Activity

Target activities are identified based on the candidate adjusting interface activity set (CAIAS) introduced in §6.3.3 with inclusion of all relevant private activities associated with each CAIA through data dependencies. Target activity is addressed in association with the three types of CAIAs.

### 6.4.2.1 Candidate Removing Interface Activities and their Data Successors

According to Definition 6-5, a candidate removing interface activity is a unique interface activity found only in the Host's DIAS. Because only data demand type interface activities can survive the pre-treatment procedure (introduced in §6.3.1), a CRIA will be of data demand type for certain. This implies that among its control flow successors there must be some activity(ies) depending on the data once it has been delivered to the CRIA.

To reconcile the discrepancy caused by a CRIA, the CRIA needs to be removed from the Host's workflow definition. However, before doing that, its data successor(s) needs to be

visited and the data link(s) should be reviewed by users. If and only if users agree to discard all these data links connected to the CRIA, the CRIA can then be safely removed. Since the traverse of the activities associated with data flow provides users with complete information when faced with any CRIA-related situation, the data successors of a CRIA should be treated as target activities together with the CRIA.

#### **6.4.2.2 Candidate Moving Interface Activities and their Data Predecessors**

When identifying a candidate moving interface activity (CMIA) as a target activity, all relevant private data predecessors associated with the CMIA need to be identified as target activities as well for the purpose of conducting complete adjustment in the initial workflow definition.

##### **Candidate Moving Interface Activity (CMIA)**

According to Definition 6-6, since a candidate moving interface activity (CMIA) has already existed in the Host's process, the only concern is its position. Being moved into an appropriate place will contribute to the reconciliation of the current discrepancy.

In order to get the Host's CMIAS, activities in the Guest's DIAS are examined. The activities with common message semantics are picked out. They will then be mapped onto the counterpart activities in the Host's process, among which the most upstream positioned interface activity(ies) on each independent route will be identified and included in the Host's CMIAS. By following the upstream independent rule in identifying CMIA(s), only minimum reconciliation requirements and least complexity will be brought into to a negotiation process, which reflects the lean principle.

The examples in Figure 6-14 illustrate how CMIAs are identified. In process *H1*, counterpart interface activities of *G*'s DIAS are *H1.e* and *H1.f*. Since *H1.e* is the predecessor of *H1.f*, i.e. they are not independent, only the most upstream positioned activity *H1.e* is included in *H1*'s CMIAS. Although the comparison between process *H2* and *G* reveals the same discrepancy scene, since *H2.e* and *H2.f* are in parallel in process *H2*, meaning they are independent, both *H2.e* and *H2.f* are included in *H2*'s CMIAS.

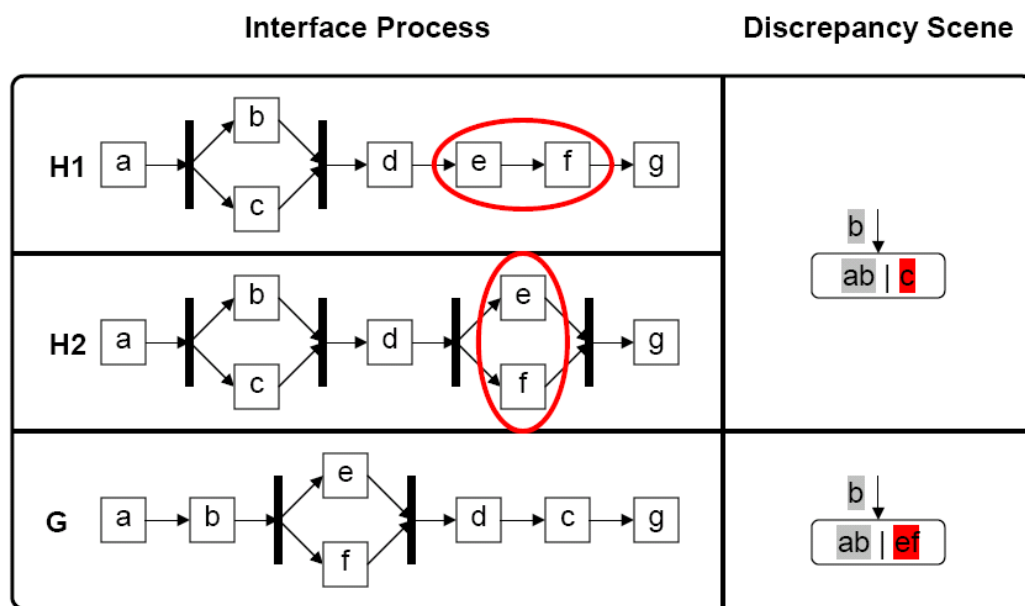


Figure 6-14. Identifying CMIAS with upstream independent rule.

### Candidate Moving Activity (CMA)

It is by the destination area and candidate moving interface activities that a given process is divided into three sections: before the destination area (I), after the destination area but before candidate moving interface activities (II) and after candidate moving interface activities (III), which is shown in Figure 6-15. For Section I, as either it has not caused any discrepancy or the discrepancy caused by the section has already been dealt with previously, it is no longer a section of interest. As to Section III, since the issue whether or not it will have something to do with any future discrepancy has not been touched and revealed by the current discrepancy scene, it will not be considered until a later stage. Due to Section II sits right in between the destination area and the CMIAS, the dynamics of the activities in this section need to be investigated and understood in relation to corresponding reconciliation suggestions.



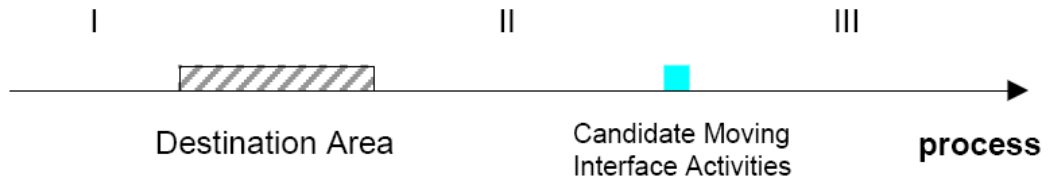


Figure 6-15. A process divided by destination area (DA) and candidate moving interface activities (CMIA).

In Section II as shown in Figure 6-15, there might well be activities acting as data predecessors of a particular CMIA. If only CMIA's are prompted and moved into the destination area but their data predecessors within Section II are left unattended, relevant data dependency would be violated, which makes the adjustments meaningless. Therefore, in order to capture all the aspects with regard to a particular reconciliation suggestion, candidate moving interface activities need to be propagated to embrace all the activities within Section II that appear in the data flow of each candidate moving interface activity. Each activity in the extended target activity set should then be prompted to users. The extended target activity set is called *candidate moving activities set* (CMAS) in this thesis. Figure 6-16 depicts the necessity of extending the CMIA's to include other candidate moving activities. It can be seen, in Host's process,  $H4$  depends on the data produced by  $H3$  and  $H.e$  depends on the data produced by  $H4$ . Since they are all located in Section II, they are identified as candidate moving activities and will all be prompted to the user. By doing so, the data dependency between  $H3 \rightarrow H4$  and  $H4 \rightarrow H.e$  will be explicitly exposed to user scrutiny and subject to user decision making, which will otherwise be hidden from the user.

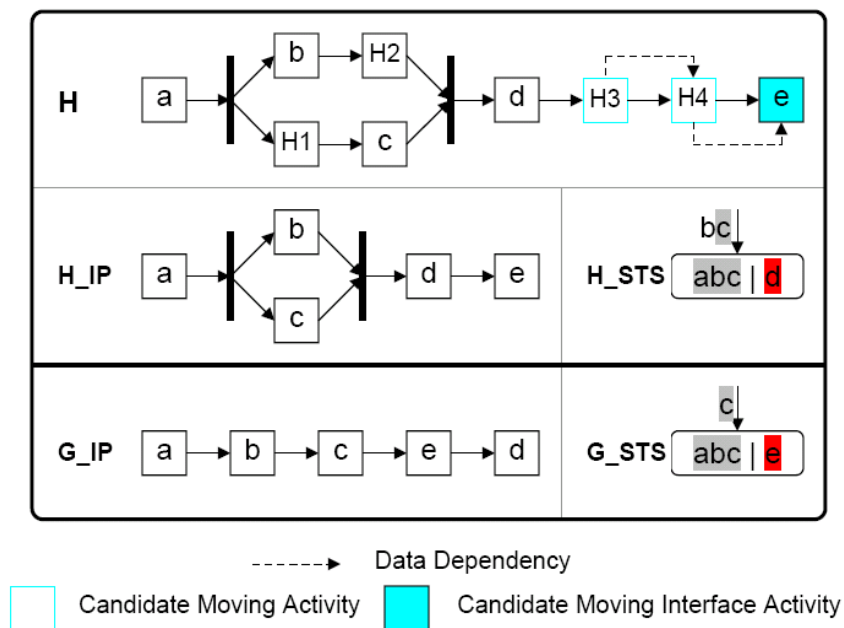


Figure 6-16. Necessity of extending CMIAS.

However, there is one exception, which is when a candidate moving activity turns out to be a data demand interface activity. In this case, the decision should be automatically made as ‘reject the control flow adjustment’ because by moving such an interface activity upstream into the destination area it will still be positioned before corresponding candidate moving interface activities, which results in an ineffective move. This will be discussed in §6.4.6.3 Manual and Automated Decision Making together with other circumstances where automated adjustment is appropriate.

### Algorithm for Capturing Candidate Moving Activity Set

The algorithm for capturing candidate moving activity set is shown in Figure 6-17.

```

getCMAS(AD_Wf, CMIAS): CMAS

CMAS = createList();
FOR each v in AD_Wf
    IF v is of DBA Type 1
        FOR each cmia in CMIAS
            IF isDataPred(v, cmia, AD_Wf) //v precedes cmia in DF

```

```

                                CMAS.add(v)
                                BREAK
ELSE
    OuterLoop:
    FOR each dba in DBA Type 1 and 2
        IF isControlSucc(v, dba, AD_Wf) //v succeeds dba CF
            FOR each cmia in CMIAS
                IF isDataPred(v, cmia, AD_Wf)
                    CMAS.add(v)
                    BREAK OuterLoop

```

Figure 6-17. Algorithm for capturing candidate moving activity set.

#### **Algorithm for Associating CMAs with their CMIA**

The algorithm for associating CMAs with their CMIA is shown in Figure 6-18.

```

associateCMAwithCMIA(CMIAS, CMAS, AD_Wf): update CMIAS

FOR each cmia in CMIAS
    decisionNodes = createList();
    FOR each cma in CMAS
        IF isDataPred(cma, cmia, AD_Wf)
            decisionNodes.add(cma)
    decisionNodes.add(cmia)
    cmia.decisionNodes = sort_CF(decisionNodes, AD_Wf)

```

Figure 6-18. Algorithm for associating CMAs with their CMIA.

#### **6.4.2.3 Candidate Inserting Interface Activities and their Data Generation Activities**

With user consent, a candidate inserting interface activity (CIIA) is created in the Host's process for the purpose of communicating relevant data demanded by the Guest's process. At this stage, since the data that a CIIA depends on is not available in the Host's process, the data generation activity itself needs to be created as well and inserted together with the CIIA if the user on the Host's side wants to conform to the Guest's demand.

### 6.4.3 Hypothetical Target Position

A hypothetical target position (HTP) denotes the position in a destination area (DA), to which target activities of certain types should be placed to reconcile the current discrepancy. However, given the homogeneity of all the positions in a DA in discrepancy reconciliation, the next question is, “Within a destination area are all the positions equal in terms of the impact on the initial process if each of them is chosen as a hypothetical target position (HTP)?” The question is answered from two perspectives, namely sequence and connectivity.

#### 6.4.3.1 Sequence Perspective

In a process, due to the causal relationship between neighbouring activities from the start to the end, as long as an activity is moved upstream (towards the starting point), control flow dependency is inevitably violated. However, due to the fact that the successful completion of a process is realised by incremental completion of each activity in the process, it is discovered that the further an activity is moved upstream, the more likely the resulting control flow violation cannot be tolerated. Using the classic ‘make a cup of tea’ process as an example (shown in Figure 6-19), if the task *Dispose The Tea Bag* is moved upstream, unless it is put before *Pour In Boiled Water*, a cup of tea can still be made, although the result becomes increasingly less ideal. Therefore, if a task must be moved upstream, keeping its target position as downstream as possible can increase the chances of tolerance regarding control flow violation.

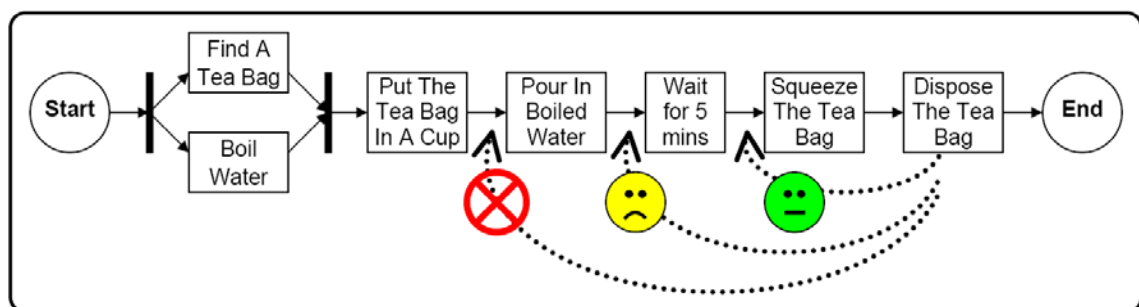


Figure 6-19. Can we still get a cup of tea?

Also, the further an activity is moved upstream, the more likely data flow dependency is violated. Therefore, if possible, keeping hypothetical target positions as downstream as possible can also mitigate the possibility of data dependency violation, hence resulting in

less impact on the initial process. As Figure 6-20 illustrates, after interface process comparison, four *candidate hypothetical target positions* (CHTP) are identified. From the data dependency aspect, in the destination area, candidate moving interface activity *H.e* could depend on any data produced by activities that sequentially precedes it in control flow, which in this case are *H.b*, *H.c*, *H2* and *H3*. Among all the CHTPs, the only position that will not cause any potential data flow violation in the destination area is CHTP4.

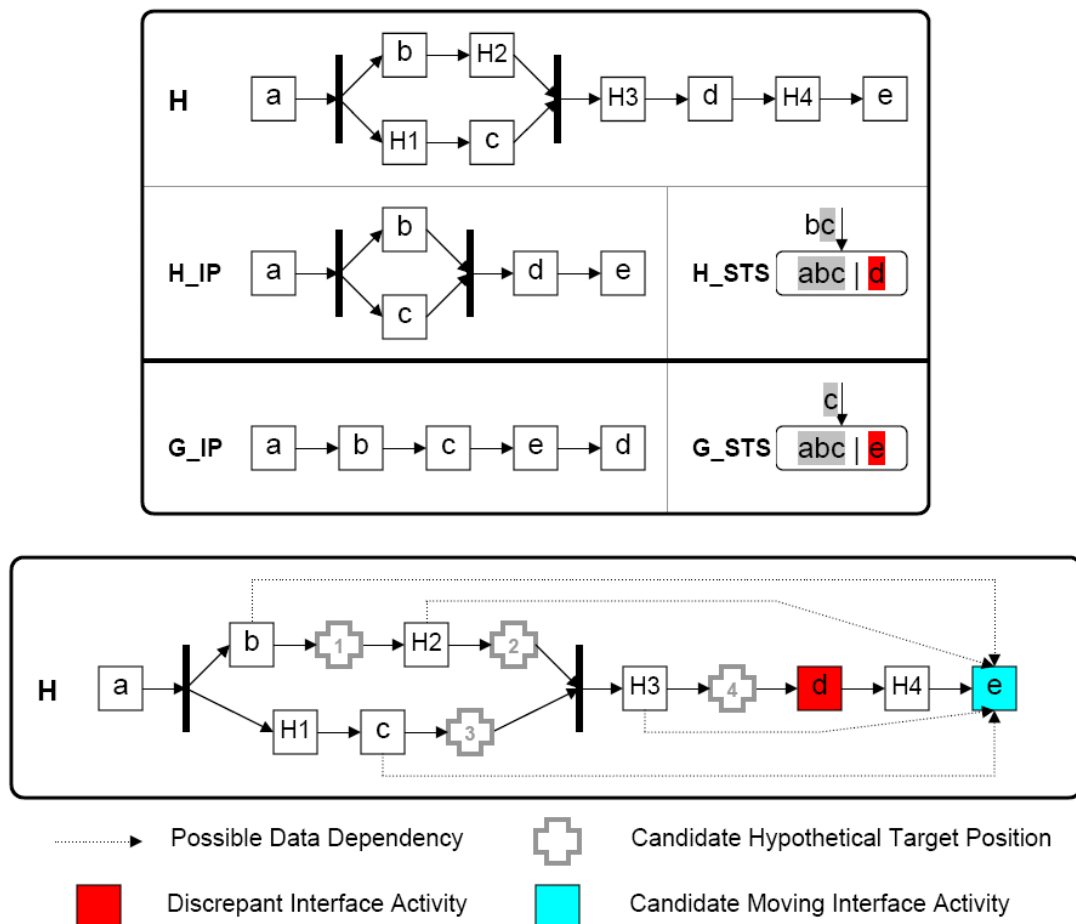


Figure 6-20. Choice of hypothetical target position.

From the explanation, it can be seen that hypothetical target positions should be located next to downstream boundary activities, which are the most downstream positions available within a given destination area. The question now becomes “How to identify and express these positions in the form of the two types of downstream boundary activities?” For a given DBA Type 1, it can be seen that the position ‘right before’ it is inside the destination area

and is the most downstream position of its kind. As to a given DBA Type 2, the position ‘right after’ it would be the most downstream position inside the destination area. By using the word ‘right’, it means the position is sequentially immediately next to a downstream boundary activity. The two types of hypothetical target positions are illustrated in Figure 6-21. At the current discrepancy scene, *H.c* is identified as DBA Type 1 whilst *H2* is identified as DBA Type 2. Corresponding HTPs are identified as ‘right before *H.c*’ (*HTP1*) and ‘right after *H2*’ (*HTP2*).

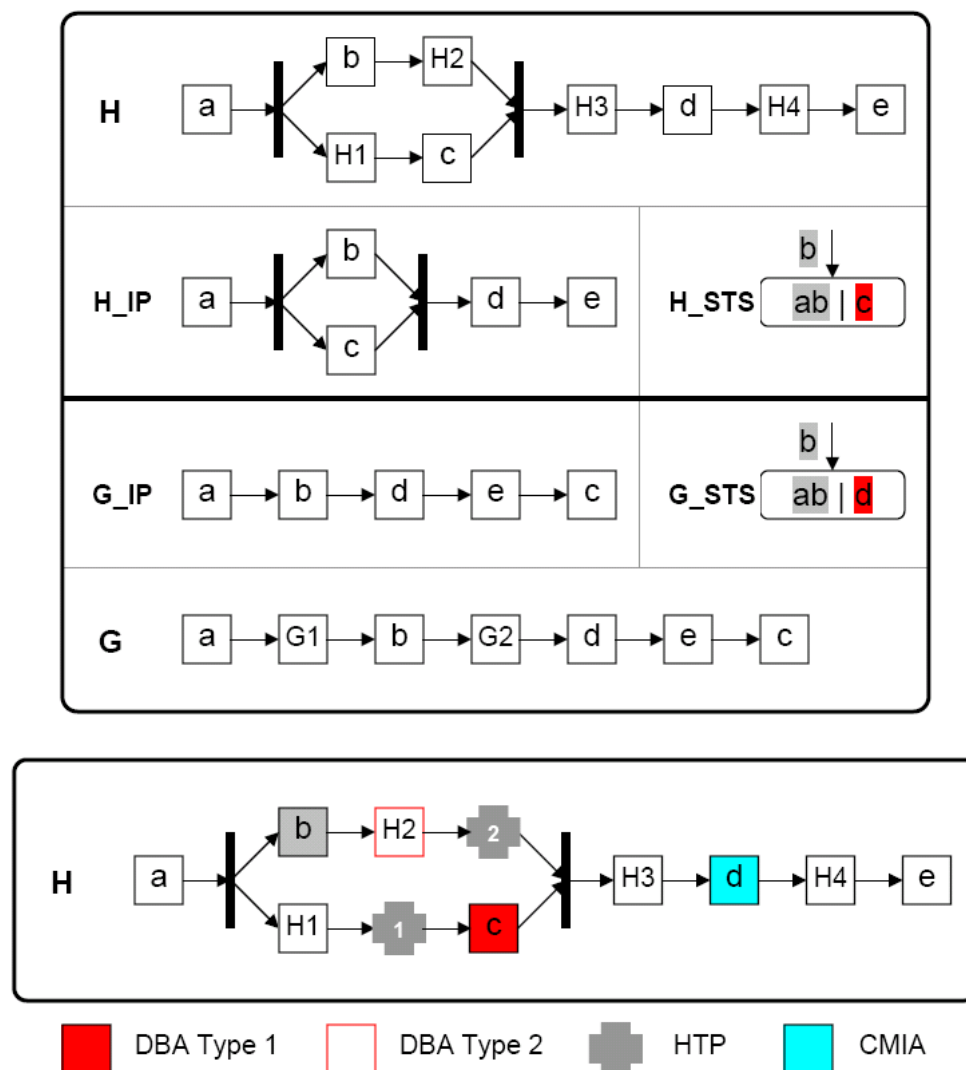


Figure 6-21. DBA Type 1 and 2 and their HTPs.

### 6.4.3.2 Connectivity Perspective

In line with the minimum impact principle, another issue is to logically pair each candidate moving activity (CMA) with its corresponding hypothetical target positions (HTP). This is carried out based on the initial control flow connectivity specified in an initial workflow definition. It is assumed that if control flow connectivity exists between two activities in an initial workflow definition, unless the user explicitly decide otherwise, the connectivity should be maintained after any suggested adjustment. In order for the user to have the flexibility in changing the initial control flow connectivity arrangement when necessary, such connectivity information for each candidate moving interface activity should be prompted to the user and let them make decisions on whether they want to keep the connectivity. In Figure 6-22, candidate moving activities are identified as *H7*, *H9* and the candidate moving interface activity is *H.e*. According to the definition in sequence perspective, positions of ‘right before *H.c*’, ‘right after *H6*, *H11* and *H12*’ are identified as the HTPs. By taking connectivity perspective into account, HTPs for each candidate moving activity can be worked out as shown in Table 6-1.

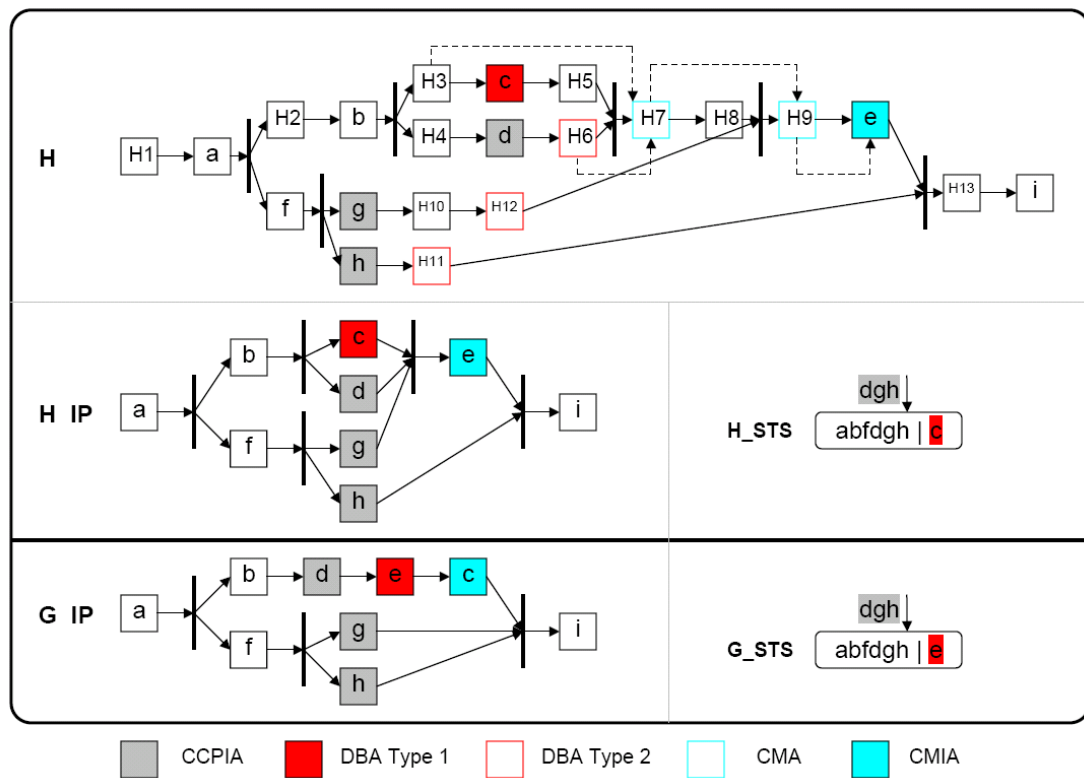


Figure 6-22. Connectivity perspective – CMA/CMIA and HTP(s).

Table 6-1. CMA/CMIA and their HTPs.

	<b>CMA</b>		<b>CMIA</b>
<b>HTP</b>	<i>H7</i>	<i>H9</i>	<i>H.e</i>
right before <i>H.c</i>	√	√	√
right after <i>H6</i>	√	√	√
right after <i>H11</i>	–	–	–
right after <i>H12</i>	–	√	√

From the table it can be seen that each CMA/CMIA is paired with relevant HTPs. Because there is no connectivity between the HTP ‘right after *H11*’ and any of the CMA/CMIAs, the HTP is not associated with any CMA/CMIA.

For any currently identified CIIA, since it does not exist in an initial workflow definition, there is no connectivity perspective for it to refer to. It is decided that all the current HTPs are applicable to each of the currently identified CIIA.

#### 6.4.3.3 Effective Adjustment with Minimum Impact

Control flow-driven adjustments aim to reconcile any behavioural discrepancy without causing any new difference. Control flow connectivity testing tries to contain the changes within a necessary range so that the adjustment has as little impact as possible on the hosting process. Combining both measures, the definition of *effective adjustment with minimum impact* summarises the process adjustment requirements for a reconciliation option should it is accepted.

Definition 6-9 Effective Adjustment with Minimum Impact denotes an adjustment on a hosting workflow that can effectively reconcile the currently identified discrepancy and has minimum impact on initial control/data flow dependencies.

By observing this guideline corporate knowledge and asset can be preserved as much as possible.

#### 6.4.4 Information Prompting

Information prompting is crucial to support a user’s decision-making process, which links all the key factors mentioned above in this section. Any discrepancy and options needs to be



presented to a user in an understandable format and provides appropriate means of interaction. As stated in §4.2.2.3 Decision Making Support, sufficient information in terms of reconciliation options and resulting implication on both control flow and data flow needs to be prompted to the user in an interactive and timely manner. Due to the stages involved in the reconciliation process and the natures of the target activities, the information required to be prompted to the user is different.

#### **6.4.4.1 Option at Activity Bipartite Comparison Stage**

At activity bipartite comparison stage (or pre-treatment stage for process comparison stage), whenever a unique data demand interface activity (UDDIA) is encountered in the Guest's interface process, the data demanded needs to be searched in the Host's workflow definition. If a match is found, it means that the Host's process has already generated such data for internal use. However, whether it is permitted for the data to be shared with the Guest is unclear. If the data can be shared, a data supply interface activity will be inserted right after the data generation activity. Therefore, the Guest's UDDIA is no longer unique and is converted into a common interface activity. If the user decides not to share the data, current round of negotiation on the Host's side terminates. Such an option prompt takes the form of:

“Is it allowed for the data to be passed to the partner?”

If no match is found, no option will be prompted at this stage. The UDDIA will be recorded for later use.

#### **6.4.4.2 Reconciliation Option for CRIA**

Reconciliation options in association with a candidate removing interface activity (CRIA) only exist on the data flow dimension. The decision on whether to remove the CRIA from control flow can be deduced from the decisions made on all data flow options and is revealed by real-time messages.

#### **Formulation of Data Flow Reconciliation Option**

For a CRIA, since the decision on whether to have it removed depends on the decision(s) on whether to discard the data links between the CRIA and its data successor(s), the reconciliation options is formulated as:

“Data dependency from [*current CRIA*] to [*a data successor of the current CRIA*] should be discarded.”

The decision on the CRIA will then be automatically deduced based on the decisions made by users on each of the data successors.

### **Real-Time Interactive User Prompt**

Based on the decision having been made on the data flow, either of two implication messages will be prompted instantly. Firstly, when all the options are accepted, the message shown is:

“Based on current decisions, [*current CRIA*] can be removed”.

Secondly, if at least one of the options is rejected, the message will appear as:

“Based on current decisions, [*current CRIA*] cannot be removed”.

The decisions at this stage are provisional, which allows the user to evaluate all the implications on data flow and provides the user with the chance to change their decisions if necessary. When the user is satisfied with their decisions, they can proceed to commit these decisions.

#### **6.4.4.3 Reconciliation Option for CMA/CMIA**

Reconciliation options in association with a candidate moving activity (CMA) exist on both control flow and data flow. Relevant real-time messages are provided to facilitate user decision-making tasks. The handlings of candidate moving interface activities (CMIA) are similar to those of CMAs except that those in relation to CMAs’ data successors are not applicable to CMIA as in a CMA/CMIA set a CMIA is the last activity in the data flow.

### **Formulation of Control Flow Reconciliation Option**

Before any control flow reconciliation option can be prompted to the user, it needs to be formulated based on the information including the current CMA and its corresponding hypothetical target positions (HTP) in the destination area. The generic format is:

“[*current CMA*] should be put at [*a HTP*]”.

When the target position is one of the identified HTPs, a control flow reconciliation option is formulated in one of the following two forms corresponding to each of the two types of downstream boundary activities (DBA):

- “[*current CMA*] should be put right before [*DBA Type 1*]”,
- “[*current CMA*] should be put right after [*DBA Type 2*]”.

However, for situations where there is any previously accepted CMA positioned as an immediate control flow predecessor of the current CMA in the current CMA/CMIA set, to echo the user’s previous decision at real time, any control flow reconciliation option of the current CMA that shares the same HTP as the previously accepted CMA will be prompted as a ‘Recommended’ option and is formulated as one of the following two forms:

- “[*current CMA*] should be put right before [*DBA Type 1*] – **Recommended**”,
- “[*current CMA*] should be put right after [*DBA Type 2*] – **Recommended**”.

This is because by accepting the recommended options, initial data dependency between CMAs can be preserved. However, it is still up to the users whether to follow the suggestion based on their own judgment.

### **Formulation of Data Flow Reconciliation Option**

If all the control flow reconciliation options regarding a CMA are rejected, data flow reconciliation options need to be prompted as:

“Data dependency from [*current CMA*] to [*current CMA’s data successor in current CMA/CMIA set*] should be removed”.

### **Real-Time Interactive User Prompt**

For each CMA, as its HTPs are independent (i.e. in parallel with each other in terms of control flow), each control flow reconciliation option occupies a single entry. The user can make individual decision on each option entry as either ‘accept’ or ‘reject’ based on their knowledge, business rules and organisational policies. Accompanied by the user’s current decisions, relevant implication messages on control flow and data flow are displayed in real time to fully reveal the decision and option dynamics.

Two types of control flow implication messages (CFIM) are identified, namely CFIM-1 and CFIM-2.

### *CFIM-1*

If a decision on a certain control flow reconciliation option implies the same type of decision on other option(s), the message takes the form of:

“Decisions on options with regard to [*implied HTP(s)*] should be in consistency with the current decision on the option regarding [*HTP of current option*]”.

### *CFIM-2*

Following the rejection of all available control flow options, data flow options regarding the current CMA and its data successors are prompted. When any of the data flow option is rejected, CFIM-2 is displayed in the form of:

“In order to avoid further control flow restrictions, all the above options need to be accepted. Otherwise, [*rejected data successor of the current CMA*] will be set as a non-moveable activity if current decision(s) are confirmed.”

Three types of data flow implication messages (DFIM), namely DFIM-1, 2, 3, are identified for two situations. The first situation is where at least one control flow reconciliation option is accepted and certain data dependency is violated as a direct result of rejecting some of the control flow options.

### *DFIM-1*

If the violated data dependency is between the current CMA and its data predecessor located before Section II (mentioned in §6.4.2.2), DFIM-1 takes the form of:

“As a prerequisite, data dependency from [*a data predecessor of current CMA*] to [*current CMA*] needs to be removed based on current decision(s)”.

### *DFIM-2*

If the violated data dependency is between the current CMA and any provisionally accepted CMA that immediately precedes the current CMA, DFIM-2 is shown in the form of:

“With regard to the [*provisionally accepted CMA*]: As a prerequisite, data dependency from the [*provisionally accepted CMA*] to [*current CMA*] needs to be removed based on current decision(s)”.

### DFIM-3

The second situation is where all the control flow reconciliation options have been rejected and the current CMA itself is barred from moving. DFIM-3 appears as:

“In order for the negotiation to carry on, since none option is accepted, data dependency from [*current CMA*] to [*current CMA's data successors*] should be removed if current decision(s) is confirmed”.

The decisions at this stage are still provisional, which allows the user to evaluate all the implications on control flow and data flow and provides the user with the chance to change their decisions if they think necessary. When the user is happy with their decisions, they can proceed to commit them.

In all cases, real-time control/data flow implication is provided based on current users' decisions. Once decisions are committed by the user, the relevant results and implications are recorded and will be referred to in subsequent stages of reconciliation options prompting.

#### 6.4.4.4 Reconciliation Option for CIIA

Reconciliation options in association with a candidate inserting interface activity (CIIA) exist on control flow. Relevant real-time messages are provided to facilitate user decision-making.

#### Formulation of Control Flow Reconciliation Options

Due to the data dependency between the CIIA and the underlying data generation activity as its data predecessor, the two activities will appear as a group in the form of ‘data generation activity → CIIA’. Since the group has never existed in the Host's process, there is no control flow constraint from the HTPs on the inserting group. Full flexibility is granted to the user by providing the options of inserting the group at all available HTPs. The following two forms correspond to the two types of downstream boundary activities (DBA):

- “[*current inserting group*] should be put right before [*DBA Type 1*]”,
- “[*current inserting group*] should be put right after [*DBA Type 2*]”.

### **Real-Time Interactive User Prompt**

When at least one of the control flow reconciliation options is accepted, the inserting group will be inserted and an implication message will appear as “New activities will be inserted”. Otherwise, when no option is accepted, the inserting group will not be inserted and an implication message will appear as “No activity will be inserted”. The decisions at this stage are still provisional, which allows the user to evaluate all the implication and provides the user the chance to change their decisions if necessary. When the user is happy then they can proceed to commit these decisions.

#### **6.4.4.5 Sequence of Reconciliation Options**

For each negotiation cycle, if there are any options available at the activity bipartite comparison stage, they are prompted to the user first. Then comes the process behaviour comparison. If any discrepancy is encountered at this stage, options available are to remove, move or insert certain target activities to reconcile currently identified discrepancy. This may introduce new activities to, or detach existing activities from, the process definition. Due to the fact that the number of activities has a direct correlation to the complexity of the hosting process, which could affect the effort required from the user during the course of decision-making, the sequence of the options should be carefully arranged in order to keep the process complexity at its lowest possible level at any time to minimise the complexity involved. This point reflects the lean principle and is explained through the following examples.

In the interface processes depicted in Figure 6-23, it is assumed that the unique data demand interface activity of the Host,  $H.c[d]$ , is to be removed. Although the end results will be the same, different sequences of option prompting result in different courses and user experiences. If the moving option is prompted before the removal option with regard to  $H.c[d]$ , it will take two more extra reconciliation cycles (including process comparison, option prompt, decision-making and process adjustment – ‘move  $d$  right before  $c$ ’ and ‘move  $e$  right before  $c$ ’) at  $H$  before  $H.c[d]$  can be finally removed. These steps can be avoided if the sequence of the option prompt is rearranged.

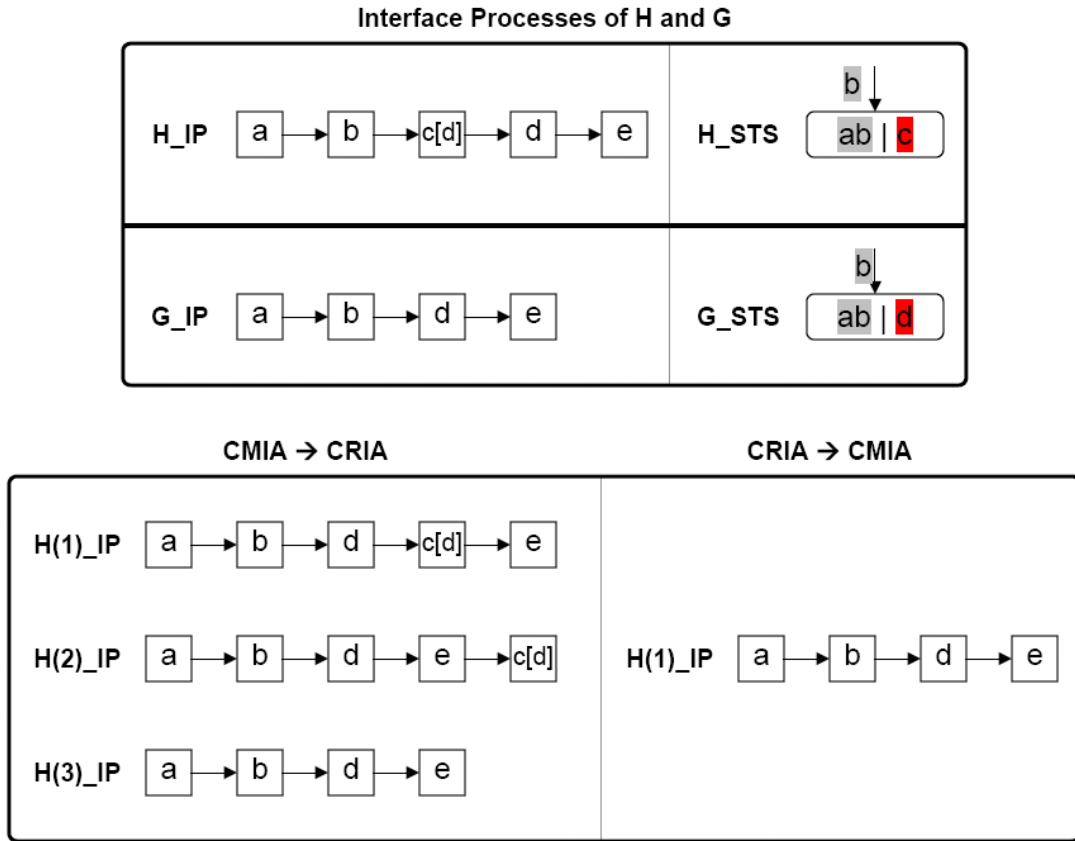


Figure 6-23. Sequence of reconciliation options: CRIA → CMIA.

When an interface activity needs to be inserted into a process, it will always be of data supply type in order to meet the partner's corresponding data demand requirement. Since the data to be sent needs to be generated in the process, the data generation activity needs to precede the data supply interface activity in the control flow. However, these could add unnecessary complexity for later options.

Consider Figure 6-24 (a), assuming that  $c[s]$ 's data generation activity  $X$  does not exist in  $H$  previously, Figure 6-24 (b) shows that following the interface process comparison, with users' consent, after  $H.X$  and  $H.c[s]$  are created and inserted, data dependency in association with  $H.X$  should be reviewed and specified by the user. It is decided a new data dependency needs to be added as  $H.X \dashv\vdash H.I$ . Then, after the next interface process comparison, the discrepancy is encountered and the option is prompted as ' $H.I$  should be put right before  $H.d$ ;  $H.I$  should be put right before  $H.f$ '. If due to some other concerns the user decides to

reject Option[1] whilst accept Option[2], the data dependency  $H.X \rightarrow HI$  is violated as shown in Figure 6-24 (c). Although with the real time prompt, such a violation can be detected, it adds unnecessary overhead to the user's decision-making process. This is all because additional activities are introduced earlier than they need to be. If both  $c[s]$  and  $X$  are inserted after the decisions on  $HI$  and  $e[s]$  then user will not experience self-contradictory options with regard to the data flow specification.



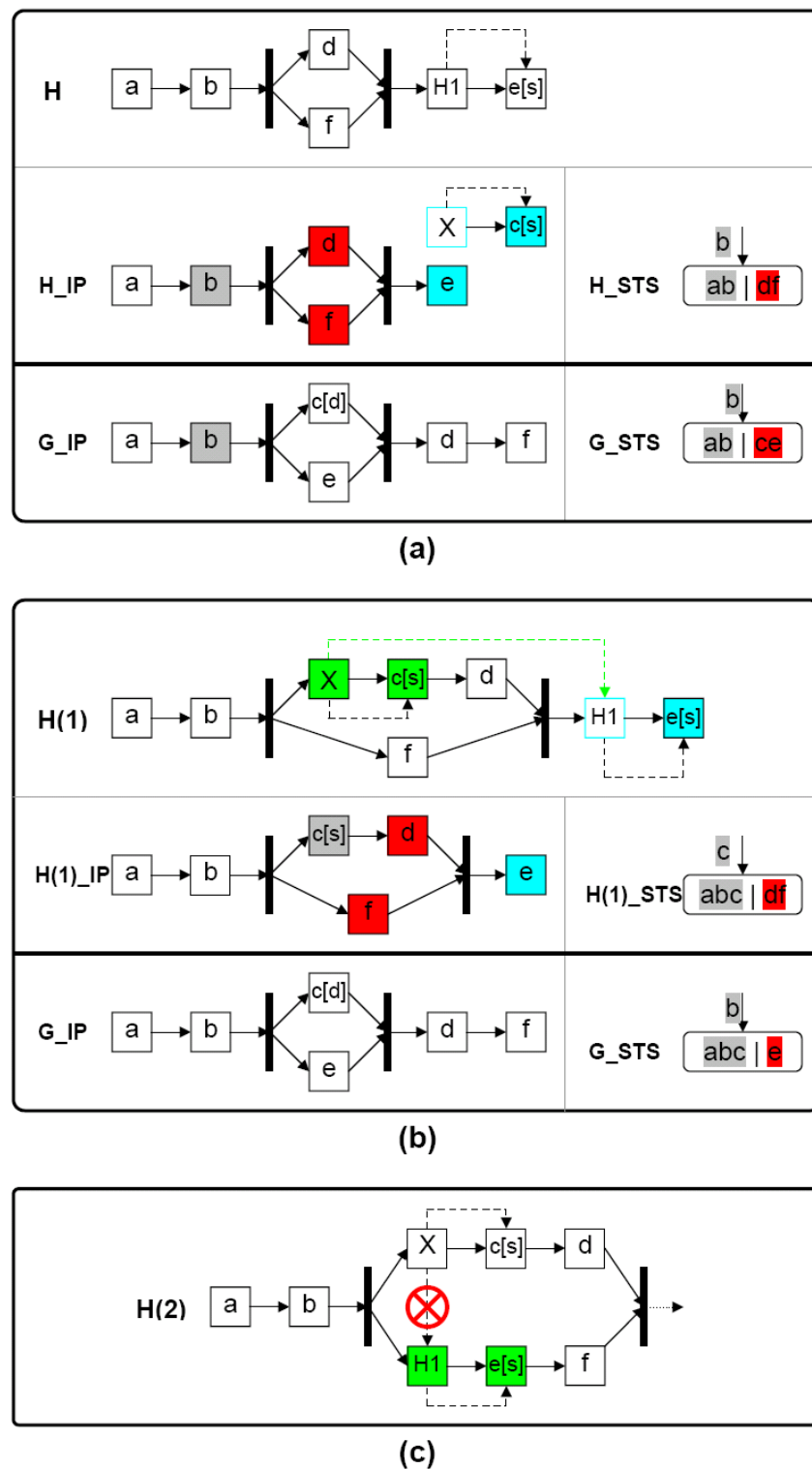


Figure 6-24. Sequence of reconciliation options: CMIA → CIIA.

Therefore, for options involving multi-independent candidate adjustment interface activities (CAIA), the sequence of option prompting should follow the types of involved CAIAs. It is

decided that CRIA-related options has the highest priority to be prompted and dealt with followed by CMIA-related options, and lastly CIIA-related ones. The underlying reasons for this arrangement are: firstly, since an option with regard to a CRIA represents an opportunity to remove an interface activity by users on the Host's side, it has the potential to reduce the number of activities and thus the complexity of the hosting process. Secondly, since CMIA related options would not lead to any change in process complexity in terms of the number of activities, they have the second priority. Lastly, due to options targeting CIIAs have the potential to add new interface activities and/or other related information generation activities, which adds the complexity into the hosting process, they are the last batch of options to be prompted to users after all other activities have been decided and settled down at a given discrepancy scene.

### **6.4.5 Option and Decision Dynamics**

Due to the control flow and data flow dependencies between activities in a process, the user's decision (or combination of decisions) may influence other decisions or even educe new options. Such interactions are named as *option and decision dynamics* in this thesis. They are centred on the three types of target activities: CRIA, CMA/CMIA and CIIA.

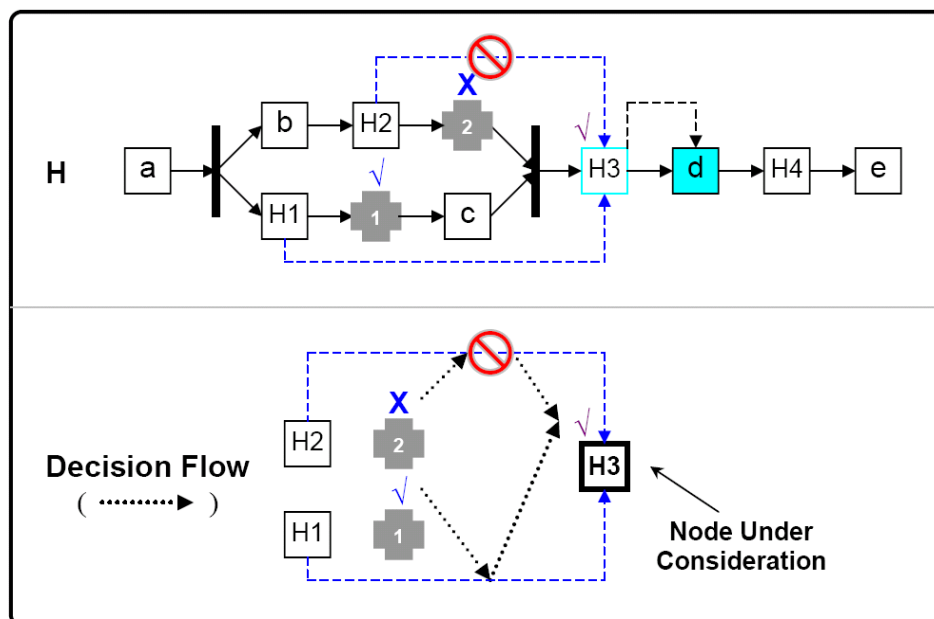
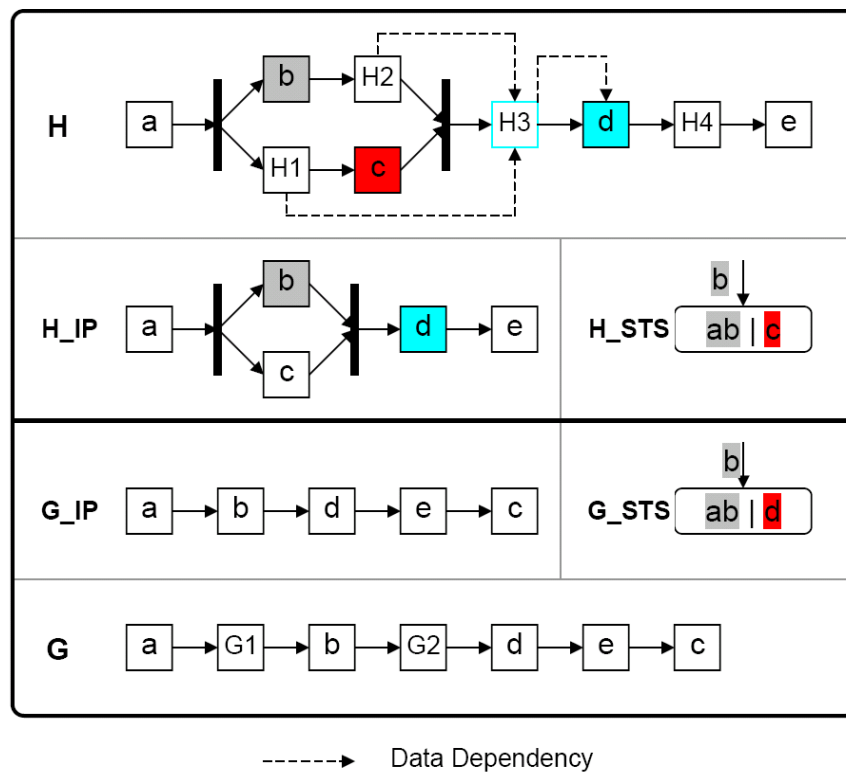
#### **6.4.5.1 CRIA**

The decisions on whether to retain the data links between a CRIA and all its data successors determine whether the CRIA itself can be safely removed. As a prerequisite, only if none of initial data successors has ever depended on the CRIA, the CRIA can be freed and removed. Otherwise, the CRIA must stay.

#### **6.4.5.2 CMA and CMIA**

Things become more complicated for CMA and CMIA. So far, all the reconciliation options have targeted control flow. What will happen if any control flow-oriented option is rejected? Two types of scenarios should be differentiated and considered. Firstly, if not all control flow reconciliation options are rejected in relation to corresponding candidate moving activities, for each rejected control flow option with regard to a particular HTP, a data flow violation might arise. This is because when one of the control flow connectivity between the CMA/CMIA and the HTP is no longer retained, if the data dependency between the current CMA/CMIA and either any of its data predecessors located before Section II (refer to

§6.4.2.2) or a provisionally accepted CMA that immediately precedes the current CMA/CMIA is attached to the same rejected control flow, the data dependency will be violated. As shown in Figure 6-25,  $H3$  depends on data generated by  $H1$  and  $H2$  and  $H.d$  depends on the data generated by  $H3$ . When interface process  $H$  is compared with  $G$ ,  $H3$  and  $H.d$  are identified as CMA and CMIA respectively for  $H$ . If  $HTP2$  is rejected as one of  $H3$ 's target positions, data dependency between  $H2$  and  $H3$  can no longer be retained and thus violated, which needs to be brought to the user's attention.



Users' decision on control flow (✓, X) Decision deduced on control flow (✓, X)

Users' decision on data flow ( retain -----> , remove -----> )

Figure 6-25. Data flow implication between current CMA/CMIA and its data predecessor before DBA when not all options are rejected.

After  $H3$  is accepted, during the decision-making for  $H.d$  with regard to  $HTP1$  and  $HTP2$  as shown in Figure 6-26, if  $HTP2$  is accepted but  $HTP1$  is rejected, data dependency between  $H.d$  and the provisionally accepted node  $H3$  is violated. The user needs to decide whether to discard the data dependency between  $H3$  and  $H.d$  or adjust their decisions in order to retain the data dependency.

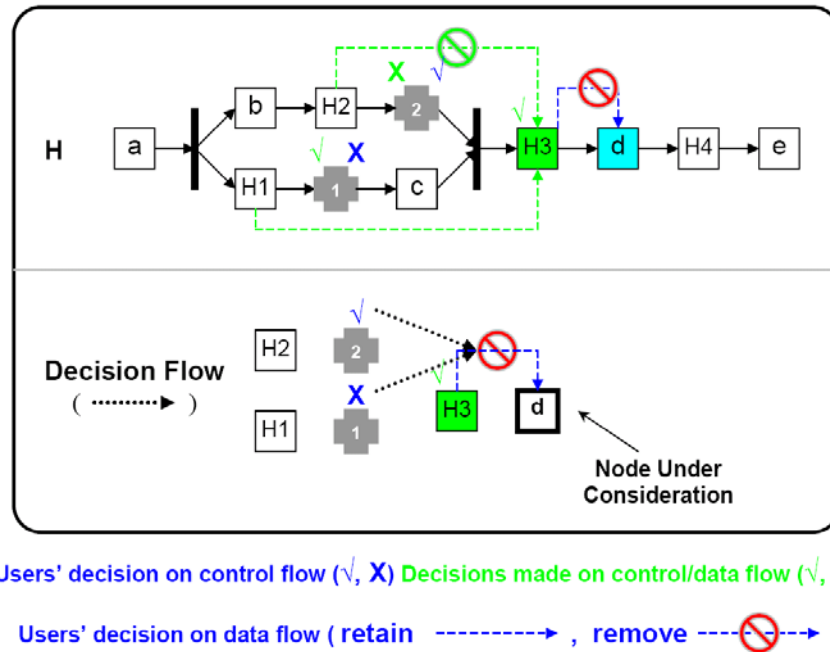


Figure 6-26. Data flow implication between current CMA/CMIA and immediately preceding CMA that was provisionally accepted when not all options are rejected.

Secondly, for a candidate moving activity, if all the control flow reconciliation options are rejected, as a result the candidate moving activity itself will be barred from moving. The data dependency between the rejected CMA and its data successors (if there is any) should be checked and prompted to the user for them to be aware of any potential impact on the next stage of decision-making. If it is decided to retain the data dependency, the associated data successor should be automatically marked as 'non-moveable'. This is shown in Figure 6-27. The same example is used as in Figure 6-25 but this time, all options with regard to  $H3$  are rejected which results in  $H3$ 's rejection. Since  $H.d$  is  $H3$ 's data successor and the decision regarding the data dependency is made as to retain it,  $H.d$  is automatically marked as 'non-moveable' to prevent any conflicting attempt to move it upstream into the

destination area in a later stage. Such an automated operation will be addressed in §6.4.6.3 Manual and Automated Decision Making.

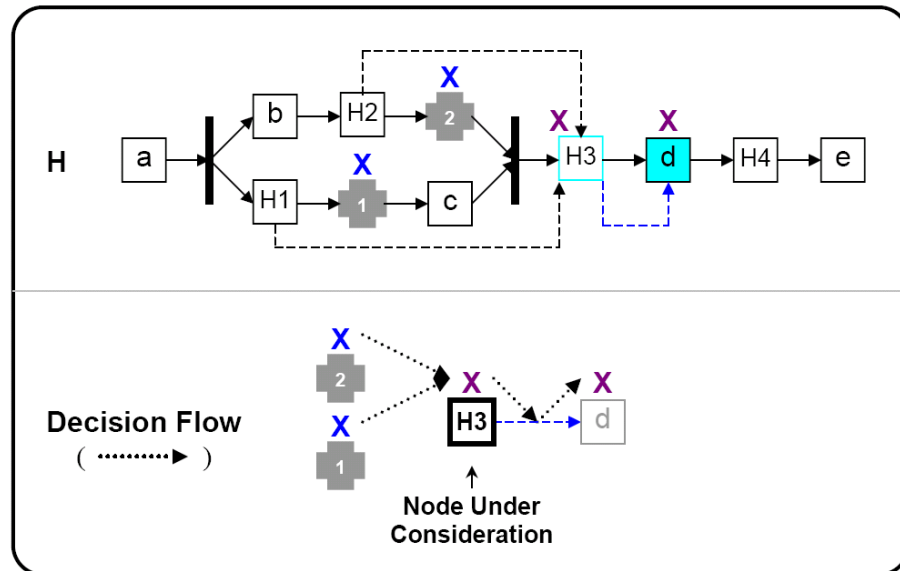


Figure 6-27. Data flow implication when all options are rejected.

### 6.4.5.3 CIIA

For a CIIA, it is rather straightforward. If it is decided that a CIIA needs to be inserted, together with the CIIA, its data generation activity needs to be created and inserted right before the CIIA. Only when both of the activities are successfully inserted, the operation for the CIIA is regarded as complete.

## 6.4.6 Decision Making

Being faced with reconciliation options, the user needs to make decisions according to business rules and organisational policies. From a reconciliation support viewpoint, the decision-making process includes two aspects (control flow and data flow), two stages (assessment and confirmation), and two methods (manual and automated).

#### **6.4.6.1 Control Flow and Data Flow**

Since the reconciliation options cover control flow and data flow, decisions need to be made on the two aspects. In order to guide the user to make consistent decisions, decisions regarding the same issue should only be made once. Due to the fact that different CMIA might share the same CMAs in control flow, measures are introduced to make sure that each control flow decision regarding the same CMA is made only once. This is by introducing a list to hold all the visited CMAs without duplication. Whenever a CMA is encountered, the list is checked to find whether the CMA has already been included. If not yet, the option-prompting and decision-making procedure will proceed; otherwise, previous decision will be retrieved and used. As to data flow options, since each of them is specific and unique for each pair of activities, there is no chance of making conflicting decisions.

#### **6.4.6.2 Assessment and Confirmation**

Due to the causal relationship between activities as well as the option and decision dynamics (mentioned in §6.4.5) involved in process changes, before any decision is confirmed, the user should be fully aware of the prerequisite and result of their decisions. Also, being provided with such implications can improve user understanding of the dynamics involved and help with option evaluation to identify their desired moves. For these purposes, the decision-making process is divided into two stages: decision assessment and confirmation. During the assessment stage, users are free to try different decisions or combination of decisions, study the real time feedback and implication of their current choices, and understand what each decision means and implies. They need to confirm their decisions only when they are fully satisfied.

#### **6.4.6.3 Manual and Automated Decision Making**

In general, users need to go through decision-making manually. However, in order to keep users' decisions consistent and minimise their workload, several scenarios are identified, where automated decisions can be made.

#### **Eliminating Unique Data Supply Interface Activity during Activity Bipartite Comparison**

During activity bipartite comparison, if any unique data supply interface activity (UDSIA) is encountered, no matter which side it belongs to, the interface activity can be safely

eliminated without users' intervention because data supplied by such an activity has not been demanded.

### **Rejecting Option Regarding Data Demand Interface Activity Acting as Data Predecessor in CMA/CMIA Set**

Among all the candidate moving activities (CMA) associated with a given candidate moving interface activity (CMIA), if any CMA is a data demand interface activity, when it comes to the decision-making regarding control flow reconciliation options for the CMA, the decision should be automatically made as rejection. This is because even if the decision is made as acceptance, the adjustment of the preceding interface activity will not contribute to the reconciliation of the discrepancy, as it will still be located before the CMIA.

### **Subsequent Rejections in Control and Data Flow Reconciliation Options**

When the control flow reconciliation options of the current candidate moving activity (CMA) are all rejected, the current CMA is barred from moving in the control flow. If any of the subsequent data flow reconciliation options with regard to discarding data dependencies between current CMA and succeeding CMA/CMIA is rejected, the corresponding succeeding CMA/CMIA is to be marked as 'non-moveable' automatically. This is because if the succeeding CMA/CMIA is to be moved, it will be positioned upstream into the destination area, i.e. a position before the current CMA. In order to back such a move, relevant data link should be discarded. Since the user decides to retain the data link, in order to prevent inconsistent decisions from being made at a later stage, the succeeding CMA/CMIA is disqualified for moving at this stage.

## **6.5 Process Change**

Process change is carried out at the level of digraph representation and under the consideration of integrity of relevant digraphs, which means initial process digraphs must not be decomposed into disconnected subsets.

### **6.5.1 Conceded Target Activity**

Following users' decision-making activity, for all the target activities (covered in §6.4.2), wherever a users' decision is made in favour of reconciliation in control flow (if the target activity is a candidate moving activity, its corresponding candidate moving interface activity



needs to be set as ‘accepted to move’), i.e. a concession is made, the target activity will be marked as a *conceded target activity* (CTA). Only CTAs will be committed the proposed adjustment. For different target activities, concession means differently and leads to different operations as shown in Table 6-2.

Table 6-2. Concession and operation for different target activities.

Target Activity	Concession	Operation
<i>candidate removing interface activity (CRIA)</i>	agree to abandon the data demand	to be removed
candidate moving activity (CMA) and candidate moving interface activity (CMIA)	agree to move the activity upstream	to be removed and inserted
candidate inserting activity set (including candidate data generation activity and candidate inserting interface activity)	agree to insert the whole set into a target position	to be inserted

### 6.5.2 Control Flow Change Operations

There are two fundamental types of change operations in control flow: removal and insertion. When used individually, they can perform change operations following concessions related to CRIA and CIIA. When used collectively (removal followed by insertion), they can deal with required operations following concessions related to CMIA.

#### Operation Removal

Operation removal uses the following steps to remove a conceded target activity (CTA).

- Identify and record each immediate predecessor and successor of the CTA in control flow,
- Disconnect the CTA from its immediate predecessor(s) and successor(s),
- Reconnect the CTA’s immediate predecessor(s) and successor(s),
- Remove the disconnected CTA from the vertex list.

During reconnection stage, similar to the removal of non-interface activity mentioned in §6.2.1, if the CTA due to be removed:

- is the very first activity and it has multiple successors,
- is the very last activity and it has multiple predecessors,
- has both multiple predecessors and successors,

a control flow route activity needs to replace the outgoing CTA to keep the resulting process definitions syntactically and semantically correct.

### **Operation Insertion**

Operation insertion uses the following the steps to insert a conceded target activity (CTA).

- Locate each destination position called target predecessor (TP) that is expressed as the CTA's immediate predecessor, which has been recorded right after each decision on a control flow reconciliation option is committed,
- Connect the CTA to each identified TP,
- Identify each immediate successor called target successor (TS) in the current process definition for each recorded TP,
- Connect the CTA to each identified TS by setting the corresponding value in the control flow adjacent matrix as 1,
- Disconnect each TP from its corresponding TS by setting the corresponding value in the control flow adjacent matrix as 0.

### **6.5.3 Data Flow Change Operations**

Change operations for data flow are much simpler. These operations include deletion and addition.

#### **Operation Deletion**

When a decision on discarding a data link is confirmed, operation deletion will be performed to delete the data dependency by setting the corresponding value in the data flow adjacent matrix from 1 to 0.

#### **Operation Addition**

Whenever needed, data dependency can be added into the data flow adjacent matrix by setting the value of the intersection from the data predecessor to the data successor from 0 to

1. However, since any data dependency must be backed by the underlying control flow, before operation addition, existence of control flow needs to be checked.

## **6.6 Summary and Conclusion**

In this chapter, key enabling techniques for workflow negotiation and reconciliation are explained, which include interface process extraction, process comparison, and formulating, prompting and decision-making of reconciliation options, and process change operations.

The interface process extraction technique enables the collaboration interests to be represented in the form of interface processes, which are used in the process comparison stage to reveal (dis)similarity between the two interface processes. In the case of discrepancy, reconciliation options need to be formulated, prompted to the Host user and decisions need to be gathered. This is the most complicated but crucial step because the choice of target activities and destination areas as well as the understanding and capture of the dynamics between options and decisions directly relate to the quality of the end result of the reconciliation task. Finally, computerised process change operations free the user from the trivial but error-prone task.

# Part III

## Design, Implementation and Evaluation

*"It always seems impossible until it's done."*

*– Nelson Mandela*

# Chapter 7

## System Design and Implementation

### 7.1 Introduction

This chapter describes the system architecture of the prototype system COWCO-Guru, which is based on the client/server paradigm. Negotiation protocol is implemented in the form of a pair of compatible workflows, which enables the build-time workflow negotiation services to share a unified collaboration enactment infrastructure with the run-time agreement fulfilment services.

§7.2 explains the function of each constituent component from a system viewpoint and the workflow implementation of the negotiation protocol. §7.3 describes the run-time collaboration enactment infrastructure with the foci on collaboration messages and their exchange mechanism. §7.4 demonstrates how the system works at build time. §7.5 summarises the chapter.

### 7.2 A Prototype System for Cross-Organisational Workflow Collaboration

In order to demonstrate the effectiveness of the integrated approach (Chapter 4) and the comprehensive framework (Chapter 5), key enabling techniques (Chapter 6) are implemented as part of the prototype system, COWCO-Guru.

#### 7.2.1 Basic Facts Explained

Before introducing the implementation of the prototype system, several basic facts are explained.

### **Exclusive OR Structure Free at Build Time**

As mentioned in §4.2.1, at collaboration build time, the processes have been pre-treated to remove XOR structures. However, XOR and iteration structures can be dealt with at run time.

### **Client/Server Architecture**

The classic client/server architecture with rich client is selected as the basis of implementation of the prototype system. The underlying reasons are: firstly, workflow management systems have already existed on clients' sites. Due to the inherent close associations between WfMSs and some services, e.g. user interface and agreement fulfilment, these services are preferable to be located on clients' sites, which makes the clients rich by nature. Secondly, by deploying the core module – collaboration negotiation and reconciliation – as part of the client functions, collaboration messages can be significantly reduced based on the analysis in §5.3.3 Implementation Architectures. Thirdly, privacy protection is in favour of rich client architectural arrangement over the thin client and the SOA because there is no need for partners to submit their full workflow definitions to the collaboration server. However, possible ways of adapting the Framework to thin client C/S and SOA has been described in §5.3.3, which can be used as a guidance when exploring other implementations.

### **Shark Workflow**

Shark Workflow is chosen as the workflow management system in this thesis for four reasons. First, Shark is a fully functioning open source workflow system. Secondly, it complies with WfMC's standards, which makes the assumptions, techniques and approaches that are based on it more generic and representational. Thirdly, several application Tool Agents have been implemented for Shark by third parties and they comply with WfMC's relevant standard as well. Fourthly, Shark can interpret XPDL – one of the most popular process definition languages.

However, XPDL does not provide means to explicitly specify data flow dependencies. Although data flow dependencies can still be identified and extracted from an XPDL workflow definition by tracing the same workflow data in control flow, for the purpose of simplicity, in this thesis, they are explicitly defined in a separate XPDL file where all the

vertices are kept the same as in the original workflow definition but transitions stand for data flow dependencies only.

## 7.2.2 System Architecture

The prototype system, named COWCO-Guru, is implemented based on the conventional client/server architecture of rich client type to demonstrate the effectiveness of key services of the cross-organisational workflow collaboration (COWCO) supporting framework. Derived from the rich client implementation architecture shown in Figure 5-11, the system architecture of COWCO-Guru is illustrated in Figure 7-1.

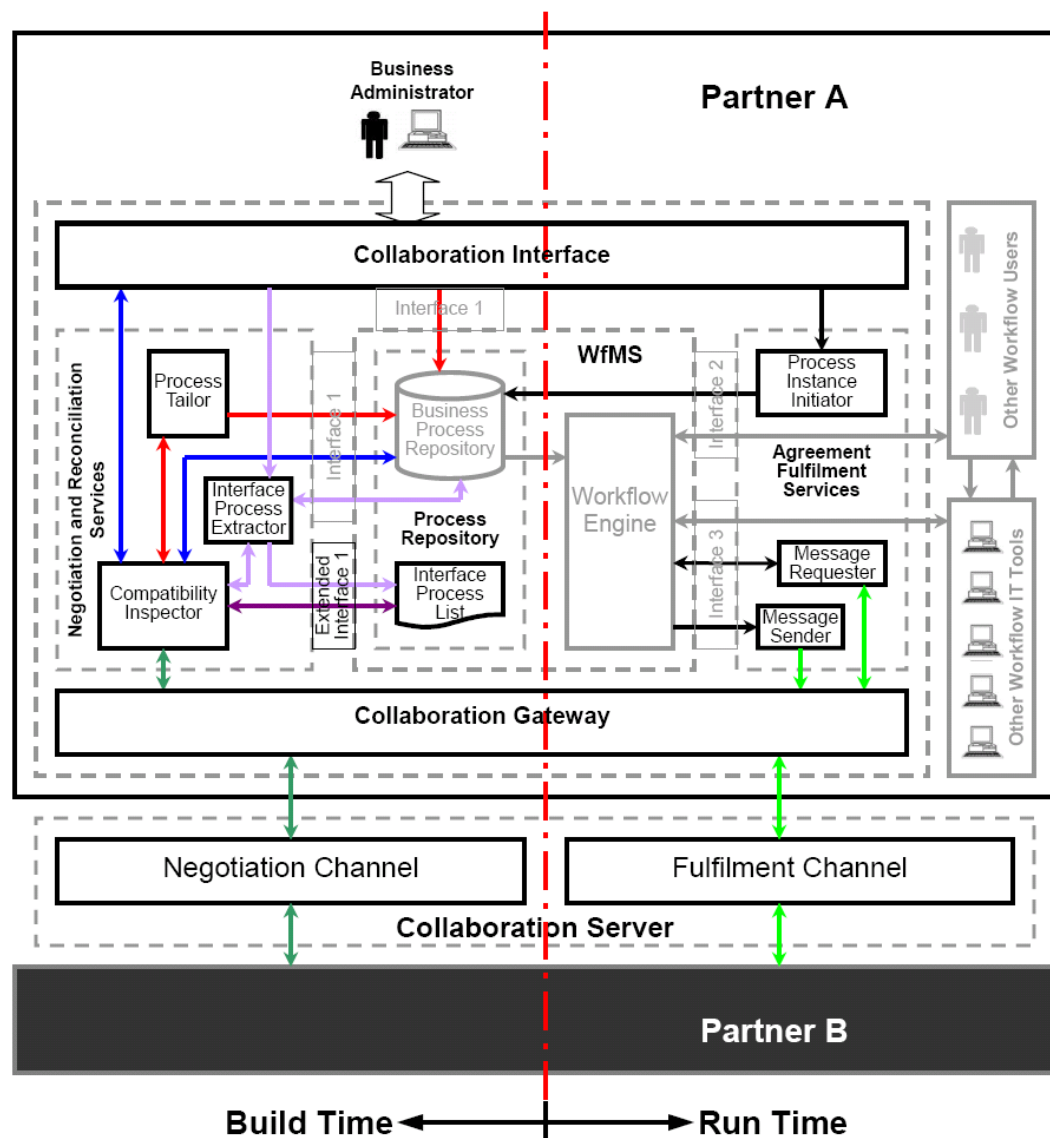


Figure 7-1. System architecture.

Interfaces with regard to a workflow management system defined by the Workflow Management Coalition (WfMC) are included in the figure. Communication between functional modules are colour coded, which will be referred to later in this sub-section.

### 7.2.2.1 Collaboration Interface

The *Collaboration Interface* implements the Collaboration Interface Service in the service stack of the Framework and serves as the user interface at both collaboration build time and run time. At build time, the interface comprises two main parts – the primary interface and a series of dialogue boxes. The primary interface provides users with the options of extracting an interface process from a designated workflow definition in an XPDL file and displaying diagrams of the Host's initial and resulting business processes, both the Host's and the Guest's interface processes, and the corresponding state transition system diagrams. A process editor is integrated as well. Figure 7-2 shows the primary interface.

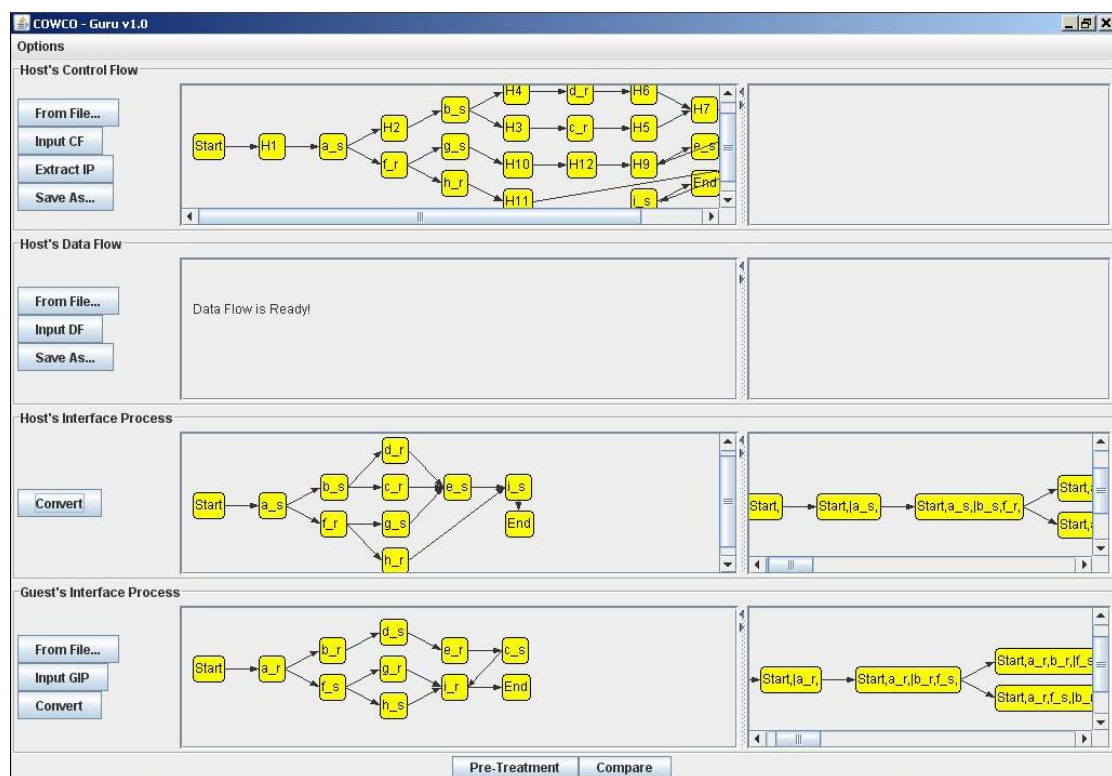


Figure 7-2. Primary interface.

Generally speaking, a dialogue box is in charge of prompting formulated reconciliation options (in both control flow and/or data flow) in the form of check boxes and displaying



real time implication message based on users' current decision combinations in an active text area. These are illustrated in Figure 7-3.

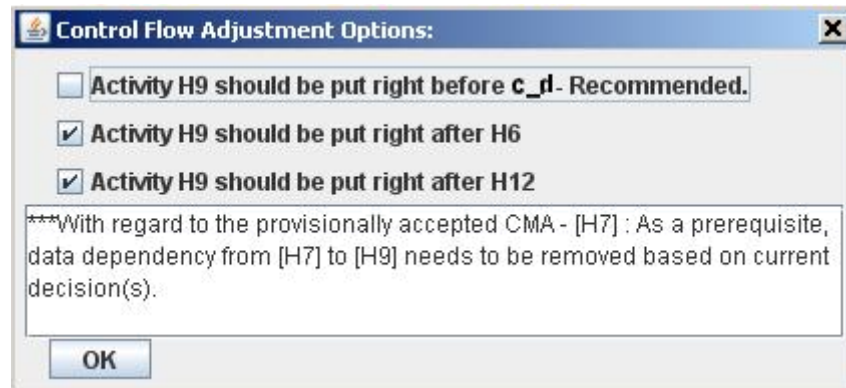


Figure 7-3. Control flow options dialogue.

At run time, users instantiate a desired workflow through an integrated workflow administrative interface to initiate the agreement fulfilment stage as shown in Figure 7-4. Shark's administration interface is adopted in this project.

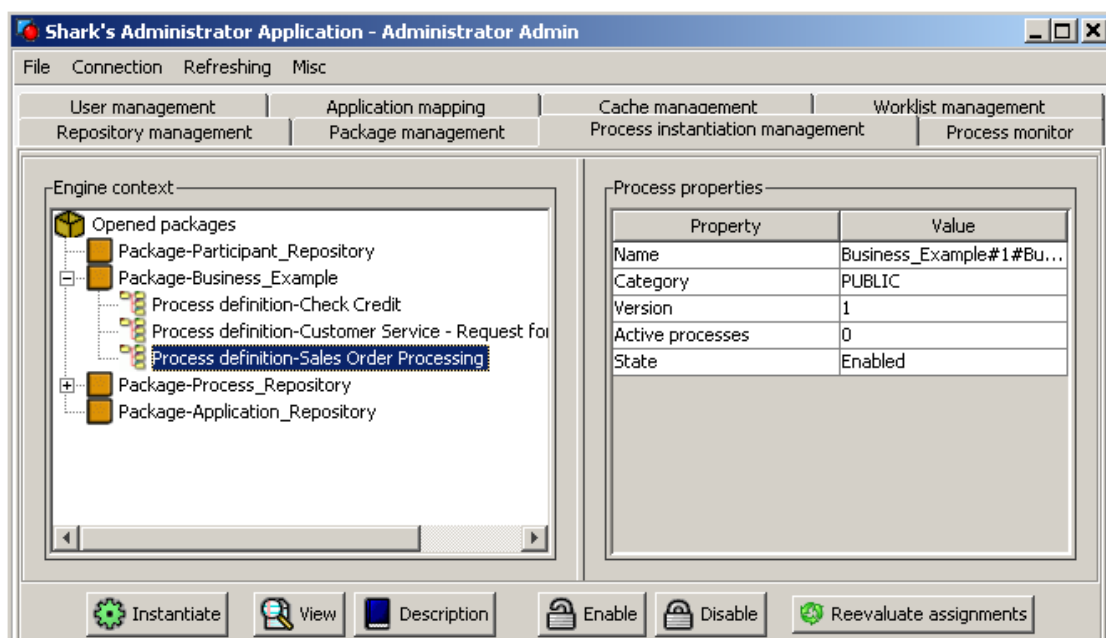


Figure 7-4. Shark Workflow's administration interface.

### 7.2.2.2 Interface Process Extractor

The *Interface Process Extractor* is the implementation of the interface process extraction service. Following users' demand, the *Interface Process Extractor* picks a definition from the *Business Process Repository*, extracts the interface process, stores the result into the *Interface Process List* and passes a copy to the *Compatibility Inspector* for further process (shown as the lavender arrows in Figure 7-1).

### 7.2.2.3 Compatibility Inspector

The *Compatibility Inspector* is the core module in negotiation and reconciliation services. Its roles include:

- exchanging interface processes and users' decisions with negotiation partners through the *Collaboration Gateway* (shown as the green arrows in Figure 7-1),
- comparing interface processes of the Host and the Guest,
- prompting users relevant reconciliation options based on the result of the comparison and the initial workflow definition of the Host (shown as the blue arrows in Figure 7-1),
- effecting process changes based on the users' decisions on the reconciliation options (shown as the red arrows in Figure 7-1).

In order to facilitate these tasks, temporary flow models of the Host's corresponding workflow need to be created and stored locally. The flow models include control flow and data flow. Explanation of detailed techniques and procedures can be found in §6.3 Interface Process Comparison and §6.4 Reconciliation Option.

### 7.2.2.4 Process Tailor

For each identified discrepancy, users' decision on the reconciliation options will finally be returned to the *Compatibility Inspector*, where process change instructions will be synthesised and passed onto the *Process Tailor*. According to the process change instructions (covering both control flow and data flow aspects), the *Process Tailor* updates the temporary flow models of the workflow in question. After all the currently identified discrepancies have been gone through, the temporary flow models are saved locally and waiting for the next round of negotiation. When the negotiation is fully accomplished and results in a success, the temporarily saved flow models will be converted to back into XPDL, stored in the *Business Process Repository* and ready for the run-time instantiation.

More details about techniques used by the *Process Tailor* can be found in §6.5 Process Change.

#### **7.2.2.5 Collaboration Gateway**

Every communication message coming into, and going out of, a collaboration partner flows through the Collaboration Gateway situated on each client side. The gateway has two main purposes – mapping between the internal terms and the ones in a commonly agreed ontology and converting the internal flow models into a commonly agreed process exchange format.

As communication takes place across organisational boundaries, terms used in describing concepts of common interest need to be understandable on both sides. An ontology is designated by a partner with the consent from the other. In this thesis, since the examples are from the e-commerce domain, the ontology is chosen as the widely accepted term dictionary – RosettaNet Business Dictionary (RosettaNet, 2002). Semantic identifiers of both the incoming and outgoing messages pass through the gateway are translated back and forth between internal and common terms.

Since at the negotiation stage, the core information containing in the message is the flow model of an interface process, the flow model also needs to be converted from an internal format to a common process exchange format and vice versa. The common format is chosen as the XML process definition language (XPDL) because it is sufficient to represent control flow required by an interface process.

#### **7.2.2.6 Process Repository**

The *Process Repository* is a component of a workflow management system. It is used to store workflow definitions that will be used for instantiation and enactment by a workflow engine. In this architecture, the repository is extended to store definitions of interface processes together with the identifiers of their original workflow definitions. An interface process is a process definition containing only interface activities that are extracted from its original workflow. As a subset of the original flow model, it is non-executable, i.e. cannot be instantiated and executed by a workflow engine. The interface processes are organised in the *Interface Process List* according to process-level service descriptions, which means under the same service description, there might be different interface processes in storage.

The purpose of such a collection of interface processes is to let the *Compatibility Inspector* to lookup in the list first and decided whether there is any existing successful match (shown as the purple arrow in Figure 7-1). For any successful match, the interface process extraction procedure can be skipped. For multiple unmatched interface processes, as long as their *estimated effort of design from scratch* (EEDS) is lower than the predefined threshold, the one with the lowest EEDS will be selected for the collaboration negotiation to start with.

#### 7.2.2.7 Agreement Fulfilment Services

When the negotiation and reconciliation services successfully lead to an agreement, it moves on to the agreement fulfilment stage. With users' consent, the *Process Instance Initiator* loads the designated executable workflow into the workflow engine and instantiates it. While the process instance is actively running, the *Message Sender* gets the data from the workflow instance and passes it on to the *Fulfilment Channel*. The *Message Requester* firstly gets the semantic identifier from the process instance and registers it with the *Fulfilment Channel*. When the required data arrives, it will be passed back to the *Message Requester* before being further relayed to the process instance. All the communications between the *Workflow Engine* and the *Message Sender/Requester* are through the WfMC's Interface 3. All the communication between the *Message Sender/Requester* and the *Fulfilment Channel* is through the *Collaboration Gateway*, where translations and conversion between local and common terms are carried out. These run-time message movements are marked by bright green arrows in Figure 7-1.

#### 7.2.2.8 Collaboration Server

As the link and mediator between the two partners, *Collaboration Server* implements the message relay facility, including *Negotiation Channel* and *Fulfilment Channel*. These two channels are used for their specific purposes respectively. The *Negotiation Channel* relays interface processes and users' *willingness to stay* in a negotiation at the collaboration build time. The *Fulfilment Channel* mediates the exchange of collaboration messages between the two running workflow instances. They will be discussed with more details in §7.3 Workflow Collaboration Enactment Infrastructure.

### 7.2.3 Implementation of the Negotiation Protocol

As the set of rules governing the negotiation interaction, negotiation protocol needs to be implemented in line with the client/server architecture and imposed on both partners. A pair of collaborative negotiation workflows are defined and implemented to realise the desired negotiation protocol as introduced in §5.2.1.2 Capturing Communication. Each negotiation workflow is specified by using the hosting partner's WfMS and should then be able to be enacted by the workflow engine of the same WfMS. As to collaboration messages, the intermediate interface processes and decisions about *willingness of staying* need to be exchanged between collaboration partners through the collaboration server.

Please note, due to the overall negotiation process (the backbone of the Negotiation and Reconciliation Services) is implemented as a pair of collaborative workflows, both the build-time Negotiation/Reconciliation Services and the run-time Agreement Fulfilment Services can share the same workflow collaboration enactment infrastructure (to be introduced in §7.3).

## 7.3 Workflow Collaboration Enactment Infrastructure

The purpose of the Workflow Collaboration Enactment Infrastructure (or the Infrastructure in a short form) is to support the negotiation and reconciliation services at collaboration build time and enact the pair of compatible workflows at run time. As described in §7.2.3, since the overall negotiation process is implemented in the form of a pair of collaborative workflows, the Infrastructure can be unified and utilised at both build time and run time. It comprises the *Message Sender/Requester* on the client sides and the *Collaboration Server*.

### 7.3.1 Collaboration Message

Messages, as data bearing vehicles, are key for any distributed systems. For COWCO-Guru – the prototype system, messages are the objects being passed back and forth between client applications through the server. They are discussed from three aspects: type of data content, format and ontology.

### 7.3.1.1 Types of Message Content

Messages are needed at both collaboration build time and run time. At build time, as shown in Figure 7-5, two types of data need to be carried around by messages, namely interface processes and willingness of staying (in the negotiation). An interface process is the offer/counteroffer generated by each partner to represent its interests. As a process, normal method for internal representations (e.g. vertex list and adjacent matrix) is not suitable. A file that contains a vertex list and a transition list suits the task. In this thesis, an interface process is saved in the form of an XPDL file before it is exchanged between partners. In contrast, the willingness of staying can be well represented by plain text. At run time, workflow relevant data needs to be exchanged. It takes the form of either a binary file (e.g. a purchase order or a design drawing) or plain text (a reference number or the amount certain goods) representing various business objects.

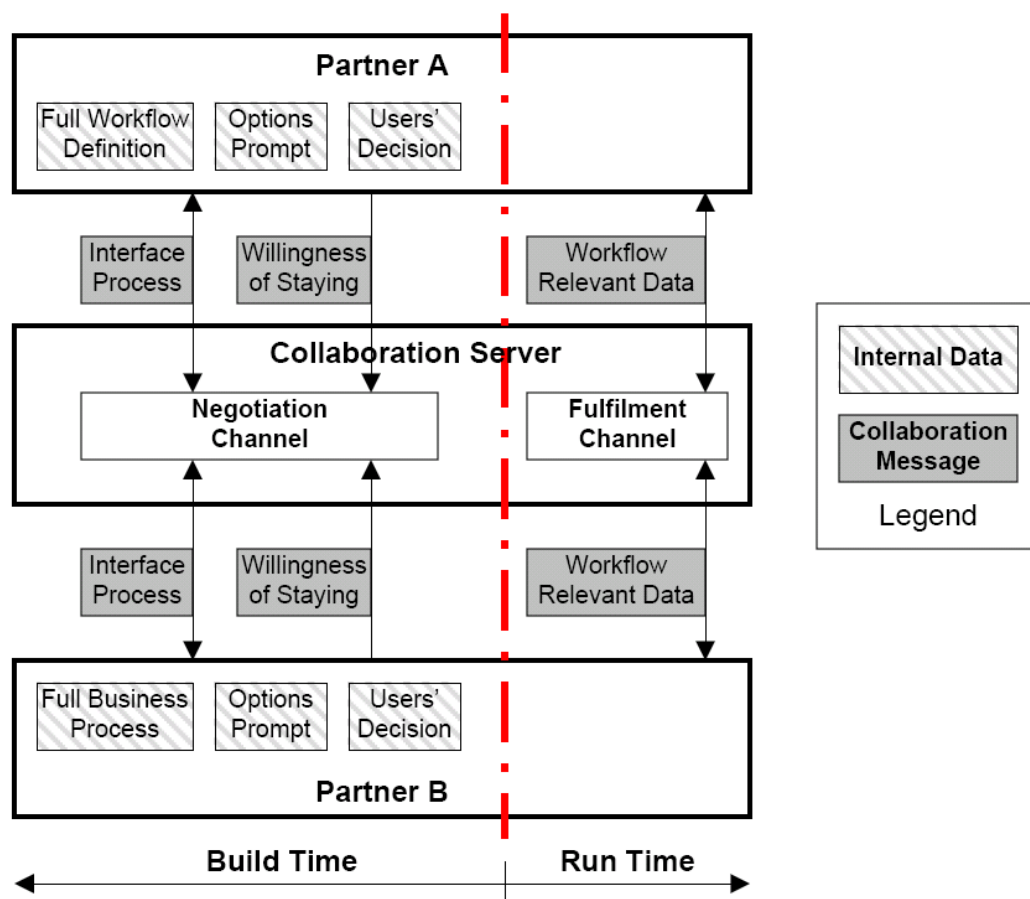


Figure 7-5. Message exchanged in a rich client type of client/server architecture.

### 7.3.1.2 Message Format

Quite often, a message needs to carry more than one piece of information. Different information is assembled at one end of communication to form a message and extracted at the other end. As a result, a message format is required to define the information arrangement within the message. In this thesis, five different message formats are defined, which will be introduced in §7.3.2 and §7.3.3.

### 7.3.1.3 Data Ontology

Two sets of data ontology are adopted in this thesis for negotiation/reconciliation stage and agreement fulfilment stage respectively. Two terms compose the former ontology, namely *interfaceProcess* and *willingnessOfStay*. For the latter, terms in the commonly recognised RosettaNet Business Dictionary are adopted.

## 7.3.2 Message Sender/Requester

Message Sender and Requester are implemented as workflow application tools. They are attached to and invoked by corresponding interface activities through Tool Agents at WfMC's Interface 3. The specific tool agent is chosen as the Java Class Tool Agent (a third party implementation for the Shark workflow engine) because it supports application tools that are implemented in user defined Java classes.

### 7.3.2.1 Message Sender

As described in §4.3.3 Effectiveness of Distributed Coordination, the task of a *data supply activity* (DSA) is to have a message associated with it sent as well as to allow the hosting process to move on to the next activity. This is implemented as the invocation of the attached application tool – Message Sender by a DSA in an asynchronous manner, which are all standard workflow functions.

As a result, the key information that a Message Sender needs to send to the server is the transaction semantic identifier (SI) and the data itself. Since the data contained in a message can be either text data or a file, the data type also needs to be indicated by the Message Sender and recorded in the message. The three pieces of information are organised in either Format (1) or (2) as shown in Figure 7-6 depending on the type of data involved.

Message sent from a client:

Format (1) Message containing a file: File SI++++ext followed by the file;  
Format (2) Message containing text: Text SI\*\*\*\*\*text.

Figure 7-6. Data Supply Message: Message Format (1) and (2)

### 7.3.2.2 Message Requester

The task of a *data demand activity* (DDA) is to block itself and wait until the required data comes. However, in a decentralised and loosely coupled system, no party could possibly know someone is waiting for something unless the waiting party informs its desire to others. Therefore, a DDA needs to have the data it is interested registered with the server before blocking itself until the data arrives, which requires the attached application tool – Message Requester – to be invoked to send the request message in a synchronous manner. Furthermore, since iteration might exist in an executable workflow, it is important to differentiate different rounds of an iteration structure. An iteration flag is required by a Message Requester to clarify this issue.

As a result, the information that a Message Requester needs to send to the server is only the transaction semantic identifier. The information a Message Requester is expecting is the required data and the iteration flag. Since the two pieces of information are wrapped in the message sent by the server, the Message Requester is equipped with an *Incoming Message Parser* to parse the received message and extract each required portion. The outgoing message takes Format (3) shown in Figure 7-7.

Message sent from a client:

Format (3) Message containing only the semantic identifier: SI

Figure 7-7. Data Request Message: Message Format (3)



### 7.3.3 Collaboration Server

The Collaboration Server is designed to mediate the collaboration communication. Rather than acting only as a message repeater, the server needs to record the message traffic because useful information can be reasoned about from the message log. A unified data structure (a form) is introduced to record desired information. The server is implemented based on a generic blackboard system – GBBOpen because of its suitability for distributed problem solving in terms of the provision of an information storage facility and the availability of an effective event triggering mechanism.

#### 7.3.3.1 Collaboration Form

A form is designed to record and forward information occurred in the interaction between the clients. Table 7-1 describes the headings of the form.

Table 7-1. Headings of the collaboration form.

Column Heading	Description	Example	Info Supplier
data type	type of data contained in the message	File or Text	Sender
semantic identifier (SI)	transaction identifier	interfaceProcess, purchaseOrder, shippingAdvice, etc.	Sender, Requester
file extension	the extension of the file	doc, xls, pdf, etc.	Sender
data	the binary sequence of a file or text	binary sequences of a file, text string	Sender
consumption flag	denote whether the data has been consumed, i.e. delivered to corresponding Data	true, false	Server

	Demand Activity		
iteration flag	denote whether an iteration is detected	true, false	Server

Among the six headings, the semantic identifier is the field used for message matching under a decentralised environment. Information under the first four headings is supplied by either of the participating collaboration workflows. Information under the last two headings is deduced by the server based on the progress of each collaboration instance. The forms used at both the negotiation/reconciliation stage and the agreement fulfilment stage are essentially the same except that they are called *Negotiation Form* and *Fulfilment Form* respectively due to their purposes. Table 7-2 shows an example of a fulfilment form in the middle of a collaboration instance.

Table 7-2. An example of agreement fulfilment form.

Data Type	Semantic Identifier	File Extension	Data	Consumption Flag	Iteration Flag
...					
Text	quotationEnquiry	nil	cs0011	true	nil
File	quotation	doc	the file in binary stream	true	nil
...					

### 7.3.3.2 Form Filling

For each process collaboration case, an empty form is created on the *blackboard*. As the collaboration progresses, the form is expanding and being updated by the information coming from both participating workflows as well as that deduced by the server based on current information history in the form.

Whenever the server receives a message, information is parsed and extracted from the message by an *Incoming Message Parser*. Firstly, it is checked whether an unconsumed

entry marked by the same semantic identifier exists. If no match is found, a new entry will be created. If the reason of no matching is due to an existing entry having been consumed, the iteration flag of the newly created entry will be set as *true*. Furthermore, if the current incoming message comes from a Message Sender, information under the first four headings will be recorded in the new entry accordingly. If the current incoming message is from a Message Requester, since there is no matched entry at the moment, the thread associated with the Message Requester will enter the sleeping state with the waking-up condition set as any later message coming from a Message Sender.

If there is a match for the semantic identifier on an unconsumed entry, the information corresponding to the first four columns will be used to formulate an outgoing message in either Format (4) or (5) as shown in Figure 7-8 depending on the data type (File or Text) and the consumption flag will be updated as *true*. Furthermore, if the current incoming message is from a Message Requester, the newly formulated outgoing message will be sent directly back to the Message Requester. If the current incoming message is from a Message Sender, such an event will wake up all the currently sleeping threads and let them check whether they are interested in the newly arrived data. If yes, the newly formulated outgoing message will be delivered to the associated Message Requester who is waiting and threads of all other unmatched entries return to the sleeping state.

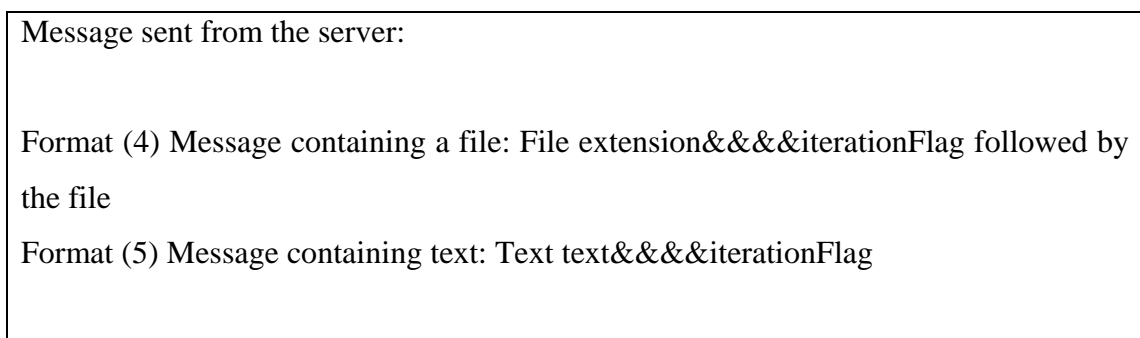


Figure 7-8. Data response message: Message Format (4) and (5).

By combining the procedures of the clients and the server, the overall run-time message exchange mechanism is illustrated in Figure 7-9.

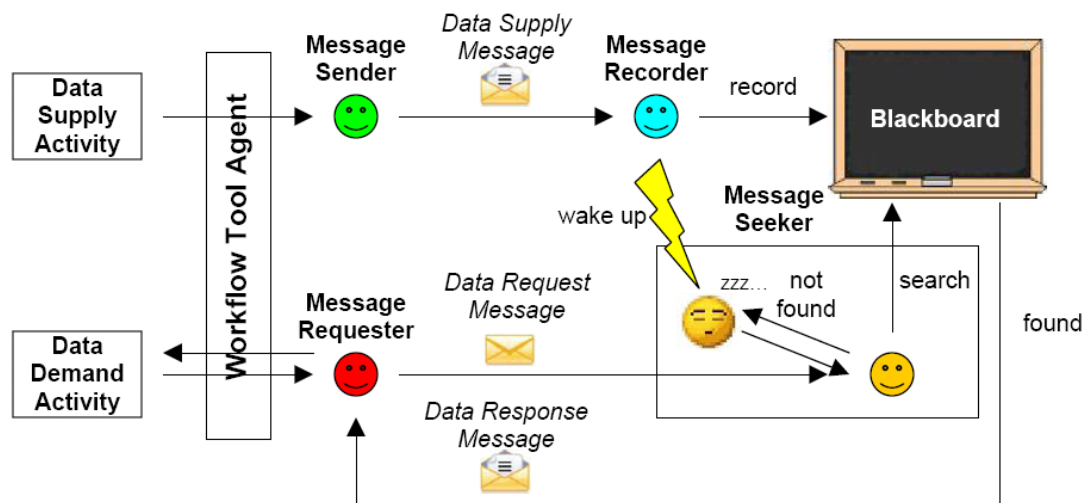


Figure 7-9. Run-time message exchange mechanism.

## 7.4 A Demonstration of the Prototype System at Build Time

In this section, a pair of workflows belonging to Partner A and B respectively are used to demonstrate how the prototype system works at build time. Unless stated otherwise, the legends are adopted according to the arrangement shown in Figure 7-10. The meaning of the abbreviations are: CCPIA – current common preceding interface activity, DBA – downstream boundary activity, CMA – candidate moving activity and CMIA – candidate moving interface activity.

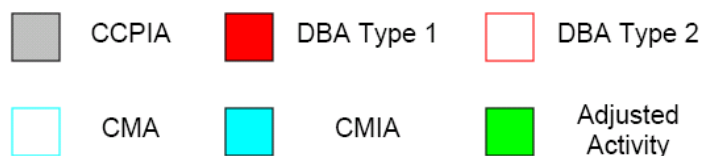


Figure 7-10. Legends adopted in the demonstration.

#### 7.4.1 On Partner A's Site

Assuming that Partner B is the collaboration initiator and Partner A is the responder, as shown in Figure 7-11, the process comparison first starts at A's site after the interface process of B (B\_IP) is received by A. The discrepancy is encountered when A\_IP reaches *A.b[d]* and B\_IP reaches *B.c[d]*. The position right before *A.b[d]* in A's initial workflow is

identified as the hypothetical target position (HTP). Due to the data dependency between A3 and A.c[s], A3 is identified as candidate moving activity (CMA) of the candidate moving interface activity (CMIA) – A.c[s].

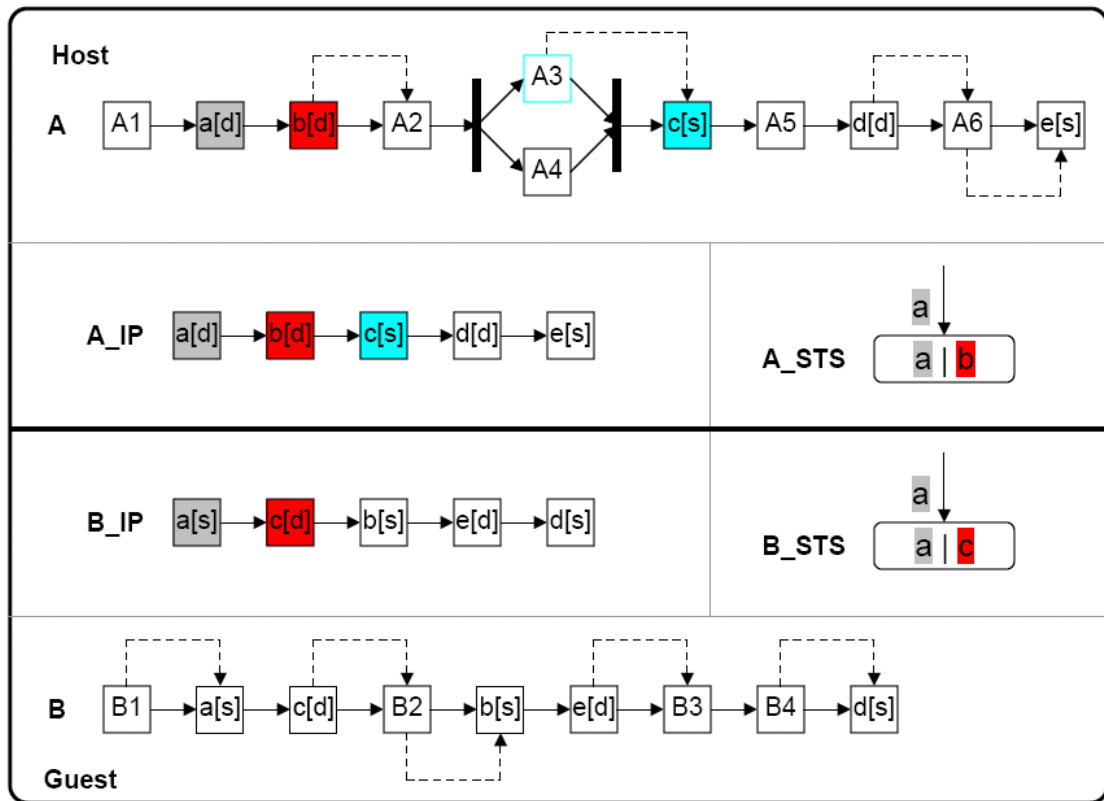


Figure 7-11. First round of comparison at Partner A's site.

The option regarding A3 in control flow is prompted to the user as shown in Figure 7-12. Following the user's decision of acceptance, the option for A.c[s] is prompted as shown in Figure 7-13, which is also accepted by the user.

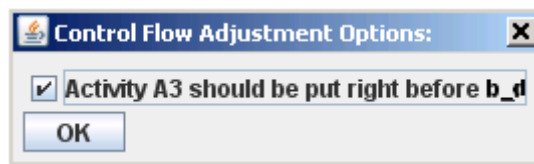


Figure 7-12. Reconciliation option for A3 in control flow.

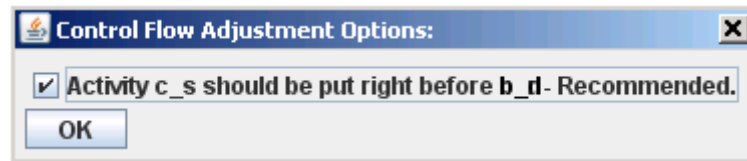


Figure 7-13. Reconciliation option for A.c[s] in control flow.

Up to this stage, the current discrepancy is reconciled and Workflow A is transformed to Workflow A(1) accordingly as shown in Figure 7-14. After the extraction of A(1)'s interface process A(1)\_IP, the second round of interface process comparison starts. This time the discrepancy is encountered when A(1)\_IP reaches  $A(1).d[d]$  and B\_IP reaches  $B.e[d]$ . The position right before  $A(1).d[d]$  in A(1) is identified as the hypothetical target position (HTP). Due to the data dependency between  $A(1).d[d]$ , A6 and  $A(1).e[s]$ ,  $A(1).d[d]$  and A6 are identified as the candidate moving activities (CMA) of the candidate moving interface activity (CMIA) –  $A(1).e[s]$ .

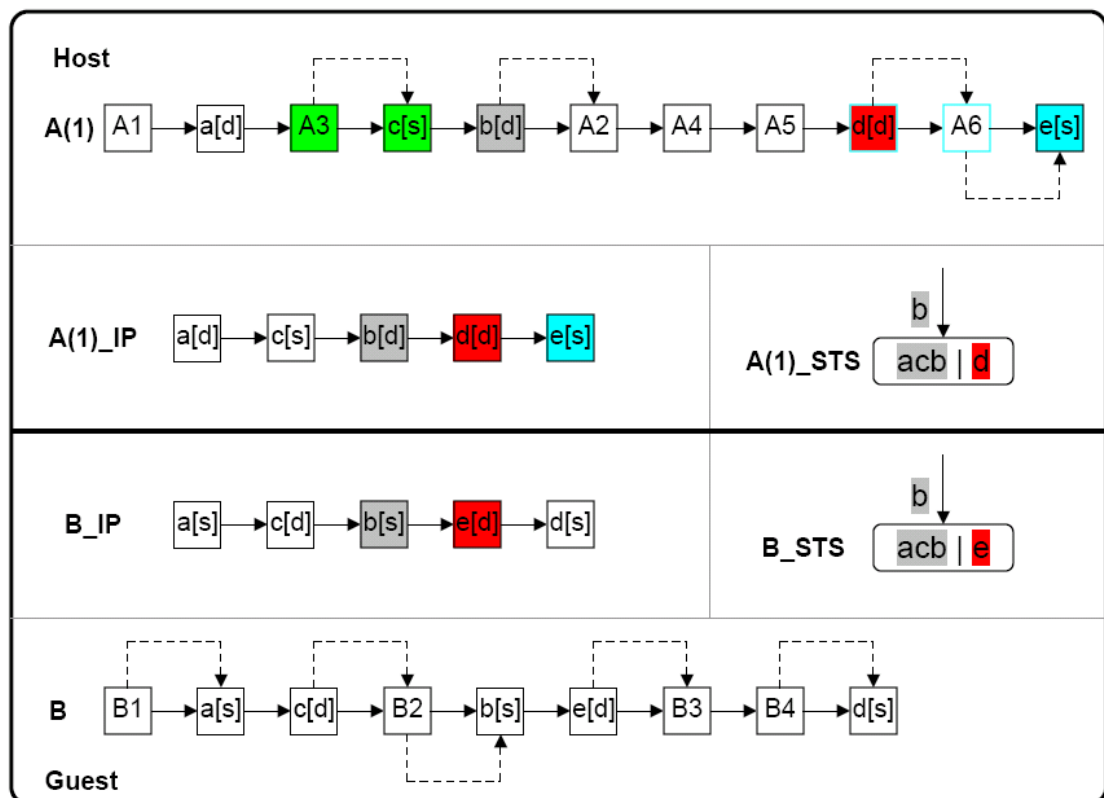


Figure 7-14. Second round of comparison at Partner A's site.

As  $A(1).d[d]$  is an interface activity, one of the automated decision making rules (described in §6.4.6.3 Manual and Automated Decision Making) applies, which prevents  $A(1).d[d]$  from being moved. The user is prompted with a reconciliation option regarding  $A(1).d[d]$  and A6 in data flow as shown in Figure 7-15. Being informed with relevant implications, the user decides to reject the option as shown in Figure 7-16.

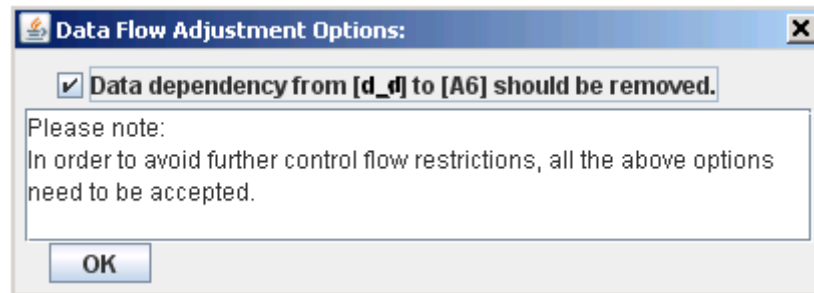


Figure 7-15. Reconciliation option regarding  $A(1).d[d]$  and A6 in data flow.

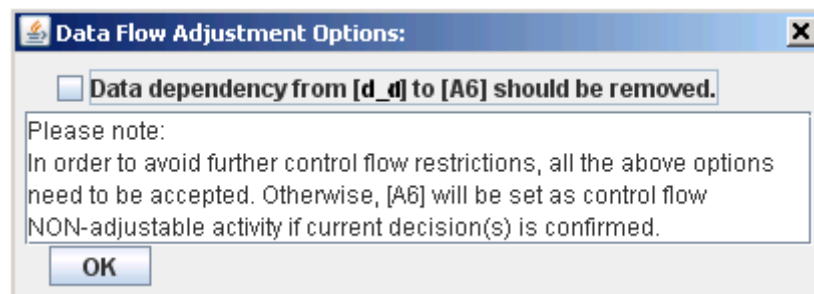


Figure 7-16. Rejection of the reconciliation option regarding  $A(1).d[d]$  and A6 in data flow.

Following the rejection decision, the user is prompted with another reconciliation option regarding A6 and  $A(1).e[s]$  in data flow as shown in Figure 7-17. The user decides to retain the initial data dependency and thus rejects the option as shown in Figure 7-18. As a result, the CMIA  $A(1).e[s]$  is rejected according to one of the automated decision making rules (described in §6.4.6.3 Manual and Automated Decision Making).

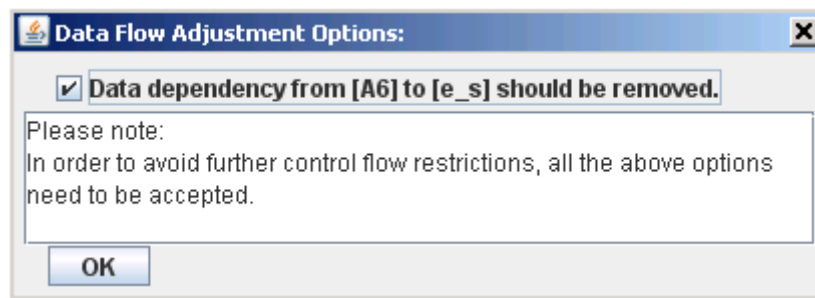


Figure 7-17. Reconciliation option regarding A6 and A(1).e[s] in data flow.

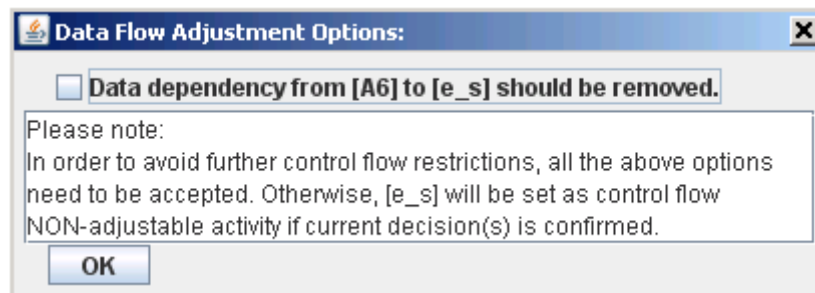


Figure 7-18. Rejection of the reconciliation option regarding A6 and A(1).e[s] in data flow.

At this stage, Partner A could give no further concession and the reconciliation is completed on A's site. Workflow A(1)'s interface process – A(1)\_IP – is extracted, converted to the format of XPD L and transferred to B's site.

### 7.4.2 On Partner B's Site

On receiving A(1)'s interface process, process comparison starts on B's site as shown in Figure 7-19. A discrepancy is encountered when B reaches *B.e[d]* and A(1) reaches *A(1).d[d]*. The position right before *B.e[d]* is identified as the hypothetical target position (HTP) in B's initial workflow. Due to the data dependency between *B4* and *B.d[s]*, *B4* is identified as the candidate moving activity (CMA) of the candidate moving interface activity (CMIA) – *B.d[s]*.



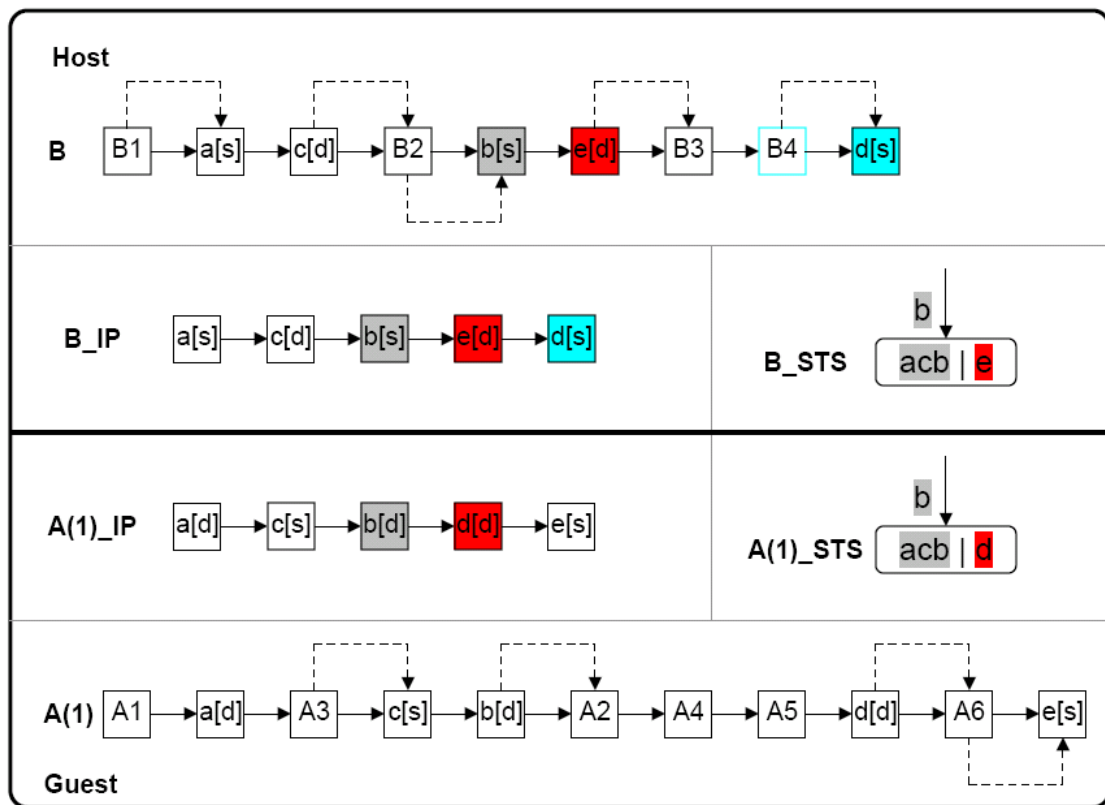


Figure 7-19. First round of comparison at Partner B's site.

The option regarding *B4* in control flow is prompted to the user as shown in Figure 7-20. Following the user's decision to accept *B4*'s move, the option for *B.d[s]* is prompted as shown in Figure 7-21, which is also accepted by the user.

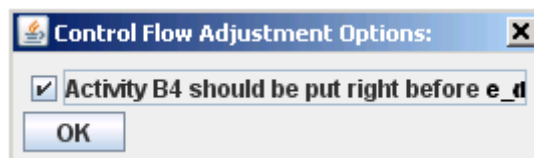


Figure 7-20. Reconciliation option for B4 in control flow.



Figure 7-21. Reconciliation option for B.d[s] in control flow.

Workflow B is then transformed into Workflow B(1) accordingly. Since no further discrepancy can be found, Workflow B(1) and A(1) are a pair of compatible workflows (as shown in Figure 7-22) and is ready for execution.

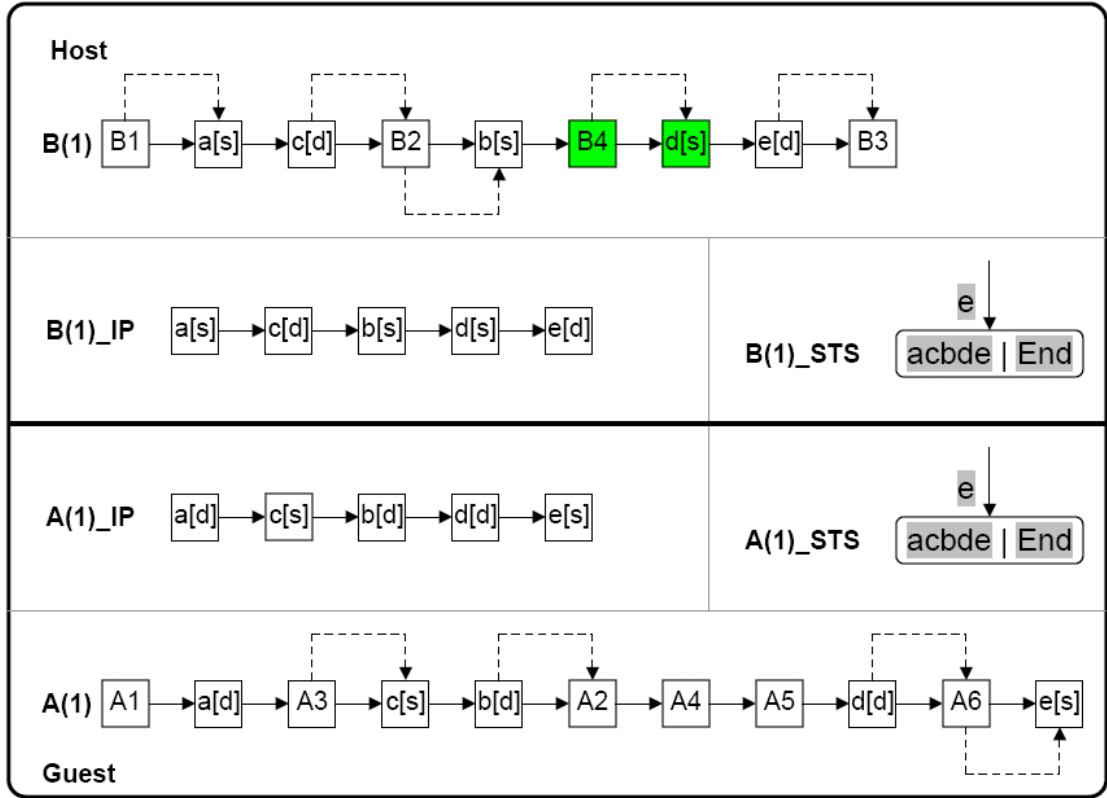


Figure 7-22. Compatible workflows B(1) and A(1).

## 7.5 Summary and Conclusion

In this chapter, the design and implementation of the prototype system COWCO-Guru is introduced. The prototype is implemented by following the classic client/server paradigm with underlying reasons explained. The overall negotiation process is implemented as a pair of collaborative negotiation workflows, which makes the build-time negotiation services be able to share a unified workflow collaboration enactment infrastructure with the run-time agreement fulfilment services. Various formats of collaboration messages are defined according to their roles and the data they carry. GBBopen, an open source blackboard system, is utilised for the implementation of the message relay facility due to its suitability

for the task of distributed problem solving. An example is used to demonstrate how the system works at build- time.

# Chapter 8

## Testing and Evaluation

### 8.1 Introduction

Key enabling techniques of COWCO-Guru prototype system are put into testing and evaluation in this chapter. §8.2 introduces an evaluation system comprising eight criteria covering both collaboration build time and run time. §8.3 forms Case Study 1, in which the overall procedure is evaluated by two examples. §8.4 discusses Case Study 2 that groups five examples to evaluate all the seven build-time criteria. §8.5 describes how Case Study 3 uses two examples to evaluate Criterion 8 – the effectiveness of the run-time collaboration enactment infrastructure. §8.6 compares the COWCO-Guru approach with four other relevant approaches to reveal differences as well as similarities. §8.7 summarises the chapter.

### 8.2 Evaluation System

Before any evaluation task takes place, an evaluation system needs to be set up, which includes a set of criteria that are distilled from the challenges and requirements set earlier in this thesis. These criteria cover both the collaboration build time and run time.

#### 8.2.1 Evaluation Criteria at Collaboration Build Time

At collaboration build time, the proposed change should aim at reconciling the currently encountered discrepancy. Conducting change demands flexibility to a certain level. Having minimum impact requires initial processes to be preserved as much as possible. Options as well as the way they are presented should be human-centric, which can be measured by criteria including provision of decision guidance, ability of decision assessment and level of noise reduction during decision-making.

### **Overall Drive and Guidance**

When two processes are compared at collaboration build time, two possible results can be reached. If the two processes are found as compatible, no reconciliation needs to be conducted and they can proceed directly to collaboration enactment stage. However, when behavioural discrepancies occur, reconciliation options need to be generated and users' decisions are expected. In each case, the system should be able to guide users through different stages, help them to understand relevant reconciliation options, respond to their decisions and drive the negotiation process to reach one of the final results – an agreement or rejection.

#### **Criterion 1 – Process Preservation (General Principle)**

As mentioned in Definition 6-9 (Effective Adjustment with Minimum Impact), if any of the reconciliation option is accepted, it should be able to deliver effective change with minimum impact on the hosting processes in terms of control and data dependencies, i.e. preserving initial processes as much as possible.

#### **Criterion 2 – Users' Flexibility of Choice (General Principle)**

In a dynamic and changing environment, wherever adjustment is required, flexibility should be accompanied. Users' flexibility of choice puts users in control in deciding which one or many options will be accepted from an option list. The level of flexibility that users have is determined by the amount of independent options available on the list, which is decided by relevant control flow dependencies.

#### **Criterion 3 – Provision of Option Recommendation (Human Centric)**

Because of the preference of process preservation and the causal relations between activities in a process, a decision on a preceding activity is very likely to have implication on the construction of later options. Therefore, where appropriate, recommendation should be identified and attached to certain options based on previous decisions. The recommended options are in line with the principle of process preservation.

#### **Criterion 4 – Ability of Decision Assessment (Human Centric)**

With flexibility of choice, users are able to override any recommendation and change the process the way they want. However, due to the causal relationship and the resulting option

and decision dynamics mentioned in §6.4.5 - Option and Decision Dynamics, such flexibility needs to be closely monitored and safeguarded. Also, in order for users to make fully informed and truly desired decisions on any option at real time, decision assessment support is crucial. This is because it allows users to fully practice flexibility of choice, try out different decision combinations, see through resulting implications and evaluate different choices in order to decide which option to accept and which one to reject. For this purpose, comprehensive feedback information should be provided in a timely manner, which covers every implied prerequisite and/or result by a certain decision. Also, a two-phase trial-and-confirmation mechanism should be in place to make decision assessment a reality.

#### **Criterion 5 – User Interface (Human Centric)**

In order to communicate reconciliation options and recommendations, and accommodate a test bed, effective user interfaces need to be designed. It should be able to communicate with users effectively, i.e. allow users to express themselves freely and delivery the right information to users at the right time. Real-time reaction and relevance of feedback are among the most important.

#### **Criterion 6 – Noise Reduction of Options (Human Centric)**

Good quality of options should provide necessary information whilst reduce the level of noise in terms of complexity, confusion and potential mistake faced by users. Since any piece of extra information beyond sufficiency and necessity can be deemed as noise, measures are expected to reduce the noise to the lowest possible level by identifying unnecessary information, dealing with it in the background and keeping it out of the sight of users. Also, it is found that different arrangement and presentation of the same information can bring different levels of option noise. Hence, the most appropriate option arrangement in terms of sequence and combination of options needs to be identified and adopted.

#### **Criterion 7 – Correctness of Process Change (Correctness)**

After all the reconciliation options have been generated and presented to the user and all the decisions have been made with regard to the current discrepancy, initial processes are expected to undergo certain changes to reflect those decisions. There are several requirements that assure the correctness of such process change. Firstly, the change should

not leave any approved activity, or group of activities, disconnected from the main process. Secondly, due to operations being moved or removed, new route activities need to be introduced where appropriate to make the resulting process syntactically correct. Thirdly, redundant transitions should be eliminated to reduce complexity. Fourthly, data flow should be duly updated to maintain consistency with control flow changes and users' decisions. Lastly, the control and data flow changes should be carried out automatically where possible by following users' decisions without users' direct manipulation to reduce users' workload as well as the chance of human error.

Criteria 1 – 7 are summarised in Figure 8-1. The direction from the centre to the periphery represents the direction from principles to practical measures.

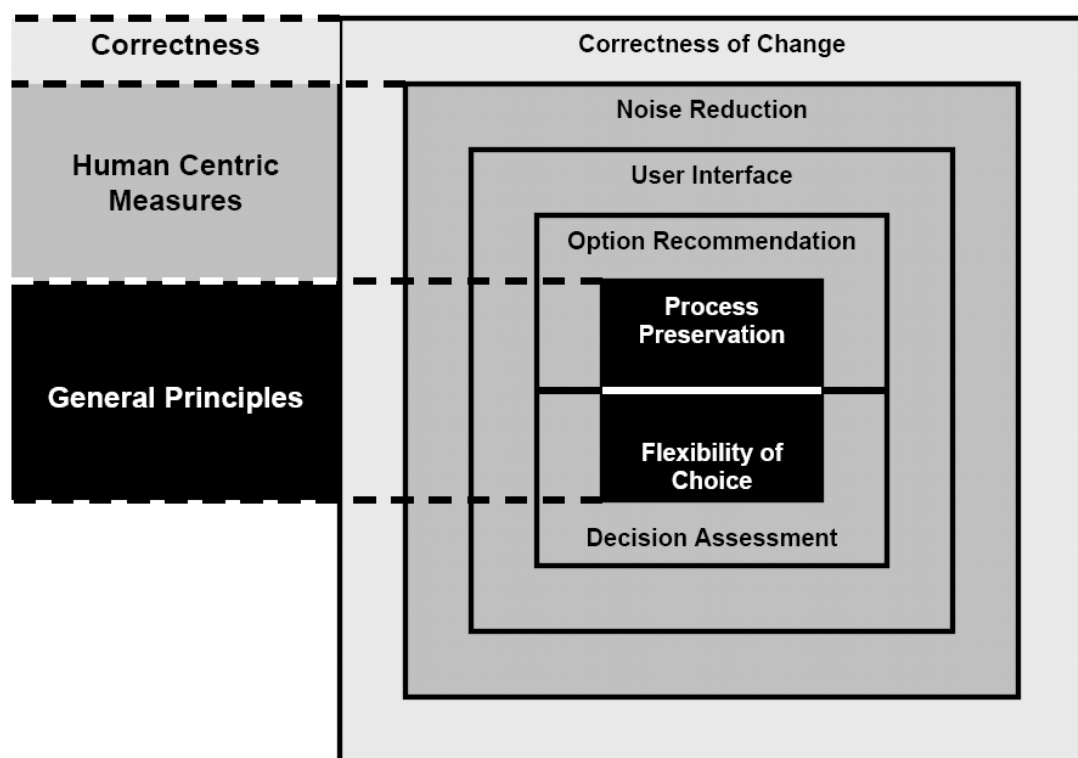


Figure 8-1. Evaluation criteria at collaboration build time.

### 8.2.2 Evaluation Criteria at Collaboration Run Time

At collaboration run time, since the main concern for the enactment infrastructure is to provide collaboration enactment support to any collaborative executable workflows in a distributed environment, the last criterion is identified as:

**Criterion 8** – the capability of the infrastructure to correctly enact and mediate collaboration between workflows that contain sequential, XOR branching, iterative and parallel structures.

### 8.2.3 Testing Plan

A set of evaluation criteria is summarised in Table 8-1 together with the cases and scenarios to be used for testing.

Table 8-1. Testing plan.

Criteria	Examples (Targeted Aspect)
<i>Collaboration Build Time</i>	
Overall Procedure	<ul style="list-style-type: none"><li>• C1.E1 (handling of compatible workflows)</li><li>• C1.E2 (handling of incompatible workflows)</li></ul>
Process Preservation (1) (control and data flow)	<ul style="list-style-type: none"><li>• C2.E1.DS1 (option preparation)</li><li>• C2.E1.DS1 (preservation of rejected CMA/CMIA and disqualified CMA)</li><li>• C2.E2.DS1 (preserve control/data flow based on previous decisions)</li></ul>
Users' Flexibility of Choice (2)	<ul style="list-style-type: none"><li>• To be observed in all examples</li></ul>
Option Recommendation (3)	<ul style="list-style-type: none"><li>• C2.E2.DS1 (recommendation based on previous decisions)</li></ul>
Decision Assessment (4)	<ul style="list-style-type: none"><li>• C2.E1.DS1 (prerequisite implication – between non-CMA and CMA/CMIA)</li><li>• C2.E1.DS1 (prerequisite implication – between CMAs)</li><li>• C2.E1.DS1 (resulting implication –</li></ul>



	<p>following control flow adjustment rejection)</p> <ul style="list-style-type: none"> <li>• C2.E1.DS1 (resulting implication – following data flow adjustment rejection)</li> <li>• C2.E1.DS2 (independent options)</li> <li>• C2.E1.DS3 (interdependent options)</li> </ul>
User Interface (5)	<ul style="list-style-type: none"> <li>• To be demonstrated by the use of various dialogue boxes in all examples (dynamic option interface in terms of the Test Bed facility and real time feedback display)</li> </ul>
Noise Reduction (6)	<ul style="list-style-type: none"> <li>• C2.E3 (CMIA identification rule)</li> <li>• C2.E4 (non-duplication control flow decision making on CMA as predecessor of multiple CMIA's)</li> <li>• Examples in Figure 6-23 and 6-24 of §6.4.4.5 Sequence of Reconciliation Options</li> <li>• Scenarios identified in §6.4.6.3 Manual and Automated Decision Making</li> </ul>
Correctness of Process Change (7)	<ul style="list-style-type: none"> <li>• C2.E5 (insertion of route activity)</li> <li>• C2.E5 (elimination of redundant transition)</li> </ul>
<i>Collaboration Run Time</i>	
Correctness of enactment between a pair of compatible workflows with sequential, XOR branching, iterative and structures parallel (8)	<ul style="list-style-type: none"> <li>• C3.E1 (collaboration between compatible workflows with sequential, XOR branching, iterative and parallel structures)</li> <li>• C3.E2 (collaboration between compatible workflows with parallel structures)</li> </ul>
<p><i>Abbreviation Explained</i></p> <p><b>C:</b> Case Study; <b>E:</b> Example; <b>DS:</b> Decision Sequence</p> <p>E.g., decision sequence 1 of example 1 in case study 1 is represented as C1.E1.DS1.</p>	

### 8.3 Case Study 1: E-Commerce Case Study in Casual Trade

In this section, fictitious examples based on the transactions in international trade community are used to demonstrate the effectiveness (General Principles) of negotiation and reconciliation services. Emphasis is put on the alternate unilateral decision making procedures. Intermediate tasks such as ontology mapping and interface process representation conversion are omitted. In this case study, it is assumed that *Customer* takes the role of collaboration initiator and *Vendor* acts as the responder.

In §8.3.1, the example mentioned in §3.4 is used to demonstrate the handling of a pair of compatible business processes from *Vendor* and *Customer\_1*. In §8.3.2, a variation of the example that involves *Vendor* and *Customer\_2* explains how discrepancies are discovered and reconciled. For situations where the direction of message is crucial, [s] and [d] representing data supply and data demand respectively will be elaborated; otherwise, they will be omitted.

#### 8.3.1 Example 1: Compatible Workflows (C1.E1)

The first example in Case Study 1 uses the same case as described in §3.4. The interface processes extracted from both *Vendor* and *Customer\_1*'s initial business processes are shown in Figure 8-2. They are then converted to state transition systems and compared for the first time on the site of the collaboration responder - *Vendor*.

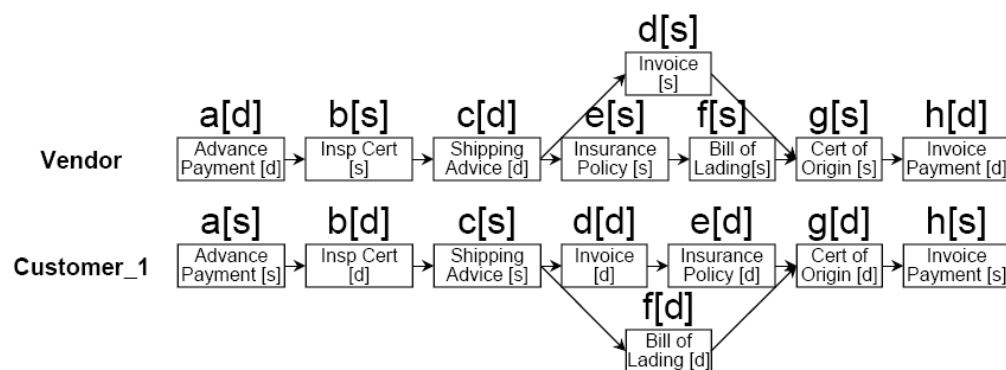


Figure 8-2. Interface processes extracted from Vendor and Customer\_1's initial workflows.

The comparison result is illustrated in Figure 8-3. For the purpose of a clearer representation in the figure, each message name borne by an interface activity is represented as a unique letter, e.g. *Advance Payment* represented by letter *a*, *Insp Cert* by letter *b* and so on. Grey shades mark common trace between the two state transition systems. A common trace can be seen from the start to the end, which means the two interface processes and their initial hosting workflows are compatible.

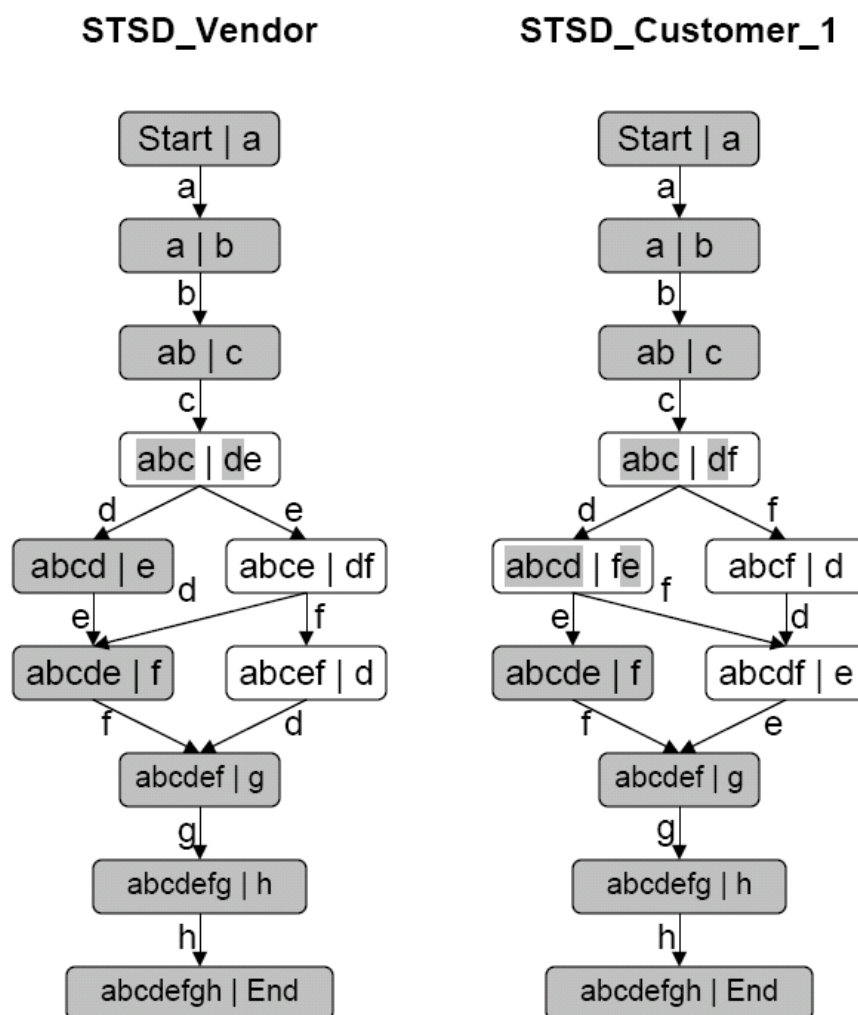


Figure 8-3. STS comparison and result for Vendor and Customer\_1.

### 8.3.2 Example 2: Incompatible Workflows (C1.E2)

Example 2 involves two partners, namely *Vendor* and *Customer\_2*. *Vendor*'s workflow is the same as the one in Example 1 but *Customer\_2*'s is different due to a different underlying

business process. Again, *Customer\_2* is an overseas importer. After effecting the advance payment to a desired vendor, it is waiting for the *Certificate of Origin*, which is required to acquire an import permit from its local authority. It then needs to review the *Inspection Certificate* issued by the vendor as a proof of the quality of the goods. As part of the duty of a Cost and Freight (C&F) trade term, *Customer\_2* needs to arrange the insurance itself and issue a *Shipping Advice* to inform the vendor to ship the goods. It then needs to use the *Commercial Invoice* received from the vendor to declare the goods at its Customs and waits for the *Bill of Lading* to get the goods from the shipper. After the goods are delivered, *Customer\_2*'s own inspection will be carried out. With its satisfaction of the result, final invoice payment will be approved and the rest of the invoice amount will be paid to the vendor. Both *Vendor* and *Customer\_2*'s workflow definitions are shown in Figure 8-4. Their interface processes are in Figure 8-5.

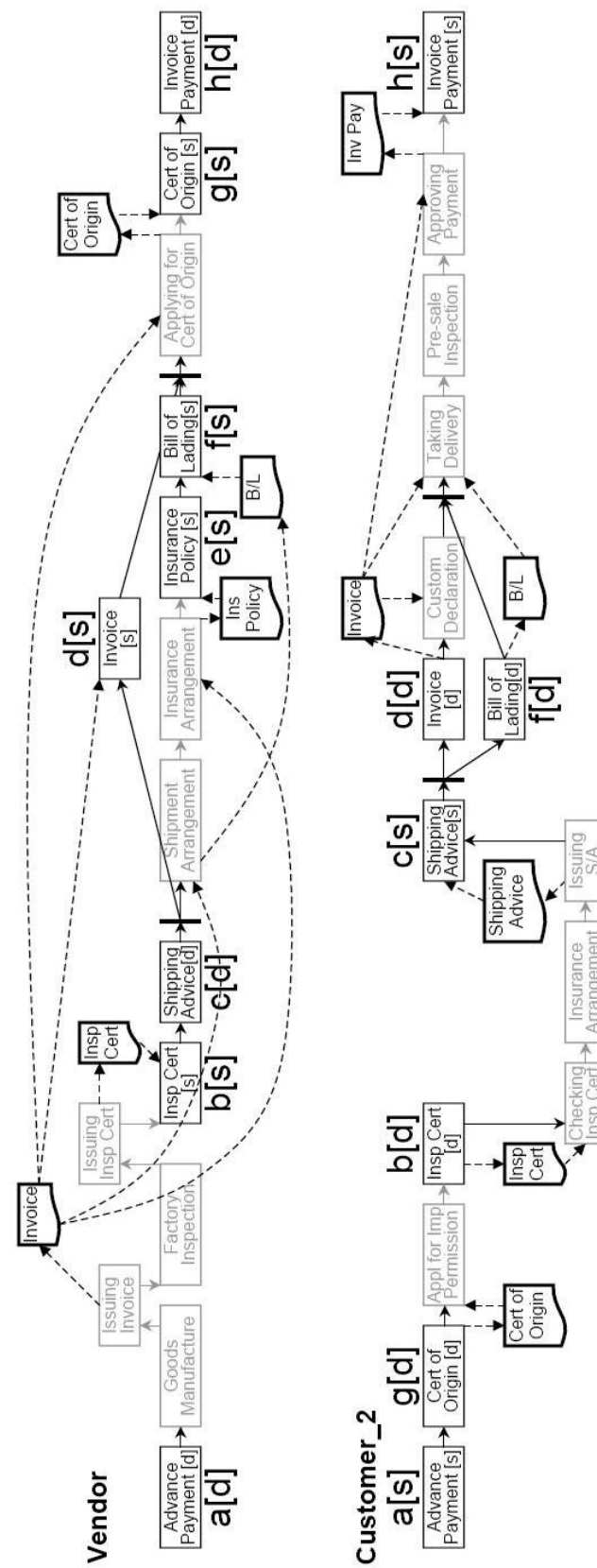


Figure 8-4. Example 2: Workflows of Vendor and Customer\_2.

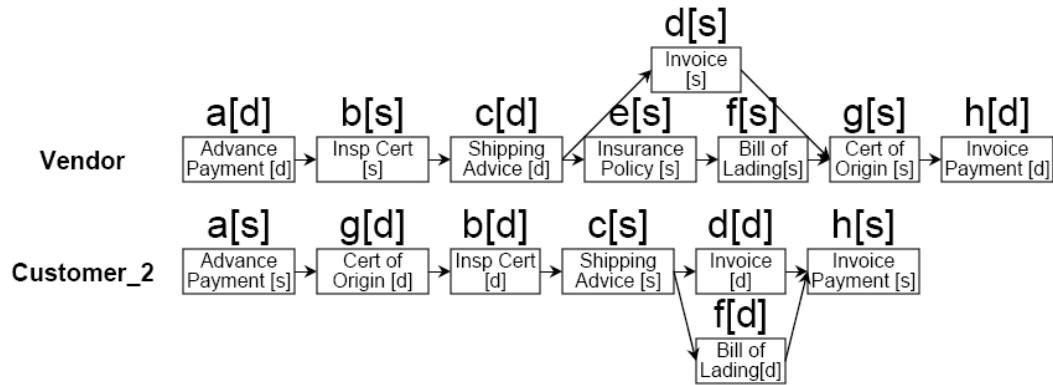


Figure 8-5. Interface processes extracted from Vendor and Customer\_2's initial workflows.

Pre-treatment immediately finds that *Vendor.Insurance Policy [s]* is a unique data supply interface activity (UDSIA) by following the procedure mentioned in §6.3.1.1 Unique Collaboration Message/Interface Activity. It is therefore removed automatically from *Vendor*'s initial process immediately. Its data predecessors *Vendor.Insurance Arrangement* and *Vendor.Issuing Invoice* are then prompted for the users to decide whether to remove them as well. In this example, since the only reason for the presence of *Vendor.Insurance Arrangement* is *Vendor.Insurance Policy [s]*, *Vendor.Insurance Arrangement* has no reason to remain given *Vendor.Insurance Policy [s]* being removed. For the activity *Vendor.Issuing Invoice*, because it supplies data (*Invoice*) not only to *Vendor.Insurance Arrangement* but to several other activities, it should remain. As a result, *Vendor*'s process is changed to *Vendor(1)* as shown in Figure 8-6. After being converted into state transition systems, the two processes are compared on the site of the collaboration responder - *Vendor*. The comparison result is illustrated in Figure 8-7.

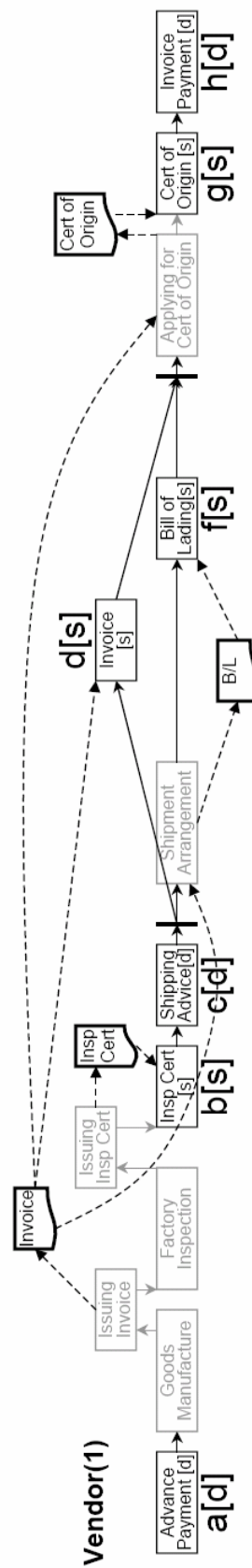


Figure 8-6. Process of Vendor(1).

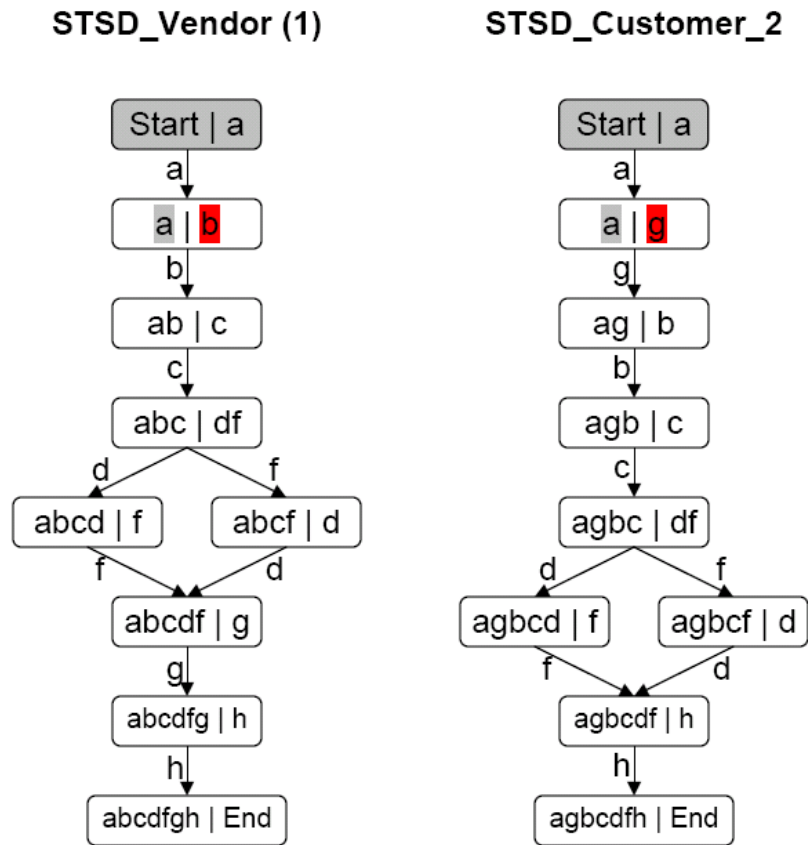


Figure 8-7. 1st round STS comparison result between Vendor(1) and Customer\_2.

A discrepancy occurs when *Vendor(1)* reaches *Insp Cert [s] (b[s])* and *Customer\_2* reaches *Cert of Origin [d] (g[d])*. A data flow relevance check (as mentioned in §6.4.2.2 – Candidate Moving Interface Activities and their Data Predecessors) for *Vendor(1)*’s *Cert of Origin [s] (g[s])* identifies *Vendor(1)*’s *Applying for Cert of Origin* is *Cert of Origin [s]’s* data predecessor. In order to reconcile the discrepancy, at *Vendor*’s site, the first reconciliation option (1st\_Option) is prompted as ‘Move *Applying for Cert of Origin* right in front of *Insp Cert [s] (b[s])*’.

If 1st\_Option is accepted, the following option (2nd\_Option) is prompted as ‘Move *Cert of Origin [s] (g[s])* right in front of *Insp Cert [s] (b[s])*’.

### Likely Result

If 2nd\_Option is accepted as well, the reconciliation adjustments will be committed, which results the *Vendor*’s business process *Vendor(2)* as shown in Figure 8-8. After the



concession made by *Vendor*, further comparison on *Vendor*'s site between the interface process of *Vendor(2)* and *Customer\_2* reveals no more discrepancy and a pair of compatible business process is reached. It can proceed to collaboration enactment stage.

### **Unlikely Result**

Otherwise, if 2nd\_Option is rejected, no concession has been made in *Vendor*'s current unilateral decision-making cycle and no reconciliation adjustment will be committed. *Vendor(1)* is kept the same. A counter offer – the interface process of *Vendor*'s current business process *Vendor(1)* – is prepared and sent to *Customer\_2*'s site. Then the same decision-making process takes place there. Although in practice, it is unlikely for users to reject 2nd\_Option given the previous decision that 1st\_Option is accepted, user still has the flexibility to say no based on new information.

Otherwise, if 1st\_Option is rejected, the following option (2nd\_Option) is prompted in the form of a data flow option as 'In order for the negotiation to carry on, since none option is accepted, data dependency from *Applying for Cert of Origin* to *Cert of Origin [s] (g[s])* should be removed if your current decision is confirmed.'

### **Unlikely Result**

If 2nd\_Option is accepted, the data dependency between *Applying for Cert of Origin* and *Cert of Origin [s] (g[s])* will be discarded. However, in this example, it is unlikely for users to make such a decision because with out *Applying for Cert of Origin*, there will be no *Cert of Origin* let alone having it sent.

### **Likely Result**

Otherwise, if 2nd\_Option is rejected, *Cert of Origin[s] (g[s])* will be automatically set as 'non-moveable' in control flow (as mentioned in §6.4.6.3 – Manual and Automated Decision Making), which marks the end of the decision making process. Since no concession has ever been made during *Vendor*'s unilateral decision making, *Vendor(1)* is kept the same and a counter offer – the interface process of *Vendor*'s current business process *Vendor(1)* – is prepared and sent to *Customer\_2*'s site. Then the same decision making process takes place there.

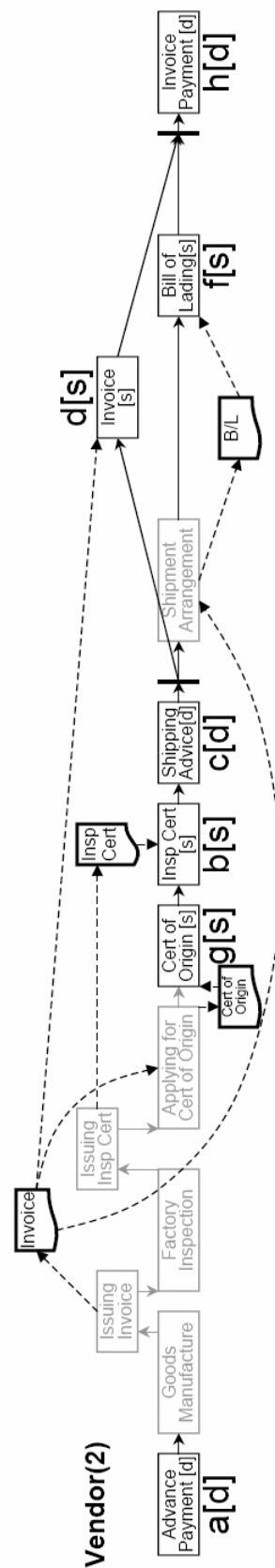


Figure 8-8. Process Vendor(2).

Examples in §8.3.1 and §8.3.2 demonstrate the effectiveness of COWCO-Guru’s ability to deal with both compatible and incompatible situations. However its scale advantage might not be clear yet. Imagine if *Vendor* has hundreds of customers, and each has a somewhat different workflow or their workflows are constantly changing, the value of the support provided by COWCO-Guru will be very obvious.

## 8.4 Case Study 2: Process Negotiation and Reconciliation

In this case study, a series of abstract examples are constructed to evaluate key enabling techniques against criteria (1) – (7) set in §8.2.1 Evaluation Criteria at Collaboration Build Time. For the purpose of clarity, collaboration messages are identified as single characters (e.g. a, b, c etc.) whilst the Host’s private activities are identified as ‘H’ followed by numbers (e.g. H1, H2, H3 etc.). For situations where the direction of message is crucial, [s] and [d] representing data supply and data demand respectively will be elaborated; otherwise, they will be omitted. Unless stated otherwise, the legends are adopted according to the arrangement shown in Figure 8-9. The meaning of the abbreviations are: CCPIA – current common preceding interface activity, DBA – downstream boundary activity, CMA – candidate moving activity and CMIA – candidate moving interface activity.

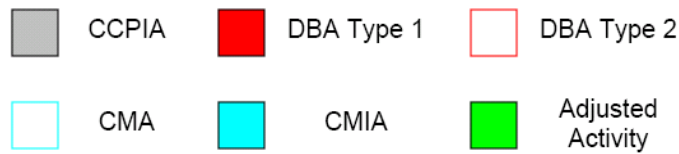


Figure 8-9. Legends adopted in the examples.

### 8.4.1 Examples

Examples that will be used in this section are all described in this subsection and will be referred to later in the discussion.

#### 8.4.1.1 Example 1 (C2.E1)

Example 1 is depicted in Figure 8-10. Current discrepancy occurs, as shown in the figure, when the Host’s STS reaches the state {abfdgh | c} and the Guest reaches {abfdgh | e}. Therefore, the candidate moving interface activity for the Host is identified as *H.e*. By

following the rules on the sequence dimension (§6.4.3.1 – Sequence Perspective), hypothetical target positions (HTP) are identified as:

- *HTP1*: Right before *H.c*,
- *HTP2*: Right after *H6*,
- *HTP3*: Right after *H11*,
- *HTP4*: Right after *H12*.

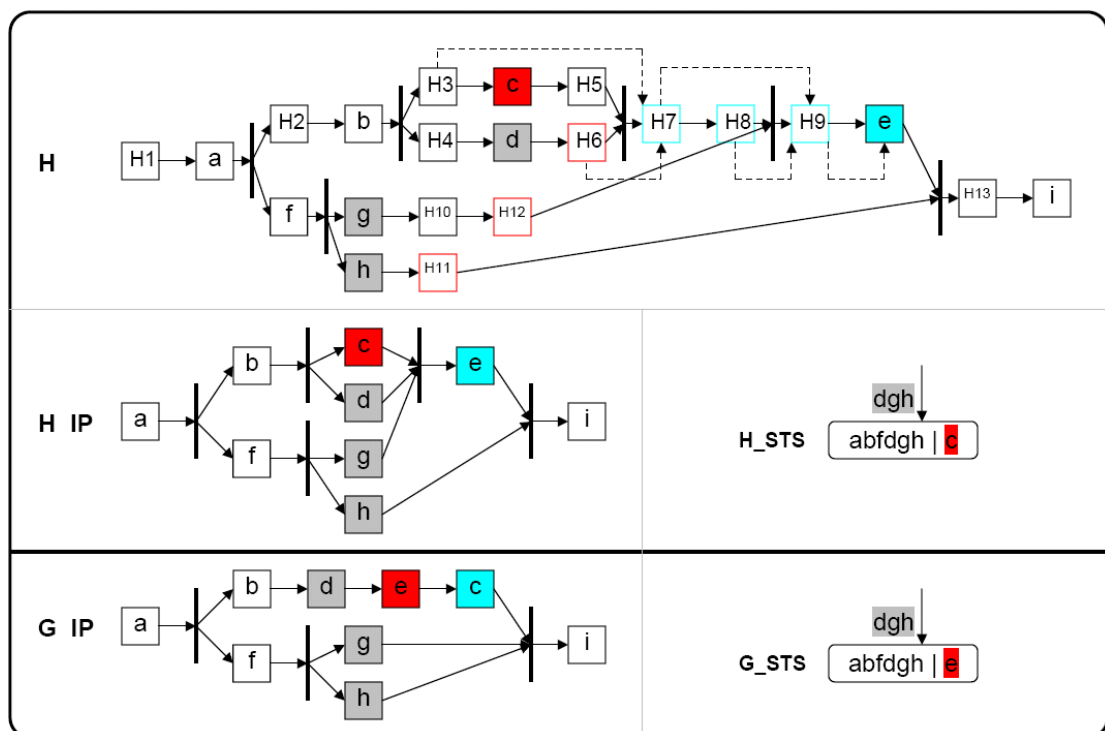


Figure 8-10. Case 2 Example 1 – CMAs and Their HTPs.

The sample decisions for Example 1 are listed in the decision sequence shown in Table 8-2, Table 8-3 and Table 8-4.

Table 8-2. Decision Sequence 1 of Example 1 (C2.E1.DS1).

Option ID	Option	Implication
<b>Candidate Moving Activity (CMA): <i>H7</i></b>		
1	[✓] <i>H7</i> should be put right before <i>c</i> ( <i>HTP1</i> ).	As a prerequisite, data dependency from <i>H6</i> to

2	[x] <i>H7</i> should be put right after <i>H6</i> ( <i>HTP2</i> ).	<i>H7</i> needs to be removed based on current decision(s).
Candidate Moving Activity (CMA): <i>H8</i>		
1	[x] <i>H8</i> should be put right before <i>c</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	In order for the negotiation to carry on, since none option is accepted, data dependency from <i>H8</i> to <i>H9</i> should be removed if current decision(s) are confirmed.
2	[x] <i>H8</i> should be put right after <i>H6</i> ( <i>HTP2</i> ).	
Implied Data Successor Adjustment Option: <i>H8</i>		
1	[√] Data dependency from <i>H8</i> to <i>H9</i> should be removed.	
Candidate Moving Activity (CMA): <i>H9</i>		
1	[x] <i>H9</i> should be put right before <i>c</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	With regard to the provisionally accepted CMA – <i>H7</i> , as a prerequisite, data dependency from <i>H7</i> to <i>H9</i> needs to be removed based on current decision(s).
2	[√] <i>H9</i> should be put right after <i>H6</i> ( <i>HTP2</i> ).	
3	[√] <i>H9</i> should be put right after <i>H12</i> ( <i>HTP4</i> ).	
Candidate Moving Interface Activity (CMIA): <i>e</i>		
1	[x] <i>e</i> should be put right before <i>c</i> ( <i>HTP1</i> ).	
2	[√] <i>e</i> should be put right after <i>H6</i> ( <i>HTP2</i> ) – <b>Recommended.</b>	

3	[√] <i>e</i> should be put right after <i>H12</i> ( <i>HTP4</i> ) – <b>Recommended.</b>	
---	---	--

Table 8-3. Decision Sequence 2 of Example 1 (C2.E1.DS2).

Option ID	Option	Implication
Candidate Moving Activity (CMA): <i>H7</i>		
1	[√] <i>H7</i> should be put right before <i>c</i> ( <i>HTP1</i> ).	As a prerequisite, data dependency from <i>H6</i> to <i>H7</i> needs to be removed based on current decision(s).
2	[x] <i>H7</i> should be put right after <i>H6</i> ( <i>HTP2</i> ).	
Candidate Moving Activity (CMA): <i>H8</i>		
1	[x] <i>H8</i> should be put right before <i>c</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	
2	[√] <i>H8</i> should be put right after <i>H6</i> ( <i>HTP2</i> ).	
Candidate Moving Activity (CMA): <i>H9</i>		
1	[x] <i>H9</i> should be put right before <i>c</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	With regard to the provisionally accepted CMA – <i>H7</i> , as a prerequisite, data dependency from <i>H7</i> to <i>H9</i> needs to be removed based on current decision(s).
2	[√] <i>H9</i> should be put right after <i>H6</i> ( <i>HTP2</i> ) – <b>Recommended.</b>	
3	[√] <i>H9</i> should be put right after <i>H12</i> ( <i>HTP4</i> ).	
Candidate Moving Interface Activity (CMIA): <i>e</i>		
1	[√] <i>e</i> should be put right before <i>c</i> ( <i>HTP1</i> ).	

2	[√] <i>e</i> should be put right after <i>H6</i> (HTP2) – <b>Recommended.</b>	
3	[√] <i>e</i> should be put right after <i>H12</i> (HTP4) – <b>Recommended.</b>	

Table 8-4. Decision Sequence 3 of Example 1 (C2.E1.DS3).

Option ID	Option	Implication
Candidate Moving Activity (CMA): <i>H7</i>		
1	[√] <i>H7</i> should be put right before <i>c</i> ( <i>HTP1</i> ).	As a prerequisite, data dependency from <i>H6</i> to <i>H7</i> needs to be removed based on current decision(s).
2	[x] <i>H7</i> should be put right after <i>H6</i> ( <i>HTP2</i> ).	
Candidate Moving Activity (CMA): <i>H8</i>		
1	[√] <i>H8</i> should be put right before <i>c</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	
2	[√] <i>H8</i> should be put right after <i>H6</i> ( <i>HTP2</i> ).	
Candidate Moving Activity (CMA): <i>H9</i>		
1	[x] <i>H9</i> should be put right before <i>c</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	Options with regard to <i>H6</i> ( <i>HTP2</i> ) should be rejected as well.  With regard to the provisionally accepted
2	[√] <i>H9</i> should be put right after <i>H6</i> ( <i>HTP2</i> ) – <b>Recommended.</b>	

3	[√] <i>H9</i> should be put right after <i>H12</i> ( <i>HTP4</i> ).	CMA – [ <i>H7</i> , <i>H8</i> ], as a prerequisite, data dependency from [ <i>H7</i> , <i>H8</i> ] to <i>H9</i> needs to be removed based on current decision(s).
<b>Candidate Moving Interface Activity (CMIA): <i>e</i></b>		
1	[√] <i>e</i> should be put right before <i>c</i> ( <i>HTP1</i> ).	
2	[√] <i>e</i> should be put right after <i>H6</i> ( <i>HTP2</i> ).	
3	[√] <i>e</i> should be put right after <i>H12</i> ( <i>HTP4</i> ) – <b>Recommended.</b>	

#### 8.4.1.2 Example 2 (C2.E2)

Example 2 is depicted in Figure 8-11. Current discrepancy occurs, as shown in the figure, when the Host's STS reaches the state  $\{ab \mid c\}$  and the Guest reaches  $\{ab \mid d\}$ . Therefore, the candidate moving interface activity for the Host is identified as *H.d*. By following the rules on the sequence dimension (§6.4.3.1 – Sequence Perspective), HTPs are identified as:

- *HTP1*: Right before *H.c*.

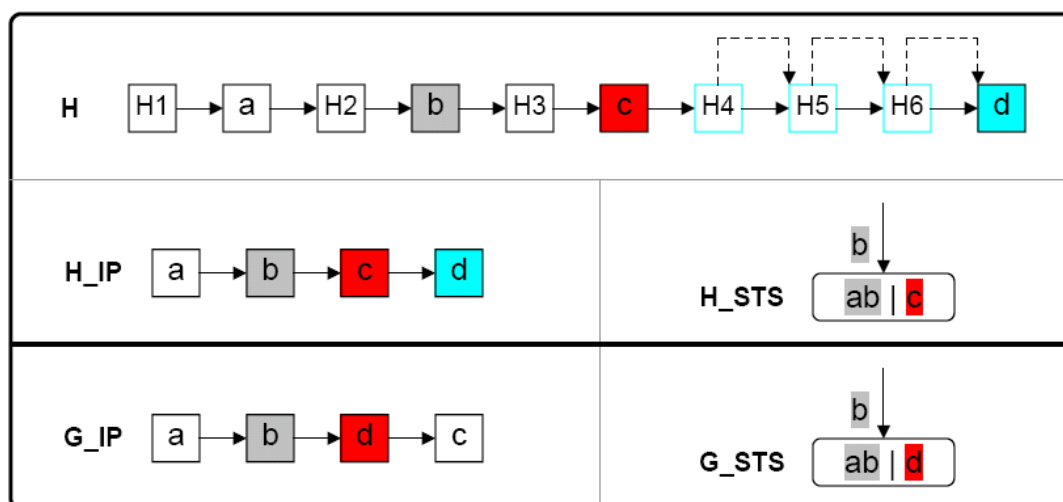


Figure 8-11. Case 2 Example 2.



The sample decisions for Example 2 are listed in the decision sequence shown in Table 8-5.

Table 8-5. Decision Sequence of Example 2 (C2.E2.DS1).

Option ID	Option	Implication
<b>Candidate Moving Activity (CMA): <math>H4</math></b>		
1	[√] $H4$ should be put right before $c$ (HTP1).	
<b>Candidate Moving Activity (CMA): <math>H5</math></b>		
1	[x] $H5$ should be put right before $c$ (HTP1) – <b>Recommended.</b>	In order for the negotiation to carry on, since none option is accepted, data dependency from $H5$ to $H6$ should be removed if current decision(s) are confirmed.
<b>Implied Data Successor Adjustment Option: <math>H5</math></b>		
1	[√] Data dependency from $H5$ to $H6$ should be removed.	
<b>Candidate Moving Activity (CMA): <math>H6</math></b>		
1	[√] $H6$ should be put right before $c$ (HTP1).	
<b>Candidate Moving Interface Activity (CMIA): <math>d</math></b>		
1	[√] $d$ should be put right before $c$ (HTP1) – <b>Recommended.</b>	

#### 8.4.1.3 Example 3 (C2.E3)

Example 3 is depicted in Figure 8-12. Current discrepancy occurs, as shown in the figure, when the Host's STS reaches the state  $\{ab \mid d\}$  and the Guest reaches  $\{ab \mid ce\}$ . Therefore, by applying the upstream independent rule (mentioned in §6.4.2.2 Candidate Moving

Interface Activities and their Data Predecessors), the candidate moving interface activity for the Host is identified as *H.c*. By following the rules on the sequence dimension (§6.4.3.1 – Sequence Perspective), HTPs are identified as:

- *HTP1*: Right before *H.d*.

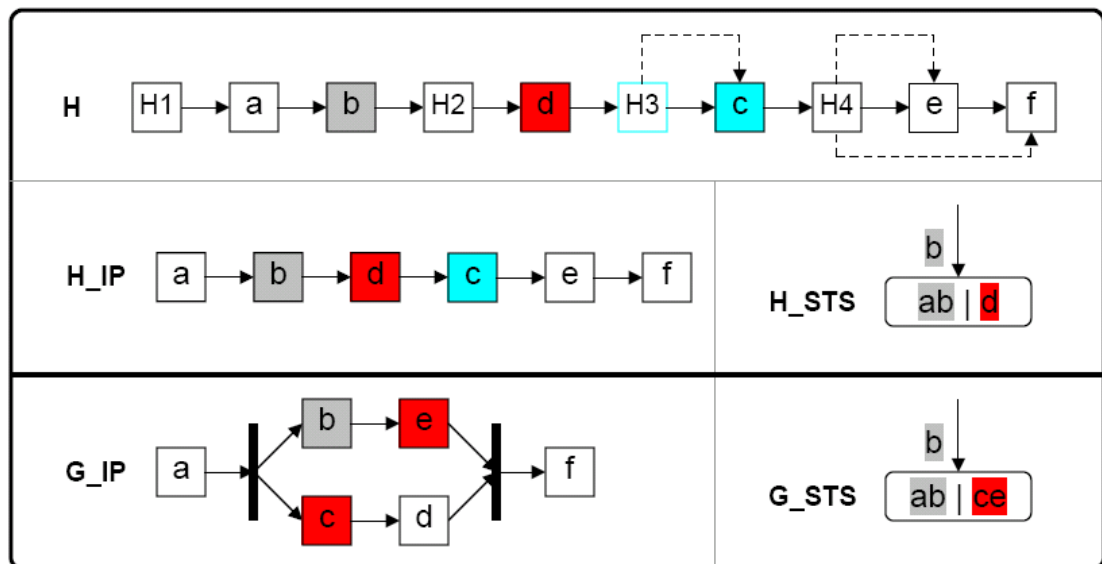


Figure 8-12. Case 2 Example 3.

The sample decisions for Example 3 are listed in the decision sequence shown in Table 8-6.

Table 8-6. Decision Sequence of Example 3 (C2.E3.DS1).

Option ID	Option	Implication
<b>Candidate Moving Activity (CMA): <i>H3</i></b>		
1	[✓] <i>H3</i> should be put right before <i>d</i> ( <i>HTP1</i> ).	
<b>Candidate Moving Interface Activity (CMIA): <i>H.c</i></b>		
1	[✓] <i>c</i> should be put right before <i>d</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	

#### 8.4.1.4 Example 4 (C2.E4)

Example 4 is depicted in Figure 8-13. Current discrepancy occurs, as shown in the figure, when the Host's STS reaches the state  $\{a \mid b\}$  and the Guest reaches  $\{a \mid cd\}$ . Therefore, by applying the upstream independent rule (mentioned in §6.4.2.2 Candidate Moving Interface Activities and their Data Predecessors), the candidate moving interface activity for the Host is identified as  $H.c$  and  $H.d$ . By following the rules on the sequence dimension (§6.4.3.1 – Sequence Perspective), HTPs are identified as:

- *HTP1*: Right before  $H.b$ .

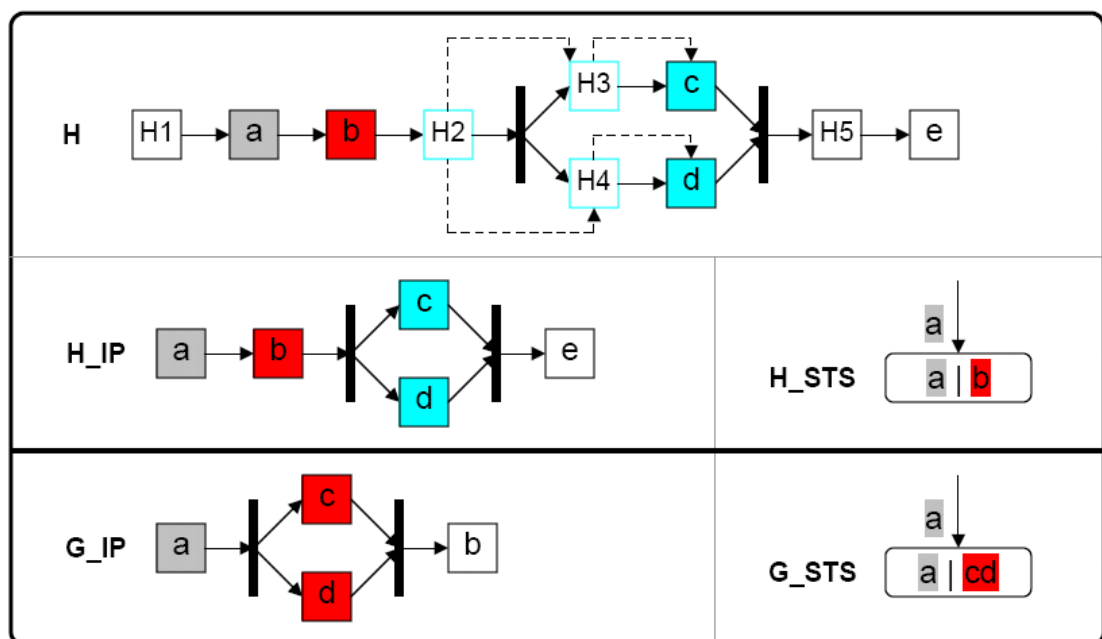


Figure 8-13. Case 2 Example 4.

The sample decisions for Example 4 are listed in the decision sequence shown in Table 8-7.

Table 8-7. Decision Sequence of Example 4 (C2.E4.DS1).

Option ID	Option	Implication
<b>Candidate Moving Activity (CMA): <i>H2</i></b>		
1	[x] <i>H2</i> should be put right before <i>b</i> ( <i>HTP1</i> ).	In order for the negotiation to carry on, since none option is accepted, data dependency from <i>H2</i> to [ <i>H3</i> , <i>H4</i> ] should be removed if current decision(s) are confirmed.
<b>Implied Data Successor Adjustment Option: <i>H2</i></b>		
1	[√] Data dependency from <i>H2</i> to <i>H3</i> should be removed. [√] Data dependency from <i>H2</i> to <i>H4</i> should be removed.	
<b>Candidate Moving Activity (CMA): <i>H3</i></b>		
1	[√] <i>H3</i> should be put right before <i>b</i> ( <i>HTP1</i> ).	
<b>Candidate Moving Interface Activity (CMIA): <i>c</i></b>		
1	[√] <i>c</i> should be put right before <i>b</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	
<b>Candidate Moving Activity (CMA): <i>H4</i></b>		
1	[√] <i>H4</i> should be put right before <i>b</i> ( <i>HTP1</i> ).	
<b>Candidate Moving Interface Activity (CMIA): <i>d</i></b>		
1	[√] <i>d</i> should be put right before <i>b</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	

### 8.4.1.5 Example 5 (C2.E5)

Example 5 is depicted in Figure 8-14. Current discrepancy occurs, as shown in the figure, when the Host's STS reaches the state  $\{a \mid bc\}$  and the Guest reaches  $\{a \mid d\}$ . Therefore, the candidate moving interface activity for the Host is identified as  $H.d$ . By following the rules on the sequence dimension (§6.4.3.1 – Sequence Perspective), HTPs are identified as:

- *HTP1*: Right before  $H.b$ ,
- *HTP2*: Right before  $H.c$ .

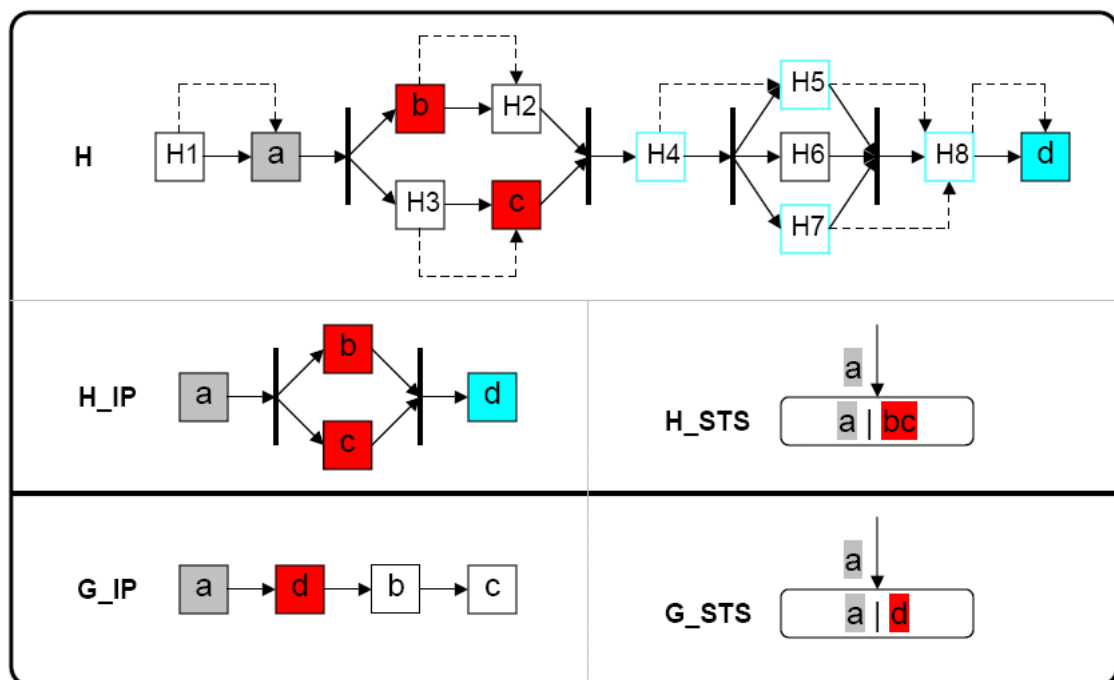


Figure 8-14. Case 2 Example 5.

The sample decisions for Example 4 are listed in the decision sequence shown in Table 8-8.

Table 8-8. Decision Sequence of Example 5 (C2.E5.DS1).

Option ID	Option	Implication
<b>Candidate Moving Activity (CMA): <math>H4</math></b>		
1	[√] $H4$ should be put right before $b$ ( <i>HTP1</i> ).	
2	[√] $H4$ should be put right before $c$ ( <i>HTP2</i> ).	

Candidate Moving Activity (CMA): <i>H5</i>		
1	[√] <i>H5</i> should be put right before <i>b</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	
2	[√] <i>H5</i> should be put right before <i>c</i> ( <i>HTP2</i> ) – <b>Recommended.</b>	
Candidate Moving Activity (CMA): <i>H7</i>		
1	[x] <i>H7</i> should be put right before <i>b</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	In order for the negotiation to carry on, since none option is accepted, data dependency from <i>H7</i> to <i>H8</i> should be removed if current decision(s) is confirmed.
2	[x] <i>H7</i> should be put right before <i>c</i> ( <i>HTP2</i> ) – <b>Recommended.</b>	
Implied Data Successor Adjustment Option: <i>H7</i>		
1	[√] Data dependency from <i>H7</i> to <i>H8</i> should be removed.	
Candidate Moving Activity (CMA): <i>H8</i>		
1	[√] <i>H8</i> should be put right before <i>b</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	
2	[√] <i>H8</i> should be put right before <i>c</i> ( <i>HTP2</i> ) – <b>Recommended.</b>	
Candidate Moving Interface Activity (CMIA): <i>d</i>		
1	[√] <i>d</i> should be put right before <i>b</i> ( <i>HTP1</i> ) – <b>Recommended.</b>	

2	[√] <i>d</i> should be put right before <i>c</i> ( <i>HTP2</i> ) – <b>Recommended.</b>	
---	--	--

## 8.4.2 Control Flow Preservation

Efforts and results of control flow preservation can be observed on several occasions in C2.E1, e.g. when preparing reconciliation options and handling rejected/disqualified CMA after users' decision-making.

### 8.4.2.1 Effort in Reconciliation Option Preparation

#### Between CMA and HTP

The hypothetical target positions (*HTP1* – *HTP4*) are the most downstream positions available in the currently identified destination area. Recall the 'Make a cup of tea' example introduced in §6.4.3.1, the more downstream an activity is kept in an adjustment, the more the control flow is preserved, the less disruption such an adjustment will cause, and the more likely such an option and the implied concession will be accepted. The choices of HTPs on the sequence dimension preserve the control flow as much as possible, which contributes to a higher likelihood of acceptance to these options.

Since *H7*, *H8* and *H9* are on the data flow path of *H.e* – the CMIA, and located after the current destination area, they are identified as candidate moving activities (CMA) according to §6.4.2.2 – Candidate Moving Interface Activities and their Data Predecessors. For each CMA/CMIA, relevant HTP(s) are identified in order to formulate corresponding reconciliation options. This is done by following the rules of preservation on connectivity dimension of control flow (introduced in §6.4.3.2 – Connectivity Perspective), which is able to maintain the control flow dependency in terms of connectivity relations between a CMA/CMIA and its corresponding HTP(s) unless users decide otherwise. For *H7*, since in initial process definition, only *HTP1* and *HTP2* can reach it, they are identified as *H7*'s HTPs. Similarly, *HTP1* and *HTP2* are identified as *H8*'s HTPs, and *HTP1*, *HTP2* and *HTP4* are identified as *H9* and *H.e*'s HTPs. Because *HTP3* can reach none of the current CMAs, it is excluded from any reconciliation options.

### Between CMAs

In C2.E1.DS1, after the decisions on the reconciliation options regarding CMA *H7* have been made and as a result *H7* is provisionally accepted as a moving activity, CMA *H8*'s reconciliation options need to reflect this progress. Since in the initial process definition, *H8* is a control flow successor of *H7*, such a dependency is preferred to sustain unless users decide otherwise. In order to maintain the control flow dependency, at least *H8* needs to follow *HTP1* because *H7* has been set to follow *HTP1*, which sets the Option[1] for *H8* – “*H8* should be put right before *c (HTP1)*” as a recommended option for users to consider acceptance with priority.

#### 8.4.2.2 Result after User Decision Making

##### Preservation of Rejected CMA

In C2.E1.DS1, since both options for *H8* are rejected, *H8* becomes a rejected CMA. It is therefore excluded from the final list of moving activities and stay untouched during process change stage following user decision making.

##### Result after User Decision Making – Preservation of Disqualified CMA

In C2.E1.DS1, although *H7* is accepted in an earlier stage, due to the later rejection decision to Option[1] for *H9* ([x] *H9* should be put right before *c (HTP1)* – **Recommended**), the data dependency between *H7* and *H9* is discarded with users' consent. Base on such a data flow change, *H7* is disqualified as a CMA and thus is excluded from the final list of moving activities. COWCO-Guru's real-time response is able to pick up the data flow update and spare *H7* from being adjusted. As a result, relevant control flow (*H5* → *H7* → *H8*) is preserved.

#### 8.4.3 Data Flow Preservation

Data flow preservation is observed on several occasions in Example 1, e.g. when preparing reconciliation options and after users' decision-making.

##### 8.4.3.1 Effort in Reconciliation Option Preparation

In C2.E1.DS1, taking *H7* for example, because *H3* and *H6* are data predecessors of *H7* in the destination area, the proposed adjustment should avoid the introduction of any new data flow conflict between the CMA and activities in the destination area. This means *H7*'s ideal



location in the destination area should be at least after *H3* and *H6*. The HTP identification rule states that an HTP should be identified at the latest available position in its destination area, which ensures that an HTP is positioned after any activity in the current CMA's destination area and thus after any data predecessor (*H3* and *H6*) of *H7*. As a result, no new data flow conflict will occur by following the proposed moves and initial data flow can be preserved (Criterion 1).

#### **8.4.3.2 Result after User Decision Making**

Assuming the same decision sequence in C2.E1.DS1 is followed as shown in Table 8-2, due to rejection towards Option[2] for *H7*, the data dependency from *H6* to *H7* is set to be removed if the adjustment of *H7* is executed after the decision-making process. However, since later decisions (rejection to Option[1] for *H9*) result in the cancellation of *H7*'s move, the data dependency between *H6* and *H7* is restored and preserved at the end automatically with no conscious effort from the user.

#### **8.4.4 Considering Previous Decisions**

In order to assist users with decision-making, recommendations can be attached to reconciliation options under certain circumstances indicating the acceptance of the option complies with some initial control flow dependency based on the decisions made so far and thus has a higher priority to be accepted. Marking an option as 'recommended' needs two prerequisites. Firstly, relevant HTPs should appear in the accepted options for provisionally accepted CMAs that immediately precede the current CMA/CMIA. Secondly, the immediately preceding CMAs should still be qualified as conceded target activities (CTA). By 'qualified', it means so far there has not been any decision having been made that results in the removal of the provisionally accepted CMA(s) from the current CMIA's preceding data flow.

In C2.E2.DS1, it can be observed that Option[1] for *H5* is marked as recommendation. This is due to the acceptance of *H4* and for the purpose of preserving the initial control flow dependency between *H4* and *H5*. It can also be observed that although *H4* is the provisionally accepted CMA that immediately precedes *H6*, Option[1] for *H6* is not marked as 'recommended'. This is because in earlier decisions, *H5* is rejected and the data dependency between *H5* and *H6* is discarded, which breaks both *H4* and *H5* away from

*H.d*'s preceding data flow. As a result, *H6* no longer has any provisionally accepted preceding CMA and thus no option for *H6* should be recommended.

#### 8.4.5 User Flexibility

The number of reconciliation options available to a CMA/CMIA is determined by control flow connectivity between the CMA/CMIA and its preceding HTPs in the initial workflow definition. On the one hand, such a rule provides users with the choice that fully complies with the initial control flow dependencies between the CMA/CMIA and all of its HTPs as desired by process preservation (Criterion 1) if all the available options are accepted. On the other hand, users still possess the right to make changes to the initial control flow dependencies by rejecting certain or all option(s), which provides users with the flexibility (Criterion 2) to respond to new issues occurred. For example in C2.E1, as a CMA, *H7* has two options:

[√] *H7* should be put right before *c* (*HTP1*)

[√] *H7* should be put right after *H6* (*HTP2*).

By default, they are all accepted, which would make *H7* succeed both *HTP1* and *HTP2*, and thus complies with initial control flow dependencies between *H7* and *HTP1*, *HTP2* respectively. Based on users assessment, either or both of the options can be rejected by users in order to suit for certain business requirement.

#### 8.4.6 Implication Accompanied by User Flexibility

Users are flexible in decision-making and free to change the initial process. However, due to the causal relationship and the resulting option and decision dynamics mentioned in §6.4.5 - Option and Decision Dynamics, such flexibility needs to be closely monitored and safeguarded. This leads to the *Test Bed* function that supports users' decision-making, i.e. before any decision is confirmed, whenever a decision or a combination of decisions implies certain prerequisite or resulting change to initial control/data flow dependencies, such implications are made available to users in real time for assessment purposes (Criterion 4). Guiding information generated by the *Test Bed* can be observed in the form of prerequisite implication and resulting implication.

### **Prerequisite Implication – Between Non-CMA and CMA/CMIA**

In C2.E1.DS1, for *H7*, when Option[1] is accepted but Option[2] is rejected, *H7* is provisionally set as a moving activity and should succeed *H3* but not *H6* in control flow. However, this is in direct conflict with the data dependency between *H6* and *H7* in initial workflow definition. In order to uphold the current decision, data dependency between *H6* and *H7* should be discarded as prompted by the Guru system:

“As a prerequisite, data dependency from *H6* to *H7* needs to be removed based on current decision(s).”

Being faced with such prerequisite implication, if users think the data dependency between *H6* and *H7* can be discarded, they can stick to their current decision. Otherwise, if they think *H7* must rely on the data produced by *H6*, they should alter their decision.

### **Prerequisite Implication – Between CMAs**

In C2.E1.DS1, for *H9*, when Option[2] and Option[3] are accepted but Option[1] is rejected, *H9* is set to move but should not be put onto the control flow path that contains *H3* as predecessor. However, since following a previous decision (on Option[1] for *H7*), *H7* has been provisionally accepted as a moving activity and set as *H3*’s immediate successor in control flow, *H9* cannot not be put onto the control flow path coming from *H7*. As a result, data flow dependency between *H7* and *H9* must be removed. Therefore, a prerequisite data flow implication is prompted as:

“With regard to the provisionally accepted CMA – *H7*, as a prerequisite, data dependency from *H7* to *H9* needs to be removed based on current decision(s).”

Being faced with this prerequisite implication, if users think the data dependency between *H7* and *H9* can be discarded, they can stick to their current decision. Otherwise, if they think *H9* must rely on the data produced by *H7*, they should alter their decision.

### **Resulting Implication – Following Control Flow Adjustment Rejection**

In C2.E1.DS1, since both options for *H8* are rejected, *H8* cannot be moved upstream. As *H8*’s data successor, if *H9* is about to be moved before *H8*, the data dependency between them should be discarded as prompted by the Guru:

“Please note: In order for the negotiation to carry on, since none option is accepted, data dependency from *H8* to *H9* should be removed if your current decision is confirmed.”

Being faced with this resulting implication, if users think the data dependency between *H8* and *H9* can be discarded, they can stick their current decision. Otherwise, if they think *H9* must rely on the data produced by *H8*, they should try to alter their decision by accepting at least one option for *H8*. If that is not an option either, they should reject all the options and leave the decision to the following step where data flow adjustment options are prompted.

### **Resulting Implication – Following Data Flow Adjustment Rejection**

In C2.E1.DS1, following the rejection of both of the control flow adjustment options for *H8*, a data flow adjustment option is prompted as “Data dependency from *H8* to *H9* should be removed”. Different from the decision made in Table 8-2, if users insist that *H9* should still rely on the data produced by *H8* and thus to retain the data dependency, they need to reject this option. The rejection implies that *H9* should not be moved upstream either in order to keep the data dependency. That is to say the possibility of accepting *H9* as a moving activity is no longer available as prompted in the resulting implication:

“In order to avoid further control flow restrictions, all the above options need to be accepted. Otherwise, *H9* will be set as a non-moveable activity if current decision(s) are confirmed.”

### **Independent and Interdependent Options**

Decision Sequence 2 of Example 1 (C2.E1.DS2) illustrates the scenario of independent options. For CMA *H9*, both recommended Option[1] and Option[2] are independent options. Decision on one of the options does not imply certain decision to the other. The underlying reason is: by following the decisions made on *H7* and *H8*, *H9*’s provisional immediate predecessors are *H7* and *H8* that appear on the routes of *HTP1* (right before *H.c*) and *HTP2* (right after *H6*) respectively. Rejection to *H9* taking the route of either HTP only denies *H9* to reside on the same route as either *H7* or *H8*.

On the contrary, in Decision Sequence 3 of the same example (C2.E1.DS3), interdependent options are encountered. For CMA *H9*, recommended Option[1] and Option[2] are interdependent because the decision of acceptance on Option[1] demands the same decision

on Option[2] and vice versa. The underlying reason is: by following the decisions made on *H7* and *H8*, *H9*'s provisional immediate predecessor is *H8* that spans across parallel routes of *HTP1* (right before *H.c*) and *HTP2* (right after *H6*). Rejection to *H9* taking the route of either *HTP1* or *HTP2* denies *H9* to reside on the same route as *H8*, which implies *H9* should not appear after the other HTP either. For example, when only Option[1] is rejected, the system's response to the interdependent options is to prompt:

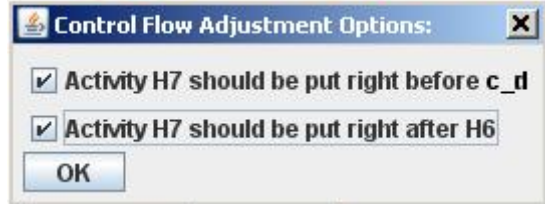

“Decisions on options with regard to *H6* (*HTP2*) should be in consistency with the current decision on the option regarding *H.c*”.

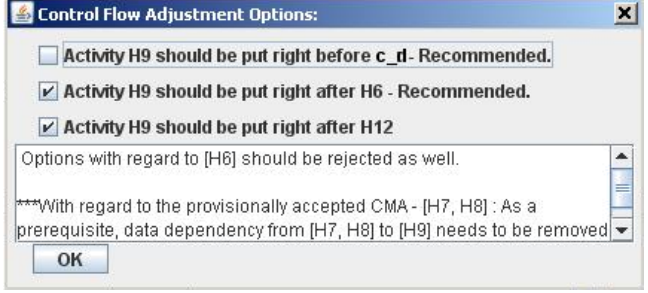
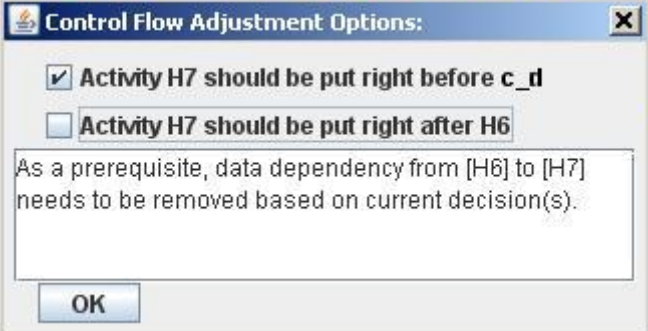
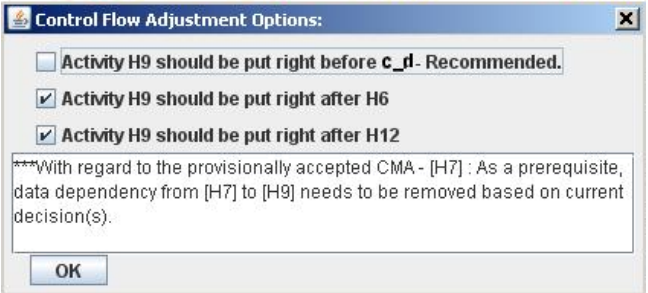
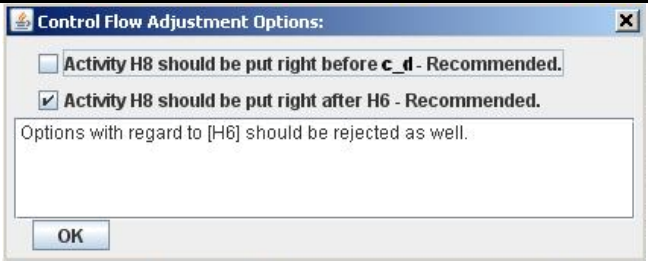
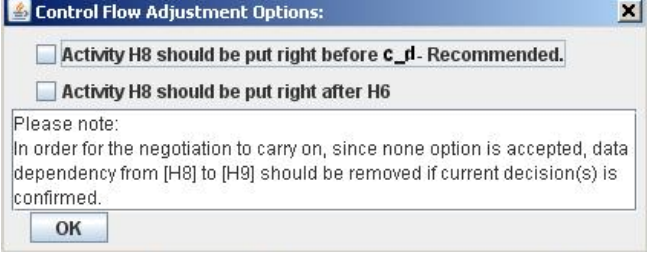
Also, from the two scenarios, it can be seen users' previous decisions can shape succeeding options.

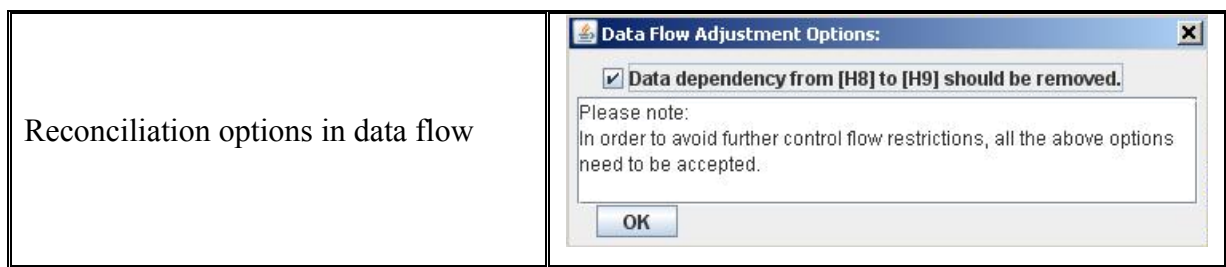
### 8.4.7 User Interface

The most important part of the user interface is the active dialogue box. It not only prompts users options and recommendations but also acts as a test bed for users to try out different decisions, gain a thorough understanding of the dynamics of reconciliation, assess different choice and make up their minds to confirm their decisions. The real-time feedback is displayed in an active text area. Dialogue boxes for different situations are summarised in Table 8-9.

Table 8-9. Dialogue boxes.

Situation	Dialog Box
Reconciliation options in control flow	
Option with recommendation	

Feedback in active text area	 <p>Control Flow Adjustment Options:</p> <p><input type="checkbox"/> Activity H9 should be put right before c_d- Recommended.</p> <p><input checked="" type="checkbox"/> Activity H9 should be put right after H6 - Recommended.</p> <p><input checked="" type="checkbox"/> Activity H9 should be put right after H12</p> <p>Options with regard to [H6] should be rejected as well.</p> <p>***With regard to the provisionally accepted CMA - [H7, H8] : As a prerequisite, data dependency from [H7, H8] to [H9] needs to be removed</p> <p>OK</p>
Data dependency violation (between CMA and data predecessor before DBA)	 <p>Control Flow Adjustment Options:</p> <p><input checked="" type="checkbox"/> Activity H7 should be put right before c_d</p> <p><input type="checkbox"/> Activity H7 should be put right after H6</p> <p>As a prerequisite, data dependency from [H6] to [H7] needs to be removed based on current decision(s).</p> <p>OK</p>
Data dependency violation (between CMAs)	 <p>Control Flow Adjustment Options:</p> <p><input type="checkbox"/> Activity H9 should be put right before c_d- Recommended.</p> <p><input checked="" type="checkbox"/> Activity H9 should be put right after H6</p> <p><input checked="" type="checkbox"/> Activity H9 should be put right after H12</p> <p>***With regard to the provisionally accepted CMA - [H7] : As a prerequisite, data dependency from [H7] to [H9] needs to be removed based on current decision(s).</p> <p>OK</p>
Interdependent options	 <p>Control Flow Adjustment Options:</p> <p><input type="checkbox"/> Activity H8 should be put right before c_d- Recommended.</p> <p><input checked="" type="checkbox"/> Activity H8 should be put right after H6 - Recommended.</p> <p>Options with regard to [H6] should be rejected as well.</p> <p>OK</p>
Data flow implication (after rejection of current CMA)	 <p>Control Flow Adjustment Options:</p> <p><input type="checkbox"/> Activity H8 should be put right before c_d- Recommended.</p> <p><input type="checkbox"/> Activity H8 should be put right after H6</p> <p>Please note: In order for the negotiation to carry on, since none option is accepted, data dependency from [H8] to [H9] should be removed if current decision(s) is confirmed.</p> <p>OK</p>



## 8.4.8 Noise Reduction

### 8.4.8.1 Identification of CMIA

Example 3 (C2.E3) justifies the CMIA identification rule introduced in §6.4.2.2 Candidate Moving Interface Activities and their Data Predecessors. According to the rule, only *H.c* is identified as the current CMIA rather than both *H.c* and *H.e*. By following the decision sequence shown in Table 8-6, *H3* and *H.c* are put in front of *H.d*, which makes the resulting process compatible with the Guest's. By dealing with *H.c* alone as a CMIA first, not only less complexity is introduced, but also initial process is preserved from potential unnecessary change.

### 8.4.8.2 Non-Duplication Control Flow Decision Making

Example 4 (C2.E4) depicts a scenario where a preceding CMA (*H2*) is shared as a data predecessor by succeeding CMAs (*H3* and *H4*) that are acting as data predecessors for different CMIA (*H.c* and *H.d*) respectively. In this case, from the decision sequence log, it can be seen *H2* is prompted only once, which prevents inconsistent control flow decision regarding *H2* from being made and thus eliminates potential confusion that otherwise could be passed on to the users.

### 8.4.8.3 Sequence of Reconciliation Option

According to the reasons explained in §6.4.4.5 Sequence of Reconciliation Options, the sequence of reconciliation options should be arranged in the order of CRIA-related → CMIA-related → CIIA-related. The underlying principle is to reduce the noise level by decreasing the process complexity as early as possible and increasing it as late as possible. Examples found in Figure 6-23 and 6-24 are referred to, which verify the legitimacy of such an arrangement.

#### 8.4.8.4 Automated Decision Making

As mentioned in §6.4.6.3 Manual and Automated Decision Making, several scenarios qualify automated decision-making. When any of these is encountered, they will be dealt with without demanding users' conscious effort. The scenarios and their purposes are summarised in Table 8-10.

Table 8-10. Scenarios and purposes of automated decision making.

Scenarios of Automated Decision Making	Purposes
Eliminating unique data supply interface activity during activity bipartite comparison	To save users' effort
Rejecting option regarding data demand interface activity acting as data predecessor in CMA/CMIA set	To save users' effort
Subsequent rejections in control and data flow reconciliation options	To prevent contradicting decisions from being made

#### 8.4.9 Correctness of Change

From the result of Example 5 (C2.E5) as shown in Figure 8-15, it can be seen there is no isolated subsections. Two route activity *R1* and *R2* are inserted when *H4* and *H.d* are removed and relocated to ensure the syntactical correctness of the resulting process. A redundant transition from *H3* to *H.c* is identified and removed. Data flow dependency from *H7* to *H8* is removed to echo users' decisions on the rejection of *H7*'s suggested movement and the acceptance of discarding data flow dependency from *H7* to *H8*. All other data flow dependencies are intact. During the course, it can be seen there is no user involvement.



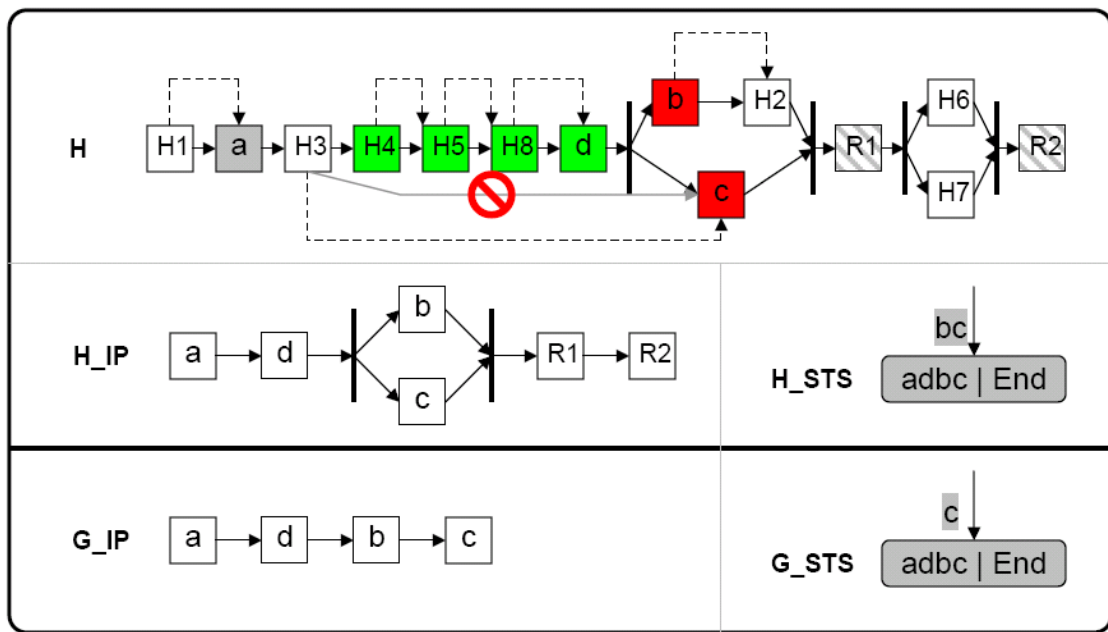


Figure 8-15. Result of Example 5 (C2.E5).

#### 8.4.10 Exceptional Circumstances

In theory, there exists a type of scenarios where the processes still need to go through the reconciliation procedure although there is neither possible occurrence of unnecessary delay nor enactment deadlock. Such an example is shown in Figure 8-16.

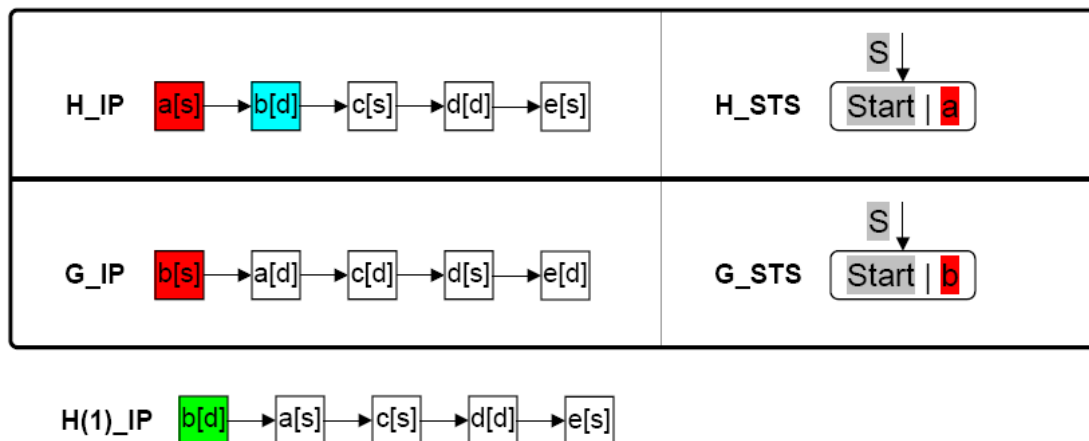


Figure 8-16. Exceptional Circumstances.

Clearly, the two initial business processes based on the interface processes *H\_IP* and *G\_IP* can collaborate with each other smoothly with no need of undergoing any reconciliation adjustment. With a closer look, firstly it can be seen the first interface activity of both sides are of data supply type. Although in theory such an arrangement is possible, in practice based on the different roles required in a particular collaboration, it is a relatively rare situation. Secondly, even with the unnecessary reconciliation options and possible adjustment following such options, the result as shown in *H(1)* will not introduce any negative effect (e.g. deadlock or unnecessary delay) into the collaboration apart from the change of the initial process. In fact, what the adjustment (moving *H.b[d]* upstream) means for *H* is simply to get hold of the required data earlier than previously, which should be accepted by users.

## 8.5 Case Study 3: Collaboration Enactment

In this case study, the capability of Guru Run Time (Criterion 8) is demonstrated and evaluated through two examples. Example 1 (C3.E1) looks into the collaboration enactment between two processes with sequential, exclusive OR branching, iterative and parallel structures. Example 2 further examines processes with parallel structures. It is assumed that all the collaborative processes used in this case study are compatible regardless how they are constructed – by either following Guru Build Time or being defined from scratch.

### 8.5.1 Example 1 (C3.E1): Sequential, XOR Branching, Iterative and Parallel Structures

#### 8.5.1.1 Example Description

In this example, a manufacturer (*Manufacturer*) exports a component to a retailer (*Retailer*). *Manufacturer*'s sales process starts by waiting for the arrival of a *Quotation Enquiry*, shown as *QE*, from a potential retailer. After receiving the enquiry, *Manufacturer*'s component database is checked and relevant information, especially the price of the component, is retrieved and put into a *Quotation Result*, shown as *QR*. After sending the result to *Retailer*, the process starts waiting for *Retailer*'s decision on whether extra information is needed. If the *Extra Information Request*, shown as *Extra?*, received from *Retailer* indicates some *Extra Information*, shown as *ExtraInfo*, is needed, *Manufacturer* prepares the information and sends it to *Retailer*. With *Retailer*'s decision regarding '*Extra?*' indicating no more

extra information needed and its process moving on to purchase order preparation, *Manufacturer's* process leaves the iteration and starts waiting for a *Purchase Order*, shown as *PO*, from *Retailer*. On receipt of the *PO*, *Manufacturer* verifies it. Once the *PO* is satisfactory, an *Acceptance Notice* of the *Purchase Order*, shown as *POAcpt*, is sent to *Retailer*. Then *Manufacturer* starts its manufacturing activity. After the *Commercial Invoice*, shown as *Invoice*, is issued, it is sent to *Retailer*. Simultaneously, shipment and insurance are arranged, which is followed by the sending of the *Bill of Lading*, shown as *B/L*, and the *Insurance Policy*, shown as *Ins Policy*, as fulfillment of the CIF (Cost, Insurance and Freight) trade term on *Manufacturer's* side. At last, *Manufacturer* waits for the invoice payment, shown as *Inv Pay*. The activity diagram is depicted in Figure 8-17, which includes sequential, XOR branching, iterative and parallel structures.

In *Retailer's* purchasing process, a *Quotation Enquiry* is composed for a certain component and sent to *Manufacturer*. After a *Quotation Result* is received, *Retailer* evaluates the result and decides whether further information is needed. If more information is required, it sends an *Extra Information Request* to *Manufacturer*. On receipt of the *Extra Information*, the quotation is re-evaluated. The iteration of requesting extra information stops when *Retailer* is satisfied with the result. It then starts preparing a *Purchase Order*. When the order is ready, it is sent to *Manufacturer*. After receiving *Manufacturer's Acceptance Notice of Purchase Order*, *Retailer* needs to use the *Commercial Invoice* received from *Manufacturer* to declare the goods at its Customs and waits for the *Bill of Lading* to get the goods from the shipper. Since *Retailer* also adopts CIF as its trade term, the original copy of the *Insurance Policy* is expected to arrive too. After the goods are delivered, *Retailer* carried out its own inspection. At last, upon satisfaction of the inspection, *Retailer* effects the payment as per the *Commercial Invoice*. The activity diagram is shown in Figure 8-17, which includes all four structures as well.

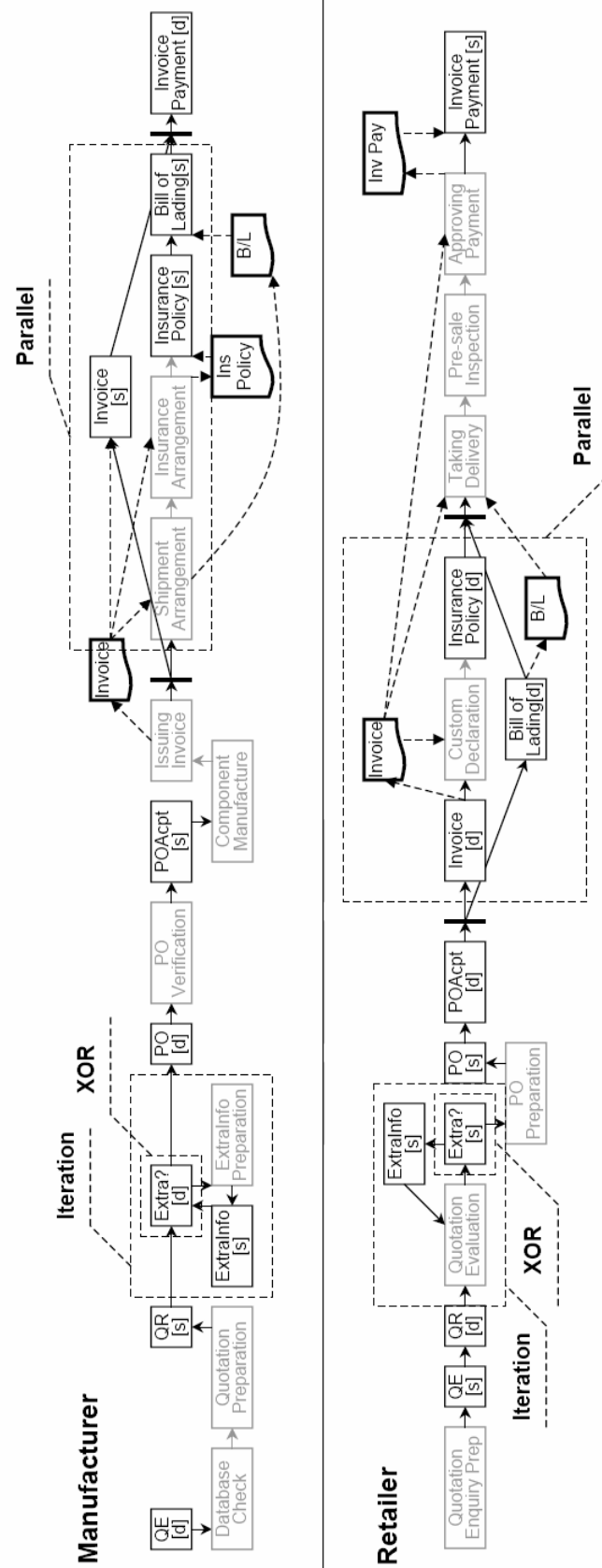


Figure 8-17. Example 1 (C3.E1): Manufacturer and Retailer.

### 8.5.1.2 Findings

The two processes are defined in XPDL and instantiated by two Shark workflow engines on two separate computers. The collaboration server of Guru Run Time is started and running on a third machine. Having been configured to communicate with the server, clients of Guru Run Time pass collaboration messages back and forth between Shark workflow engines and the server. The enactment follows all the three structures as defined in the processes. In order to distinguish different rounds of the iterative structure, an iteration flag is updated based on the reasoning of the message log on the server and passed to users through the client of Guru Run Time. Both the data request message (containing plain text) and data supply/response messages (containing either plain text or document as attachment) can be observed passed back and forth correctly from the document (or text message) received on both sides. Therefore, it is concluded that Guru Run Time that is implemented in the client-server architecture is capable of dealing with sequential, XOR branching and iterative structures contained in a pair of compatible workflow definitions. For parallel structures, it is running well in this example but some discoveries have been made and will be explained in §8.5.2 Example 2 (C3.E2): Revisit Parallel Structures.

### 8.5.2 Example 2 (C3.E2): Revisit Parallel Structures

Example 2 (C3.E2) is constructed based on Example 1 (C3.E1). In *Manufacturer's* process, before the *Insurance Policy* and *Bill of Lading* are sent, a *Shipping Advice* is expected from *Retailer*. In *Retailer's* process, before receiving the *Bill of Lading* from *Manufacturer*, *Retailer* needs to inform *Manufacturer* to start the shipment by sending a *Shipping Advice*. The activity diagrams are shown in Figure 8-18. Although the two processes are still compatible following this alteration, deadlock appears during collaboration enactment when *Manufacturer's* sales process has reached *Shipping Advice[d]* but not *Invoice[s]* yet and *Retailer's* purchase process has reached *Invoice[d]* but not *Shipping Advice[s]* yet. After careful investigation and tests, the reason for the deadlock is discovered as neither the invalidity of the compatibility definition nor the incapability of the collaboration enactment infrastructure but the fact that Shark's implementation of invocation of parallel activities is based on single thread. Therefore, when the current parallel branch is blocking, no other branches can be executed. This is a design defect of the Shark workflow engine.

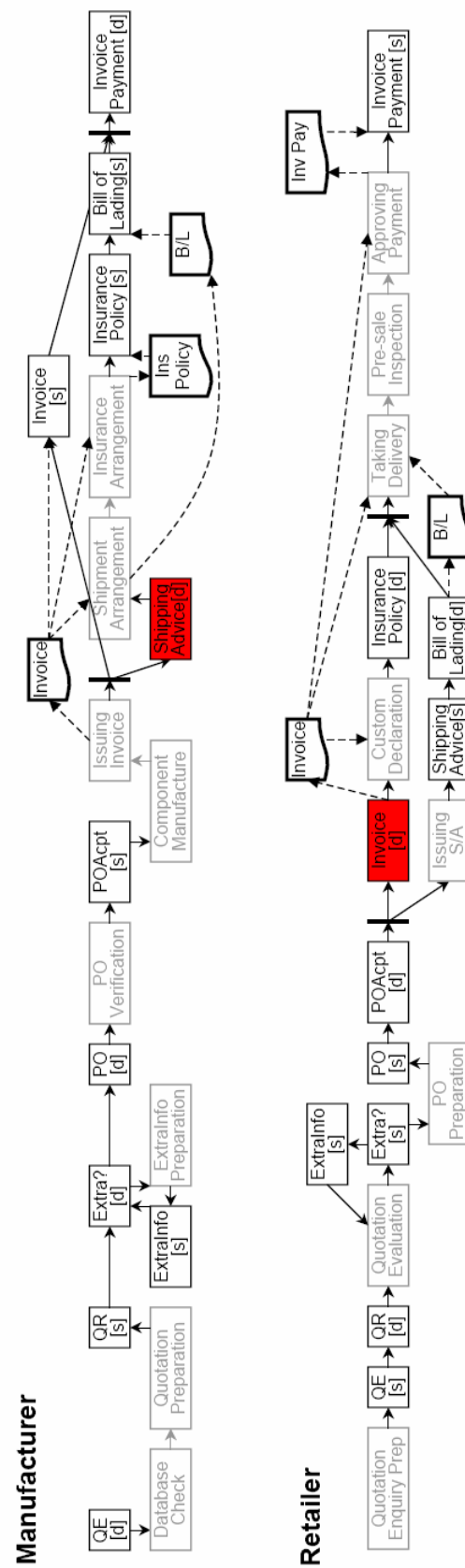


Figure 8-18. Example 2 (C3.E2): revisit parallel structures.

## **8.6 Approach Comparison**

COWCO-Guru is designed to provide process comparison and reconciliation support at collaboration build time and enactment support at run time, which are named as Guru Build Time and Guru Run Time respectively. To the best knowledge of the author, there are no other approaches or systems that provide B2B collaboration support in such a comprehensive manner covering the two stages. In this section, comparisons are made between Guru and the other approaches/systems introduced in Chapter 3 with key similarities as well as differences highlighted.

### **8.6.1 openXchange Project Vs Guru**

openXchange (Krukkert, 2002) is similar to Guru only in the way that process comparison is conducted at build time. The key differences are: firstly, openXchange Project approaches B2B collaboration from a service discovery angle. Based on the discussion about negotiation approaches in §4.4.3 Approaches of Negotiation, it can be seen that service discovery is actually an implementation of the mutual gain negotiation approach. Compared with Guru's combined approach of mutual gain and concession convergence, openXchange is unable to deal with situations where incompatible processes are encountered, i.e. no common interface activity sequence exists. Secondly, when compatible processes are encountered, openXchange creates a new common process based on the commonly identified activity sequences, which is viewed as unnecessary by Guru. Thirdly, openXchange does not provide collaboration enactment solution at run time as Guru Run Time does.

### **8.6.2 Workflow View Approach Vs Guru**

At build time, the differences are: firstly, Schulz's (2002) approach needs the support of an explicitly pre-defined coalition workflow. The defining procedure is not covered by the approach itself. Secondly, the mapping and dependency from coalition workflow through workflow views to private workflows means that the private workflows need to be created from scratch by using corresponding workflow view as guidance. Or, only if private workflows have already been compatible with their corresponding workflow views, they can be re-used, which, in reality, leaves the chance of reuse very slim.

At collaboration run time, the architecture proposed by Schulz is similar to Guru's as they are all supported by a message mediation facility. However, there is no further implementation available for the purpose of system evaluation.

### **8.6.3 InDiA Vs Guru**

At build time, firstly, Interoperability Dialogue and Agent (InDiA) (Biegus and Branki, 2004) needs the support of an explicitly pre-defined dialogue definition, which oversees and coordinates the collaboration. Secondly, the dialogue is defined by extending the process definition language adopted. As a result, interoperability during process enactment can be affected and current workflow engines need to be adjusted before they can interpret such an extended PDL. Similar to InDiA's send-type and wait-type activities, interaction points in Guru are modeled as data supply and data demand activities. However, the difference lies in that InDiA's send-type activity is defined as sending a message and blocking itself whilst Guru's data supply activity is defined as sending the associated message and moving on because there is no need to wait for any result from the coordination definition checking.

### **8.6.4 Automatic Process Negotiation Vs Guru**

The automatic process negotiation framework proposed by Bye, Piccinelli and Lamersdorf (2002) targets the collaboration build time. The key differences are firstly, the negotiation approach adopted by automatic process negotiation is mutual gain only and thus the range of situation it can deal with is limited. Secondly, based on the reconciliation techniques it adopted, only differences caused by different function of activity can be tackled through methods called blocking and hiding. Confined by such a reconciliation technique, only minor change can be made with all the activity sequencing related differences being left and marked as 'irreconcilable'. Lastly, it is not clear which process comparison technique is adopted to reveal the difference between the initial process and the automatically proposed process. As a result, it can be seen that the effectiveness of the automatic process negotiation is limited compared with Guru.



## **8.7 Summary and Conclusion**

In this chapter, an evaluation system is established, which contains eight criteria covering both collaboration build time and run time. A testing plan is outlined specifying which example will be used to evaluate which specific aspect of which criterion.

Case Study 1 tests the effectiveness of the prototype system at collaboration build time. Two fictitious examples based on the practice in trade community are provided to evaluate Guru's ability to cope with compatible as well as incompatible workflow collaboration scenarios. Having shown the ability to support individual collaboration effectively, it can be envisaged that the potential value of the Guru approach lies in the scale effect, where an organisation has hundreds of business partners and collaboration transactions to manage.

Change and preservation, flexibility and control, are contradicting pairs. They are treated as balanced issues in this thesis. Change starts in a preservation-oriented manner, e.g. options are formed based on initial control and data flow dependencies. Without flexibility, specific requirement and change cannot be accommodated. However, as flexibility is provided to users, measures in revealing implication of each potential change have to be developed and put into place to guide and safeguard users' decision-making. A number of scenarios found in five examples are given in Case Study 2 to demonstrate Guru's ability in coping with the dynamic interaction between change and preservation as well as between flexibility and control. All the build-time criteria (Criteria 1-7) are evaluated with satisfactory results.

In Case Study 3, collaboration enactments between compatible workflows are carried out. Except for a design defect found in Shark Workflow's workflow engine that affects workflows with parallel structures, all other aspects are running as expected. As a result, the prototype system passes the test of Criterion 8.

Finally, the Guru approach is compared with four other related approaches with (dis)similarities revealed.

# Chapter 9

## Conclusion and Future Work

### 9.1 Introduction

In this chapter, §9.2 reviews this thesis. §9.3 summarises main contributions. §9.4 outlines limitations and directions for future work. §9.5 gives an overall conclusion.

### 9.2 Thesis Review

The aim of this thesis was set as:

*How to provide comprehensive IT support for business collaboration in the form of an integrated cross-organisational workflow collaboration supporting framework, which comprises support for compatible business process acquisition at build time and a loosely-coupled infrastructure for workflow collaboration enactment at run time.*

Benefit from a globalised economy and the ubiquity of information technology, organisations are able to reach more collaborative partners and opportunities than ever before. However, the exponential growth in client base and the ever-changing nature of marketplace poses major challenges to organisations in terms of prompt and cost-effective response to collaboration opportunities. Having been equipped with workflow management systems, organisations are eager to see what role WfMSs can play in business collaboration. Previous approaches of workflow-based business collaboration are mainly developed following a series of isolated tasks, including conventional negotiation, implementation of the resulting agreement to form cross-organisational workflows, and subsequent workflow

enactment. The whole process of establishing workflow collaboration is viewed as expensive, resource-intensive and error-prone. It is envisaged that if IT support could be brought into business collaboration from an earlier stage, it would transform the conventional business collaboration experience in terms of shortening the response time to new collaboration opportunities and improving overall quality of business automation.

In order to achieve the aim of the thesis, workflow technology was reviewed, focusing on WfMC's generic workflow paradigm, activity model, workflow data model and workflow interoperability due to their relevance to the topic of workflow-based business collaboration.

For the purpose of having a clear understanding of how workflows from different organisations interact with each other, cross-organisational workflows were studied within the context of business collaboration. Three patterns of cross-organisational workflows were identified as hierarchical, composite and peer-to-peer. Also, previous techniques that tackle workflow collaboration were reviewed and categorised into several approaches. Build-time approaches include concrete process modelling, abstract interaction modelling, service discovery and automated reconciliation. Run-time approaches are centralised workflow enactment, sub-flow invocation, workflow case transfer and coordinated data exchange.

Based on the mismatch between the task of cross-organisation workflow collaboration and the previous approaches discovered from the reviews, requirements for new approaches and techniques were identified, which are loose coupling-based solution, negotiation support, decision-making support, process change support and unified collaboration enactment infrastructure.

To realise the requirements, an integrated approach was adopted, which takes into account of three parallel dimensions: workflow (build/run time and control/data flow), organisation (knowledge preservation, lean principle and user centric) and operating environment (distribution and dynamism).

As part of the preparation work, a new distributed workflow coordination mechanism based on loosely coupled data dependency was designed by modelling an interaction point as a

pair of interface activities. Negotiation was reviewed with the emphasis on negotiation principles, approaches and the operational aspect. Three key elements in the operational aspect were identified as interests, communication and options. A behaviour-oriented process matching technique was chosen to conduct interface process comparison after the introduction to generic comparison approaches and the review of process comparison techniques.

A comprehensive and novel framework was constructed to capture and organise the requirements as well as facilitate the integrated approach. Most importantly, the three key elements regarding negotiation operation were captured from a workflow perspective. Interests were represented as interface processes; communication was interpreted as a series of process comparison under the control of a workflow-driven negotiation protocol; options took the form of workflow change suggestions that could reconcile currently identified discrepancies. Logic components that compose the framework were decided as Collaboration Negotiation Block (comprising Process Matching, User Interaction and Process Change), Agreement Fulfilment Block (if an agreement is reached), and Collaboration Enactment Infrastructure (containing the message relay facility). The Framework is also viewed as service stack expanding across both partners and the message relay facility. The ways to adapt the Framework to different implementation architectures were described.

Following the identified requirements, the integrated approach and the comprehensive framework, key enabling techniques for workflow negotiation and reconciliation were explained in details. The interface process extraction technique enables the collaboration interests to be represented in the form of interface processes, which are used in the process comparison stage to reveal (dis)similarity between the two interface processes. In the case of discrepancy, reconciliation options need to be formulated and presented to users of the Host and decisions need to be gathered from the users. This is the most complicated but crucial step because the choice of target activities and destination areas as well as the understanding and capture of the dynamics between options and decisions directly relates to the quality of the end result of the reconciliation task. Finally, computerised process change operations further free users from the trivial but error-prone task, which echo's Michael Melenovsky's

view on the ultimate goal of workflow as returning business professionals the ability to modify their processes from IT organisations (Swenson, 2007).

The design and implementation of the prototype system, COWCO-Guru, was described with the functions of all the components explained. The workflow-based implementation of the negotiation protocol made the build-time negotiation services able to share a unified workflow collaboration enactment infrastructure with the run-time agreement fulfilment services. Various formats of collaboration messages were defined according to their roles and the data they carry. Collaboration message exchange mechanism was explained and GBBopen, an open source blackboard system, was utilised for the implementation of the message relay facility due to its suitability for the task of distributed problem solving.

Following the implementation of the prototype system, key enabling techniques were put into testing and evaluation. An evaluation system was set up, which comprises eight criteria covering collaboration build time and run time. These criteria are process preservation, users' flexibility of choice, provision of option recommendation, ability of decision assessment, user interface, noise reduction, correctness of process change, and effectiveness of run-time collaboration enactment. A testing plan was outlined to specify the criterion and example pairs. Three case studies were constructed and carried out to test and evaluate all the eight criteria together with the overall procedures. Also, the Guru approach was compared with four other approaches targeting similar research areas to reveal the differences.

### **9.3 Summary of Contributions**

The contributions of the thesis can be summarised as:

- A new modelling approach that implicitly specifies cross-organisational workflow interaction. It provides an alternative to conventional explicit flow modelling and suits the need for loose coupling in a decentralised and distributed environment.
- Through the abstraction of interface process and comparison of different criteria adopted by existing process compatibility definitions, a new workflow collaboration compatibility is defined.
- Clarify and enhance the understanding of techniques applied in process comparison.

- Capture key operational elements (interests, communication and options) of a business negotiation with workflow technology.
- A novel integrated negotiation approach that implements lean thinking. It is able to bring about a cost-effective negotiation outcome.
- A novel approach that provides decision-making support to non-technical users during collaboration negotiation.
- A flexible cross-organisational workflow collaboration (COWCO) supporting framework that can be adapted to different implementation architectures.
- A new blackboard-based infrastructure provides distributed workflow enactment support at run time.
- Enhance the matching and reconciliation capability of web services where services are defined and described in the form of flow models.
- Preserve corporate intellectual asset in terms of business process/workflow as much as possible.
- Have the potential to facilitate B2B e-business in a dynamic market place from transaction negotiation to agreement fulfillment.

## 9.4 Limitations and Future Work

This section clarifies the boundary and limitations of the proposed Guru approach. Some future work directions have been identified and outlined as well.

### 9.4.1 Suitability in the Targeted Area

As mentioned on several occasions in this thesis, due to business process collaboration has various motivations, involves different types of partners and takes a number of forms, no single IT solution is able to cover all these aspects. The Guru approach is most suitable for collaboration of casual trade type between business partners with equal partnership relations. However, it is worth noting that the growth of such type of collaboration is accelerating in the Internet era and the requirements for this type of collaboration is in line with the trend of agility and flexibility demanded in the business world. Also, the *estimated effort of design from scratch* (as mentioned in §6.3.1.2 Estimated Effort of Design from Scratch) should be reasonably low for this approach to be more cost-effective. It can be seen, with these conditions attached, the Guru approach is still not a silver bullet to every

issue in cross-organisational workflow collaboration (COWCO) rather it targets the area that is thriving due to a globalised economy but has been largely neglected by the research community.

#### **9.4.2 Fine Tuning Compatibility Definitions and Comparison Techniques**

In order to deal with the exceptional situation identified in §8.4.10 Exceptional Circumstances more effectively, other compatibility definition as well as comparison approaches could be further explored.

#### **9.4.3 Automated Risk Assessment and Decision Making**

Whenever a discrepancy is encountered, decision on the reconciliation option should be based on the assessment between the risk associated with the implied concession and the risk as a result of the conflict. At the moment, the risk assessment is carried out by human users. With the development, enrichment and integration of automated business rule bases, the risk assessment and decision-making could be automated on a larger scale, which can make the whole business collaboration negotiation more intelligent.

#### **9.4.4 Multi-Lateral Process Negotiation**

For the purpose of simplicity, it is assumed that the approach is applied between two business partners. However, in the real world, it is not uncommon for a business transaction to involve more than two partners. Techniques and procedures for multi-lateral negotiation support and agreement fulfilment could possibly be different from the ones for a bilateral situation, e.g. the definition of compatibility, the way to reach an agreement and the infrastructure may need to be extended or altered to accommodate multi-lateral process negotiation.

#### **9.4.5 Bring back XOR Structure**

As mentioned previously, the Guru approach removes XOR structures before entering process comparison stage. It may be worth of exploring whether it is possible to carry out process comparison and reconciliation when XOR structures are attached.

#### **9.4.6 Evaluation for other Workflow Systems**

Due to the time and financial scale of the project, the Guru approach can only be tested on Shark Workflow. More mainstream workflow management systems could be included in future testing to justify the generality of the approach.

#### **9.4.7 Evaluation by Real Cases**

Due to the scale of the project, fictitious examples are used to test, demonstrate and evaluate key components of the prototype. In order to further evaluate the prototype, real cases from industrial partners would be of great value.

#### **9.4.8 Potential Industrial Partner and Knowledge Transfer**

Given the nature of small and medium-sized enterprises (SMEs), transactions among them are likely to benefit from the outcome of the research. Large companies can also benefit by adopting the IT-supported approach to reduce the workload associated with their huge transaction volume, especially when they collaborate in equal partnership, e.g. collaboration between BAE Systems and Rolls-Royce. It will be ideal that the prototype system can be tested by a selection of industrial partners (both SMEs and large companies) in order to further evaluate the possibility of knowledge transfer.

#### **9.4.9 Potential Contribution to Standards**

This thesis discovers a number of properties with regard to workflow collaboration. In order to elaborate them, new terms are defined, which have the potential to form and complement relevant standards in the field of collaborative workflow.

### **9.5 Overall Conclusion**

A comprehensive and integrated solution to B2B e-business collaboration is proposed in this thesis. It is concluded that the design and implementation of several novel enabling techniques and a collaboration enactment infrastructure, which form the key parts of the Framework, has realised the aim and objectives set at the beginning of the thesis based on the result of a series of testing and evaluation.



## References

Aalst, van der, W.M.P., 1999. Process-oriented architectures for electronic commerce and interorganizational workflow. *Information Systems*, Vol. 24, No. 8, 1999. pages 639-671.

Aalst, 2001. The P2P Approach to Interorganizational Workflows. In Weske, M. Ed.: *Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE'01)*, Volume 2068 of Lecture Notes in Computer Science.

Aalst, van der, W.M.P., 2003. Workflow Patterns. *Distributed and Parallel Databases*, 14, 2003, pages 5–51.

Aalst, van der, W.M.P. and Anyanwu, K., 1999. Inheritance of Interorganizational Workflows to Enable Business-to-Business E-commerce. In *Proceedings Of the Second International Conference on Telecommunications and Electronic Commerce (ICTEC'99)*, Nashville, Tennessee, pages 141-157.

Aalst, van der, W.M.P. and Hee, K.v., 2002. *Workflow Management: Models, Methods, and Systems*. MIT Press, Cambridge.

Ader M., 1997, *Seven Workflow Engines Reviewed*, Document World, vol. 2, no 3.

Australian Taxation Office, 2008.

<<http://www.ato.gov.au/corporate/content.asp?doc=/content/00167245.htm>> , (accessed on 8.2.2008)

Baeyens, T., 2004. The state of workflow,

<<http://jbpm.org/2/state.of.workflow.html#tombaeyens>>. (accessed 5.5.2005)

Biegus, L., Branki, C., 2004. InDiA: a Framework for Workflow Interoperability Support by Means of Multi-Agent Systems. *Engineering Applications of Artificial Intelligence*, 17 (7) 2004, pages 825-839.

Boivie, C.A., 2007. Cross-organizational collaboration: from dating to tying the knot, <[http://www.backbonemag.com/Magazine/CIO\\_View\\_11080701.asp](http://www.backbonemag.com/Magazine/CIO_View_11080701.asp)>. (accessed 8.11.2007)

Bolero.net, 2008. Bolero.net, <<http://www.bolero.net/>>. (accessed, 28.10.2008)

Boyes-Schillers, S., 2003. Putting Workflow and BPM standards into context. *Presentation at e-Science Workflow Services*, Edinburgh 3 December 2003.

Byde, A., Piccinelli, G. and Lamersdorf, W., 2002. Automating Negotiation over B2B Processes. In *Proceedings of the 13<sup>th</sup> International Workshop on Database and Expert Systems Applications (DEXA'02)*.

Chen, D. and Vernadat, F., 2002. Enterprise Interoperability: A Standardisation View. In Kosanke, K. et.al Eds.: *Proceedings of the ICEIMT'02 Conference: Enterprise Inter- and Intraorganizational Integration, Building International Consensus*, 2002, ISBN 1-4020-7277-5.

Chen, Q. and Hsu, M., 2001. Inter-Enterprise Collaborative Business Process Management. In *Proceedings Of 17th International Conference on Data Engineering (ICDE'01)*, Heidelberg, Germany, April 2001, IEEE Computer Society, pages 253-260.

Chung, P.W.H. and Chen, X., 2008. Reconciling and Enacting Cross-Organisational Workflow for B2B E-Commerce. In Fischer, L. Ed.: *2008 BPM and Workflow Handbook Digital Edition v2*, 2008, pages 347-360.

Collins Cobuild English Dictionary, HarperCollins Publishers Ltd, London, 1995, page 1105.

Fancey, J., 2005. What is BPEL4WS? Build Better Business Processes with Web Services in BizTalk Server 2004, <<http://msdn.microsoft.com/en-gb/magazine/cc163820.aspx>>. (accessed on 16.10.2008)

Fisher, R., 2008. *The 7 Elements of Negotiation*, <<http://www.pon.harvard.edu/hnp/theory/tools/7elements.shtml>>. (accessed 1.5.2008)

Follett, M. P., 1942. In Metcalf, H. C.& Urwick, L. Eds.: *Dynamic administration: The collected papers of Mary Parker Follett*, New York: Harper & Row.

Gartner, 2006. Gartner Survey of 1,400 CIOs Shows Transformation of IT Organisation is Accelerating, <[http://www.gartner.com/press\\_releases/asset\\_143678\\_11.html](http://www.gartner.com/press_releases/asset_143678_11.html)>. (accessed on 13.7.2006)

Goranson, H.T., 1999. *The Agile Virtual Enterprise - Cases, Metrics, Tools*. Quorum Books, London.

Guillaume, F., 2003. Trying to Unify Entity-based and Activity-based Workflows, <<http://www.zope.org/Wikis/DevSite/Projects/ComponentArchitecture/TryingToUnifyWorkflowConcepts>>. (accessed 3.7.2005)

Hamner, W. Clay and Gary A. Yukl. 1977. The Effectiveness of Different Offer Strategies in Bargaining. In Daniel Druckman, Ed.: *Negotiations: Social-Psychological Perspectives*, Beverly Hills, Calif.: Sage.

Hicks, J.R., 1932. *The Theory of Wages*, London: Macmillan and Co., 1932.

Hiltrop, J., Udall, S., 1995. *The Essence of Negotiation*, Prentice Hall, London, 1995.

International Accounting Standards Board (IASB), 2008. The IASC Foundation publishes the IFRS Taxonomy Guide,

<<http://www.iasb.org/News/XBRL/The+IASC+Foundation+publishes+the+IFRS+Taxonomy+Guide.htm>>. (accessed 25.10.2008)

Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C. and Wooldridge M., 2001. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2), 2001, pages 199-215.

Juan, Y.C. and Ou-Yang, C., 2005. A Process Logic Comparison Approach To Support Business Process Benchmarking. *International Journal of Advanced Manufacturing Technology*, Vol. 26, 2005, pages 191-210.

Keen, M., Chin, H. H., Ganapathi, C., Ghazaleh, D., Krogdahl, P., Neave, W., Sahni, M. and Thorwart, J., 2006. *Patterns: Extended Enterprise SOA and Web Services*, IBM Redbooks.

Krukkert, D., 2003. Matchmaking of ebXML Business Processes – Technical Report IST-28584-OX\_D2.3\_v.2.0, openXchange Project, Oct 2003.

Lax, D. A. and Sebenius, J. K., 1986. *The manager as negotiator: Bargaining for cooperation and competitive gain*. New York: Free Press.

Leymann, F., Roller, D., and Schmidt, M.-T., 2002. Web Services And Business Process Management. *IBM Systems Journal*, vol 41, NO 2, 2002, pages 198-211.

Li, C., Giampapa, J.A., and Sycara, K., 2003. A Review of Research Literature on Bilateral Negotiations – Technical Report CMU-RI-TR-03-41, Robotics Institute, Carnegie Mellon University, November, 2003.

Longman Dictionary of Contemporary English, Essex England: Pearson Education Limited, 2002, page 1100.

Orriëns, B. and Yang, J., 2005. Establishing and Maintaining Compatibility in Service Oriented Business Collaboration. In *Proceedings of the 7th International Conference on Electronic Commerce (ICEC05)*, Xi'an, China, August 2005.

The Oxford Compact English Dictionary, Oxford University Press, Oxford, 2000, page 216.

Poppendieck, M., 2002. Principles of Lean Thinking,  
<<http://www.poppendieck.com/papers/LeanThinking.pdf>>. (accessed on 1.9.2008)

Pratt, V.R., 1991. Modeling concurrency with geometry. In *Proceedings of the 18<sup>th</sup> Ann. ACM Symposium on Principles of Programming Languages*, pages 311-322, January 1991.

Pruitt, D.G. and Carnevale, P.J., 1993. *Negotiation in Social Conflict*, Buckingham: Open University Press.

Rosenschein, J.S. and Zlotkin G., 1994. *Rules of Encounter*, MIT Press.

RosettaNet, 2002. RosettaNet Business Dictionary, RNBD v2.1,  
<<http://www.rosettanet.org/cms/sites/RosettaNet/Standards/RStandards/dictionary/business/index.html>>. (accessed 5.10.2008)

RosettaNet, 2008. RosettaNet Standards,  
<<http://www.rosettanet.org/cms/sites/RosettaNet/Standards/index.html>>.  
(accessed 5.10.2008)

Rubin, J. Z., 1994. Models of Conflict Management. *Journal of Social Issues*, v50 n1, 1994, pages 33-45.

Schulz, K.A., 2002. *Modelling and Architecting of Cross-Organisational Workflows* – PhD Thesis at the University of Queensland, Australia, the School of Information Technology and Electrical Engineering, 2002.

Schulz, K.A., Orlowska, M.E., 2004. Facilitating Cross-Organisational Workflows with a Workflow View Approach. *Data & Knowledge Engineering*, 51 (1), 2004, pages 109-147.

Schurter, T., 2006. BPM - Processes and Market Directions. Bloor Research, 2006.

Shapiro, R., 2001. A Comparison of XPDL, BPML and BPEL4WS,  
<[http://www.ebpml.org/A\\_Comparison\\_of\\_XPDL\\_and\\_BPML\\_BPEL.doc](http://www.ebpml.org/A_Comparison_of_XPDL_and_BPML_BPEL.doc)>. (accessed 6.7.2005)

Smith, H., 2002. A Systems Integrator's Perspective on Business Process Management, Workflow and EAI,  
<<http://www.fairdene.com/processes/CSC-SI-Perspective-BPM-v1.pdf>>. (accessed 5.5.2005)

Swenson, K.D., 2007. The Business Value of Workflow and BPM. In Fischer, L. Ed.: *2007 BPM and Workflow Handbook*.

SWIFT, 2008. SWIFT Solution – Business Transactions,  
<[http://www.swift.com/index.cfm?item\\_id=60000](http://www.swift.com/index.cfm?item_id=60000)>. (accessed 5.5.2008)

Tagg, R., 2001. Workflow in Different Styles of Virtual Enterprise. In *Proceedings of the workshop on Information technology for virtual enterprises, SESSION: Automation of business activities*, Queensland, Australia, pages 21 - 28.

Tracy, L. and Peterson, R.B., 1986. A behavioral theory of labor negotiations: How well has it aged?, *Negotiation Journal*, 2, 1986, pages 93-108.

Walton, R. E. & McKersie, R. B., 1965. *A behavioral theory of labor negotiations: An analysis a social interaction system*. Ithaca, NY: ILR Press.

Wombacher, A., 2005. *Decentralized establishment of consistent, multi-lateral collaborations* – PhD Thesis at Technical University Darmstadt, Faculty of Informatics, 2005.

Workflow Management Coalition, 1995. The Workflow Reference Model – Technical Report WFMC-TC-1003. Workflow Management Coalition.

Workflow Management Coalition: Workflow Management Application Programming Interface (Interface 2&3) – Technical Report WFMC-TC-1009. Workflow Management Coalition.

Workflow Management Coalition, 1999a. Terminology & Glossary – Technical Report WFMC-TC-1011. Workflow Management Coalition.

Workflow Management Coalition, 1999b. Workflow Standard – Interoperability Abstract Specification – Technical Report WFMC-TC-1012. Workflow Management Coalition.

Yang, J. and Papazoglou, M., 2000. Interoperation Support for Electronic Business, *Communications of the ACM*, June Vol. 43, No. 6, 2000, pages39-47.

Yeoh, M.L., Chung, P.W.H., Anumba, C.J., El-Hamalawi, A., and Motawa, I.A., 2004. Process Change Identification Using Workflow Specification Matching. In *Proceedings of the Tenth Americas Conference on Information Systems*, New York, August 2004.

Zartman, I.W., 1977. Negotiation as a Joint Decision-Making Process. Zartman, I.W., Ed.: *The Negotiation Process: Theories and Applications*, London: Sage Publications, 1977.

Zelm, M. and Kosanke, K., 2007. Standardisation in Interoperability. *Presentation at IMS Workshop*, Zürich, Nov 2007,

<[http://www.cimosa.de/Standards/Standardisation\\_in\\_Interoperability.pdf](http://www.cimosa.de/Standards/Standardisation_in_Interoperability.pdf) >. (accessed on 5.10.2008)

# Appendices



## **Appendix 1: List of Abbreviations**

B2B – Business to Business  
CAIA – Candidate Adjusting Interface Activity  
CAIAS – Candidate Adjusting Interface Activity Set  
CCPIA – Current Common Preceding Interface Activity  
CCPIAS – Current Common Preceding Interface Activity Set  
CFIM - Control Flow Implication Message  
CHTP – Candidate Hypothetical Target Position  
CIIA – Candidate Inserting Interface Activity  
CIIAS – Candidate Inserting Interface Activity Set  
CMA – Candidate Moving Activity  
CMAS – Candidate Moving Activities Set  
CMIA – Candidate Moving Interface Activity  
CMIAS – Candidate Moving Interface Activity Set  
CMP – Collaborative Process Manager  
COWCO – Cross-Organisational Workflow Collaboration  
CPDL – Collaborative Process Definition Language  
CRIA – Candidate Removing Interface Activity  
CRIAS – Candidate Removing Interface Activity Set  
CTA – Conceded Target Activity  
DA – Destination Area  
DBA – Downstream Boundary Activity  
DDA – Data Demand Activity  
DFIM - Data Flow Implication Message  
DIA – Discrepant Interface Activity  
DIAS – Discrepant Interface Activity Set  
DSA – Data Supply Activity  
E-Business – Electronic Business  
E-Commerce – Electronic Commerce  
EEDS – Estimated Effort of Design from Scratch

HTP – Hypothetical Target Position

IA – Interface Activity

IASB – International Accounting Standards Board

IFRS – International Financial Reporting Standards

IP – Interface Process

M&A – Merger and Acquisition

MSC – Message Sequence Charts

PDL – Process Definition Language

PIP – Partner Interface Processes

QoS – Quality of Service

RNBD – RosettaNet Business Dictionary

SME – Small and Medium-Sized Enterprise

SOA – Service Oriented Architecture

SOAP – Simple Object Access Protocol

STS – State Transition System

SWIFT – the Society for Worldwide Inter-bank Financial Telecommunication

TA – Target Activity

UBA – Upstream Boundary Activity

UDDI – Universal Description Discovery and Integration

UDDIA – Unique Data Demand Interface Activity

UDSIA – Unique Data Supply Interface Activity

UML – Unified Modeling Language

VE – Virtual Enterprise

WfMC – Workflow Management Coalition

WfMS – Workflow Management System

XBRL – eXtensible Business Reporting Language

XOR – eXclusive OR

XPDL – XML Process Definition Language

## Appendix 2: List of Publications

Chen, X. and Chung, P., 2006. Cross-Organisational Workflow Enactment Via Progressive Linking by Run-Time Agents. In Ali, M. and Dapoigny, R. Eds.: *Proceedings of 19th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems*, IEA/AIE 2006, Annecy France, June 2006, LNAI 4031, pages 54-59.

Chen, X. and Chung, P.W.H., 2007. A simulation-based difference detection technique for bottom-up process reconciliation, in *Proceedings of 9th International Conference on Enterprise Information Systems (ICEIS 2007)*, Funchal, Madeira – Portugal, June 2007.

Chen, X. and Chung, P.W.H., 2007. A Framework for Cross-Organizational Workflow Collaboration. In Zhang, Z., and Zhang, R. Eds.: *Proceedings of 13th Cross-Strait Academic Conference On Information Management Development & Relevant Strategy*, Beijing, China, 2007.

Chen, X. and Chung, P.W.H., 2008. Facilitating B2B E-Business by IT-Supported Business Process Negotiation Services. In *Proceedings of 2008 IEEE International Conference on Service Operations and Logistics, and Informatics*, Beijing, October 2008, pages 2800-2805.

Chung, P.W.H. and Chen, X., 2008. Reconciling and Enacting Cross-Organisational Workflow for B2B E-Commerce. In Fischer, L. Ed.: *2008 BPM and Workflow Handbook Digital Edition v2*, 2008, pages 347-360.