Ghent University
Faculty of Economics and Business Administration
Department of Management Information Science and Operations Mgt.

The Resource-Event-Agent Ontology

as a Foundation for Business Modeling

Wim Laurier

Advisor: Prof. dr. Geert Poels

Submitted to the Faculty of Economics and Business Administration of Ghent University in fulfillment of the requirements for the degree of Doctor in Applied Economics
August 2010

Ghent University
Faculty of Economics and Business Administration
Department of Management Information Science and Operations Mgt.

Advisor: Prof. dr. Geert Poels

Ghent University
Faculty of Economics and Business Administration
Department of Management Information Science and Operations Management
Tweekerken 2, B-9000 Gent, België Tel.: +32-9-264.35.17
Fax.: +32-9-264.42.86

Submitted to the Faculty of Economics and Business Administration of
Ghent University in fulfillment of the requirements for the degree of Doctor
in Applied Economics
August 2010

Doctoral Jury

Prof. dr. Marc De Clercq (Dean)
Prof. dr. Patrick Van Kenhove (Academic Secretary)
Prof. dr. Geert Poels (advisor)
Em. prof. dr. Roland Paemeleire
Prof. dr. ir. Birger Raa
Prof. dr. Jan Verelst (Universiteit Antwerpen)
Prof. dr. ir. Hajo Reijers (Technische Universiteit Eindhoven)
Prof. dr. William E. McCarthy (Michigan State University)

Dankwoord

Een doctoraat schrijf je niet alleen. Een doctoraat schrijf je met de mensen om je heen, met je vrienden en familie, met je promotor en collega's. Een doctoraat schrijf je met alle mensen die je maken wie je bent. Met je ouders, broers en zussen, met je vrienden — de goede en de slechte — met je gewezen proffen, leerkrachten, juffen, meesters — sympathieke en minder sympathieke — en alle anderen die je levenspad kruisten. Daarom wil ik iedereen bedanken die mij gevormd heeft tot de persoon die ik nu ben. Hoewel de lijst van personen die mijn pad kruisten lang is, zijn er toch enkele mensen die er echt uitspringen, mensen die het verdienen om met naam genoemd en persoonlijk bedankt te worden.

Eerst en vooral wil ik mijn ouders — Paul en Vivianne — bedanken, omdat ze me 27 jaar lang het kind hebben laten uithangen. Ik wil hen bedanken voor de vele ritjes van en naar Gent, de fluo-patatjes, de propere kleren en vooral, de warme thuis. Ten tweede wil ik ook mijn zus — Lien — bedanken om twee jaar lang een geweldig kotgenootje te zijn. Het is fijn om 's avonds te kunnen thuiskomen en een vertrouwd gezicht te zien. Bedankt om twee jaar lang dat gezicht te zijn. Ten derde wil ik ook nog mijn vrienden uit Oudenaarde bedanken, omdat ze na al die jaren — waarin ik niet altijd heb uitgeblonken in aanwezigheid en beschikbaarheid — nog steeds mijn vrienden zijn. Pieter, Stefaan, Dorothée, Delphine, Steven, Thomas, Koen, Griet, Eva, Leen en José bedankt voor de babbels, de terrasjes, de etentjes, de filmpjes, de pastisjes en alle andere kleine dingen die het leven boeiend maken. Speciale dank ook aan Dorothée om de laatste tijd het organiseren van de filmclub-avondjes over te nemen, je doet het geweldig. Ten vierde wil ik ook nog mijn vrienden uit Gent bedanken voor de leuke studententijd en de tijd erna. Alexander, Karel, Natasja, Leentje, Willem, Ine, Bart, Joke bedankt voor de fijne studententijd en de bijhorende uren in het Salon, de Bluesette en andere eet- en drankgelegenheden. Willem ook bedankt voor het organiseren van de waterpolo en andere sportactiviteiten, het werd zeer geapprecieerd.

Dan wil ik ook nog mijn collega's en begeleiders aan de universiteit bedanken voor de goede begeleiding en de aangename werkomgeving. Geert, jou wil ik bedanken voor de kans die je me gegeven hebt, de kans om te doctoreren. Doctoreren was nooit mijn ambitie tot jij me overtuigde. Het was een goede beslissing en vandaag ben ik blij dat ik ze genomen heb. Ik wil je ook bedanken voor je geduld, de vrijheid en de goede begeleiding die je me gegeven hebt in de loop van dit doctoraat. Frederik, jou wil ik bedanken om m'n 'promotor-bis' te zijn. Er was geen vraag — technisch of theoretisch — waar je geen blijf mee wist. Je kennis en vaardigheid worden nog steeds geapprecieerd. Ken, jou wil ik bedanken voor

de aangename werksfeer en het delen van de liefde voor warm eten 's middags. Ook dank aan Thomas, Maxime, Christophe en Jeroen om ervoor te zorgen dat de middagen geen lange reeks tête-à-têtes met Ken zijn geworden. Ook de kapoentjes om vijf uur zal ik missen. Ook wil ik nog Ann en Martine bedanken voor de goede zorgen, het boeken van menig reisje en het overnemen van vele administratieve taakjes. Ik zal de kwinkslagen missen. Ook wil ik nog de leden van m'n begeleidings-, lees- en examencommissie (Em. Prof. Dr. Roland Pameleire, Prof. Dr. Jan Verelst, Prof. Dr. William E. McCarthy, Prof. Dr. Ir. Hajo Reijers, Prof. Dr. Ir. Birger Raa) bedanken voor de vele tips en richtlijnen die dit doctoraat tot een nog hechter geheel hebben gemaakt. Jullie bijdrage wordt gewaardeerd.

En tot slot — last but not least — wil ik nog een heel bijzonder iemand bedanken. Adel, bedankt om er op de cruciale momenten te zijn. Ik hoop dat er nog vele cruciale momenten mogen komen en dat we die samen kunnen vieren.

Gent, 25 Augustus 2010
Wim Laurier

Table of Contents

Doctoral Jury i

Dankwoord — Dutch — i

List of Figures xi

List of Tables xiii

List of Acronyms xv

Samenvatting — Dutch — xvii

1 Introduction 1
- 1.1 On the resource-event-agent (REA) ontology 5
- 1.2 On ontologies in conceptual modeling 9
- 1.3 On the data model section 11
- 1.4 On data model formalization 13
- 1.5 On the simulation model section 17
- 1.6 On the simulation model formalization 19
- 1.7 On the papers in this dissertation 20

I Data Models 25

2 Enterprise Ontology-Based Structuring of Conceptual Data Modeling Patterns 27
- 2.1 Introduction 29
- 2.2 Building Blocks for a Structuring Framework 30
  - 2.2.1 Enterprise Pattern 31
  - 2.2.2 Abstraction Pattern 34
  - 2.2.3 Composing the Framework 36
- 2.3 Positioning Batra's patterns in the framework 36
  - 2.3.1 TransactionPatterns 36
  - 2.3.2 Generic Patterns 45
- 2.4 Example application scenario: extending the Datawarehouse Pattern 48
- 2.5 Conclusion and Future Research 53

2 Summary **57**

3 A Resource-Event-Agent Reference Information Model for Representing Value Chains and Systems **59**
3.1 Introduction 61
3.2 Introduction to the REA ontology 64
3.3 The REA reference information model 66
3.4 Concept models for business operations 69
3.5 Concept models for business planning 77
3.6 Comparative analysis of related conceptualizations 80
3.6.1 McCarthy's reference models 80
3.6.2 Hruby's reference model 87
3.6.3 e3-Value Ontology 90
3.6.4 SCOR 93
3.7 Conclusion and Future Research 95

3 Summary **99**

4 Tracking and Tracing Future, Present, and Past Product and Money Flows **101**
4.1 Introduction 103
4.2 Reference Model for Tracking and Tracing 105
4.3 Prototype Application: Baking Bread and Pizza, From Farm to Customer 113
4.4 Related work 122
4.4.1 Tracking and Tracing Models 122
4.4.2 REA Models 127
4.5 Conclusions, Limitations and Directions for Future Research . . . 134

4 Summary **137**

II Simulation Models **139**

5 Simulating Liquidity in Value and Supply Chains **141**
5.1 Introduction 143
5.2 Atomic REA Model Construction Patterns 144
5.3 Exchange Configurations 147
5.4 Value Chain Configurations 148
5.5 Supply Chain Pattern 155
5.6 Discussion 158
5.7 Conclusion and Future Research 163

5 Summary **165**

6 A Simulation Model Articulation of the REA Ontology 167
6.1 Introduction 169
6.2 Overview of the REA Ontology and the ExSpect Simulation Language 170
6.2.1 The ExSpect Simulation Language 170
6.2.2 The REA Ontology 171
6.3 Economic Rationale Models 172
6.4 Exhibit: alternative Value Chain Designs 175
6.5 Conclusion and Future Research 179

6 Summary 181

7 Conclusion 183

References 189

A Appendix chapter 2 209

B Appendix chapter 3 213

List of Figures

1.1 Trading partner view REA model for Elmo's cookie store 7
1.2 Trading partner view model for Elmo's acquisition and revenue cycle 8
1.3 Independent view REA model for Elmo's cookie store 10
1.4 UML class diagram for car ownership with association 14
1.5 UML class diagram for car ownership with association class . . . 14
1.6 UML object diagram for car ownership, the case of me and my car 15
1.7 Annotated UML class diagram for relational database design . . . 15
1.8 Mutually exclusive sub-classes, with abstract super-class 16
1.9 Non-mutually exclusive sub-classes 17
1.10 An example Petri-net 20
1.11 An example colored Petri-net 21

2.1 Employment patterns proposed by Fowler and Eriksson & Penker 32
2.2 Enterprise pattern 33
2.3 Abstraction pattern 35
2.4 Structured enterprise pattern for individuals 37
2.5 Structured abstraction pattern for resources 37
2.6 Generic transaction pattern 38
2.7 Discrete product pattern 40
2.8 Time-based transaction pattern 41
2.9 Registration as a time-based transaction 42
2.10 Subsequent transaction pattern 43
2.11 Data warehouse pattern 44
2.12 Bill of materials 45
2.13 Container-content hierarchy 46
2.14 Type pattern for power-typing 47
2.15 Generalization pattern 48
2.16 Entity-event pattern 49
2.17 Recursive one-many relationship 50
2.18 Extended data warehouse pattern 52

3.1 Informative component of the reference information model 68
3.2 Economic rationale component of the reference information model 70
3.3 Archetypal object model for the transformation event pattern . . . 72
3.4 Archetypal object model for the transfer event pattern 73

3.5 Archetypal object model for the buyer side of the exchange pattern (i.e. acquisition cycle) 74
3.6 Archetypal object model for the seller side of the exchange pattern (i.e. revenue cycle) 75
3.7 Archetypal object model for the exchange pattern 76
3.8 Archetypal object model for the contracted exchange pattern . . . 78
3.9 Claim and liability object model template 79
3.10 Settlement of transfer commitment object model template 81
3.11 Completion of scheduled transformation event object model template 82
3.12 McCarthy's trading-partner view reference model 83
3.13 McCarthy's independent view reference model 85
3.14 McCarthy's trading-partner view exchange object model template 86
3.15 McCarthy's independent view exchange object model template . . 86
3.16 Hruby's REA reference model 88
3.17 Hruby's REA exchange object model template 89
3.18 Hruby's REA conversion object model template 90
3.19 The e3-value ontology 92
3.20 The top-level SCOR model 94

4.1 Tracking and tracing conceptual data model structure 107
4.2 Exchange model with two mirrored transfer events 109
4.3 Conversion model with one transformation event 110
4.4 Conversion model with one transportation event 111
4.5 Bread baking commitment 112
4.6 Example food supply chain 114
4.7 Application data model for traceability 115
4.8 Event summaries 116
4.9 Transaction summaries 118
4.10 Resource trace 119
4.11 Event track 120
4.12 Commitment trace 123
4.13 Commitment summaries 124
4.14 Reference data model for tracking and tracing 126
4.15 Traceability data model 128
4.16 McCarthy's trading-partner view REA model (confer fig. 3.12) . . 130
4.17 McCarthy's independent view REA model (confer fig. 3.13) . . . 132
4.18 Hruby's REA reference model (confer fig. 3.16) 133

5.1 Event template 146
5.2 Claim duality template 147
5.3 Liability duality template 147
5.4 Buyer liable 149
5.5 Seller liable 150
5.6 Economic unit liable to suppliers and customers 152
5.7 Economic unit liable to suppliers, claim on customers 153

5.8 Economic unit claim on suppliers, liable to customers 154
5.9 Economic unit claim on suppliers and customers 156
5.10 Duality lifecycle template 157
5.11 Supply chain link, claim on suppliers and liable to customers . . . 159
5.12 Supply chain link, claim on customers and liable to suppliers . . . 160
5.13 Exhibit: supermarket and retail store 162

6.1 Collaboration space model for an enterprise with one supplier and one customer 173
6.2 Value chain model for the enterprise in the collaboration space model for fig. 6.1 176
6.3 Business process model for the conversion cycle of the enterprise in the value chain model of fig. 6.2 177

A.1 Fully developed combination of the enterprise (fig. 2.2) and abstraction (fig. 2.3) pattern 211

B.1 Fully developed combination of the informative (fig. 3.1) and economic rationale component (fig. 3.2) of the reference information model 215

List of Tables

1.1 Model quality variables as listed by Moody and Shanks [MS03] . 2

6.1 Examples of basic REA Concepts 172
6.2 Simulation results for alternative value chain designs 178

List of Acronyms

B

BMM	Business Motivation Model
BMO	Business Modeling Ontology

D

DEMO	Design & Engineering Methodology for Organizations

E

ebXML	Electronic Business using eXtensible Markup Language
EDI	Electronic Data Interchange
EO	Enterprise Ontology
ERP	Enterprise Resource Planning

I

ISO	International Organization for Standards

K

KBS	Knowledge Base System

N

NEF	Need for External Funding

O

OeBTO	Open-edi Business Transaction Ontology
OMG	Object Management Group

R

REA	Resource Event Agent

S

SCOR Model	Supply Chain Operations Reference Model

U

UFO	Unified Foundational Ontology
UML	Unified Modeling Language
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business

V

VMI	Vendor Managed Inventory

W

W3C	World Wide Web Consortium
WIP	Work In Progress

Samenvatting

In de praktijk zijn bedrijfsinformatiesystemen — zoals een boekhoudsysteem, een voorraadbeheersysteem, een kostprijscalculatiesysteem, een ordermanagementsysteem — onderdeel van een veel grotere informatieverwerkende omgeving. Bijgevolg is het volgens Moody en Shanks onnuttig om bedrijfsinformatiesystemen los van elkaar te bekijken. In [MS03] stellen en tonen ze aan dat het beschouwen van informatiesystemen binnen deze bredere context cruciaal is voor het verkrijgen van kwaliteitsvolle informatiesystemen. Daarom voegen zijn 'integratie'toe aan een lijst van reeds bestaande kwaliteitsfactoren voor het ontwerpen van informatiesystemen. Met integratie bedoelen ze de mate waarin de onderliggende infrastructuurmodellen van verschillende bedrijfsinformatiesystemen met elkaar overeenstemmen, wat de samenwerking tussen en het uitwisselen en hergebruik van data door deze systemen moet vergemakkelijken.

Bij het ontwerpen van een informatiesysteem starten we met een conceptueel ontwerp. Zo'n conceptueel ontwerp stelt het beoogde informatiesysteem voor zonder details die specifiek zijn voor de gekozen technologie — zoals een webpagina, een database, software — te tonen. Een conceptueel model focust dus op het voorstellen van een conceptuele oplossing voor een bestaand probleem nog voor de technologische oplossing wordt ontworpen. Hoewel zo'n conceptueel model slechts een deel van de uiteindelijke oplossing voorstelt, stelt het de essentie van de oplossing voor — die dan eventueel in verschillende technologieën tegelijk kan worden uitgewerkt. Omdat deze oplossing essentieel is, moet een conceptueel model aan een reeks kwaliteitsparameters voldoen. Zo moet een conceptueel model een zo eenvoudig mogelijke maar toch volledige oplossing bieden voor een probleem. Ook moet de oplossing, voorgesteld in het conceptueel model, verstaanbaar zijn voor anderen dan de ontwerpers — bijvoorbeeld de programmeurs die de praktijkoplossing uitwerken of de klant die het systeem besteld heeft en zal gebruiken. Vaak probeert men ook te bereiken dat een systeem flexibel is, waarmee men bedoelt dat de oplossing rekening houdt met mogelijke veranderingen in de omgeving — zonder in te gaan tegen wat als gangbaar wordt beschouwd binnen de bredere informatieverwerkende omgeving of zonder moeilijk te verwezenlijken te zijn binnen het vooropgestelde budget en de beoogde planning.

Zonder deze kwaliteitsfactoren uit het oog te verliezen, focussen we in deze doctoraatsthesis op de integratie van informatiesysteemontwerpen, die door Moody en Shanks werd voorgesteld en waarvan zij substantiële voordelen aantoonden. Aangezien er vele verschillende soorten conceptuele modellen bestaan, kiezen wij er in deze thesis twee specifieke soorten uit, namelijk conceptuele datamodellen en

simulatiemodellen. We kiezen datamodellen omdat zij een van de meest essentiële soorten conceptuele modellen voorstellen. Conceptuele datamodellen stellen immers de basis van elke informatiesysteem voor. Bijgevolg is de keuze voor of het ontwerp van een datamodel een van de meest cruciale stappen in het ontwerp van een informatiesysteem die invloed heeft op o.a. de uiteindelijke kost van het project, de flexibiliteit van het resulterende systeem, de klantentevredenheid en de integratie met andere (bestaande) informatiesystemen. Waar conceptuele datamodellen ons toelaten een ontwerp te maken voor de data die in informatiesystemen worden bijgehouden over de huidige toestand en historiek van een voor ons relevant deel van de realiteit, laten simulatiemodellen ons toe een ontwerp — dat met potentiële toekomstige situaties kan omgaan — te formuleren van een voor ons relevant deel van een omgeving. Samen laten conceptuele data- en simulatiemodellen ons dus toe het verleden, heden en de toekomst van een voor ons relevant deel van de realiteit voor te stellen.

In deze thesis is dat relevant deel van de realiteit de bedrijfseconomische context. Aangezien deze bedrijfseconomische context dynamisch — en dus inherent onstabiel — is met steeds wijzigende en slechts gedeeltelijk gespecificeerde systeemvereisten — wat flexibiliteit vereist — is een speciale aanpak noodzakelijk. Deze aanpak heet 'Design Science' [HMJR04] en houdt ook rekening met hoe succesvolle informatiesysteemontwerpen afhankelijk zijn van de cognitieve en sociale vaardigheden van de mensen die deze systemen gebruiken, ontwikkelen en ontwerpen. Om deze flexibiliteit en sociale en cognitieve vaardigheden te ondersteunen, gebruiken we een ontologie als basis voor het ontwerpen van de conceptuele data- en simulatiemodellen in deze doctoraatsthesis. Zo'n ontologie is een gedeelde beschrijving van het probleemdomein (bijvoorbeeld de bedrijfseconomische realiteit). Dat deze beschrijving gedeeld wordt, ondersteunt de sociale en cognitieve vaardigheden van de groep mensen die een informatiesysteem ontwerpt, omdat de beschrijving kan gebruikt worden als een referentiekader voor deze groep mensen. Daarenboven kan dit delen ervoor zorgen dat de beschrijving hergebruikt en verbeterd wordt bij opeenvolgende projecten door verschillende groepen. Dit hergebruik van de beschrijving van het probleemdomein ondersteunt dan impliciet de integratie tussen verschillende projecten, aangezien ze gebaseerd zijn op (verschillende delen van) dezelfde probleembeschrijving.

De ontologie die wij in deze thesis hoofdzakelijk hanteren is de Resource-Event-Agent (REA) ontologie. Deze ontologie werd begin jaren '80 ontwikkeld voor boekhoudinformatiesystemen, met het oog op een gedeelde dataomgeving waarin boekhouders en niet-boekhouders informatie delen over dezelfde economische gebeurtenissen — zoals aankopen, verkopen, productie. De 'Resources' beschrijven de producten (i.e. goederen en diensten) die verhandeld en geproduceerd worden. De 'Events' beschrijven de gebeurtenissen die de voorraden van deze producten veranderen. Bijvoorbeeld, een verkoop van producten vermindert de voorraad van de verkoper en vermeerdert de voorraad van de aankoper. De 'Agents' beschrijven de economische actoren die de goederen en diensten produceren, verkopen en aankopen. Ondertussen werd de REA-ontologie al gebruikt als basis voor boekhoudinformatiesystemen en een ISO open-edi standaard [ISO07]

voor het uitwisselen van elektronische bedrijfsdocumenten, als methode voor het onderwijzen van boekhoudinformatiesystemen en nog veel meer. [GLP08]

In het eerste hoofdstuk van datamodelsectie gebruiken we de REA- en UFO-ontologie om conceptuele datamodellen te structureren en af te bakenen zodat ze makkelijker te interpreteren zijn, voornamelijk door onervaren systeemontwerpers. De UFO (i.e. Unified Foundational Ontology) is een ontologie die speciaal werd ontworpen om conceptuele modellen te duiden. [BGHS+05] Van de gestructureerde conceptuele datamodellen wordt verwacht dat ze onervaren ontwerpers helpen conceptuele datamodellen te maken die compleet zijn en geen overbodige onderdelen bevatten. Onvolledige modellen en overbodige onderdelen zijn immers de meest voorkomende fouten die onervaren ontwerpers maken. De gestructureerde datamodellen die aan de onervaren ontwerpers worden aangeboden zijn patronen waarvan al bewezen is dat ze waardevolle oplossingen zijn voor bepaalde problemen binnen een specifieke context. Door deze patronen te structureren volgens de REA- en UFO-ontologie wordt de ontwerper verwacht sneller de overbodige en ontbrekende delen van een patroon te kunnen identificeren, naargelang van het probleem dat hij wenst op te lossen. Door de aangeboden structuur kan de ontwerper ook op zoek naar ontbrekende delen van zijn oplossing in andere patronen, die hij dan kan integreren in zijn bestaande onvolledige oplossing. Dit integreren van patronen heeft twee voordelen. Ten eerste wordt de integratie tussen systemen die (delen) van dezelfde patronen bevatten vergemakkelijkt, ten tweede wordt de kwaliteit van de modellen die werden gemaakt door onervaren informatiesysteemontwerpers verhoogd.

In het tweede hoofdstuk van de datamodelsectie gebruiken we de REA-ontologie als basis voor het ontwikkelen van een conceptueel referentiedatamodel dat geschikt is om zowel productie- als transactiedata van verschillende handelspartners voor te stellen. Dat het datamodel data van verschillende handelspartners tegelijkertijd kan voorstellen, heeft als gevolg dat de integratie tussen de bedrijfssystemen van deze handelspartners sterk vergemakkelijkt wordt. Zowel het integreren van de verschillende informatiesystemen van elke individuele handelspartner (bijvoorbeeld verkoopsysteem en voorraadbeheer) als het integreren van informatiesystemen van verschillende handelspartners (bijvoorbeeld een aankoopsysteem met een verkoopsysteem) worden vergemakkelijkt. Het voorstellen van productie- en transactiedata van verschillende handelspartners wordt verwezenlijkt door zowel het perspectief van de individuele handelspartners als het perspectief van een onafhankelijke derde partij expliciet in het datamodel te integreren. Door deze keuze is het datamodel geschikt om volledige waardesystemen voor te stellen. Zo'n waardesysteem bestaat uit de transacties tussen handelspartners in een waardenetwerk of toeleveringsketen (supply chain) en de bedrijfsprocessen die elk van deze handelspartners uitvoert.

In het derde en laatste hoofdstuk van de datamodelsectie gebruiken we het conceptuele datamodel dat werd voorgesteld in het tweede hoofdstuk om een toepassing te ontwikkelen die het mogelijk maakt historische, huidige en toekomstige product- en geldstromen te volgen in een waardesysteem. Het gekozen voorbeeld toont hoe akkerbouwgewassen worden gebruikt voor humane consumptie en vee-

voeder. De dierlijke producten worden dan weer gebruikt voor humane consumptie en samen met de landbouwgewassen verwerkt in consumentenproducten. De excrementen vloeien terug naar de akkerbouw. Het beschreven waardesysteem toont hoe goederen- en geldstromen kunnen worden gevolgd bij transacties tussen handelspartners (bijvoorbeeld akkerbouwer en voedermolen) en hoe goederenstromen kunnen worden gevolgd doorheen de productieprocessen van deze handelspartners (bijvoorbeeld het verwerken van akkerbouwgewassen van verschillende herkomst in een lading veevoer). Bovendien toont de ontwikkelde toepassing dat diezelfde data kunnen worden gebruikt om de transacties tussen handelspartners weer te geven (bijvoorbeeld een akkerbouwer die graan verkoopt aan een voedermolen en daarvoor een vergoeding ontvangt). Diezelfde informatie kan dan ook gebruikt worden om geldstromen in kaart te brengen zoals gebeurt met de goederenstromen. Daarenboven wordt gedemonstreerd hoe niet alleen huidige en gewezen, maar ook hoe toekomstige transacties en goederen- en geldstromen in kaart kunnen worden gebracht (o.a. aan de hand van contracten en productieschema's). In de levensmiddelenindustrie hebben dergelijke systemen voor het traceren van huidige en gewezen goederenstromen al hun nut bewezen wanneer gecontamineerde voedingstoffen werden aangetroffen of getransporteerd vee ziek bleek. Aan de mogelijkheden van dergelijke traceringsystemen voegen wij toe dat niet enkel de bron van de besmetting, maar ook het doel van de goederen en de geldstromen, kan worden geïdentificeerd. Hierdoor kunnen we de economische gevolgen van zo'n besmetting inschatten aan de hand van contracten en productieschema's. Met de beschikbare informatie kunnen dan eventueel noodscenario's worden uitgewerkt om de economische gevolgen van zo'n besmetting te beperken. Ook in andere sectoren kan zo'n traceringssysteem nuttig zijn. Zo kan het in kaart brengen van een volledig waardenetwerk voorkomen dat namaakgoederen in de reguliere handel terechtkomen, of dat geldstromen uit de reguliere economie worden gebruikt om illegale activiteiten wit te wassen of te financieren.

Waar het tweede en derde hoofdstuk van de datamodelsectie REA-gebaseerde datamodellen voor de integratie van de informatiesystemen van handelspartners voorstellen, bevat de simulatiemodelsectie REA-gebaseerde simulatiemodelelementen die het toelaten bedrijfsprocesmodellen over bedrijfsgrenzen heen met elkaar te integreren zodat ook simulatiemodellen voor volledige waardesystemen kunnen worden ontwikkeld. De hoofdstukken in de datamodelsectie bevatten dus geen bedrijfsprocesmodellen, maar superstructuren die het mogelijk maken bedrijfsprocesmodellen te hergebruiken en te integreren over de grenzen van de onderneming heen. Daarenboven werden deze superstructuren zo ontworpen dat ze ook als zelfstandige elementen van simulatiemodellen voor transacties tussen handelspartners kunnen worden gebruikt. Dit betekent dan wel dat men abstractie maakt van de bedrijfsprocessen die deze transacties ondersteunen.

Het eerste hoofdstuk van de simulatiemodelsectie analyseert de mogelijke configuraties (bijvoorbeeld eerst betalen dan halen, eerst halen dan betalen) voor transacties tussen handelspartners en de manier waarop deze configuraties de interne structuur van een bedrijf beïnvloeden. In deze analyse wordt er vooral gekeken naar het vermogen van een bedrijf om zijn activiteiten te financieren met het kre-

diet dat handelspartners verlenen (bijvoorbeeld de betalingstermijn op facturen, voorschotten). De structuren in dit hoofdstuk worden voorgesteld als Petri-net gebaseerde workflowmodellen. Deze workflowmodellen laten toe te evalueren of een gegeven sequentie van transacties externe financiering (bijvoorbeeld een banklening) vereist of niet. Door de verschillende configuraties voor een sequentie van transacties te evalueren kan men de meest optimale (bijvoorbeeld diegene die zo min mogelijk externe financiering vereist) selecteren.

Het tweede hoofdstuk van de simulatiemodelsectie bouwt voort op de modellen in het voorgaande hoofdstuk om ook statistische analyses van deze verschillende configuraties mogelijk te maken. In deze analyses hoeft men dus niet meer te vertrekken van een gegeven sequentie van transacties, maar kan men ook onzekerheid en variatie in rekening brengen. Door de REA-elementen toe te voegen aan simulatiemodelelementen, worden de capaciteiten van de huidige generatie statistische simulatiemodellen uitgebreid van het analyseren van logistieke processen naar het analyseren van volledige bedrijfsmodellen met inbegrip van de financiële parameters en resultaten. In tegenstelling tot de workflowmodellen in het eerste hoofdstuk van de simulatiemodelsectie, hebben de statistische simulatiemodellen een gelaagde opbouw. De bovenste laag modelleert de transacties tussen bedrijven, de middelste laag modelleert de interne financiële structuur van bedrijven en de onderste laag bevat bedrijfsprocesmodellen voor individuele bedrijfsprocessen. De gelaagde opbouw laat ons toe zowel de individuele lagen te simuleren als bedrijfsprocesmodellen in interne structuurmodellen te integreren en deze dan weer in transactiemodellen te integreren.

In deze thesis ontwikkelen we dus een benadering voor het ontwikkelen en integreren van modellen binnen een bedrijfseconomische context. Deze benadering houdt in dat we conceptuele modellen ontwikkelen vanuit en toetsen aan een bedrijfseconomische ontologie. De bedrijfseconomische ontologie die we in deze thesis gebruiken is de REA-ontologie. Deze thesis toont aan dat de REA-ontologie geschikt is als basis voor het ontwikkelen en integreren van zowel conceptuele datamodellen als simulatiemodellen binnen een bedrijfseconomische context. Door het gebruik van deze benadering zouden we de bedrijfswereld niet enkel moeten kunnen uitrusten met robuustere datamodellen voor de bedrijfsinformatiesystemen die ze dagdagelijks gebruiken, maar ook met krachtigere beslissingsondersteunende hulpmiddelen die hen voorzien van de informatie die ze nodig hebben voor het evalueren en voorspellen van de prestaties van hun bedrijf. Die informatie kan dan zowel van financiële (bijvoorbeeld winstmarges) als operationele (bijvoorbeeld wachttijden) aard zijn. Het evalueren kan gebeuren op basis van de data die zijn opgeslagen in de bedrijfsinformatiesystemen en het voorspellen kan gebeuren door het genereren van hypothetische data in simulatiemodellen voor toekomstige bedrijfsprocessen en omgevingsomstandigheden.

1 Introduction

Conceptual modeling in information systems (i.e., the creation of an enterprise model for the purpose of designing information systems [WMPW95]) is a challenging task, especially because - in practice - enterprise information systems form a small part of a much larger information processing environment. Consequently, conceptual models for the purpose of designing information systems cannot be considered standalone artifacts. Moody and Shanks [MS03] argue that considering individual systems in the context of an overall architecture is critical for developing quality information systems. Therefore, they introduce integration as a quality factor for conceptual models, adding it to a list of more traditional quality factors (Table 1.1). They also show that significant benefits can be achieved through integration.

The possession of domain specific knowledge [STR95] allows expert modelers to create conceptual models that have higher quality, for instance in terms of integration, flexibility, integrity and completeness, compared to novice modelers [Sha97]. Better integration can result from extensive experience with related models. Higher flexibility can be attributed to the genericity of the expert's domain conceptualization, meaning that his mental model is valid (i.e., showing integrity and completeness) in more than one situation, which leads to implicit integration between the situations covered by the same model and improved maintainability and robustness of expert models as these flexible models can cope with problem definition evolution without major changes to the conceptual model. [Ver05]

It is generally accepted that support (e.g., by providing expert knowledge, which is documented in a knowledge base system (KBS), to (novice) model-

Simplicity	means that the model contains the minimum possible symbols (e.g., entities and relationships in an ER diagram).
Understandability	is defined as the ease with which the concepts and structures in the model can be understood.
Integration	is defined as the consistency of the model with related models. (e.g., the rest of the organisation's data).
Completeness	refers to whether the model contains all user requirements.
Flexibility	is defined as the ease with which the model can cope with changes in the universe of discourse (e.g., business and/or regulatory change).
Implementability	is defined as the ease with which the model can be implemented within the time, budget and technology constraints of the project.
Integrity	is defined as compliance with domain axioms (e.g., business rules).
Syntactic correctness	is defined as proper use of the modeling language.

Table 1.1: Model quality variables as listed by Moody and Shanks [MS03]

ers [ABS05]) during the modeling process can improve the quality of conceptual models. [MS03] Many types of support can be identified. For example, Soffer and Hadar [HS06a, SH07] use ontology to support the identification of model variations, mitigating a novice's limited experience with a modeling language. Poels et al. [PMGP07] show how model understandability can be improved. Batra et al. [BW04] compare pattern- and rule-based modeling support for assuring model completeness with expert and novice modeler. Sugumaran and Storey [SS06] show that domain ontologies can improve model integrity. However, no model quality assurance or improvement approach specifically targeted at supporting the conceptual model quality factor 'integration' has been proposed yet.

Many of the proposed model quality assurance/improvement support methods use artifacts that capture expert knowledge. Some of them document proven solutions (e.g., patterns), others capture the shared conceptualization of the problem domain (e.g., domain ontologies). Patterns aim at reusing solutions (and solution variations) that have worked successfully for one or more experts in the past. [Bat05] Domain ontologies, on the other hand, provide a formal or at least explicit specification of a conceptualization that is shared by multiple contributors. [Bor97, Gru93] These domain ontologies represent the invariant conditions of the domain of interest and can be used to explain and integrate conceptual models. [FM07] Patterns, on the other hand, are conceptual models for solutions that have proven utility and should ideally instantiate the domain knowledge, which

can be captured in domain ontologies, for a specific purpose (e.g., recording or generating data). [Gui07] The instantiation of an ontology as a conceptual model with a specific purpose is called ontological instantiation. [AK03] The opposing relation is called the ontological commitment of a conceptual model. [Oli07] When patterns instantiate ontologies, a permissible range of pattern variations is defined by the ontology since a solution should reflect the problem domain. Therefore, the scope of a problem limits the range of valuable solutions. [FM07, Gua98] As ontologies provide a shared conceptualization they should therefore contribute to the integration, integrity, completeness, genericity and hence flexibility of their ontological instantiations.

The goal of this doctoral research is the development of a new approach for conceptual modeling of real-world situations in the business domain that specifically aims at achieving model integration by supporting the modeler's ability to creatively vary the modeling process within the limits of satisfactory constraints (i.e., implementability, understandability, syntactic correctness, simplicity, etc.). [HMJR04, Sim96] In this dissertation, we focus on satisfactory constraints for models that are imposed by the current state of domain knowledge as captured in domain ontologies (i.e., integrity and completeness). The desired consistency with the current state of domain knowledge is achieved by creating conceptual models that are ontological instantiations of the REA ontology [GM02], which captures the semantics of the business domain. Where necessary, the REA ontology will be complemented with other ontologies that capture other semantics than business semantics (e.g., the UFO foundational ontology [BGHS$^+$05], which accounts for concepts related to modeling like abstraction and representation).

The kinds of conceptual model that our research focused on are conceptual data models and business simulation models. Conceptual data models were chosen because they are prime assets for information systems development, as advocated by Moody and Shanks [MS03, p.619]

> *'The choice of an appropriate representation of data is one of the most crucial tasks in information systems development. Although data modeling represents only a small proportion of the total systems development effort, its impact on the quality of the final system is probably greater than any other phase [Sim00]. The data model is a major determinant of system development costs [ASM96], system flexibility [Gar92], integration with other systems [MS95b] and the ability of the system to meet user requirements [BK91]."*

Where data models enable us to document the current state and recall historic states of a relevant part of reality using information systems, simulation models allow us to represent potential future states of a relevant part of reality. Consequently, together data and simulation model allow us to address the past, present

and future of a relevant part of reality (e.g., an enterprise, a supply chain). Historic data allow us to analyze a relevant part of history. For example, business process intelligence [GCC+04] involves the use of historical data as a foundation for current and future decisions. Data about the present state of reality allows us to support operations. For example, business process management systems support the execution of an enterprises operations. Data about potential future states of reality can support the selection of alternatives. For example, business process simulation models allows managers to select optimized (e.g., most profitable) business process designs.

Our research method is inspired by the Design Science approach. Where Behavioral Science seeks to develop and verify theories that explain or predict human or organizational behavior (e.g., Why did or when will an integration attempt fail?), Design Science seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts (e.g., How can we create artifacts that integrate well?). [HMJR04] Unlike routine design, which applies existing knowledge (e.g., best practices) to solve problems, Design Science research addresses previously unsolved problems in unique or innovative ways or solves previously solved problems in a more effective or efficient way (e.g., expanding the problem scope). [HMJR04] Problems typically addressed by Design Science are characterized by (1) unstable requirements and constraints based on ill-defined environmental contexts, which requires the ability of experts to recognize a problem definition accurately and rapidly after recognizing a minimal number of situational clues [TRK90], (2) complex interactions among subcomponents of the problem and its solution, which may require an incremental problem solving process that gradually reduces the differences between an unsatisfactory current state and the desired goal state [Sim96], (3) inherent flexibility to change design processes as well as design artifacts, which motivates a quest for genericity [Ver05], (4) a critical dependence upon human cognitive abilities (e.g., creativity) to produce effective solutions and (5) a critical dependence upon human social abilities (e.g., teamwork) to produce effective solutions, which motivates the utility of shared conceptualizations as captured in ontologies. [HMJR04]

When designing enterprise information systems all these problem features can be recognized. Since uncertainty is an inherent characteristic of the business environment in which enterprises operate, unstable and ill-defined requirements for enterprise information systems are a logical consequence. The complex interactions between trading partners also lead to problem decomposition (e.g., a salesperson is not involved in the production process, marketing or financing of a product), although integrated approaches (e.g., business process orientation [McC01]) are gaining influence. The inherent volatility of the business environment also drives the need for flexibility in both the enterprise information design process and the resulting artifact, as information systems designs need to be adapted to changing

business requirement even during the information system design process. The dependence on human social and cognitive abilities is more obvious, since business consists of interactions between trading partners, which are (represented by) human individuals. Therefore, any business or enterprise information system needs to be able to communicate with or support the communication between human individuals.

1.1 On the resource-event-agent (REA) ontology

The REA ontology was selected because its level of abstraction, which is defined as a focus on the information relevant to a particular purpose and ignoring the remainder of the information [IEE90], fits the purpose of representing real-world situations in the business domain (i.e., economic reality). Additionally, REA is considered mature as its conceptualization has proven its robustness over the past 30 years [McC79, McC82] and because it builds upon accounting, which incorporates over 500 years of practical experience in recording economic phenomena (e.g., business transactions). The REA ontology was preferred over other ontologies because of its simplicity and expressive power. The e3-value ontology [Gor02], which is similar to REA, lacks the ability to represent individual transactions between business partners as it abstracts from them focusing on high-level business abstractions, where REA possesses the expressive power to represent both abstract transactions (i.e., types and groups of transactions) and individual transactions [GM02, GM06]. Also other ontologies look at business from a strategic perspective that abstracts from the logic of individual business transactions (e.g., BMO [Ost04], BMM [OMG08]). Dietz' enterprise ontology [Die06], which is complemented with the DEMO enterprise engineering and system development methodology, then looks at transactions from the control flow perspective, abstracting from the economic rationale that is inherent to REA. Ushold's enterprise ontology [UKMZ98], on the other hand, provides much weaker support to business modeling only providing a shared vocabulary for trading partners.

The REA ontology was introduced by W.E. McCarthy as a data model for a generalized accounting framework in a shared data environment in which both accountants and non-accountants (e.g., managers, salespeople) are interested in maintaining information about the same phenomena in the enterprise. [McC82] This data model originates in a modeling method that aimed at integrating database technology and accounting theory. [McC79] Both the method and data model address the same primitives (i.e., resource, event, agent) and the relationships between these primitives. Economic resources represent the goods, services and money that are produced (and stocked) by and flow through enterprises. Economic events represent the occurrences in time that animate these enterprises and drive those resource flows. Finally, economic agents are the natural or legal per-

sons (e.g., trading partners) that participate in those events (e.g., as initiator). As REA finds its origin in accounting it advocates the use of a particular association between economic events (i.e., duality), in addition to the flow of stocks (i.e., economic resources) that is animated by economic events and participation of economic agents in economic events. Duality relates increment events to decrement events, representing the economic rationale, which requires that all resources an enterprise gives up (e.g., by shipping them to a customer) should be replaced by resources of equal or greater value (e.g., though a cash receipt).

Later, the constructs from the data model were augmented with axioms to create the actual REA ontology. [GM04, GM00] These axioms addressed the rules that govern business seen from the perspective of a single trading partner. The first REA axiom stipulates that at least one inflow event and one outflow event exist for each economic resource and that inflow and outflow events must affect identifiable resources. [GM04] Consequently, this axiom requires that every economic resource has its origin in an inflow event (i.e., increment) and a purpose (i.e., being used in an outflow/decrement event). The second REA axiom addresses the economic rational by requiring that all events effecting an outflow must be eventually paired in duality relationships with events effecting an inflow and vice-versa. [GM04] Together, these two axioms define a healthy metabolism for an enterprise. The first axiom requires that all resources are useful and no resources will be stored perpetually. The second axiom requires the enterprise is rewarded for its efforts, preventing that its resources drain away. The third axiom then specifies that each exchange needs an instance of both the inside and outside subsets, requiring that each business transaction involves at least two trading partners (i.e., the enterprise that defines the viewpoint and an outside agent (e.g., supplier, customer)). Additionally, this axiom specifies that there is always an agent inside the enterprise (e.g., salesperson) that is accountable for the transaction.

Fig. 1.1 shows a trading partner view model for Elmo's cookie store [McC]. It show how Elmo (i.e., the entrepreneur) sells cookies to his customers (e.g., Cookie Monster). Since fig. 1.1 is a trading partner view model, it represents the perspective of one of the trading partners involved in the transaction. This model shows the transaction between Elmo's cookie store and its customers from Elmo's perspective. Conforming to the third REA axiom, it shows that the entrepreneur (i.e., Elmo) is considered to be part of the cookie store, which is represented by the inside participate association, and that customers are not (i.e., outside participate). The diagram also shows that a cookie sale decreases Elmo's cookie inventory, where a cash receipt increases Elmo's inventory of cash. Satisfying the second REA axiom, the duality association relates an inventory decreasing event (i.e. outflow) with an inventory increasing event (i.e. inflow). To satisfy the first REA axiom, the revenue cycle model in fig. 1.1, which shows Elmo in his role as cookie seller (i.e., selling cookies to his customers), has to be extended with an acquisi-

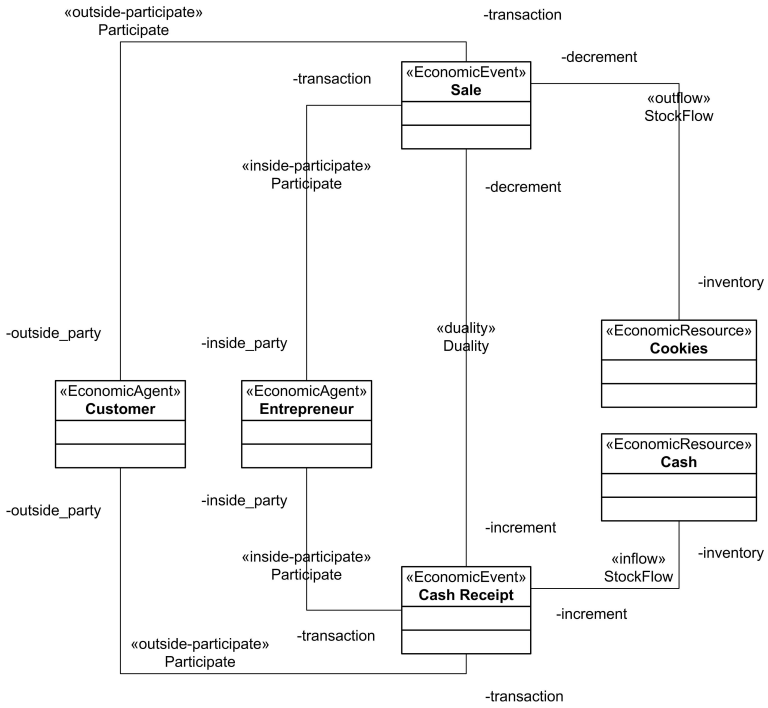

Figure 1.1: Trading partner view REA model for Elmo's cookie store

tion cycle model, which shows Elmo as a cookie buyer (i.e., purchasing cookies from his suppliers). Fig. 1.2 satisfies the first REA axiom by modeling a purchase event, which increases Elmo's cookie inventory, and a cash disbursement event, which decreases Elmo's cash inventory.

Most recently, REA's trading-partner view (i.e., from the perspective of one party in a business transaction) on the economic reality was complemented with an independent view, which focuses on the interactions between trading partners from the perspective of an independent observer that is not taking part in the business transactions. This independent view was developed for the purpose of developing an ISO standard for open-edi (i.e., electronic data interchange) that is specific for business transactions. [ISO07] Although this standard takes a totally new perspective on business, the REA primitives and axioms are also applicable in this context, since the same business reality is described and only the perspective has changed. Where the REA primitives can be transferred to this perspective without changing their actual definition, the articulation of the REA axioms is specific for the trading partner view. However, the intension of the REA axioms is also applicable in the

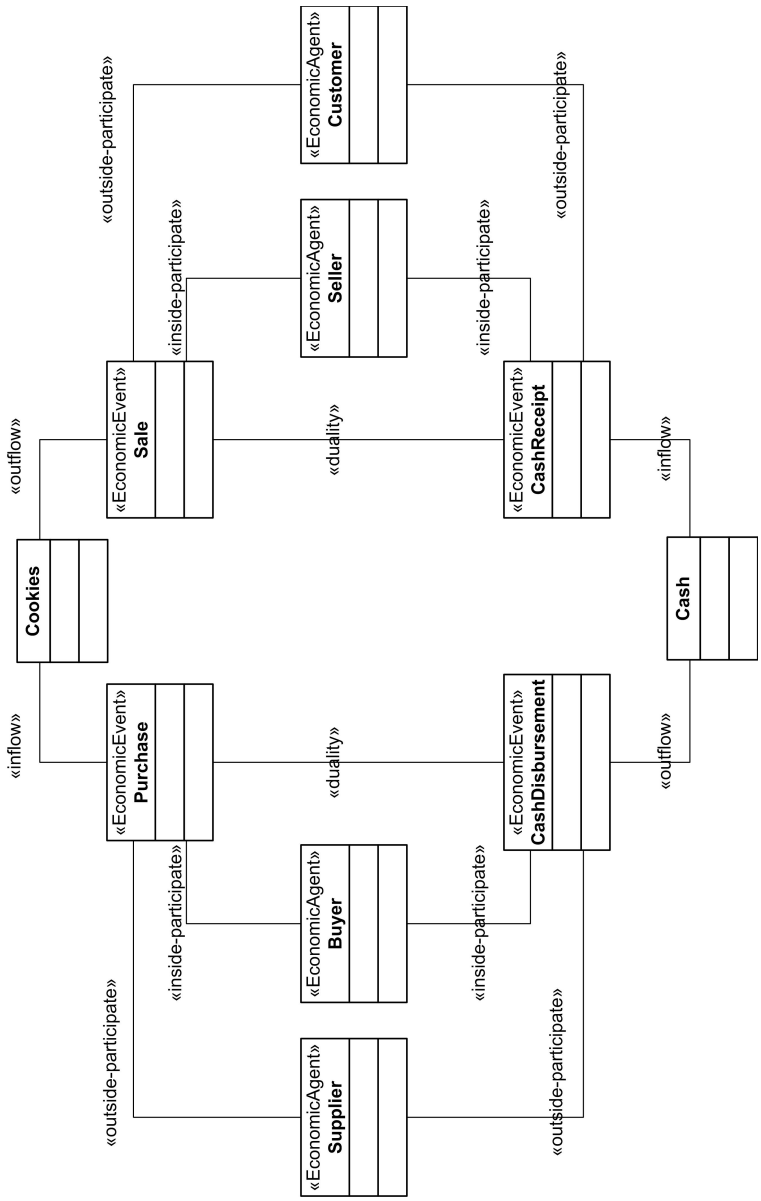

Figure 1.2: Trading partner view model for Elmo's acquisition and revenue cycle

independent view, as — even in the independent view — resources need to have an origin and a purpose, enterprises need to benefit from their activities, business transactions involve at least two trading partners and people are responsible for business transactions.

Scrutinizing fig 1.2 we see that the acquisition cycle, which shows Elmo in its role as a buyer, and the revenue cycle, which shows Elmo in its role as a seller, follow the same template. This shows us that the buyer and seller side of a transaction are congruent. When modeling a transaction from the view of both trading partners, two trading partner models should be created in which the inside and outside participate and inflow and outflow associations should be switched, indicating that the trading partner that is the inside participant in his trading partner view model is the outside participant in his trading partner's trading partner view model and that the event that represents an inflow of resources for one trading partner represents an outflow of resources for the other trading partner. To eliminate this creation of mirrored trading-partner view models when representing transactions between trading partners, independent view models were invented. Fig. 1.3 shows the independent view equivalent of fig. 1.1, representing the transactions between Elmo and his customers. Where fig. 1.1 only represented Elmo's view on cookie sales, fig. 1.3 can be read both from the cookie selling entrepreneur (i.e., Elmo) and the cookie buying customer (e.g., Cookie Monster), as the sale that decreases Elmo's cookie inventory is also the purchase that increases Cookie Monster's cookie inventory from an independent perspective. From this same independent perspective, the cash receipt that increases Elmo's cash inventory equals the cash disbursement that decreases Cookie Monster's cash inventory.

1.2 On ontologies in conceptual modeling

In conceptual modeling research, ontologies have been used in various ways. Some authors have used ontologies to evaluate and improve the expressiveness of modeling languages and their meta-models (e.g., evaluating the expressiveness of conceptual graphs [KC05] or process modeling languages [GR00], eliminating referential redundancy in UML [OHS05]) or providing theoretical guidance when using these modeling languages [Wan96]). Some have developed ontologies for the evaluation of modeling languages (e.g., evaluating ontological overlap in modeling languages [GRIM07]) and others have used ontologies as a foundation and benchmark to develop and evaluate domain-specific modeling languages [WPS09] (e.g., e3-value [Gor02], REA [GP07]). Next to the development and evaluation of modeling languages and their meta-models, ontologies have been used as a conceptual basis for model development and evaluation (e.g., conformance checking [HS06c], quality assessment [FL03, STW03], variation assessment [HS06a]). Furthermore, ontologies have been used to support the interpretation of data (e.g.,

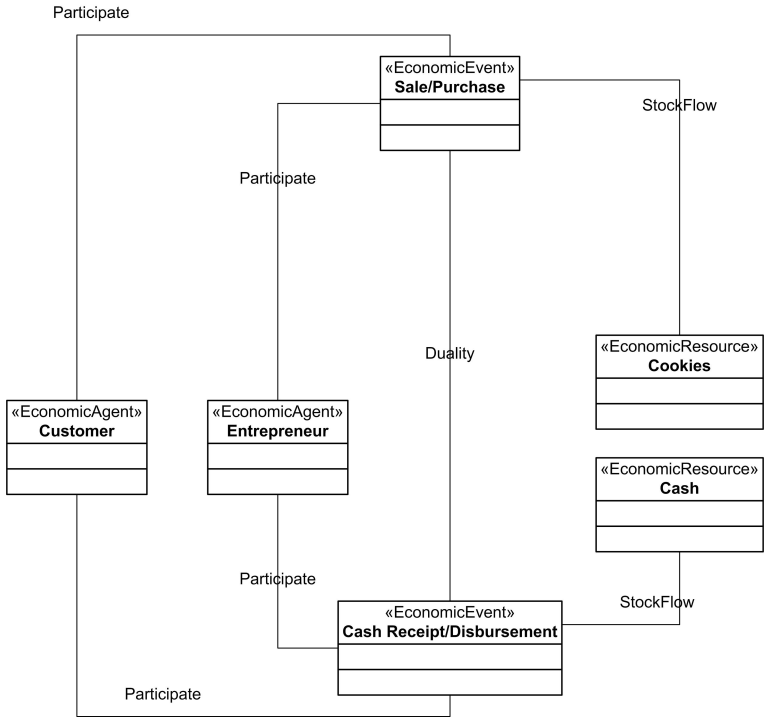

Figure 1.3: Independent view REA model for Elmo's cookie store

creating semantic interoperability between distributed information systems at data level [PR04] and schema level [CDA99]).

Ontologies also provide increasing support for interoperability between enterprise information systems, at data level, mitigating the lack of integration at the data model level. Despite these efforts, the development and effective use (e.g., through integration) of data models used by different applications within the same organization, let alone between organizations, remains an issue of concern. [KFS07] Therefore, this dissertation introduces the use of ontologies for the purpose of conceptual model integration. Although ontology-driven approaches for the conception of conceptual models exist, our approach is entirely new to the best of our knowledge as the existing ontology-driven design focuses on enhancing the integrity and completeness of conceptual models, where our approach uses the domain knowledge that is captured in ontologies to integrate conceptual models with each other. The first part of this dissertation addresses data model integration through the introduction of an ontology-based framework for data model integration and the development of an ontology-based data model that presents a unified representation of data for all trading partners, allowing for an integrated view on transactions between trading partners and the production processes that are executed by each of these trading partners. A second part of this dissertation presents ontology-based simulation models, which is expected to facilitate simulation modeling by reducing redundant effort in capturing or recreating information that has already been recorded elsewhere (e.g., by reusing a production process model, integrating it in a supply chain model to show its effect on the performance of the entire supply chain) and supporting simulation model integration, interoperability and (de)composability. [BG06, BPM06]

1.3 On the data model section

The data model section of this dissertation contains three papers. The first paper, entitled 'enterprise ontology-based structuring of conceptual data modeling patterns'demonstrates how REA can be used to analyze conceptual data modeling patterns, which is expected to support data modelers integrating conceptual data modeling patterns. Additionally it is shown that enterprise models that are conceived by REA-unaware modelers are largely REA compliant, as has also been demonstrated for SAP's data model earlier [O'L04]. This compliance validates REA as a proper conceptualization of the business domain. Where O'Leary's paper is limited to demonstrating the relevance of REA, we use REA's structuring orientation [DM97], which structures conceptual models such that their constructs can be retrieved more easily by imposing a fixed location [LS87, PMGP07]. We generalize this structuring orientation for all ontologies. Eventually, this generalized structuring orientation for ontologies should allow modelers to select the right

pattern for their purpose and support modelers while integrating multiple selected patterns to obtain a consistent model. Ultimately, such a generalized structuring principle for ontologies should allow the creation of tool support that allows modelers to define the universe of discourse as a set of ontologies according to which the patterns of interest are selected and structured, indicating how the patterns can be used and integrated correctly through their spatial location. Additionally, the structuring orientation can support pattern improvement and the creation of an extensible pattern language by providing a shared vocabulary and conceptualization accompanied by an inherent structuring mechanism. [CS95, ZBK09]

Consequently, the paper 'enterprise ontology-based structuring of conceptual data modeling patterns' demonstrates the generalized structuring orientation by structuring a validated set of frequently occurring data modeling patterns (i.e., Batra's pattern catalogue [Bat05]) according to an enterprise pattern and an abstraction pattern. The enterprise pattern is an instantiation of the REA ontology [GM02], which is used to structure the enterprise modeling constructs in Batra's conceptual data modeling patterns and facilitate their interpretation. The abstraction pattern is an instantiation of the UFO ontology [BGHS$^+$05], which is used to structure the abstraction mechanisms in Batra's patterns. Although the scale of the application (i.e., conceptual data modeling patterns for business transaction modeling at different levels of abstraction) is limited to a core of the REA and UFO ontology, the paper aims to illustrate the feasibility of such an integration supporting approach at a larger scale (e.g., using more and larger ontologies, incorporating larger pattern catalogues and languages and involving users).

The second paper of the data model section, called 'a resource-event-agent reference information model for representing value chains and systems', presents an ontological instantiation of the REA ontology that provides an observer-independent view of business, which integrates the independent and trading-partner view. Where many conceptual models limit their scope to a single enterprise and others focus solely on the supply chain, abstracting from the supply chain partner's business processes, this conceptual model incorporates both. Such a data model is expected to support the integration of data within and across enterprise boundaries and the interoperability of information systems used by different supply chain partners (e.g., the enterprise resource planning (ERP) system of a supplier and the vendor managed inventory (VMI) system of a customer). As REA has its foundations in accounting [GM86] such a boundary crossing business conceptualization should also support the creation of consolidated annual accounts, as it supports the regrouping of economic data changing the viewpoint of one trading partner into the view of a group of trading partners and is expected to facilitate the identification of transactions between consolidated trading partners.

The third and final paper of the data model section, called 'tracking and tracing future, present and past product and money flows' then presents a descriptive

evaluation of the observer-independent conceptual model, presented in the preceding paper, constructing detailed scenarios around the artifact to demonstrate its utility [HMJR04]. The presented application follows (i.e., tracks and traces) both planned and real product and money flows throughout a value system, which integrates the value chains of trading partners. Where other tracking and tracing systems limit themselves to following products though a supply chain (i.e., abstracting from the internal business processes of the respective supply chain partners) or following products within a single enterprises, the observer-independent model is shown to have the ability to track products through entire value systems (i.e., incorporating the supply chain partner's internal business processes in the supply chain model [PM85]). Our example application adds the tracking and tracing of money flows, which remunerate product flows, and future product and money flows to the product traceability capabilities that have already been developed for collaborative e-business systems [BCMT08]. Consequently, critical paths for the fulfillment of commitments (i.e., future events) can be identified and mitigation scenarios can be developed when desirable (e.g., when several food sources prove to be contaminated, alternative sources should be available to secure food supply in an area), where the current product traceability infrastructure only allows the identification of contaminated food sources and potentially contaminated products. Additionally, the registration of money flows is expected to allow for a straightforward simulation of the economic consequences of a supply chain intrusion (e.g., food safety scandal, counterfeit).

1.4 On data model formalization

In the data modeling section, the conceptual data models are represented using UML class diagrams. UML class diagrams are part of the unified modeling language (UML) that is developed and published by the object management group (OMG), which is a not-for-profit computer industry specifications consortium. The OMG is a leading proponent of business-IT integration standards that unites government agencies, small and large IT users, vendors and research institutions. UML is OMG's most used specification.

UML class diagrams consist of two main constructs (i.e., class and association). Classes represent types of things that have identity (e.g., a car, a person). Classes have attributes, which represent properties of the things that are represented by a class. Associations represent relationships between those things (e.g., a person owns a car). Associations have multiplicities which display how many instances of one class can/need to relate to an instance of an associated class. To achieve this, minimum and maximum multiplicities are shown on either side of an association. The multiplicities next to a class represent how many instances of this class can/need to relate to each instance of the class on the other end of

Figure 1.4: UML class diagram for car ownership with association

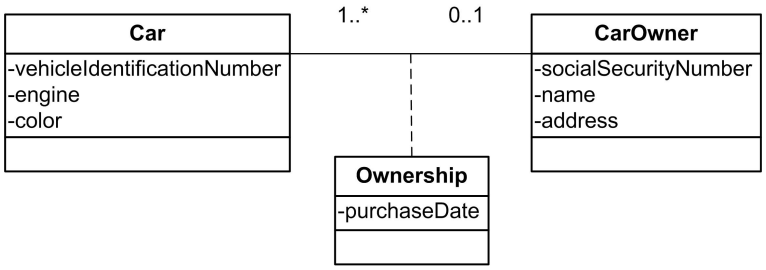

Figure 1.5: UML class diagram for car ownership with association class

the association. The minimum number of instances is followed by two dots and the maximum number of instances. When this maximum is infinite, an asterisk is shown. Fig. 1.4 shows a UML class diagram for car ownership. It shows that each car has a vehicle identification number, an engine and a color. In addition, it shows that a car owner has a social security number, a name and an address. The association (i.e., Ownership) in the model shows that cars are owned by persons which are then called car owners. The multiplicities show that a car owner must own at least one car, but is allowed to own an infinite number of them and that a car can be owned by one car owner at most, where it does not need to be owned by a car owner. It is important to note that an UML class diagram that is used as a conceptual data model represents reality at a certain point in time. Over time, a same car can be owned by multiple car owners, but at any point in time it is owned by at most one car owner.

When associations have properties of their own, they are represented as associations classes, which are associations that have attributes. Fig. 1.5 shows car ownership as an association class, which contains the attribute purchase date, which represents a property of the ownership relation and not properties of the car or the owner.

Classes and associations represent types of things and typical associations between them. Consequently, they represent sets of observations that document in-

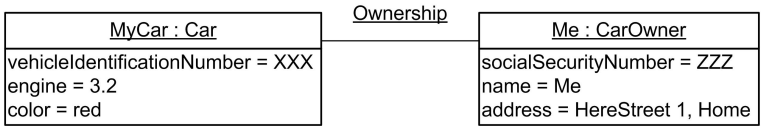

Figure 1.6: UML object diagram for car ownership, the case of me and my car

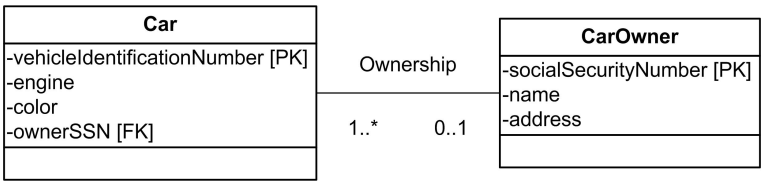

Figure 1.7: Annotated UML class diagram for relational database design

dividual real-world cases. These real-world cases are called instances, which can be modeled in UML object diagrams. Fig. 1.6 is an example object diagram that shows instances of the UML class diagram in fig. 1.4 for representing the particular case of me owning my car. I am represented by the object 'Me', which is an instance of the CarOwner class and has the attributes 'ZZZ'as social security number, 'Me'as name and 'HereStreet1, Home'as address. 'MyCar'is shown to be an instance of the Car class, that is characterized by the vehicle identification number 'XXX', has a '3.2'liter engine and is 'red'. The ownership link that instantiates the ownership association in fig. 1.4, complies with the multiplicity constraints defined in fig. 1.4 as MyCar is related to only one car owner (i.e., 'Me') and Me is related to at least one car (i.e., 'MyCar').

When a conceptual data model needs to be transformed in a relational database design, a relational data model is constructed. Many notations exist for such relational data models. In this thesis we use annotated UML class diagrams to represent relational data models. The difference with the conceptual data model in UML class diagram notation is that the relational data model contains extra attributes that represent the associations between classes. They are called foreign keys and are annotated with FK. Next to foreign keys, also primary keys (PK) are annotated in the models. Primary keys are the attributes or combinations of attributes that allow us to identify instances of a class uniquely. For example, in fig. 1.7 it is shown that a vehicle can be uniquely identified by its vehicle identification number, while a person can be uniquely identified by its social security number. Furthermore, the annotated model in fig. 1.7 shows that the attribute ownerSSN is used to represent that a car is owned by a person.

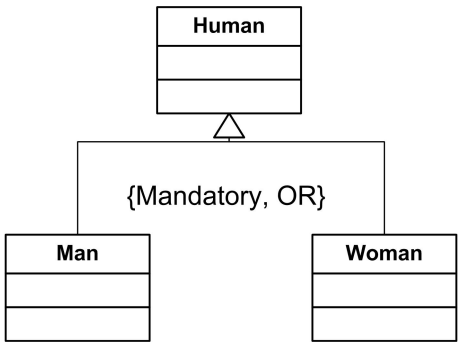

Figure 1.8: Mutually exclusive sub-classes, with abstract super-class

Where associations link instances of different classes to each other, specialization/generalization shows that instances of one class are/ can also be instances of other classes. Specialization shows that an instance of a sub-class is a particular case of an instance of super-class, where generalization shows that instances of several sub-classes share properties, which can be shared and represented in a generalizing super-class. In UML class diagrams specialization/generalization is represented with an arrow of which the hollow arrowhead point toward the super-class. In some cases, the super-class is abstract, meaning that it has no instances of its own and that it only groups properties that are shared between its sub-classes. In that case the specialization is mandatory. However, when the super-class is allowed to have instances that are not instances of one of its sub-classes the specialization is optional. When sub-classes of a super-class are mutually exclusive, the specialization/generalization is annotated with 'OR', when instances of a super-class are allowed to be instance of multiple sub-classes at the same time, the specialization/-generalization is annotated with 'AND'.

Fig. 1.8 shows the example of gender. All humans are either man or woman although they share the same set of human characteristics. That all humans are either man or woman, is represented by the mandatory annotation, which signals that no humans can exist that are not man or woman. The OR annotation shows that no humans can exist that are both man and woman at the same time. Fig. 1.9 then shows an example of non-mutually exclusive sub-classes. It shows that humans can play different roles. They can be car owners, parents and employee or neither of these. The optional annotation shows that humans can exist that are not car owners, parents or employees and the and and annotation shows that humans can exist that are multiple of those (e.g., car owner and employee).

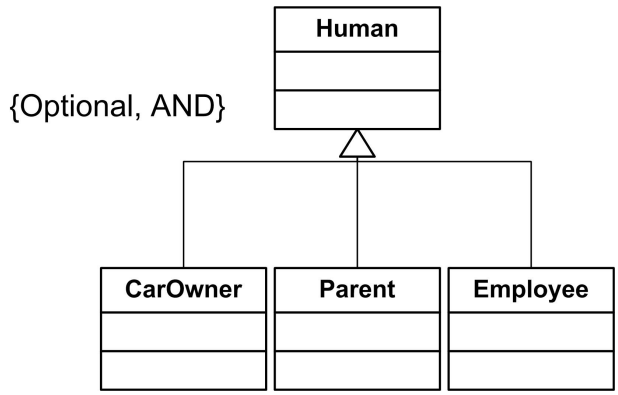

Figure 1.9: Non-mutually exclusive sub-classes

1.5 On the simulation model section

The simulation model section of this dissertation contains two papers. Both papers extend the traditional scope of business process simulation models, from logistic simulations, simulating waiting lines and delays, to financial simulations, simulating assets and liabilities, profits and losses. By creating specific REA-based financial simulation models and allowing the incorporation of logistic simulation models in these financial models, integration between financial and logistic simulation models is achieved. Where traditional business process simulations evaluate the quality of their design in terms of production time and cost, product quality and process flexibility [RLM05], our REA articulations allow modelers to convert these technical parameters in financial parameters (e.g., profit), which allows business modelers to evaluate their business process lay-outs in terms of profit, capital structure and other financial target variables. As these financial parameters determine the viability of a business (e.g., an enterprise will fail by definition when it cannot pay its suppliers but not necessarily when its cycle time is too large) and current ontology-based simulation modeling approaches are limited to logistic processes [CFMK07, DDRW06, FRM05, SLM06], REA enables business process modelers to leverage the expressive power of their models significantly.

The papers of the simulation model section present artifacts for REA-based simulation models, an approach that was pioneered by Church and Smith [CS08]. Where Church and Smith use system dynamics to create mathematics-based simulation models, the simulation model section papers introduce Petri-net based statistical simulation models, that integrate the capabilities to simulate (parts of) an enterprise's value chain, as demonstrated by Church and Smith, with the possibility to model and simulate the individual business processes that make up the value

chain and the ability to embed the value chain models of multiple supply chain partners in a complete value system model. As a result, the papers demonstrate how the REA conceptualization of business and enterprises can help to construct complete value system models, integrating business process models in value chains models and value chain models in supply chain models. The first paper chooses a workflow approach that focuses on the sequences of events, determining which event sequences are (im)possible for a given (set of) transactions. The second paper then presents discrete statistical simulations for value chains and systems based on business process characteristics (e.g., capacity, throughput time, throughput variability). Since the aim of both papers is to demonstrate the feasibility of such simulation models, the represented case examples are kept relatively simple.

The first paper, entitled 'simulating liquidity in value and supply chains' instantiates the REA ontology as a workflow model, addressing the sequence of events that is required to complete business transaction configurations (e.g., prepaid sale) and the consequences such sequences and changes to these sequences have on an enterprise's liquidity position. For example, when a supplier requires advance payment instead of granting a term of payment, its customers will need to fund their purchases with debt other than the commercial debt provided by the supplier (e.g., bank loans). The paper presents configurations for business transactions between trading partners and value chains of individual trading partners. Additionally, it is shown how these models can be integrated to construct supply chain models that show the effect of product flows and terms of payment on the liquidity position of supply chain partners (e.g., proliferation of liquidity constraints in supply chains).

The second paper of the simulation model section, called 'a simulation model articulation of the REA ontology', elaborates on the preceding paper, presenting a layered composability for REA simulation models. The top-layer models supply chains in terms of transactions between trading partners. The second layer models the value chain of individual trading partners and the third layer addresses the business processes that construct the trading partners'value chain. However, since REA abstracts from the control flow aspects that govern business processes, and other methods have been present to construct such models (e.g., DEMO [Die06], product-based workflow design [RLvdA03]), the latter level is reduced to the minimum required for addressing the effect of the duration of administrative and production processes on the need for funding of an enterprise. For example, when budgets are approved every two weeks, money received from payments stands idle for a week on average, while the idle money might be used to fulfill due payments and reduce the need for funding. Equally, this business process layer illustrates how the presented REA-based modeling method can be integrated with more traditional business process modeling approaches.

1.6 On the simulation model formalization

Both papers in the simulation model section use Petri-net based notations. The first paper uses a notation based on conventional Petri-nets, the second uses colored Petri-nets. The main difference between colored and conventional Petri-nets is that the tokens in colored Petri-nets have a color. This color is a metaphor for the data that are attached to this token. Consequently, colored Petri-nets are Petri-nets that are integrated with a data model. This data model allows us to keep track of time and model delays in a model. Additionally, it allows us to represent the value of resources as they flow through a value system that consists of business processes integrated in value chains, which are then integrated in a supply chain. This integration can be achieved by using hierarchical Petri-nets, which allow transactions in one Petri-net to consist of an entire lower-level Petri-net. Next to colored and hierarchical Petri-nets, also timed Petri-nets, in which transitions are allowed to have a duration that can cause delay, can be identified. Some Petri-nets also combine several of these features (e.g., timed and colored).

Conventional Petri-nets consist of four elements (i.e., tokens, places, arcs and transitions). Tokens represent things that have identity and flow through a modeled process (e.g., a package through a logistic process). They are often represented as dots. Places store tokens. They are often represented as circles that (can) contain dots. Transitions represent events, processing tokens by moving them from one place to another and optionally changing their color (i.e., data value). Transitions are often represented by boxes or rectangles. Finally, arcs indicate which transitions are allowed to consume tokens from a place(i.e., input arc) and which places can receive tokens from a transition (i.e., output arc). When a transition consumes a token on one or more places and one or more places receive a token from this transition, the transition is said to fire. In the original Petri-net notation, a transition is only allowed to fire when all of its input places (i.e., all places that are connected to this transition with an input arc) contain a token. When a transition fires, all of its output places (i.e., all places connected to this transition with an output arc) receive a token. Where the transitions in the original Petri-net represent AND-port semantics, firing when all their input places contain at least one token and placing a token in each of their output places, alternative Petri-net based notations may define OR-port semantics that allow a transition to fire when only one of its input places contain a token and depositing a token in only one of its output places. Even NOT-ports can be defined, requiring that a certain transition can only fire when its input port contains no token. In colored Petri-nets, it is only possible to define transition that can only fire when one or more of its input ports contain a token of a certain color (i.e., with a particular data value).

Fig. 1.10 shows an example conventional Petri-net. In the example, place P1 contains two tokens. Since P1 is an input place for transition T1, T1 is able to

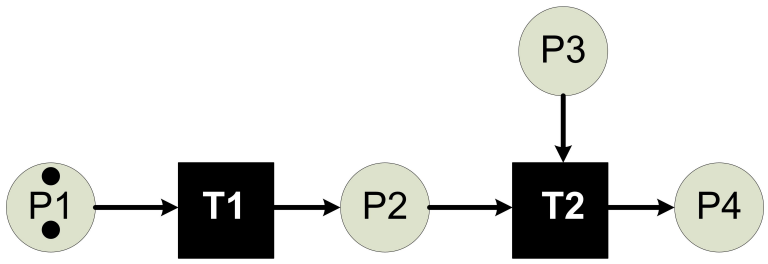

Figure 1.10: An example Petri-net

fire twice. Every time T1 is fired, one token is moved from P1 to P2 (i.e., T1's output place). T2 then has two input places (i.e., P2 and P3). Since P3 contains no tokens, T2 will not be able to fire. A transition that is not able to fire given an initial distribution of tokens is called dead. Consequently, T2 is a dead transition in this Petri-net. If P3 would contain at least one token, it would not be. [JK09]

Finally, fig. 1.11 shows an example colored Petri-net. In the example, place P1 contains three colored tokens (i.e., one red token at the top and two blue tokens at the bottom of P1). Place P1 is connected with place P2 through the transactions T1 and T2. Where this would mean that transactions T1 and T2 would be in conflict with each other (i.e. they would both be enabled to fire at the same time, which would lead to a non-deterministic choice) in a traditional Petri-net, this is not the case in this colored Petri-net as transaction T1 exclusively accepts red tokens from place P1 and transaction T2 only accepts blue tokens from place P1. Consequently, conflict is avoided, showing how a tokens color (i.e., data value) can determine its path through a colored Petri-net.

1.7 On the papers in this dissertation

Parts of this dissertation have been published, other parts are still in review. Both simulation section papers have been published as conference proceedings listed in the ISI Proceedings Index. All three papers of the data model section are still being reviewed for international peer reviewed journals that are listed in ISI Science Citation Index. The first paper of the data models section has been revised and resubmitted to Journal of Database Management (JDM). The second paper has been submitted to Journal of the Associations for Information Systems (JAIS). The third paper has been submitted to Information Systems Management (ISM).

Two other papers, only indirectly related to the doctoral research and therefore not included in this dissertation, have been published as conference proceedings listed in the ISI Proceedings Index. The first paper presents an integration between

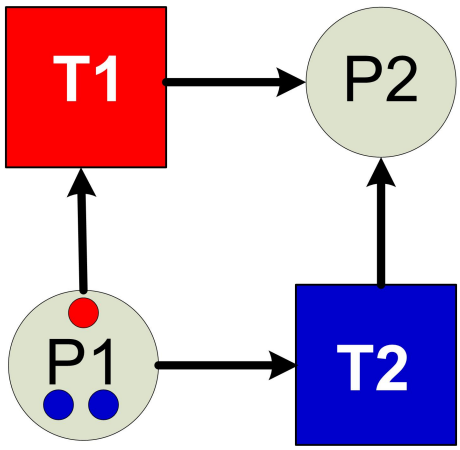

Figure 1.11: An example colored Petri-net

McCarthy's and Hruby's REA reference models and was presented at the ICEIS conference. The second paper was presented at the ISAMI symposium and addresses the use of the MERODE modeling methodology and algebra [DS95,SD98, SLG+04,Sno99,SD96,SP00] to strengthen the flexibility and reliability of context-aware pervasive applications implemented using the ReWiRe framework [VLC08] through the use of ontology-based context models. Additionally, one conference proceedings paper has been published as CEUR workshop proceedings and is still in review for the 'Transactions on Pattern Languages of Programming'(T-PLoP) subseries of 'Lecture Notes in Computer Science'(LNCS). This paper presents a pattern language that should help starting entrepreneurs to conceive a proper business plan when no business plan template fits their business idea (e.g. because it is too innovative). Moreover, Wim Laurier has been a coauthor of 'Positioning and Formalizing the REA enterprise ontology'. This paper, which presents a more complete and precise graphical REA ontology specification using a UML modeling profile that is also machine readable, has been published in 'Journal of Information Systems'(JIS), which is the academic journal of the Information Systems Section of the American Accounting Association (AAA).

Below, we list all publications and conference contributions that were published in the course of the doctoral program.

Publications in international journals

- Gailly, F., Laurier, W., & Poels, G. (2008). Positioning and Formalizing the REA enterprise ontology. Journal of Information Systems, 22(2), 219-248.

Publications in international conference proceedings

- Laurier, W. & Poels, G. (2009) Simulating Liquidity in Value and Supply Chains. CIAO! 2009 and EOMAS 2009, Amsterdam, 2009, pp. 40-54. Lecture Notes in Business Information Processing 34.
- Laurier, W. & Poels, G. (2009) A Simulation Model Articulation of the REA Ontology. On the Move to Meaningful Internet Systems: OTM 2009 Workshops, 2009, pp. 554-563. Lecture Notes in Computer Science 5872.
- Laurier, W., Hruby, P., & Poels, G. (2009). Business Plan Conception Pattern Language. Paper presented at the 14th European Conference on Pattern Languages of Programming (EuroPLoP), Irsee, Germany.
- Laurier, W., M. Bernaert, and G. Poels. (2010) A Consolidated Enterprise Reference Model, Integrating McCarthy's and Hruby's Resource-Event-Agent Reference Models. 12th International Conference on Enterprise Information Systems, Funchal, Madeira - Portugal, 2010 pp. 159 -164. Information Systems Analysis and Specification 3.
- Laurier, W., G. Vanderhulst, G. Poels, and K. Luyten. (2010) Rewiring Strategies for Changing Environments. Ambient Intelligence and Future Trends-International Symposium on Ambient Intelligence (ISAmI 2010), 2010, pp. 45-53. Advances in Soft Computing 72.

Preparing the preceding publications, together with the papers that are still in review, earlier versions have been presented at various workshops and conferences. We list these earlier versions below.

Other conference and workshop contributions

- Laurier, W., & Poels, G. (2007, June 13-15). Three fundamental REA business patterns. Paper presented at the REA-25 Conference - Special track on REA business patterns Newark, USA.
- Laurier, W., Poels, G., & Gailly, F. (2007, June 13-15). REAP the fruits of REA: An ontological evaluation of Firstness. Paper presented at the REA-25 A Celebration of the REA Enterprise Model.
- Laurier, W. (2008, June 15th). Applying Guarino's Aufbau principle for ontologies to REA-based applications. Paper presented at the 3rd REA Technology Workshop, Montpellier, France.
- Laurier, W., & Poels, G. (2009, February 9-10). Conceiving e3-value Business Plans. Paper presented at the Workshop on Value Modeling and Business Ontologies (VMBO), Stockholm, Sweden.

- Laurier, W., & Poels, G. (2009, October 23). A Framework for Open Pattern Languages of Reconfigurable Conceptual Models in MERODE. Paper presented at the 4th SIKS/BENAIS Conference on Enterprise Information Systems (EIS09), Nijmegen, The Netherlands.
- Laurier, W., & Poels, G. (2009, December 21-22). A Synthesis of REA Reference Models. Paper presented at the 4th International Workshop on Value Modeling and Business Ontologies (VMBO), Amsterdam, The Netherlands.
- Laurier, W., & Poels, G. (2009, December 21-22). Extending REA Models with a Reference Model for Abstraction Mechanisms. Paper presented at the 4th International Workshop on Value Modeling and Business Ontologies (VMBO), Amsterdam, The Netherlands.
- De Coster, M., Laurier, W., & Poels, G. (2009, December 21-22). An e3-Value Pattern for Valuing Customer Retention Initiatives. Paper presented at the 4th International Workshop on Value Modeling and Business Ontologies (VMBO), Amsterdam, The Netherlands.

Part I

Data Models

2

Enterprise Ontology-Based Structuring of Conceptual Data Modeling Patterns

Abstract

This paper argues that the scope of conceptual data modeling patterns can be identified, clarified, and explicitly represented by positioning them into an ontology-based framework. A clear and explicit definition of scope could help novice modelers in better deciding which patterns match which parts of the domain to be represented. We demonstrate our argument by positioning the conceptual data modeling patterns selected by Batra into a two-dimensional structuring framework that is constructed using two ontology-derived benchmark patterns: an enterprise pattern for representing transactions (derived from the REA domain ontology) and an abstraction pattern for representing reality at different levels of abstraction (derived from the UFO foundational ontology). By means of an application scenario dealing with the conceptual design of a transactional enterprise data warehouse, we illustrate how modelers can use the framework to evaluate the relevancy and completeness of candidate patterns with respect to the problem at hand.

Keywords

Conceptual data model, pattern, ontology, structuring orientation

2.1 Introduction

Conceptual data modeling patterns are used to support the conceptual design of databases. A conceptual data modeling pattern describes a reproducible solution to a general conceptual database design problem, which can be used in a specific context by customizing its classes, associations, and attributes [Bat05, RBS09]. Many proposed conceptual data modeling patterns exist [CNM97, Fow97, GHJV95]. The list of eleven patterns proposed by Batra [Bat05] synthesizes these proposals, with a focus on patterns that can be used for the conceptual design of enterprise databases. The selection of patterns made by Batra was justified by showing that they occur frequently in three influential sources for conceptual data modeling: the enterprise reference models of Scheer [Sch98] and Silverston et al. [SIG97] and the casebook of Whitlock et al. [WNP03].

Research has shown that the use of conceptual data modeling patterns increases the productivity of database designers, especially novices [ABS05]. However, it has also been observed that the use of patterns stimulates the inclusion of superfluous concepts, which are copied from the selected pattern even if they are irrelevant for solving the problem (Batra and Wishart, 2004). This inclusion of irrelevant concepts is explained by the human tendency to anchor to an initial solution [TK74].

Whereas the use of patterns might result in models with irrelevant elements, a prime problem observed with models produced by novice modelers is model incompleteness [MS03], meaning that the conceived model does not fully represent the domain concepts and relations that are deemed relevant for storing data about in the database. What seems to be important to address this dual problem of irrelevancy/incompleteness in models produced by novices, is being able to accurately assess the scope of the domain to be represented as well as that of the patterns (or other instruments like reference models) that are used to guide the modeling process. Ideally, the scope of conceptual data modeling patterns is explicitly defined and represented when offering them to novice modelers, e.g., as part of a pattern catalogue or repository. An explicit indication of scope could help novices in deciding which patterns match which parts of the domain to be represented, and as such improve model relevancy and completeness.

As a solution to this problem, we propose an ontology-based framework to define the scope of the conceptual data modeling patterns that are intended for a particular domain in terms of the generic domain knowledge that is captured by domain ontologies. Patterns can be positioned in this framework by ontologically classifying the elements they contain, which helps clarifying their scope (i.e., what part of the domain they cover).

The structuring orientation [DM97] capability of domain ontologies has already been shown to improve the understandability of models that are constructed

using ontology-derived patterns [PMGP07]. We believe ontologies can not only support the interpretation of patterns, but also the structuring of pattern catalogues and repositories, helping modelers to find the patterns that are relevant for the problem at hand. The positioning of patterns in the framework is expected to mitigate relevancy and completeness errors as the framework then visualizes the parts of the domain for which a solution has been suggested (e.g., by a pattern from a catalogue) and the parts for which no solution has been presented yet.

We have applied our ideas for an ontology-based structuring of conceptual data modeling patterns to Batra's selection of patterns [Bat05]. As about half the patterns in Batra's catalogue address the modeling of different kinds of transactions, we selected the REA ontology [GM02] as domain ontology because REA is a widely accepted enterprise ontology that describes the business logic of transactions (e.g., the ISO has recognized REA as a standard for a business transaction ontology [ISO07]). As the other patterns in Batra's selection address the use of different abstraction mechanisms used in conceptual data modeling (e.g., generalization-specialization, type-instance, and part-whole relationships), the UFO ontology [BGHS$^+$05] was chosen to provide a second structuring dimension. UFO, which is a foundational ontology that intends to serve as a theoretical basis for conceptual modeling, explicitly addresses the different levels of abstraction at which reality can be represented (e.g., modeling physical things like customers and abstract things like customer segments) and the appropriate abstraction mechanisms to do so.

From each ontology (i.e., REA and UFO) we derived a pattern that can be used to compare with the patterns in Batra's selection. This comparison determines the position of Batra's patterns in the framework (and hence clarifies their scope). The two 'benchmark'patterns can be seen as the building blocks that compose the structuring framework and that provide orthogonal classification dimensions for conceptual data modeling pattern elements.

Section 2.2 presents the two 'building block'patterns of the framework used to classify Batra's patterns. Section 2.3 positions each of Batra's patterns in this framework. Section 2.4 employs an application scenario [HMJR04] to illustrate how this positioning can guide modelers in the use of these patterns. Section 2.5 presents conclusions and future research.

2.2 Building Blocks for a Structuring Framework

The two ontology-derived patterns used as building blocks for the framework are composed of classes for representing domain concepts and associations for representing relations between domain concepts. The data to be stored about domain concepts and their relations depend on the functional requirements of the applications that will be supported by the database. For instance, if sales transactions are

registered in a database, then for an accounting application, the cost of the sales as determined by the value of the goods sold and the cost of the selling itself, are primordial, whereas for a logistics application, distance and duration to the customer and package volume are more relevant. Although in both cases the conceptual data model representing the sales transactions has an identical structure, with the same concepts and relations, the attributes to be included are different. It is therefore not surprising to find that not all conceptual data modeling patterns found in the literature show attributes.

Furthermore, alternative patterns for a same problem may also describe their solutions using different multiplicities that represent constraints on the relations between domain concepts. However, these constraints often depend on the context of the situation that is modeled. Fig. 2.1 shows, for instance, the employment patterns proposed by Fowler [Fow97] and Eriksson & Penker [EP00]. In Fowler's pattern, an employee can have only one employer whereas multiple employers and contracts of employment are allowed in Eriksson and Penkers's pattern, which may account for situations like part-time employment with multiple employers simultaneously but also for sequential employment arrangements.

As patterns are reusable solutions, they should be useful for different application contexts. As a result, the two 'building block'patterns abstract from participation and cardinality constraints and do not contain representations of the properties of concepts and relations, even if we do suggest initial lists of attributes that are potentially relevant for the concepts and relations modeled.

2.2.1 Enterprise Pattern

The enterprise pattern provides the classification dimension with respect to the intended domain of the Batra patterns. The enterprise pattern is derived from the Resource-Event-Agent (REA) enterprise ontology [DCH05, Hru06, McC82, McC03]. REA started as a generic accounting data model for a shared data environment where both accountants and non-accountants are interested in maintaining information about the same set of economic phenomena [McC82]. REA has now grown into a comprehensive, though highly abstract, and concise ontology accepted by the ISO as a standard for describing business transactions [GM02, GM04, GM06, ISO07]. Research has shown that the REA description of business transactions can for instance be recognized in the major part of the SAP R/3 reference data model, i.e., the dominant ERP system [O'L04].

The enterprise pattern (Fig. 2.2) shows the three core REA constructs and their relationships. Since REA focuses on transactions, the scope of the enterprise pattern is the interactions between natural and/or legal persons and the goods and services they exchange and produce in an economic context.

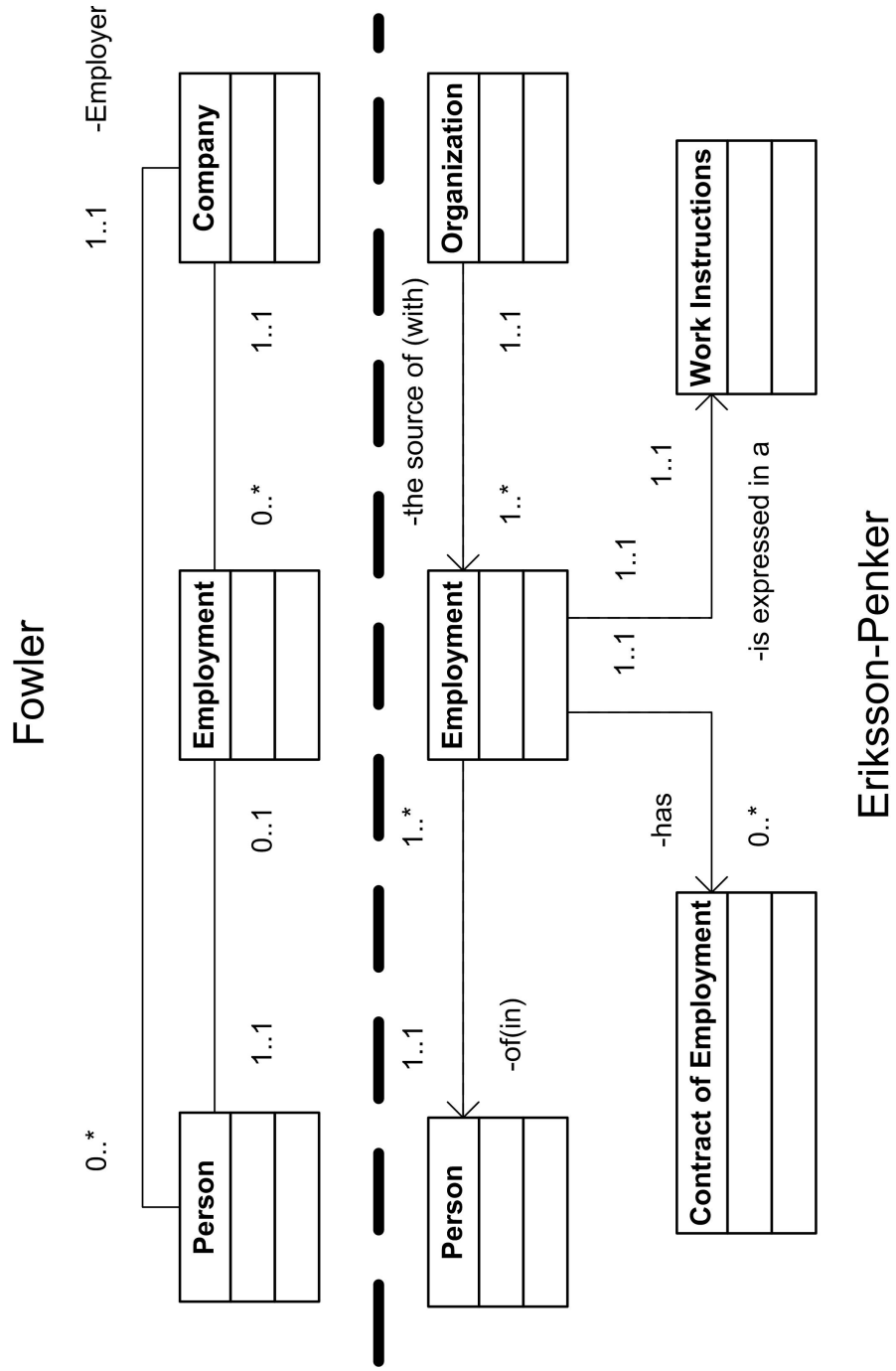

Figure 2.1: Employment patterns proposed by Fowler [Fow97] and Eriksson & Penker [EP00]

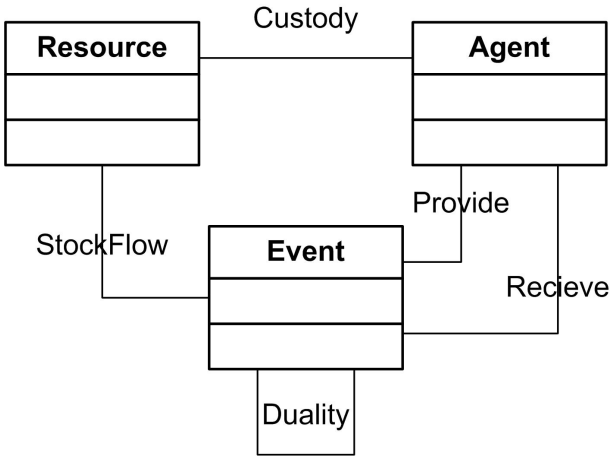

Figure 2.2: Enterprise pattern

Resources represent things that have utility or value and are under the control of a person (legal or natural) [Iji75, ISO07, McC82]. Potential attributes for the Resource class are indications for a resource's value (e.g., cost, price), utility (e.g., potential use and functionality), and characteristics (e.g., size, weight). **Events** are occurrences in time that represent changes in the utility or value of controlled resources due to ownership transfer, production, consumption, distribution and execution of rights and obligations (e.g., the right to use a rented good and the obligation to return it) [ISO07, McC82, Yu76]. Potential attributes for the Event class indicate an event's location in space and time (e.g., start and completion time). **Agents** are the natural and legal persons that participate in events (e.g., as a trading partner or executor of events) and have control over resources. Typical for the Agent class are attributes that represent properties for characterizing agents (e.g., knowledge/skills, educational level, experience level, ratings, etc.).

The **custody** relationship between agents and resources represents the control that agents have over resources. Control means that agents own the resources or otherwise have access to them so that benefits can be gained from their use. The **stockflow** relationship then shows the effect of events on resources, i.e., which resources have their value or utility changed by events. The **duality** relationship models the principle of economic reciprocity (i.e., *quid pro quo*) and identifies requiting events. If an event causes an increment in the value of the resources controlled by an agent, then economic reciprocity dictates that there will be another event causing a decrement in the value of the resources controlled by the agent. For instance, the reception of goods delivered by a supplier will lead to a payment of

these goods, meaning that the receiver of the goods loses control over the money required to pay for the goods. Finally, **participation** relationships identify the agents for whom an event is a decrement (i.e., the provider) and for whom it is an increment (i.e., the recipient).

2.2.2 Abstraction Pattern

Given the wide variety of abstraction mechanisms used in conceptual modeling, such as aggregation and composition [BHSLPLB03, OMG09, OHSB01], typification [GM06], classification [Ode98], and membership [Bro83, MPS95], modelers are likely to benefit from ontological support for correctly interpreting the meaning of these mechanisms. This support is given by the Unified Foundational Ontology (UFO) [GW05], which is a foundational ontology specifically crafted as a theoretical basis for conceptual modeling. The abstraction pattern (Fig. 2.3) that we use as a second structuring dimension of the framework is derived from UFO and captures common abstraction mechanisms that are used in conceptual modeling (e.g., aggregation and composition, generalization and specialization, classification) along with less used set-related abstraction mechanisms (e.g., membership or grouping).

In UFO, a distinction is made between three elementary conceptual modeling categories, i.e., types, groups and individuals. A **Type** is defined as *"an entity that has an extension (being a set of entities that are instances of it) and an intension, which includes an applicability criterion for determining if an entity is an instance of it"* [GW05]. Potential attributes for the Type class in Fig. 2.3 are those that represent properties which are used to define applicability criteria for determining a type's instances (e.g., planes have a wingspan). **Individuals** are *"entities that are not entity types"* [GW05], meaning that they are instances of types that have no instances of their own. Potential attributes for the Individual class point towards a unique identity (e.g., chassis number, social security number, sales order number). **Groups** map to sets, which are *"entities that have other entities as members"* [GW05] because they represent collections of individuals. In contrast to types, for which a set of individuals (i.e., extension) is defined by a types intension, groups can also be created by enumeration. For example, listing individuals of distinct types that belong together (e.g., everything that is owned by a person). Potential attributes for the Group class represent characteristics that are shared by the members (e.g., the group of all red things share the same color). They also include derived attributes that represent characteristics of the group itself (e.g., the number of members and their average or summed weight and size).

The scope of the abstraction pattern can be defined as the domain-independent conceptual relationships between objects at the same level of abstraction and at different levels of abstraction. Fig. 2.3 shows six kinds of such abstraction relation-

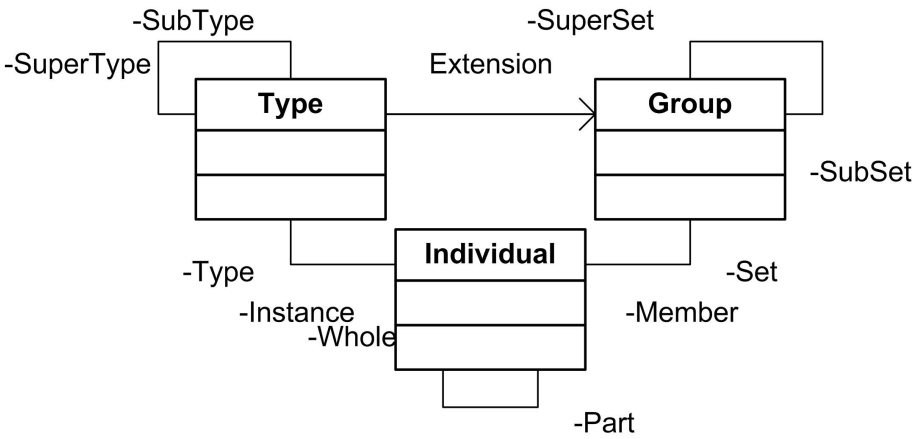

Figure 2.3: Abstraction pattern

ships. First, part-whole relationships are represented by the recursive relationship on the INDIVIDUAL class. Part-whole relationships assign PART and WHOLE roles to INDIVIDUAL objects and signify that in the real world an INDIVIDUAL object is part of another INDIVIDUAL object, which is the whole that the first INDIVIDUAL object is part of.

Second, subtype-supertype relationships are represented in Fig. 2.3 by the recursive relationship on the Type class. These relationships express specialization and generalization semantics, meaning that a supertype (identified by the SUPERTYPE role) generalizes its subtypes by emphasizing the characteristics that are common to the instances of these subtypes and a subtype (identified by the SUBTYPE role) specializes its supertype by describing additional characteristics only exhibited by the instances of the subtype considered. Subtype-supertype relationships are also known as 'is-a'relationships as each instance of a subtype is also an instance of the supertype of the subtype. As subtypes inherit the characteristics of the instances of their supertype, it is common in conceptual modeling that these inherited characteristics do not need to be specified explicitly for the subtypes.

Third, classification-instantiation relationships are recognized and represented by the relationship between the TYPE and INDIVIDUAL classes in the abstraction pattern (fig. 2.3). These relationships connect types to their instances. The INSTANCE role is assigned to INDIVIDUAL objects and the TYPE role is assigned to TYPE objects (e.g., 'John Doe'(i.e., an individual) is an instance of 'Human'(i.e., a type)).

Fourth, the membership relationship between a SET (i.e., GROUP object) and its MEMBERS (i.e., INDIVIDUAL objects) is shown in the model. This relationship

assigns MEMBER roles to INDIVIDUAL objects and SET roles to GROUP objects (e.g., 'John Doe'(i.e., an individual) is a member of the 'Flat Earth Society'(i.e., a group)). This approach to representing sets is also known as grouping [GM06].

Fifth, the abstraction pattern also contains a recursive relationship on the GROUP class that represents subset-superset relationships between GROUP objects, where one GROUP object can be a subset of another GROUP object. For instance, all members of 'Flat Earth Society'(i.e., SUBSET) are also members of 'Society'(i.e., SUPERSET)).

Sixth and finally, fig. 2.3 includes a relationship between TYPES and GROUPS. As opposed to the other relationships in this model, it can only be read in one direction (i.e., from type to group). This relationship relates a TYPES object to its extension (i.e., a GROUP object), which is the set of INDIVIDUAL objects that are instances of the TYPE object.

2.2.3 Composing the Framework

The combination of the enterprise and abstraction patterns results in a structuring framework with two orthogonal dimensions. In this framework, the enterprise pattern is developed at the individual, type, and group levels. Fig. 2.4 shows, for instance, how the enterprise pattern can be reproduced at a specific level of abstraction, in this case the level of individuals. Similarly, the abstraction pattern is developed for resources, events, and agents. Fig. 2.5 illustrates how the abstraction pattern can, for instance, be applied to the resource construct. The fully developed framework is included in Appendix A.

2.3 Positioning Batra's patterns in the framework

2.3.1 TransactionPatterns

Many patterns in Batra's collection suggest how to model transactions. The **generic transaction pattern** is meant for modeling transactions of services, which are intangible and hence difficult to individually identify, and mass-produced inexpensive tangibles, for which it is not economically efficient to register individual identities (e.g., bolts and nuts).

Fig. 2.6 shows how this pattern is positioned in our framework after classifying its elements along the two classification dimensions. The TRANSACTION class in the pattern maps to the concept of event as it is used to represent things like purchase orders, sales invoices, restaurant orders, deposits, etc. [Bat05], which are occurrences in time that involve transfer of ownership. Since this class has an attribute for identifying transactions (i.e., TRANSACTIONNO), the class represents individual transactions.

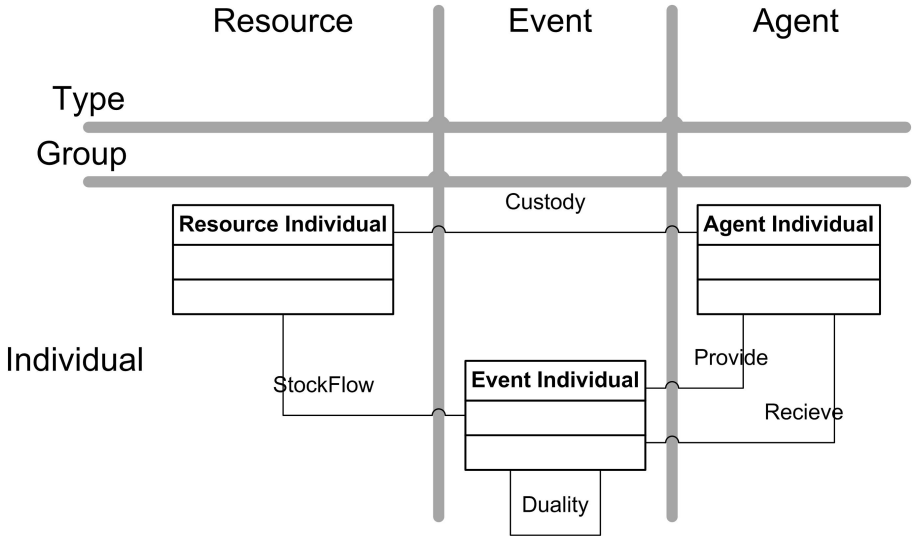

Figure 2.4: Structured enterprise pattern for individuals

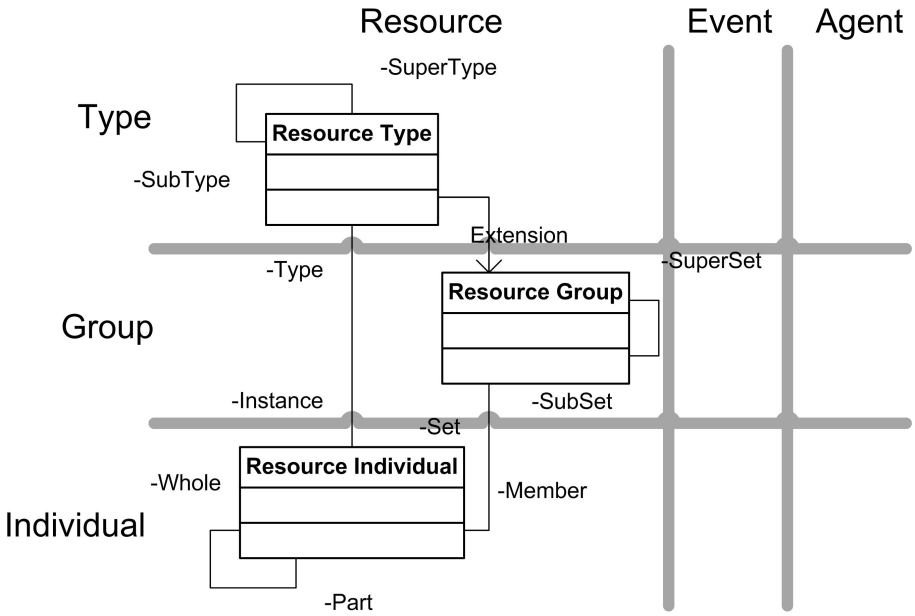

Figure 2.5: Structured abstraction pattern for resources

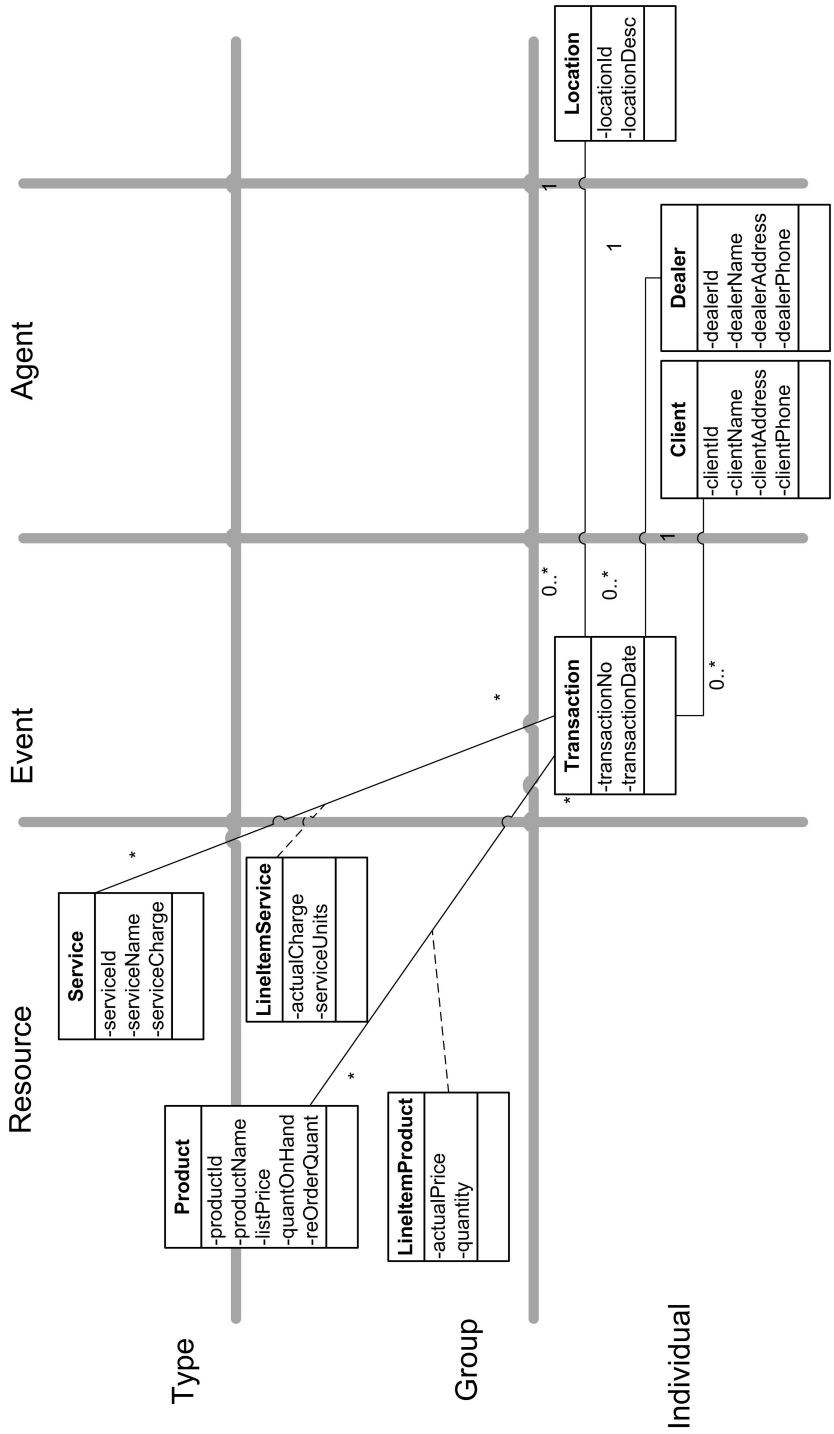

Figure 2.6: Generic transaction pattern

The transaction parties (i.e., client and dealer) embody customers, salespersons, waiters, telemarketers, etc, and can thus be classified as agents. The CLIENT and DEALER classes have an identity attribute (i.e., CLIENTID and DEALERID), which means that their instances represent individual agents. According to the enterprise pattern, the relationships between transaction individuals and client and dealer individuals are participation relationships, where clients and dealers assume opposite provider/recipient roles. For instance, in a goods delivery transaction, the client could be a customer of the enterprise, taking on the recipient role, and the dealer could be a distributor that is contracted by the enterprise, in the provider role. In the requiting payment transaction, the customer (i.e., client individual) assumes the provider role and an agent acting on behalf of the enterprise, e.g., a cashier (i.e., dealer individual), assumes the recipient role.

The remaining classes in the generic transaction pattern represent resources at the type or group abstraction levels. In Fig. 2.6, the SERVICE class represents a resource type for which implicit groups that are involved in a certain transaction are represented in the LINEITEMSERVICE association class. The classification as a resource type signifies that the objects that instantiate this class represent abstract images of services, but not the services themselves. The PRODUCT class, on the other hand, shows a merged resource type and group because some of its attributes refer to a product type (e.g., LISTPRICE), whereas others (e.g., QUANTONHAND) make only sense for a resource group (i.e., the physical stock of a product type). Hence, the PRODUCT class models a type of products together with its extension. As the pattern in Fig. 2.6 models generic transactions, there is also no explicit representation of individual products. This modeling practice, which implodes the abstraction pattern into a single class, is common for mass-produced inexpensive products, which receive no individual identity [GM06]. The individual products that are involved in a certain transaction are implicitly represented by grouping them in the LINEITEMPRODUCT association class. Each instance of this class represents a collection of individual products (of a same group/type) that are involved in a same transaction.

The LOCATION class represents individuals (given the LOCATIONID attribute) but cannot be classified according to the enterprise pattern as location is neither resource, event or agent; so with respect to this dimension falls outside the framework. There exist alternative interpretations of the REA ontology that recognize location as an additional ontological category [HDC99, ISO07].

The association between TRANSACTION and SERVICE/PRODUCT can be recognized as a variant of the stockflow relationship in which abstraction is made of individual resources. This abstraction from individuals is overcome by adding a resource group (i.e., LINEITEMSERVICE and LINEITEMPRODUCT) that collects the individuals through an attribute (i.e., SERVICEUNITS and QUANTITY).

In contrast to the generic transaction pattern, the **discrete product pattern**

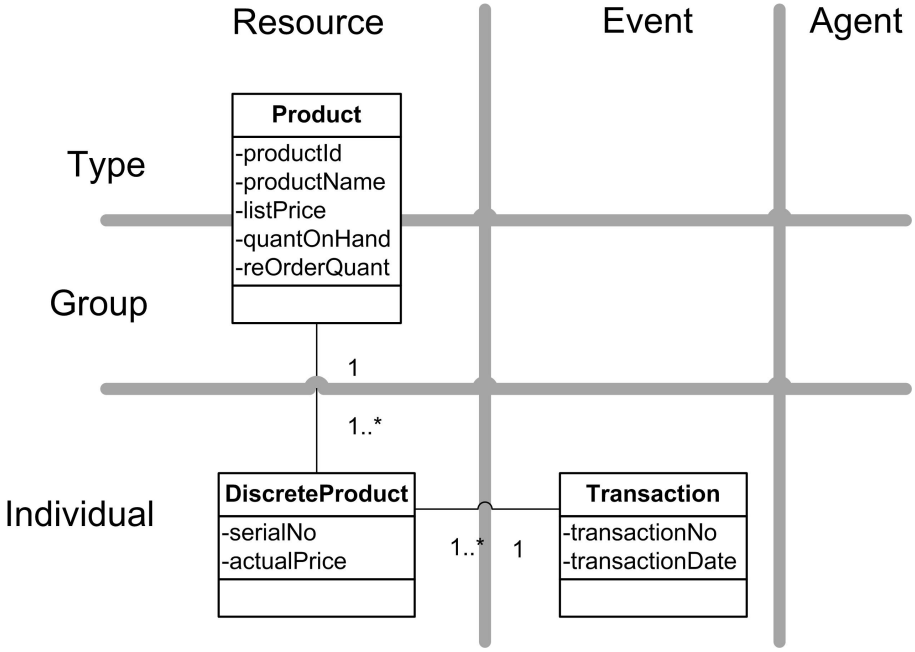

Figure 2.7: Discrete product pattern

(fig. 2.7) represents the individual resources that are involved in a transaction explicitly. The PRODUCT class shows again a merged type and group image. The DISCRETEPRODUCT class is meant for products like vehicles, airplanes, boats, computers and other products that acquire identity through a serial number. Hence, it represents resource individuals. The transaction is again an event individual. As the TRANSACTION and PRODUCT classes in fig. 2.7 copy the TRANSACTION and PRODUCT classes in fig. 2.6, it is likely that the scope of the discrete product pattern is limited to the modeling of stockflow relationships. Hence, to model transactions in which also participants and location are identified, it is necessary to extend the pattern with CLIENT, DEALER and LOCATION classes as in fig. 2.6.

The **time-based transaction pattern** (fig. 2.8) is a variation to the generic transaction pattern that has a different selection of attributes to indicate the time-based nature of some transactions (e.g., rentals). Its position in our structuring framework does not differ from the generic transaction pattern.

Batra also presents a variant of the time-based transaction pattern, i.e., the **registration as a time-based transaction** (fig. 2.9), which incorporates resource classes at all three levels in the abstraction pattern. The ENROLLMENT is the individual resource the STUDENT receives during REGISTRATION. The COURSE

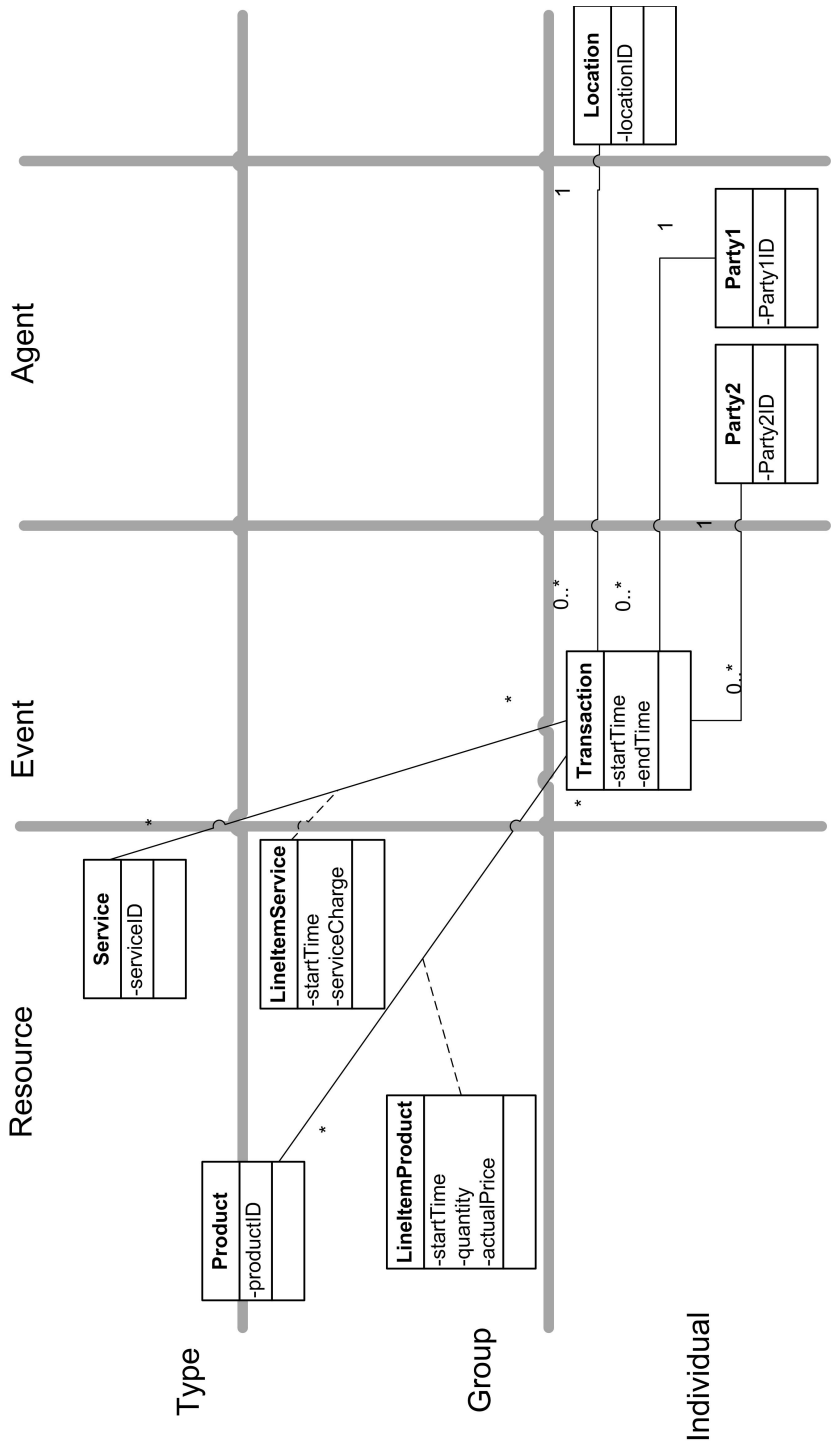

Figure 2.8: Time-based transaction pattern

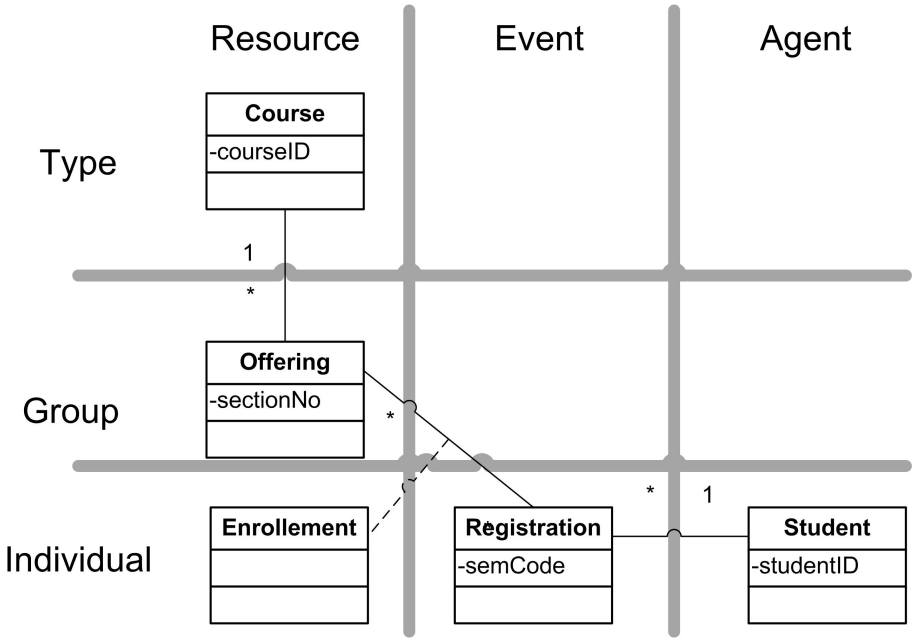

Figure 2.9: Registration as a time-based transaction

describes the type of resource that is offered to the students and the OFFERING groups all enrollments that are offered as a section of the course (implying some spatial-temporal positioning of the course, e.g., Monday morning spring semester 2010 in room 14). Compared to Fig. 2.8, it should be noticed that only the recipient participation (i.e., STUDENT-REGISTRATION relationship) is modeled. The omission of the provider participation is defendable if there is only one possible provider individual (e.g., if this pattern is used in the conceptual data model of some university), so this participation is implicit.

The **subsequent transaction pattern** (fig. 2.10) shows two subsequent transactions for which a fixed sequence has been imposed. In the example, an invoice is always preceded by an order as the invoice cannot exist without being related to an order. Although in the example, the relationship between the ORDER and INVOICE event individuals only implicitly models a future duality[1], other examples can be given (e.g., a payment in cash followed by a delivery of goods) in which the subsequent transaction pattern is used to represent the exchange of goods or services for

[1] In an accounting interpretation, the acceptance of an order commits the enterprise to deliver goods and commits the customer to pay. Even if there is no payment at the moment the invoice is sent, an accountant will book both a decrement (i.e., stock of goods is reduced) and an increment (i.e., balance of accounts receivable increases).

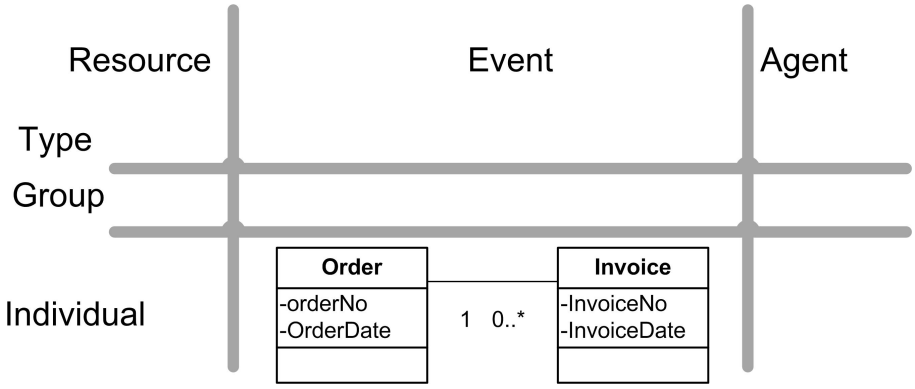

Figure 2.10: Subsequent transaction pattern

payment (or other goods and services as in barter trade) according to the economic reciprocity principle. As with the registration (fig. 2.8) and discrete product (fig. 2.7) patterns it should be mentioned that parts of the transaction pattern have been omitted (e.g., parties, location, product or service).

The **data warehouse pattern** (fig. 2.11) is a variation to the transaction pattern that summarizes transactions over a certain interval. The TRANSACTION class can here thus be classified as a representation of event groups, where each group collects all transactions between two identified agent individuals occurring in a certain time interval at a certain location and related to a certain type of product. The pattern abstracts from event individuals when representing participation relationships as it directly relates agent individuals to event groups. The TRANSACTION event group also defines an implicit resource group (i.e., the QUANTITY attribute refers to group of resource individuals whose type is described by the PRODUCT class), which results in a collapsed stockflow relation contained within the TRANSACTION class.

It is noteworthy that all transaction patterns from Batra's collection (Figs 2.6 to 2.11) abstract from agent groups (e.g., market segments) and agent types (e.g., youth, elderly), even if such concepts may prove to be useful when databases need to support marketing applications (e.g., querying transaction data related to a particular group of customers). However, using our framework (in particular the abstraction pattern), modelers can extend themselves the patterns to include agent groups and types.

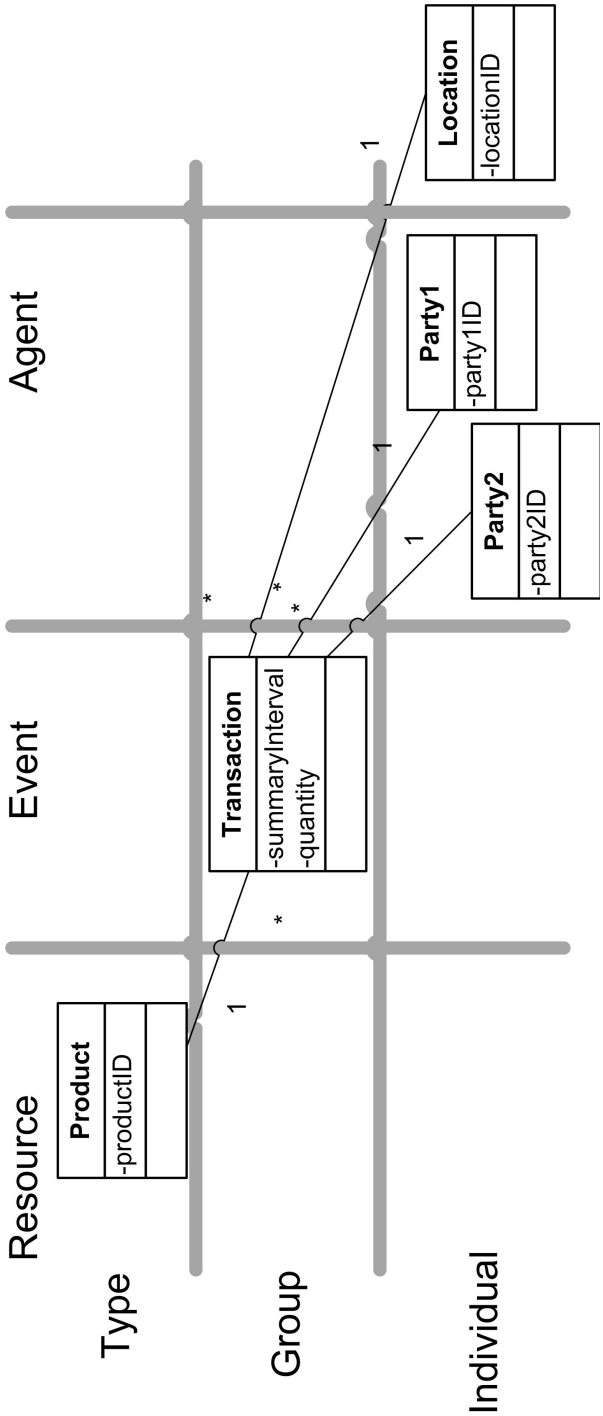

Figure 2.11: Data warehouse pattern

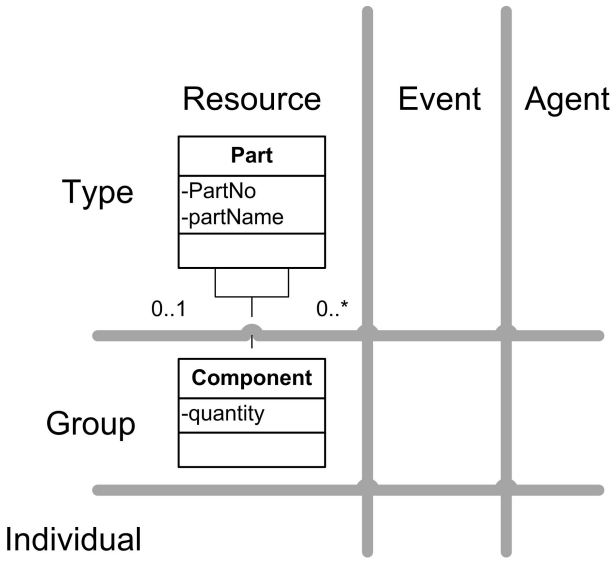

Figure 2.12: Bill of materials

2.3.2 Generic Patterns

The second part of Batra's pattern collection is designated to the use of typical abstraction mechanisms in conceptual data modeling, including various forms of part-whole relationships. Fig. 2.12 shows the **recursion pattern** as applied to a bill-of-materials that prescribes how parts are assembled using other parts. Although part-whole relationships hold between individuals, the bill-of-materials shown in Fig. 2.12 is specified at the type level (i.e., the class PART is classified as a resource type) as indicated by the presence of the quantity attribute in the COMPONENT association class (which can accordingly be classified as a resource group). As discussed before, the omission of resource individuals is common when modeling mass-produced inexpensive resources. The recursion pattern could also be applied to events and agents.

Batra's **strict hierarchy pattern** represents part-whole relationships at the level of individuals, which fully conforms to our abstraction pattern. This container-content relation (fig. 2.13) can be applied to hierarchies like corporation-company, company-division, division-department, warehouse-bin, and location-sublocation. Although in Fig. 2.13 container and content are classified as resources, they can also be events or agents. Batra's **plan pattern** is completely similar to the strict hierarchy pattern as it applies the container-content relation to a plan and the different steps that compose the plan.

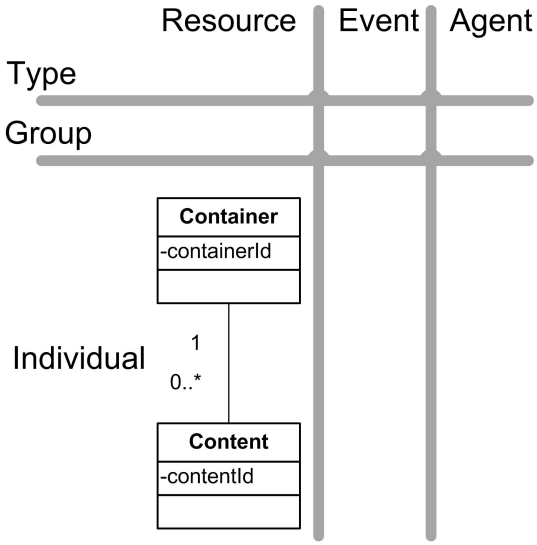

Figure 2.13: Container-content hierarchy

The **type pattern**, despite its name, is exemplified in [Bat05] using a membership relationship (fig. 2.14). The MEMBERSHIP group is defined by certain conditions (i.e., MEMBERSHIPTYPE, MEMBERTERMS and MEMBERPRICE) that apply to CLIENT individuals, which are identified and described by their characteristics (i.e., CLIENTID, CLIENNAME, CLIENTADRESS, CLIENTPHONE). Of course, this pattern can equally well be used to represent classification-instantiation relationships (e.g., Client and Client Type).

The **generalization pattern** creates a hierarchy of supertypes and subtypes. Batra illustrates this pattern using the conventional UML representation for generalization (i.e., the hollow triangle symbol) and applies it to identify different types of employees (fig. 2.15). In UML, specialization-generalization relationships are modelled between classes that represent individuals, although the semantics of these relationships are the same as the subtype-supertype relationships in our abstraction pattern (e.g., SECRETARY is a subtype of the type EMPLOYEE). This is indicated in Fig. 2.15 by the gray box in the agent-type cell of the framework.

The **entity-event pattern** is used for representing the history of entities. Fig. 2.16 shows that the participation of employees in events causes changes in the state of employees (i.e., changes in attribute values). The EMPJOBHISTORY and EMPSALHISTORY represent events, as they are defined by the REA ontology. If needed, the pattern can be further extended with classes that represent the resources (e.g., money for EMPSALHISTORY, labor for EMPJOBHISTORY) and

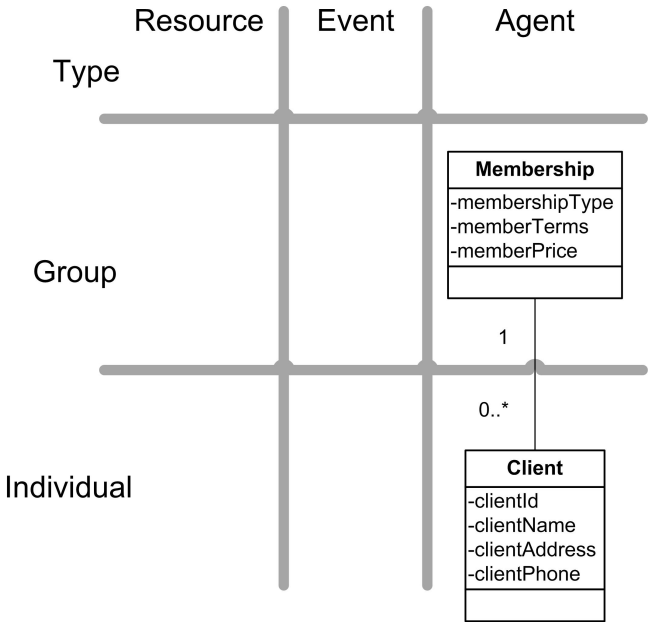

Figure 2.14: Type pattern for power-typing

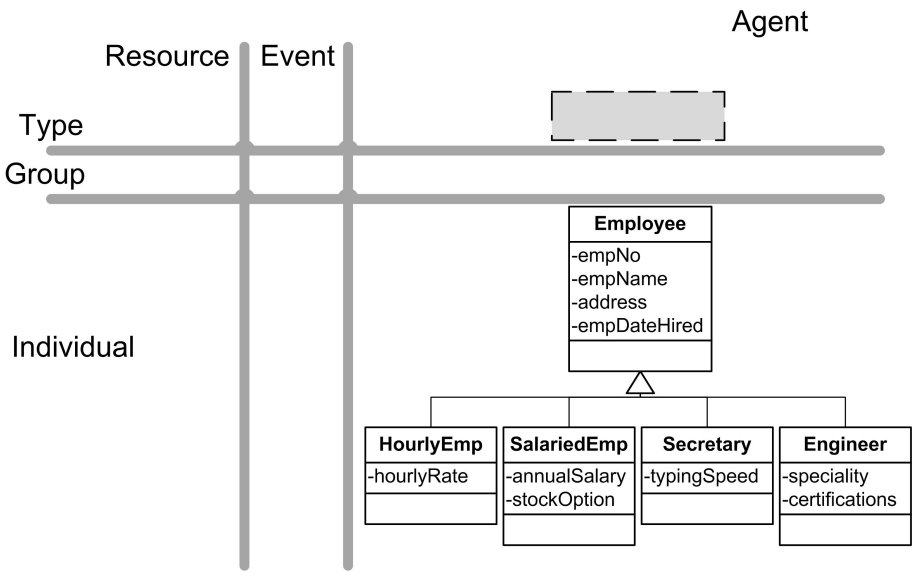

Figure 2.15: Generalization pattern

trading partners (i.e., provider, recipient) involved. The EmployeeEvent class represents non-economic events (e.g., a change of address) and thus falls outside the scope of the enterprise pattern.

Finally, Batra's collection also contains a variation to the recursion pattern (i.e., the **recursive one-many relationship** (fig. 2.17)) that shows a relationship which is not captured by the abstraction pattern, as it neither represents a subtype-supertype relationship nor an abstract part-whole relationship (as with the recursion pattern applied to a bill-of-materials). The pattern in Fig. 2.17 builds a strict hierarchy on types, with other semantics than generalization/specialization semantics (e.g., one position reports to another position), which is something that falls outside the scope of our abstraction pattern, neither is it covered by the enterprise pattern (so could be a possible extension if there is a need, for instance, for designing databases that support organizational design and management applications).

2.4 Example application scenario: extending the Dataware-house Pattern

The positioning of conceptual data modeling patterns in our ontology-based framework clarifies their scope and helps modelers decide whether a candidate pattern

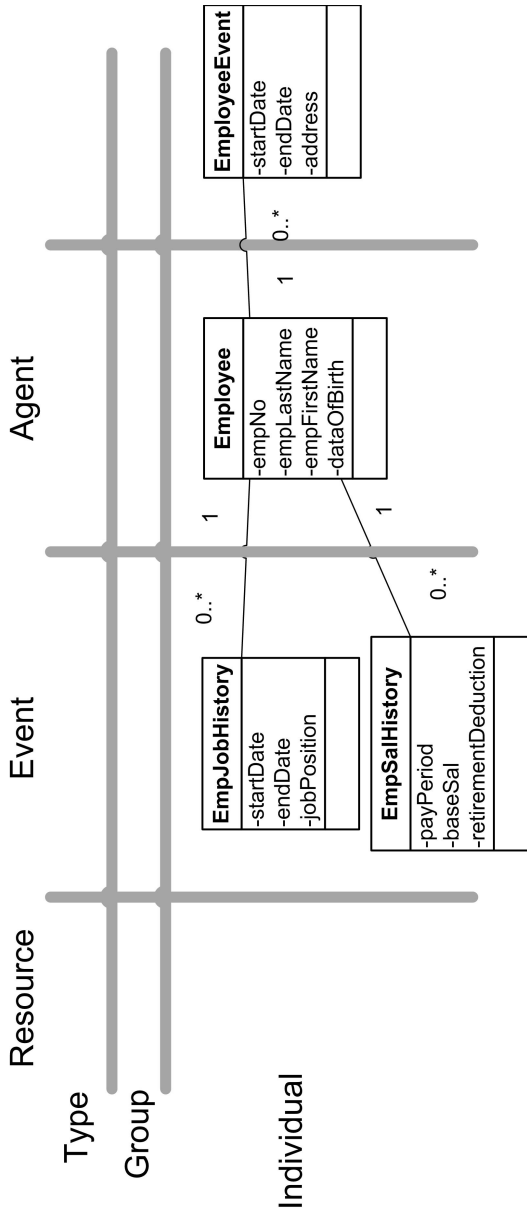

Figure 2.16: Entity-event pattern

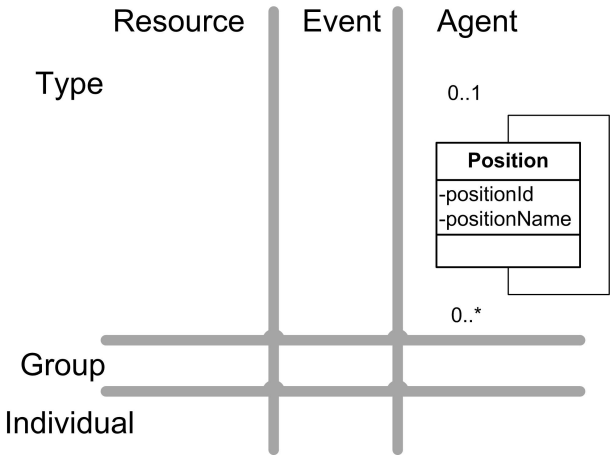

Figure 2.17: Recursive one-many relationship

offers a solution to a modeling problem. In this section, we illustrate how modelers can use the framework to evaluate the relevancy and completeness of a pattern with respect to the problem at hand. If the scope of a pattern extends to cells in the framework that are not deemed relevant for the model, then these pattern elements should not be used. On the other hand, if the scope of a pattern does not cover all cells that are deemed relevant, then the pattern should be extended. This can be done by combining it with other patterns that do cover these cells or, if no such patterns are available, by adding the relevant elements of our enterprise and abstraction patterns.

The illustration assumes an application scenario in which a conceptual data model is needed for the conceptual design of a data warehouse that stores the transactional data (e.g., sales data, purchase data) of some enterprise. This scenario is chosen to match the patterns offered in [Bat05]. We keep the premise of Batra's data warehouse pattern (fig. 2.11) that the number of transaction parties is relatively small, that each of these parties is frequently involved in transactions (i.e., once-only transactions are exceptional), and that transactions between two parties occur in a limited number of locations (possibly only one). Therefore, there is no need to classify parties as a type (e.g., business, consumer) or assign them to a group (e.g., customer groupings per country or ZIP-code), and transactions can be aggregated per recipient-provider-location triple (as in Batra's pattern). Considering that data storage and computation power are cheap, an explicit requirement for the data warehouse is that it supports drill-down retrieval functions that can trace aggregated data back to individual transactions. Furthermore, transactions only

involve products (no services), possibly of different product types.

Batra's data warehouse pattern (fig. 2.11) is an obvious candidate pattern to be used as it is a transaction pattern, and moreover it is the only transaction pattern in Batra's selection that represents transactions at an aggregate level, i.e., we classified the TRANSACTION class in the pattern as an event group. However, using this pattern as the sole foundation of the conceptual data model would result in an incomplete model as not all data warehouse requirements are met. In particular, the pattern does not contain an element in the event-individual cell of our framework, meaning that it represents transactions only at the aggregate level, whereas the required model also needs to represent individual transactions. Further, as shown by our analysis in the previous section, the TRANSACTION class incorporates an implicit resource group (through the QUANTITY attribute) which should be made explicit as in our application scenario transactions may involve products of different types (whereas the implicit resource group in Batra's pattern contains resources of the same type).

Fig. 2.18 shows how the scope of the data warehouse pattern can be extended to accommodate the data warehouse requirements of the application scenario. The TRANSACTION class is decomposed into an event group (i.e., the TRANSACTIONPERIOD class) that represents the time interval for which the transactions are summarized and a resource group (i.e., DATAWHTRANSACTION class) that aggregates the quantity of resources traded during a certain time interval, per provider, recipient, location and product type. Note that for each TRANSACTIONPERIOD object there can be many DATAWHTRANSACTION objects, each representing a group of products belonging to some product type.

To fill the empty event-individual cell, we merged the data warehouse pattern with the generic transaction pattern (fig. 2.6). The CLIENT, DEALER and LOCATION classes of the generic transaction pattern match the PARTY and LOCATION classes of the data warehouse pattern and hence provide integration points for both patterns. Since the CLIENT and DEALER classes in fig. 6 both represent transactions parties they were merged in the PARTY class, for which a PROVIDER (e.g., DEALER) and a RECIPIENT (e.g., CLIENT) role were modeled. As the PROVIDER and RECIPIENT class represent agent types, a grey box in the upper-right corner indicates that fig. 18 also contains agent types although no classes are shown in this part of the grid. The classes in the generic transaction pattern that represent services (at the group and type levels) are irrelevant, so were not included. The introduction of the TRANSACTION class of event individuals into the data warehouse pattern shows the grouping of transactional data explicitly. According to the abstraction pattern of our framework, the semantics of the relationship between TRANSACTION and TRANSACTIONPERIOD is membership. Finally, the PRODUCT class of the generic transaction pattern, which collapses a resource type and group, is matched to the PRODUCT resource type class of the data warehouse pat-

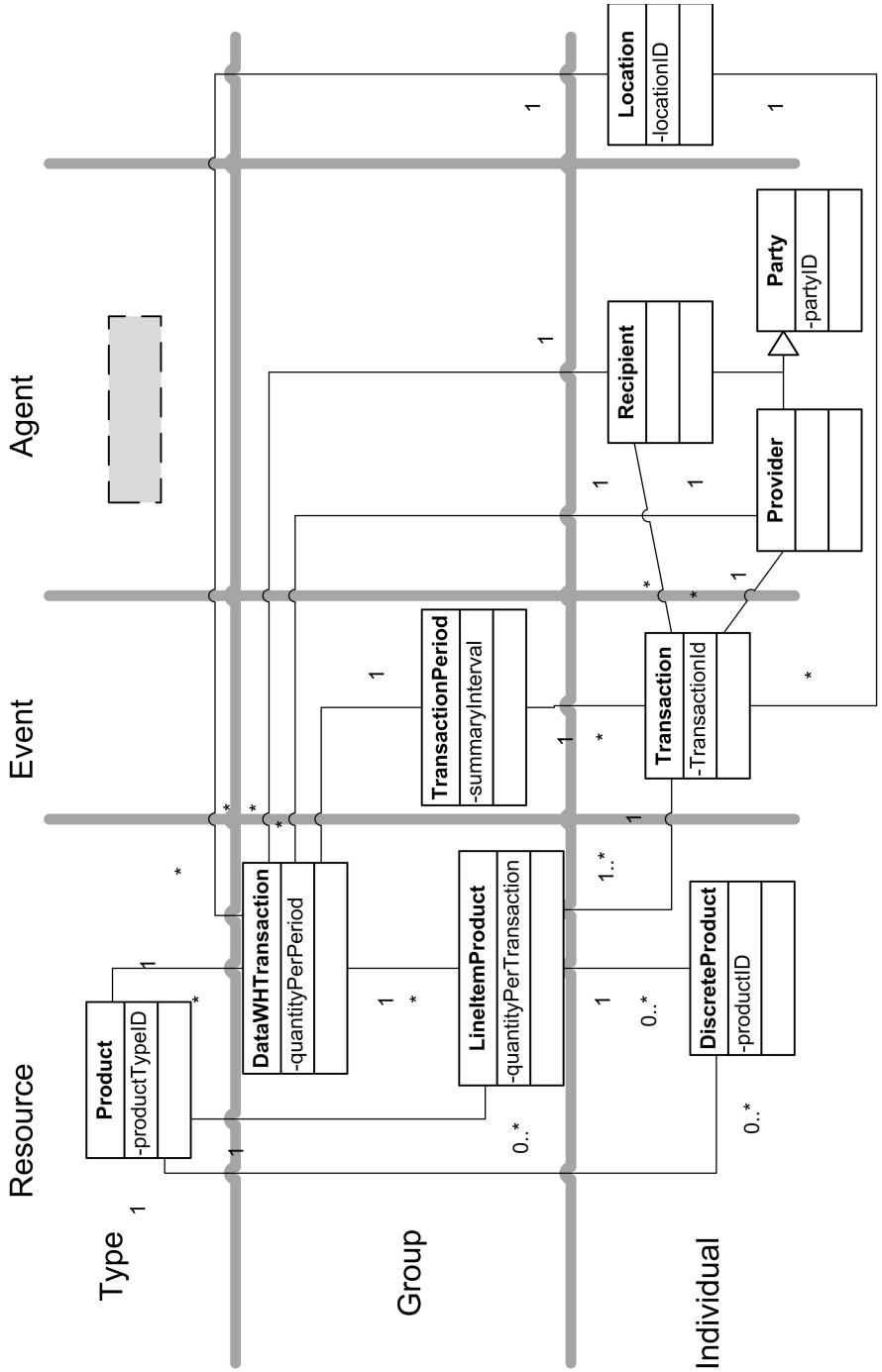

Figure 2.18: Extended data warehouse pattern

tern. The QUANTONHAND and REORDERQUANT attributes can be omitted as they are not relevant for documenting transaction history (so the PRODUCT class now falls entirely within the resource-type cell). The PRODUCT class is now indirectly related to the TRANSACTION event individual class via the LINEITEM-PRODUCT resource group class of the generic transaction pattern. The objects of this class represent groups of products that are subsets of the groups of products represented by the DATAWHTRANSACTION objects, so, conform to our abstraction pattern, the semantics of the relationship between both classes is subset-superset.

To complete the illustration, we also integrated the discrete product pattern (fig. 2.7) into the extended data warehouse pattern. The identification of the discrete products involved in a transaction (which may be another data warehouse requirement in our application scenario) is enabled by adding the DISCRETEPRODUCT class, which conserves the PRODUCT-DISCRETEPRODUCT classification-instantiation relationship from the discrete product pattern.

As shown in this example, the enterprise and abstraction patterns presented in this paper helped to identify integration points between different patterns in Batra's list (i.e., generic transaction pattern, discrete product pattern) that were used to extend and vary the original data warehouse pattern such that its scope better matches the required scope of the data warehouse conceptual data model. The changes suggested make the pattern more general (i.e., not limited to transactions dealing with single products, not limited to mass-produced inexpensive products) and allow representing transactional data at different levels of aggregation. The extended data warehouse pattern includes all kinds of enterprise concepts from the enterprise pattern and elaborates some of these concepts at different (e.g., transactions) or even all (e.g., products) levels of abstraction as given by the abstraction pattern.

2.5 Conclusion and Future Research

This paper argues that the scope of conceptual data modeling patterns can be identified, clarified, and explicitly represented by positioning them into an ontology-based framework. A clear and explicit definition of scope could help novice modelers in better deciding which patterns match which parts of the domain to be represented.

We demonstrated our argument by positioning the conceptual data modeling patterns selected by Batra [Bat05] into a two-dimensional structuring framework that was constructed using two ontology-derived 'benchmark'patterns: an enterprise pattern for representing transactions (derived from the REA domain ontology) and an abstraction pattern for representing reality at different levels of abstraction (derived from the UFO foundational ontology). We showed that Batra's

patterns can be regarded as variations on the combination of these two patterns. Based on the names of the classes and attributes in Batra's patterns, and their intended meaning (as described in (Batra, 2005)), we classified pattern elements according to the enterprise and abstraction orthogonal dimensions of our framework. Doing so, the part of the domain that is covered by the patterns (i.e., their scope) is given by the cells of the framework that are filled with pattern elements. Through this analysis, we also identified implicit modeling choices that are often made with respect to the representation of reality at different levels of abstraction. We further showed that the positioning of the patterns into our framework helps clarifying the semantics of the relationships between pattern elements.

Finally, we illustrated, by means of an application scenario dealing with the conceptual design of a transactional enterprise data warehouse, how modelers can use the framework to evaluate the relevancy and completeness of candidate patterns with respect to the problem at hand. In the example, the scope of Batra's data warehouse pattern was extended to make it both more general with respect to its application domain and more detailed with respect to different levels of aggregation in representing transactional data. The example further illustrated that the positioning of patterns in our framework helps identifying integration points between patterns that facilitate their joint use when constructing models.

For pattern developers, the framework allows interpreting and analyzing patterns created by others and thus stimulates the (partial) reuse of existing solutions, which we demonstrated when developing the extended data warehouse pattern. For modelers, we believe that the positioning of conceptual data modeling patterns in an ontology-based framework makes it easier to correctly interpret the patterns that the modelers have access to, to evaluate when and how a pattern should be used, and to jointly use patterns from the same or different catalogues.

As conceptual data modeling patterns are especially meant to support the work of novice modelers, but models produced by novices often suffer from problems related to irrelevancy and incompleteness, which we attributed to a lack of explicit scoping of patterns, our future research will focus on empirically validating the claim made in this paper. Specifically we will test whether model quality improves when novice modelers have access to Batra's patterns as they are positioned in our REA/UFO-based framework versus access to Batra's patterns without being positioned into our framework.

Our future research will also address some limitations we discovered when building and using our framework for structuring Batra's patterns. An obvious limitation of the framework is the scope of the enterprise pattern, which is limited to the interactions between natural and/or legal persons and the goods and services they exchange and produce in an economic context. To position other conceptual data modeling patterns for the enterprise domain in this framework, this scope might need to be extended to accommodate concepts like commitments, contracts,

plans, locations, business policies, organizational structures, etc.

Also, as only pattern scope is addressed, the framework does not take into account the use context of conceptual data modeling patterns. The framework, as it stands, does not support modelers in dealing with use context variations, which involves defining/customizing attributes, specifying participation and cardinality constraints, and simplifying models (i.e., modeling compromises driven by explicit choices). In the future we would like to contribute to this research by investigating the forces that drive pattern variability starting from, but not limited to, the enterprise pattern and abstraction pattern. Questions addressed by this research are, for instance, when do modelers use certain patterns and when do they modify these patterns by omitting, adding, specializing, generalizing, collapsing, etc. pattern elements? These insights would provide a basis for the further development and extension of our structuring framework.

2
Summary

Objectives

Identification, clarification and representation of conceptual date model scope, to help novice modelers, of which the models often suffer irrelevancy and incompleteness, in better deciding which patterns match which parts of the domain to be represented.

Approach

Positioning conceptual data modeling patterns in a two-dimensional structuring framework that is constructed using two ontology-derived benchmark patterns.

Findings

Batra's pattern catalogue can be positioned in a two-dimensional structuring framework that is derived from the REA domain ontology and the UFO foundational ontology, and can be regarded as variations on the combination of an REA and a UFO derived pattern. Furthermore, the structuring framework helps clarifying the semantics of relationships between pattern elements.

Contribution

This paper argues that the scope of conceptual data modeling patterns can be identified, clarified and explicitly represented by positioning them into an ontology-based framework.

Practical Implications

Modelers can evaluate the relevancy and completeness of candidate patterns with respect to the problem at hand, which is expected to facilitate the (partial) reuse of existing solution models.

3

A Resource-Event-Agent Reference Information Model for Representing Value Chains and Systems

Abstract

In business modeling the focus is shifting from the enterprises value chain to the value system (usually a supply chain) in which it is embedded. Contemporary business modeling approaches should allow each enterprise taking part in a value system to develop its own information system and at the same time support the creation of system interoperability and information sharing amongst business partners in the value system. This paper presents a general way of modeling business phenomena that is observer-independent. That is, rather than presenting a value chain model from the perspective of a company taking part in business, it presents a value chain model that will be equally valid as viewed from any partner in a value system (e.g. supplier, customer) or from a completely neutral third party. This observer-independent model, which is given strength by grounding it in the mature Resource-Event-Agent model, is shown to represent information of different supply chain partners such that it can be integrated across enterprise boundaries.

Keywords

Business ontology, business modeling, system integration, value chain, value system, value activity, supply chain, business process

3.1 Introduction

The concept of 'business model'has many different definitions and applications, determined by the context in which it is used [OPT05]. In information systems development, business modelling is usually considered as a stage in the (early) requirements engineering process for business information systems during which the application domain to be supported by the system is modelled [Gor02]. Together with the functional system requirements, the business model forms the conceptual schema of the system, which can be extended with non-functional requirements to build the complete system specification [Oli07].

In the past, business modelling focused on the enterprise's value chain [PM85] as the core structure in business that needed to be supported by information systems. These value chains consist of value activities [PM85] (e.g. logistics, operations, marketing and sales, procurement, human resources management) that are supposed to create value for the customer of which a part should return to the value creating company as a remuneration for its value creating activities. The value chain of the company then is embedded in a value system [PM85], which includes the value chains of suppliers and customers. The focus of business modeling on the enterprise's value chain resulted in the construction of enterprise models documenting the enterprise's concept or information structures, processes, activities and roles, thus providing a conceptual basis for the development of enterprise-wide management information systems, transaction-processing systems and accounting information systems. However, a continuously faster globalizing world economy and increasing cooperation among supply chain partners increases the need to model the entire value system(e.g. supply chain) and not just individual players within it [Ber08, Gia99, HS06b, SL02].

This shift from the individual players to the cooperation among them as the focus of business modeling reveals the inadequacy [Bus03, Dou07] of many academic and practitioner-defined conceptual models, which were designed to serve specific purposes in one enterprise at a time and mainly provide application-specific partial views of the business domain. Therefore, it is not surprising that their limited level of abstraction and application-specific semantics hamper the modeling of organizational units at a more aggregate level than the enterprise. The business modeling methods that do support the modeling of value systems for the purpose of information systems engineering abstract from the internal business of the involved partners, whereas conceptualizations that do address entire value systems (i.e. both inter-enterprise value chains transactions and intra-enterprise processes) do not intend to support the design of the accompanying information technology. Consequently, these conceptualizations cannot be considered business modeling methods as defined in the first paragraph. For example SCOR [Cou06] approaches the design of the value creating processes that construct a value system from an op-

erational perspective and provides a vocabulary for addressing operational issues in designing these processes, explicitly abstracting from the supporting information technology. BMO [Ost04], on the other hand, looks at these same value creating processes from a strategic perspective and provides an ontology for addressing strategic issues in process design. The e3-value ontology [Gor02], which explicitly abstracts from operational issues, then provides the tools to represent such strategic business models and predict their profitability, based on the economic reciprocity principle, as early requirements elicitation for e-commerce information systems engineering. The Business Motivation Model [OMG08], which also addresses strategic issues, looks at businesses at an even earlier stage of development (i.e. business plans). At this early stage, the architecture of the supporting information technology is irrelevant. The DEMO methodology, which complements Dietz'enterprise ontology [Die06], then looks at the control flow aspects of the transactions that construct value creating processes. Like REA, the DEMO approach includes dealing with the information such value creating processes generate. However, DEMO does not incorporate the economic rationale as a core concept (as REA does). As opposed to Dietz'enterprise ontology, the purpose of Ushold's enterprise ontology [UKMZ98] solely included enhancing communication between trading partners by providing a shared business vocabulary. Consequently, the latter enterprise ontology is more intended to be used as an artifact in an information systems and not as a support for developing information systems.

What is needed is a business modeling approach that allows each enterprise taking part in a value system to develop its own value chain information system and at the same time supports the creation of system interoperability and information sharing amongst supply chain partners. There are four prerequisites for creating system interoperability [LB09b, 20505]: (i) agreement on a standardized reference model, (ii) standardized service interface models, (iii) a standardized set of domain-specific concept models (e.g. archetypes and templates) and (iv) standardized terminologies associated with controlled vocabularies. In its current state, the business information systems engineering field lacks most of these prerequisites [Gia99, RJB+08], with the exception of standardized service interface models (e.g. web service standards [W3C], electronic data interchange standards [Eur]). First, the field does not have a generally accepted reference model that spans multiple business information system application domains, although reference models exist for broadly defined application domains (e.g. enterprise architecture [Dav08, Zac87], workflow modelling [vdAvH02]) as well as more narrow domains (e.g. product tracking and tracing [JVvDB03], e-contracting [AG08]). These single application domain reference models may enable system integration within those domains, but not across. Second, these reference models are hardly ever complemented with concept models (e.g. modeling patterns or language idioms) [GCG+99]. Third, standardized terminologies for the business domain

have been suggested in different kinds of ontologies, like thesauri, taxonomies, partonomies, semantic models, etc. [LM01], but none of them has excelled in uniting different purpose-specific sub-domains of business (e.g. accounting, business process management, enterprise governance) or has been shared extensively.

This paper presents a first step in developing a business modeling approach that can be used for both value chain and value system modeling and that provides a basis for both information systems development and integration. This first step consists of a reference information model for the business domain that overarches different application domains of business information systems. In particular, the model can be used as a reference for modeling both intra-enterprise phenomena (e.g. business processes[1]) and inter-enterprise phenomena (e.g. collaboration spaces[2]) as they appear in value systems. The paper also proposes a standardized set of business-specific concept models that instantiate the reference model. These concept models are presented as object models showing archetypal occurrences of recurring concept or information structures in business (i.e. business patterns). In future research these archetypal object models can be generalized into modeling templates which can be used as building blocks for creating business models.

The proposed reference information model was derived from the Resource-Event-Agent (REA) ontology [GM02], which we use as a controlled business vocabulary. REA is a business ontology developed from an accounting information model [McC79]. Accounting is a discipline that has been recording business transactions for more than 500 years and consequently represents a vast amount of practical experience in the domain. The accounting origins do not imply that REA is an accounting-only ontology as it has been applied as a conceptual basis for supply chain [HM00] and e-collaboration modeling , the modeling of production processes [Hru06], enterprise information systems [BS08, DCH05] and management information systems [CS08]. Moreover, previous research has shown that REA can support the integration of business processes across enterprise boundaries [GLP08] and can be used to support the evolution of database schemas [CMO95]. Because of this potential for business process integration and the already existing variety of business modeling applications, both intra-enterprise and inter-enterprise, REA was chosen as the ontological foundation for the reference information model that is proposed.

That REA is the foundation of the reference information model does not mean that it is the reference model. REA makes an explicit distinction between the trading-partner view and the independent view on business assets and transactions. The former view (as in [GM02])) specifies a business conceptualization from the

[1] *"A business process is a set of activities that takes one or more kinds of input and creates an output that is of greater value to the customer."* [McC03]

[2] Collaboration spaces, which contain the interfaces between the value chains of supply chain partners in a value system, show exchanges of valued resources independent of the perspective of any business partner. [ISO07]

sole perspective of one particular party involved in business, called the 'trading partner'(e.g. an enterprise doing business in its role of customer, producer or supplier). The latter view, put forward in the ISO standardized version of REA known as the Open-edi Business Transaction Ontology (OeBTO) , looks at business from an independent observer perspective or 'helicopter'view (e.g. business seen as flows of goods, services and money between parties that are caused by business events initiated by these parties). It is clear that both views, respectively focused on intra-enterprise modeling and inter-enterprise modeling, must be integrated if the reference model is going to serve both value chain and value system modeling purposes.

Section 3.2 reviews the REA ontology, discusses its concepts and axioms and motivates its use as a controlled vocabulary. Section 3.3 presents the REA reference information model and shows the primitives that are needed to achieve the integration between the independent view and the trading-partner view. Subsequently, sections 3.4 and 3.5 present the archetypal object models that construct a standardized set of domain-specific concept models. Section 3.4 shows archetypal models for operational purposes (e.g. exchange or conversion processes), while section 3.5 explains how the reference model addresses the planning of transactions, as specified in contracts, and the planning of production processes, as detailed in production schedules. In addition, section 3.4 shows how these archetypal models can be used for constructing and deconstructing value system models from/into value chain models, switching between the independent and trading-partner views. Next, section 3.6 compares the reference information model to related business conceptualizations Finally, section 3.7 concludes the paper and proposes ideas for future research.

3.2 Introduction to the REA ontology

The original REA generalized accounting framework [McC82] was developed to create an environment in which accountants and non-accountants can share data about a same set of business phenomena. Based on ideas taken from Chen's Entity-Relationship model [Che76], an accounting information model was proposed in which concepts were given real-world business semantics (i.e. resources, events, agents) instead of the usual debit-credit-account semantics (e.g. accounts receivable, revenues deferred) which code operational information such that it is hard to decode for most non-accountants. This REA semantic accounting model supported procedural mechanisms for taking different mutually compatible views on the same business reality. For instance, an REA data model would still contain all data required to restore the accounting view on business (e.g. calculate accounts receivable, revenues deferred, etc.) [DCH05, HDC99], but would at the same time also support other kinds of operational and managerial business applications (e.g.

stock control, policy setting, planning, management control, etc.). This grounding in real-world business semantics makes REA a prime candidate for creating a canonical conceptualization of the business domain that is useful for business modeling.

The REA ontology contains three core primitives (i.e. resource, event and agent) and one driving principle (i.e. duality) from which domain axioms are derived. **Economic Resources** (e.g. goods and services[3]) represent objects that are scarce, have utility and are under the control of an economic agent (e.g. enterprise, household) [Iji75, McC82]. The scarceness means that not every economic agent can control such resources at a certain point in time and indicates that for some economic agents trade is required to gain control over particular resources. The utility motivates why certain economic agents want to gain control over particular resources. **Economic Events** (e.g. produce, exchange, consume, distribute) result in changes (i.e. increases and decreases) of resource stocks [Yu76], whereas **Economic Agents** represent legal or natural persons that participate in economic events (e.g. performing a task, enacting a process) or have custody over resources (i.e. having physical control over resource or controlling the access to resources) [ISO07]. **Duality** then balances changes in resources due to economic activity [Iji75], which relates back to REA's accounting background. For instance, duality in market transactions dictates that when a company sells products to a customer (i.e. an economic event that decreases the value of the company's inventory of products), a requiting event like a payment or delivery of equally valued goods (e.g. as in barter trade) by the customer must follow, meaning that there is a dual economic event that balances the decrease in value caused by the sale.

REA also explicitly recognizes **Economic Commitments**, which are promises to perform economic events in the future as specified by a schedule or contract. However, commitment is not a core primitive as the creation of commitments requires participating agents and the input of other resources (e.g. labour), and the result (i.e. the promise to perform an event in the future) has value on its own, which indicates that there is a shared conceptual ground (i.e. value) for resources and commitments. For instance, accepting a sales order commits the company to a future sale, which is valuable because it also commits the customer to perform a requiting event like a payment. As another example, issuing a production schedule commits the company to produce goods, which can later be sold to generate revenues. In general, planning, which is the core purpose of the commitment construct, adds value to the company [HHKV08].

Next to the primitives that build the controlled vocabulary, the REA ontology is also complemented with domain-specific axioms that convey fundamental truths that always are observed to be valid and for which there are no counterexamples

[3]The categorization of services as resources reveals that REA takes the goods-dominant economic worldview, which sees services as intangible goods. [VL04]

or exceptions [Bah99]. This axiomatisation describes the set of models intended by the ontology [Gua98]. The axioms can be considered as constraints that have to be satisfied by models in order to be deemed valid for describing the phenomena within the domain of interest.

The REA axiomatization consists of three axioms:

The **first REA axiom** specifies that *"at least one inflow event and one outflow event exist for each economic resource; conversely inflow and outflow events must affect identifiable resources."* [GM04] Inflow (outflow) events are events that change resource stocks such that their value increases (decreases) for the economic agents that control them (e.g. when a car is sold, the buyer gains control over an additional car and the seller loses control over a car, while the seller gains control over additional money and the buyer loses control over a certain amount of money).

The **second REA axiom** requires that *"all events effecting an outflow must be eventually paired in duality relationships with events effecting an inflow and vice-versa."* [GM04]

The **third REA axiom** stipulates that *"each exchange needs an instance of both the inside and outside subset."* [GM04] In market transactions (i.e. exchanges) both parties (e.g. customer and vendor) must be present as economic agents.

Note that the application of the third axiom is restricted to exchange processes (e.g. sales, acquisition), i.e. a sub-domain within the business domain that excludes internal conversion processes (e.g. production). The first and second REA axioms hold for both exchange and conversion processes. In a production environment, the second axiom implies that producing goods requires the consumption of other resources (e.g. raw materials, energy, labour). The first axiom implies that every resource that is currently under the control of an economic agent must have been acquired before (e.g. bought, produced) and is meant to be used later (e.g. sold, consumed). The axiom also implies that something that occurs in business reality is only an economic event if at least one economic resource is identified that is affected by the event, meaning that the economic agent gains or loses control over (part of) a resource (stock) (e.g. during a production run a stock of inputs is decreased and a stock of outputs increased; when a machine is used for producing goods the owner of the machine temporary loses the ability to use it for the production of other goods; when a car is hired (out) the hirer temporary acquires the right to drive the car and the owner loses the right).

3.3 The REA reference information model

This section presents the reference information model that is based on the REA ontology. A reference information model for domain modeling prescribes a generic structure for representing a true and fair view of that domain. If an ontological

theory for the domain exists (i.e. a domain ontology [GG95]), then committing the reference information model to the conceptualization specified by the domain ontology ensures that the reference information model captures relevant domain knowledge. This knowledge includes conditions that specify the configurations in the domain that are possible and those that are not [EW05]. Since the complete reference information model, which can be found in appendix B, is relatively large, it is decomposed in an informative and an economic rationale component. The economic rationale component models the economic logic to which representations of exchanges and conversions need to comply. The informative model attributes additional information to these economic logic models.

The informative component in fig. 3.1 discriminates the classes ORGANIZATIONAL UNIT, ECONOMIC RESOURCE, ECONOMIC EVENT, ECONOMIC AGENT and ECONOMIC COMMITMENT of which the instances represent the economic resources, events, agents and commitments introduced in section 3.2. The organizational unit is a new construct that is used to represent that certain economic agents (i.e. organizational units) have control over economic resources (i.e. ownership of the right to derive economic benefit from a resource), which entails the discretionary power to use or dispose of these resources via economic events in a legal way, where other (ordinary) economic agents can only have physical access (i.e. custody) to economic resources. Organizational units represent the entities that experience the effect of economic events, whereas agents represent the entities that engage in events (e.g. an employee performs an event that affects his employers resources). So agents may have access to resources of which they are not the owner (i.e. having custody but not economic control over the resources), which means that in that case the agents act on behalf of organizational units. For example, an employee is an agent for its employer (i.e. the employee performs tasks from which the employer reaps the full benefits).

The associations between these classes then represent how they relate to each other. The RESPONSIBILITY association on the organizational unit class represents that one organizational unit can be responsible for the actions of another organizational unit (e.g., en enterprise for a department, a parent company for a subsidiary). The CONTROL association between an organizational unit and an economic resource indicates that an organizational unit either owns the economic resource or is entitled to derive economic benefit from it [ISO07]. The MEMBERSHIP association specifies that an economic agent represents the role of a person acting on behalf of the organizational unit of which it is a member. The PARTICIPATE association then indicates that this economic agent engages in economic events for which it is accountable on behalf of this organizational unit. Furthermore, the USE association shows that economic events can use economic resources without affecting their value for subsequent events (e.g. tools), next to the economic resource stocks that are irreversibly affected by those economic events (e.g.

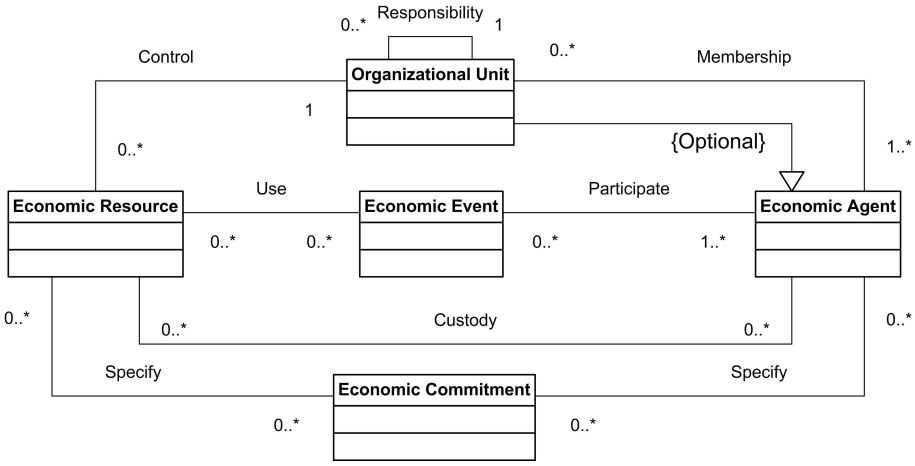

Figure 3.1: Informative component of the reference information model

inputs and outputs). The CUSTODY association then represents that an economic agent has physical control over or controls access to economic resources. [ISO07] Finally, both SPECIFY associations show that economic commitments can specify the economic resources and the economic agents for the economic events that have been committed.

Fig. 3.2 shows the economic rationale component of the REA reference information model for business modeling. The reference information model differs from the REA ontology as presented in [GM02] because it models the central role of organizational units as viewpoint determining entities in business semantics. Therefore, fig. 3.2 explicitly models the 'quid pro quo'(i.e. something in return) principle as determined by the view of a single organizational unit, which judges whether the increments and decrements it experiences are well-balanced. Consequently the reference information model has been attributed with increment and decrement classes that show how organizational units respectively gain or lose control (e.g. ownership) over resources and how the resource stocks they control respectively increase or decrease in value.

The reference model also specifies that organizational units can see events as increments, decrements or both. Increment events symbolize resource inflows, while decrement events represent resource outflows. Although events may be registered as solely increment or decrement events in the information system of one organizational unit (i.e. taking a trading-partner view), events are essentially increments and decrements at the same time signifying that no resources are allowed to occur out of nothingness or disappear into nothingness in the independent view.

Subsequently, the duality class links an event that is perceived as an increment according to the view of some organizational unit to *another* event that is perceived as a decrement according to the same organizational unit[4]. This duality concept has been applied in transaction accounting, where it models a fair trade. The model also shows that not all increment and decrement events need to be paired in duality *at any moment in time* (because in fig. 3.2 the relationships with duality are optional) as *temporal* imbalances between increment and decrement events are allowed (e.g. a prepaid or post-paid acquisition). These temporal imbalances conforms to the 'eventually'in REA axiom 2.

As commitments represent planned events, the commitment side of the reference model mirrors the event side of the reference model, showing similar structures. Resembling events, commitments can be viewed as increment, decrement or both by an organizational unit. For instance, one clause in a contract may involve the loss of resources (i.e. sale and delivery) for one organizational unit and the gain of these resources (i.e. acquisition and receipt) for its opponent, whereas another clause in the same contract specifies the amount of money to be paid by the latter to the former. As another example, a schedule specifies which resources will be consumed in return for newly produced ones. Like events, commitments are dual in nature and such commitments are said to be reciprocal [GM02]. Reciprocities, like dualities, show a balance between increment and decrement commitments as temporal imbalances are allowed for events and commitments as long as the 'quid pro quo'(i.e. something in return) principle is not violated (e.g. a promise to perform one or more decrement events is only made if one or more dual increment events are performed or a promise to perform the dual increment event(s) is made). Moreover, the reciprocity class is associated with the organizational unit class, like the duality class, showing that the appreciation of balanced increment and decrement events is subjective since it is related to the viewpoint of exactly one organizational unit (e.g. the price paid or installment plan for a car is deemed fair by the buyer, the remuneration received for a car is deemed fair by the seller). In addition, increment (decrement) commitments can be fulfilled by one or more increment (decrement) events.

3.4 Concept models for business operations

This section presents archetypal object models representing a number of concept patterns (and variants) that apply when using the REA reference information model for modeling various kinds of business transactions and enterprise operations. This section will show how trading-partner view (i.e. observer-dependent) models, which model economic reality from the viewpoint of one organizational

[4]The constraint that an increment and decrement that are related by duality should be different events is not shown in fig. 3.2 but can be expressed by means of an OCL constraint.

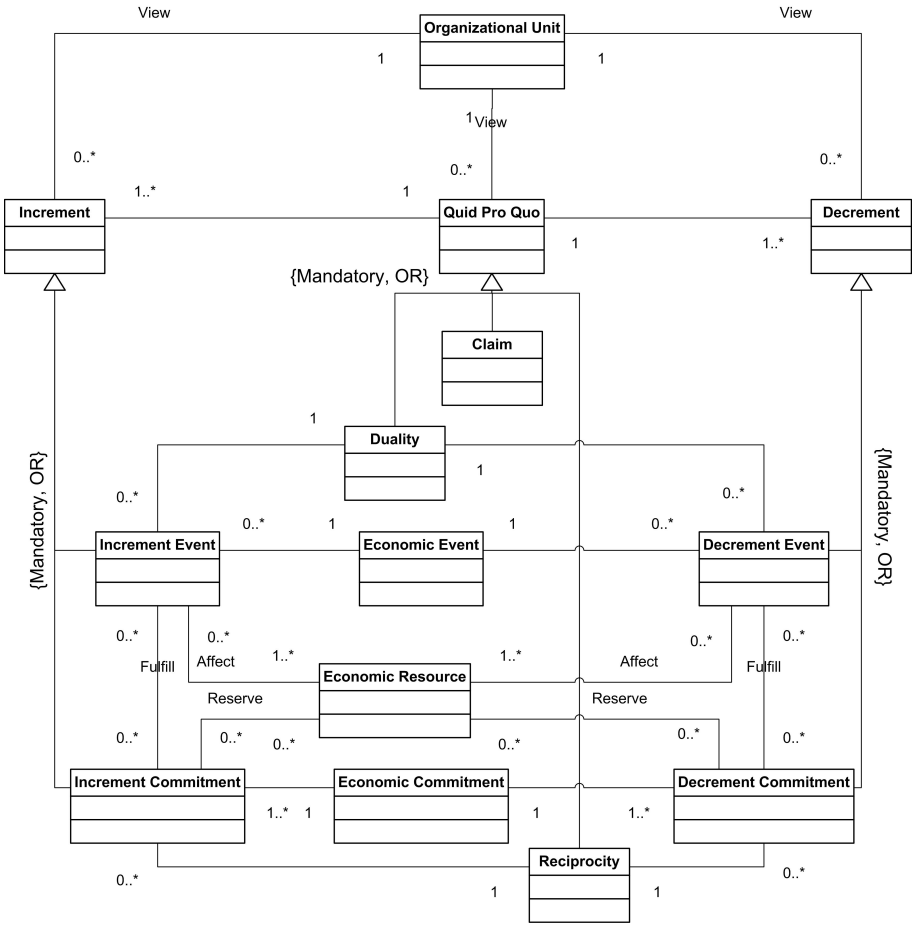

Figure 3.2: Economic rationale component of the reference information model

unit involved in events using the economic rationale component, relate to independent view (i.e. observer-independent) models. First, the trading-partner view models that construct value chains are presented. Later, they are integrated into independent view models that construct value systems.

The archetypal object models for the transformation and transfer event patterns (figs. 3.3 and 3.4) reveal the main consequence of modeling the viewpoint of an organizational unit on the economic reality explicitly, as these object models exhibit a pattern that shows two views of the same event. In the transformation event variant (fig. 3.3) one organizational unit (i.e. PIZZA RESTAURANT) perceives both sides of a transformation event (i.e. PIZZA PRODUCTION), as it observes a decrement to resources that embody inputs to the conversion (i.e. CONSUME INGREDIENTS) and an increment to the resources that represent outputs of the conversion (i.e. PRODUCE PIZZA). The transformation event pattern only partially satisfies the first REA axiom because for the involved resources either an inflow event or an outflow event is identified, but not both. In the transfer event variant (fig. 3.4) two organizational units perceive one side of the transfer event each, as the providing organizational unit (i.e. PIZZA RESTAURANT) perceives a resource decrement and the recipient organizational unit (i.e. JOHN DOE) observes a resource increment. Like the transformation event variant, the transfer event pattern partially conforms to the first REA axiom because for each organizational unit involved in the event, either an inflow or an outflow is modeled for the resource, but not both. Both variants satisfy the second REA axiom by modeling duality (i.e. transformation and transfer duality, respectively) as an inherent part of transformation and transfer events, since the increment and decrement relate to the same event. As neither variant models an exchange, the third axiom does not apply. Note that an exchange would require a transfer in both directions, whereas fig. 3.4 shows a transfer in only one direction. Consequently, when used on their own, these patterns do not build valid models according to the REA ontology.

Next to the archetypal object models for economic events above, also the duality principle for exchanges needs to be represented in object models. Both trading-partner views on an exchange (i.e. fig. 3.5 and 3.6 respectively representing the buyer and seller side, which show an acquisition and revenue cycle respectively [McC03]) connect the parts of the resource swap that is the core of any exchange, representing the duality principle for exchanges that is captured in the second REA axiom. The object model template for this exchange pattern, representing the buyer or seller side of an exchange, knows only one valid configuration, which pairs two transfer events (i.e. one viewed as increment and one viewed as decrement). These transfer events may represent a good transfer and an opposing money transfer, two money transfers (e.g. when modeling debt financing (loan proceeds and loan repayments)) or two good transfers (e.g. when modeling barter trade). The buyer and seller side of the exchange pattern only partially satisfy the

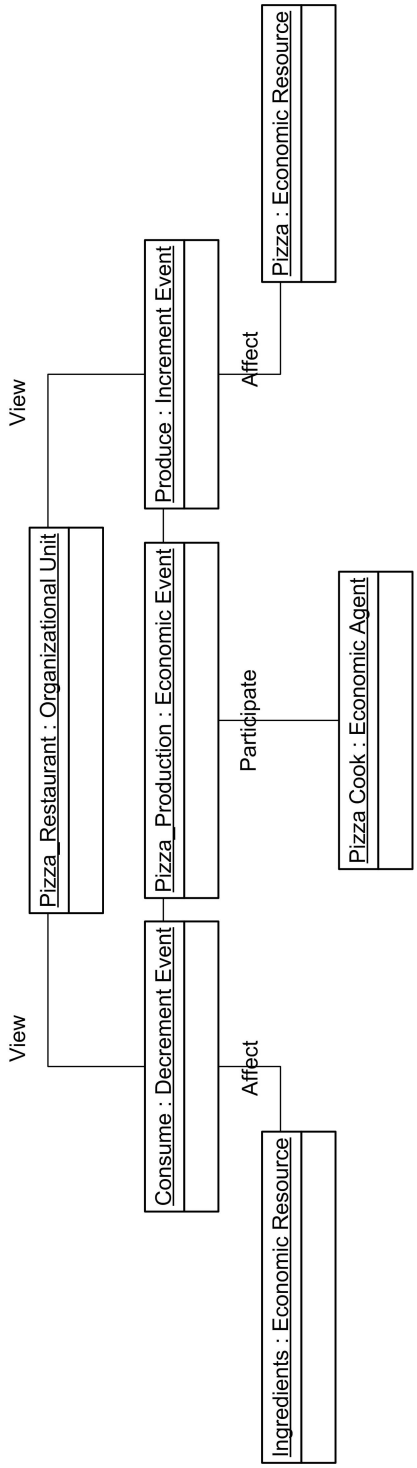

Figure 3.3: Archetypal object model for the transformation event pattern

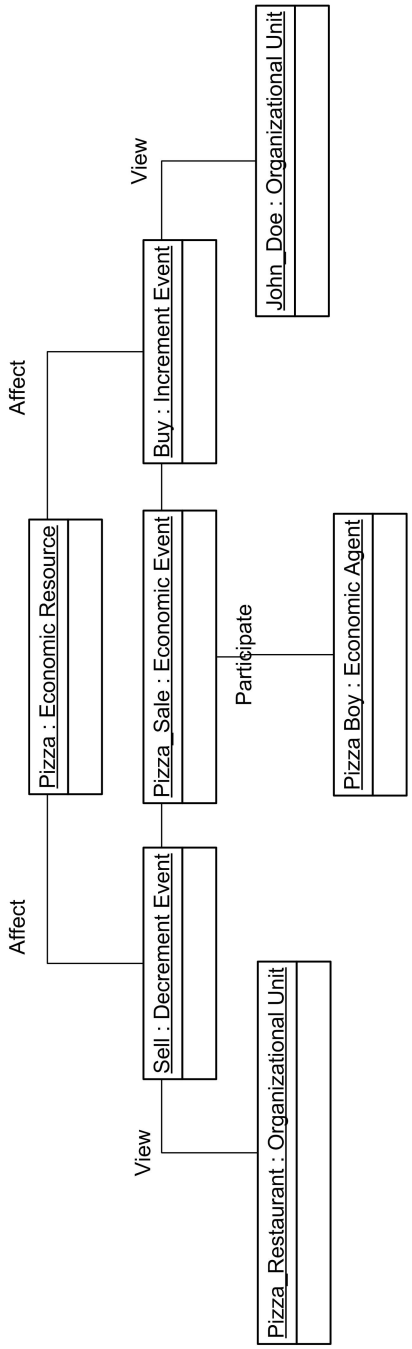

Figure 3.4: Archetypal object model for the transfer event pattern

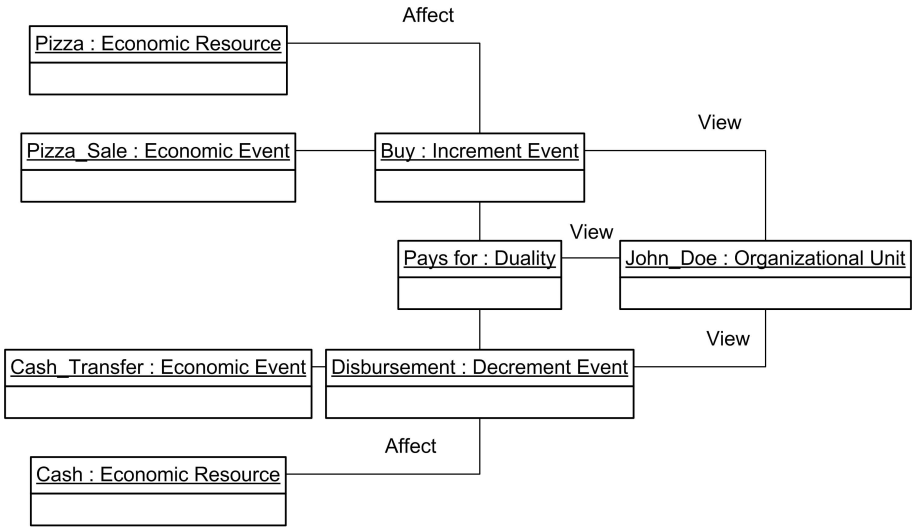

Figure 3.5: Archetypal object model for the buyer side of the exchange pattern (i.e. acquisition cycle)

first REA axiom because for the resources involved either an inflow event or an outflow event is identified, but not both. Although transfers in both directions are modeled, which is different with the transfer event pattern (fig. 3.4), the exchange pattern (i.e. fig. 3.5 and 3.6) still not satisfies the third REA axiom as the partner in the exchange (i.e. the customer or supplier of the organizational unit modeled) is not shown. So, used on its own, the exchange pattern cannot be used to build models that belong to the set of intended models of the REA ontology.

The archetypal object model for the exchange pattern (fig. 3.7) then aggregates two trading-partner view models in one independent view model. As a first step, figs. 3.5 and 3.6 show concept models representing, respectively, the buyer side (i.e. JOHN DOE) and seller side (i.e. PIZZA RESTAURANT) of an exchange. Both models conform to the second REA axiom, but violate the first and third axiom. However, the resource transfer events (i.e. PIZZA SALE and CASH TRANSFER) and transferred resources (i.e. PIZZA and SALE) can be equated (e.g. via a shared identifier (e.g. serial number)), which creates fig. 3.7 that does conform to the first axiom. The resulting model provides an independent view of the exchange by incorporating the perspectives of both parties. Therefore the exchange pattern also conforms to the third REA axiom as it identifies transfer event faces that do not belong to the inside view of the organizational unit (i.e. increment and decrement events and dualities that are not linked to a particular organizational

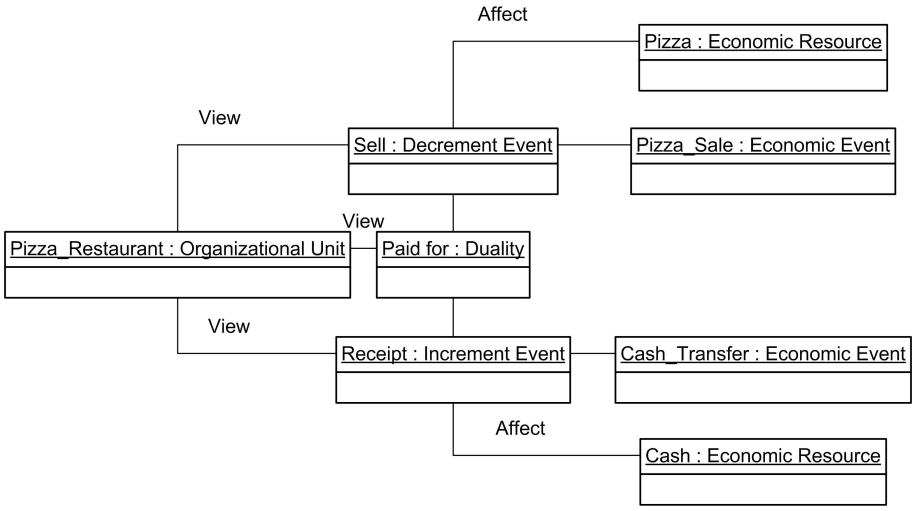

Figure 3.6: Archetypal object model for the seller side of the exchange pattern (i.e. revenue cycle)

unit are considered 'outside'the view of this organizational unit). The exchange pattern satisfies all axioms so it can be used to construct models that belong to the set of intended models of the REA ontology.

The exchange pattern represented in fig. 3.7 demonstrates that the third REA axiom (implicitly) supposes the existence of a trading-partner view on the recorded events. Business semantics are attached to this model by taking a viewpoint on the modeled events (e.g. the good transfer event is an increment called purchase for buyer and a decrement called sale for the seller). For the buyer the sell and cash receipt transfer event faces belong to the outside subset, while the buy and cash disbursement transfer event faces belong to the seller's outside subset.

The business-specific concept models of the REA reference information model that were presented in this section, are related to the enterprise information system patterns [DCH05] and business patterns [Hru06] derived from the REA ontology. However, it should be remarked that in those patterns distinct increment and decrement events were modeled due to the exchange focus and the implicit trading-partner view of the REA ontology. This implicit trading-partner view, which originates in the value chain (instead of value system) focus of business modeling, allows for the registration of one and the same transfer event as an increment event (i.e. receipt) in one system and a decrement event in another system. The event and exchange patterns of the REA reference information model, on the other hand, make it also possible to model business from an independent point of

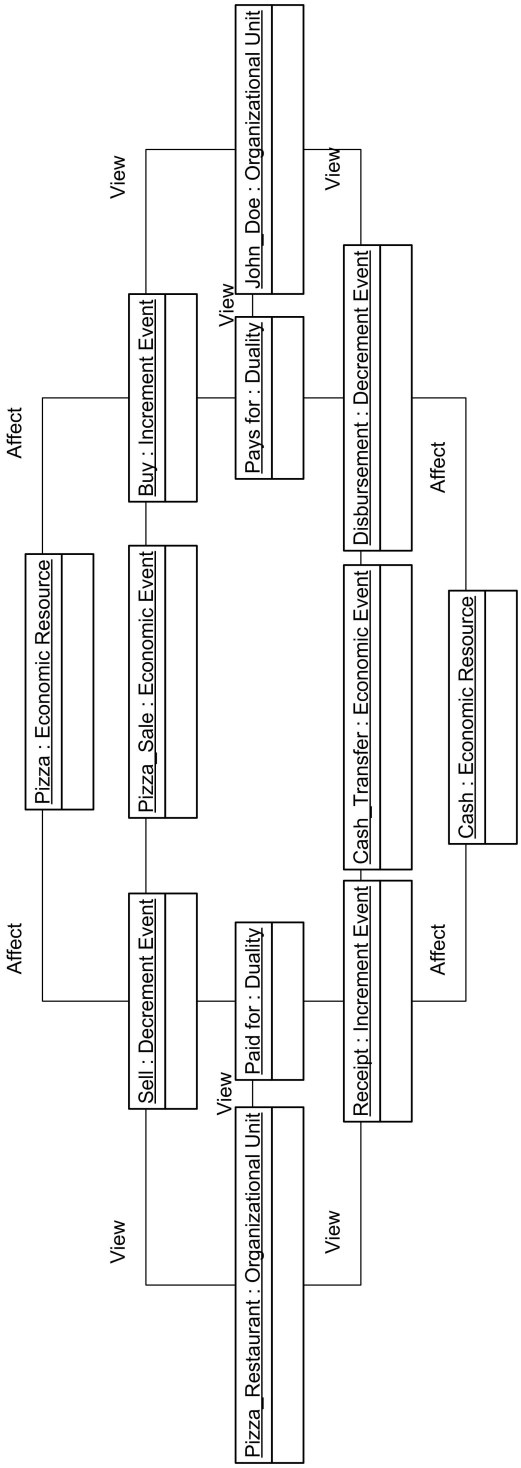

Figure 3.7: Archetypal object model for the exchange pattern

view, meaning that goods and money transfers are recognized only once (independent view) but may be observed and registered twice (as increment for one party and as decrement for the other party).

3.5 Concept models for business planning

Whereas the preceding section addressed models that document the current state and the history of an organizational unit, this section addresses models that project planned future states. Of all potential future organization unit states, the REA reference information model includes those that are desired and documented in schedules and contracts. Those schedules and contracts consist of increment and decrement commitments that are paired in reciprocity with each other and that mimic the event and duality patterns presented in the preceding section. Consequently, all archetypal object models and modeling templates of the preceding section have planning equivalents that can easily be conceived by replacing events with commitments, dualities with reciprocities, increment events with increment commitments and decrement events with decrement commitments. Therefore, commitments require a dichotomy similar to the transfer-transformation dichotomy for events (i.e. scheduled and contracted commitments). Contracted commitments are fulfilled by transfer economic events, scheduled commitments are fulfilled by transformation economic events. [GM04] Fig. 3.8 models the committed equivalent for the exchange pattern in fig. 3.7.

Since, at any stage of an exchange, there can be commitments that are not fulfilled. Fig. 3.9 represents the traditional accounting interpretation of 'positive claim'(e.g. accounts receivable) and 'negative claim'(e.g. accounts payable). The positive claim of the seller (i.e. PIZZA RESTAURANT) on the buyer (i.e. JOHN DOE) is represented by the imbalance between the decrement event (i.e. SELL PIZZA) and the unfulfilled increment commitment (i.e. CASH RECEIPT). The negative claim of the buyer to the seller is represented by the imbalance between the increment event (i.e. BUY PIZZA) and the unfulfilled decrement commitment (i.e. CASH DISBURSEMENT).

Apart from creating commitment images of the domain-specific concept models discussed in section 3.4, also other concept models can be created, which relate commitments to their fulfilling events. The settlement structure for transfers (fig. 3.10) shows a transfer event pattern that mirrors a transfer commitment pattern conveying that, at this point in time, positive and negative claims have been settled. This settlement can be partial or complete given that commitments may be fulfilled by more than one event. Finally, the completed scheduled event object model for transformation (fig. 3.11) shows a transformation event pattern that mirrors a transformation commitment pattern, representing a schedule that has been fulfilled at least partially. As there is only one organizational unit involved, mean-

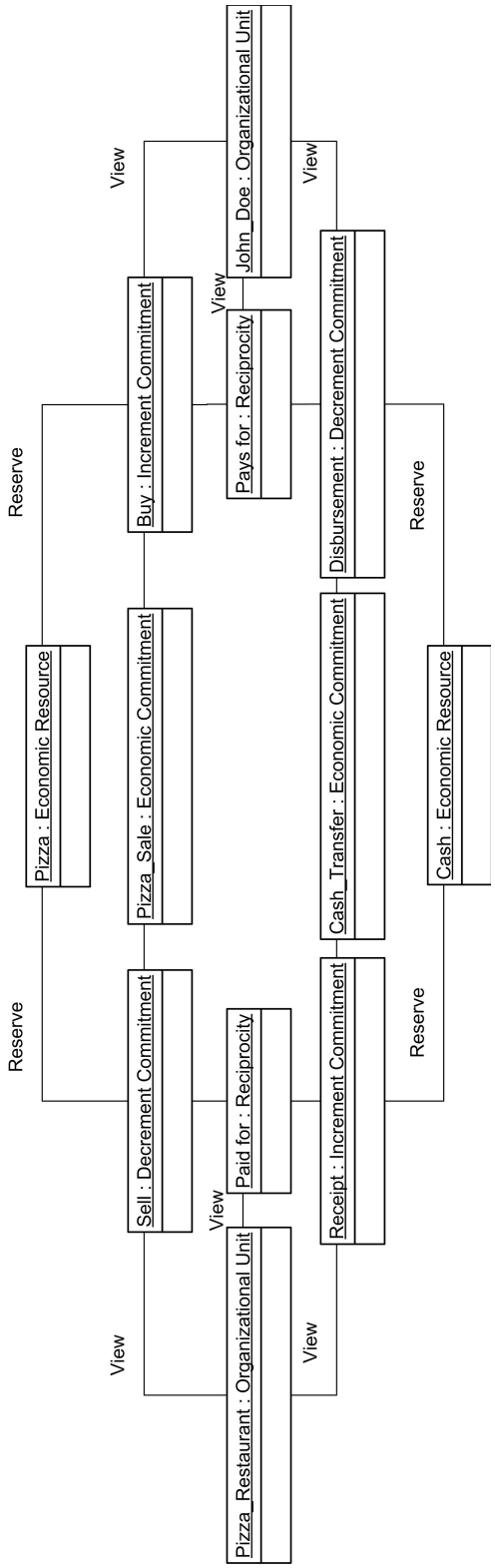

Figure 3.8: Archetypal object model for the contracted exchange pattern

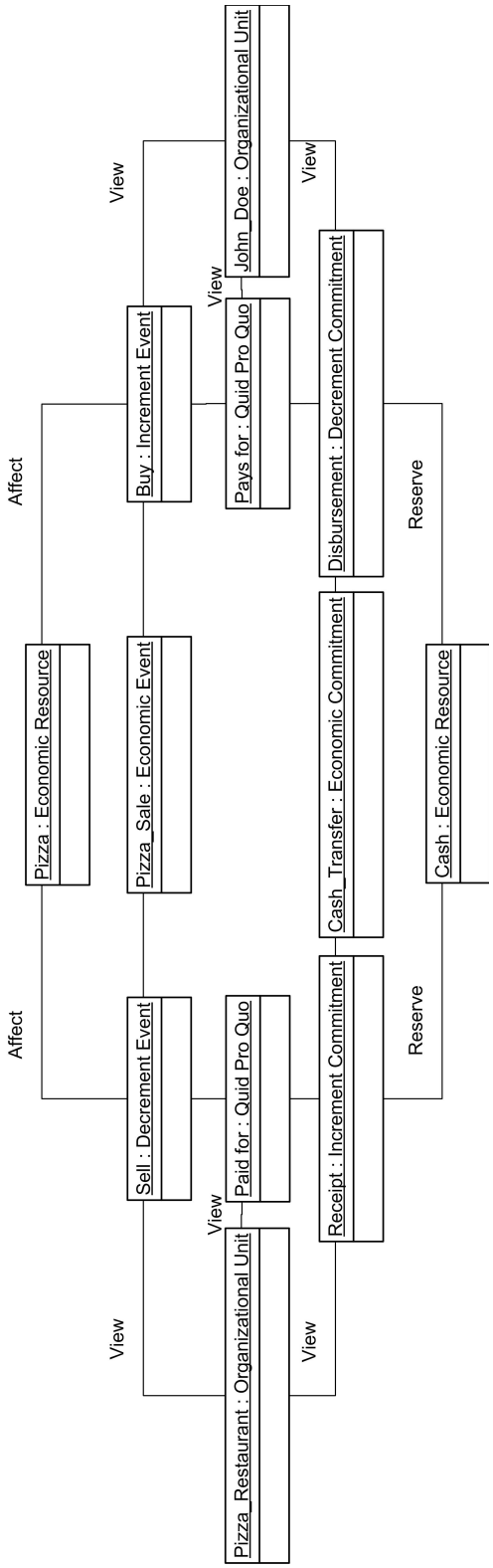

Figure 3.9: Claim and liability object model template

ing that there are no parties with opposing economic interests, positive and negative claims cannot occur for schedules.

3.6 Comparative analysis of related conceptualizations

In this section, the REA reference information model (section 3.3), which is operationalized through a set of concept models (sections 3.4 and 3.5) and conforms to the REA controlled business vocabulary (section 3.2), is shown to overarch and complement multiple value chain and value system conceptualizations. First, the merits of the reference information model presented above as opposed to other REA-based reference models for business modeling are addressed. Although the term REA-based reference model has never been coined before, we believe that REA-based reference models for the trading-partner view can be found in [McC82] and [Hru06], the former (i.e. McCarthy) focuses on transaction accounting (i.e. acquisition and revenue processes) and the latter (i.e. Hruby) on the entire value chain (i.e. including conversion processes). Furthermore, we believe a reference model for the independent view can be found in OeBTO [ISO07]. Second, the merits of the REA reference information model vis--vis the e3-value ontology are discussed. Third, the REA reference information model is shown to complement SCOR.

3.6.1 McCarthy's reference models

Fig. 3.12 shows a modernized version of McCarthy's foundational reference model [McC82] for the trading-partner view. To mitigate for Weber's critique [Web86] and comply with the analysis in [GM02], the n-ary 'control'relation has been decomposed in two separate participation (i.e. agent-event) relationships and the economic agent and economic unit have been merged in the ECONOMIC AGENT class. In this reference model, the economic rationale underlying a business model is given by the INCREMENT-DECREMENT association (representing the duality relationship between an increment and a decrement event) and the inside-outside semantics of the participation relationships. The TRANSACTION-INSIDE_PARTY association reveals the agent whose view determines which events are increments and which are decrements. The TRANSACTION-OUTSIDE_PARTY association relates the inside party's counterparty to the economic event.

Subsequently, fig. 3.13 shows McCarthy's reference model for the independent view as it appears in ISO's business transaction scenario standard [ISO07]. It should be noticed that this independent view replaces the 'increment'and 'decrement'notions by 'to'and 'from'semantics for the participation relationships. As a

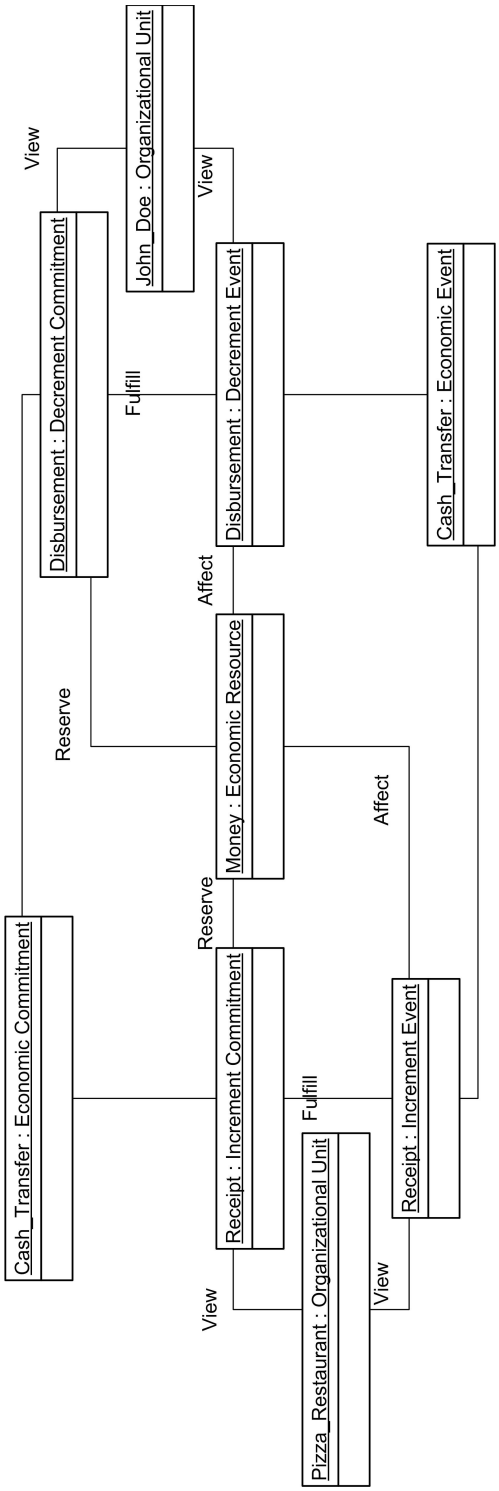

Figure 3.10: Settlement of transfer commitment object model template

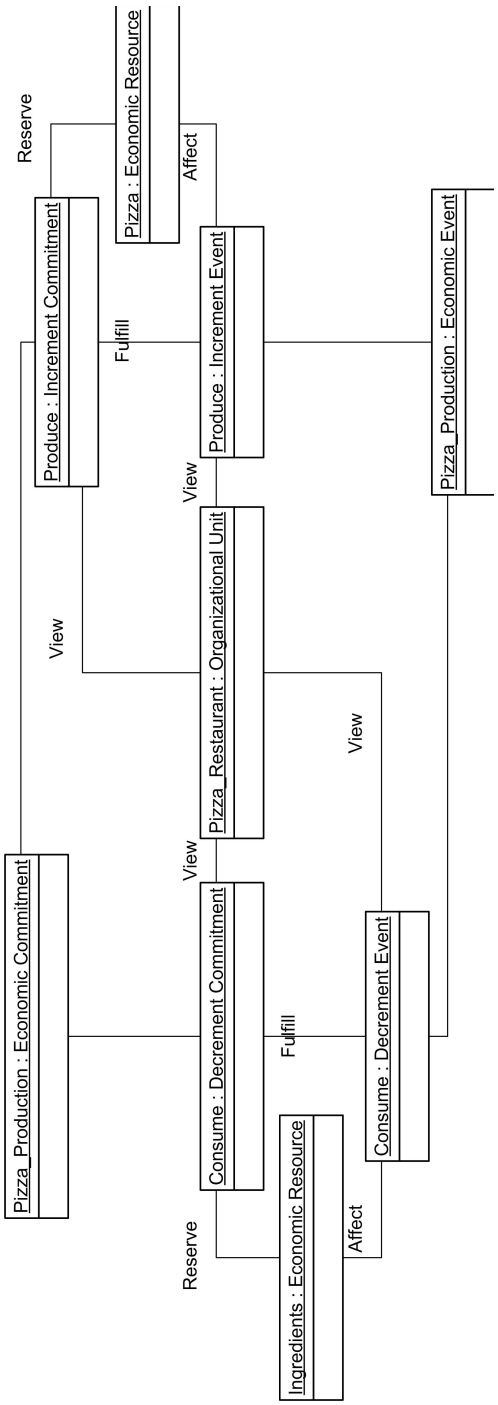

Figure 3.11: Completion of scheduled transformation event object model template

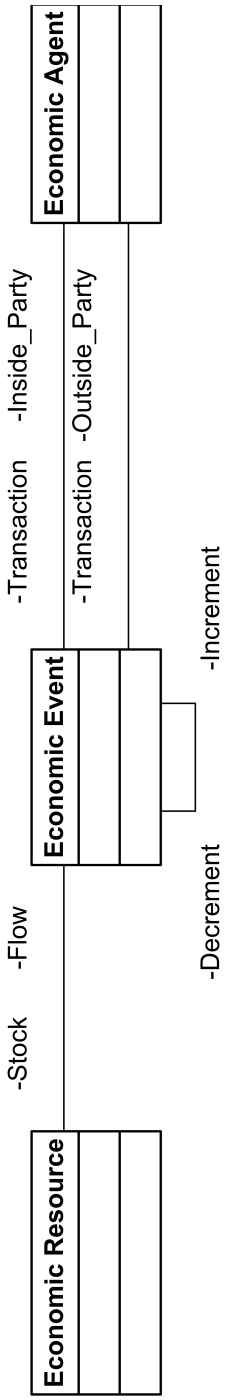

Figure 3.12: McCarthy's trading-partner view reference model

result, McCarthy's independent view does no longer contain increment and decrement semantics, which have to be derived from the TO and FROM relationships. Namely, the TO relationship relates the event to the agent that experiences an increment due to this event, the FROM relationship relates the event to the agent that experiences a decrement. As a result of this modeling choice, the duality relationship loses its increment and decrement semantics, which can no longer be represented explicitly. Consequently, both views (i.e. trading-partner and independent) cannot be integrated with each other although they can be mapped and translated, since they both represent a different view on the same reality by hiding part of the semantics (i.e. the trading-partner view hides the TO and FROM semantics, the independent view hides the increment and decrement semantics). Our reference model mitigates for this omission by representing the increment and decrement view on an event as separate classes. As a result, our reference model enables the representation of 'from'('to') semantics by relating an event to an organizational unit through a decrement (increment) event. Additionally, these relations through the increment and decrement classes of our economic rationale model conform to the inside view in McCarthy's trading-partner view as they represent which organizational unit experiences which economic event as an increment or decrement. The outside view can then be retrieved by finding the organizational unit that has the opposing view (i.e. increment or decrement) on the same economic event.

With McCarthy's reference model, conceptual models for exchanges can be constructed. Fig. 3.14 represents trading-partner view model for an exchange duality between a sale and a cash receipt. Therefore it is equivalent to the revenue cycle model in fig. 3.6. Since this object model shows the seller side of the 'cash for inventory'exchange, the seller is the inside party for both events and the buyer the outside party. Such exchange templates can be used to model all kinds of exchanges (e.g. money for product, product for money, product for product, money for money).However, because the merging model does not distinguish the organizational unit and economic agents concepts, it is hard to discriminate between the agents that participate in the exchange (e.g. a pizza boy) and the trading partners that control the resources being exchanged (e.g. pizza restaurant).

Fig. 3.15 that is equivalent to fig. 3.7, which integrates the buyer and seller side of an exchange without hiding the increment and decrement semantics as McCarthy's independent view model does, then shows the independent view equivalent of fig. 3.14. As it mimics the trading-partner view model (fig. 3.14), it also lacks the ability to discriminate between the trading partners and the event performers.

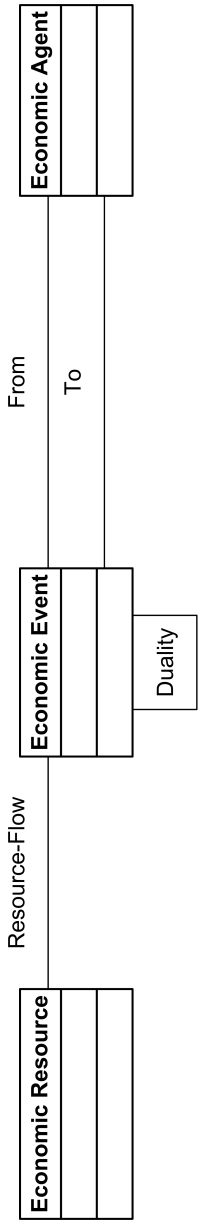

Figure 3.13: McCarthy's independent view reference model

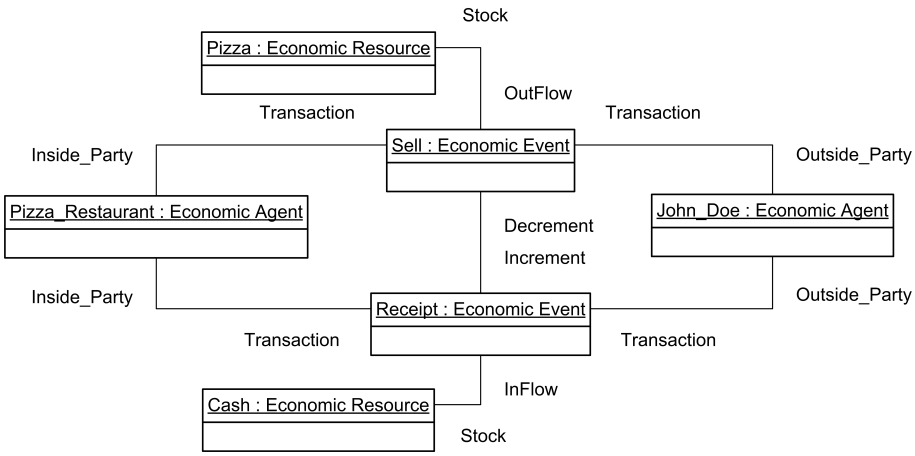

Figure 3.14: McCarthy's trading-partner view exchange object model template

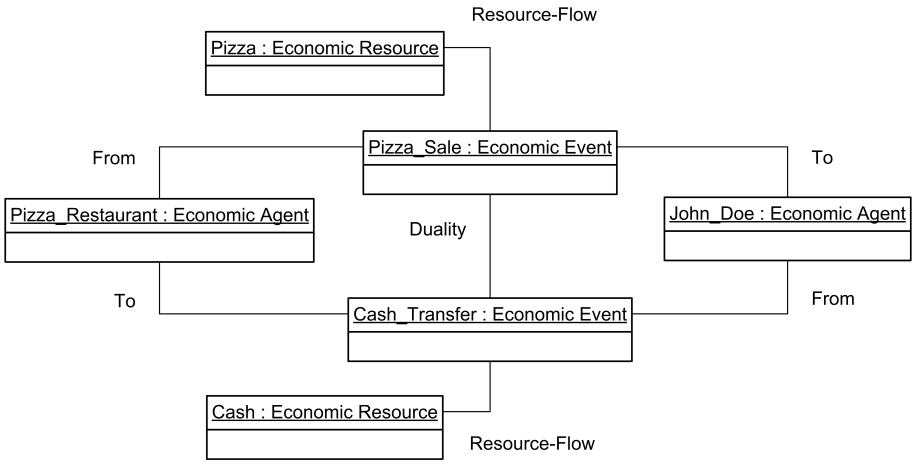

Figure 3.15: McCarthy's independent view exchange object model template

3.6.2 Hruby's reference model

Fig. 3.16 shows Hruby's reference model as it appeared in his business patterns book [Hru06]. By explicitly discriminating increment and decrement events, it differs from McCarthy's trading-partner view reference model, which chooses to represent increment and decrement as roles of events and not as kinds of events. This articulation of the REA ontology has, alike our reference model and McCarthy's trading-partner view reference model, the advantage that it explicitly shows how resources are affected by events (i.e. INFLOW and OUTFLOW), without having to rely on interpretation. The classification of events as increments or decrements depends on the viewpoint (e.g. seller, buyer) taken. This viewpoint should be unambiguous as Hruby's reference model explicitly takes the trading-partner view on economic phenomena. Nevertheless, the abstraction from view integration (i.e. trading-partner and independent) does impose considerable limitations to the use of the reference model, in particular when it is used to integrate value chain models of supply chain partners into an overall value system model. Like McCarthy's reference models, Hruby's reference model does not explicitly discriminate between the natural persons that participate in events and the natural or legal persons that experience the economic consequences of those events.

Hruby's reference model also incorporates provide (i.e. PROVIDER-PROVIDE) and receive (i.e. RECIPIENT-RECEIVE) relationships that connect agents with increment and decrement events. The provide (receive) relationship links the event to the agent that experiences a resource decrement (increment) due to the event, which makes Hruby's provide (receive) relationship equivalent with McCarthy's 'from'('to') relationship in the independent view. This indicates that the viewpoint that is taken is implicit in Hruby's conceptual models, as the viewpoint determining entity is the provider for a decrement event and the recipient for an increment event.

Finally, Hruby decomposes McCarthy's stock-flow relation into consume (i.e. OUTFLOW-CONSUME), use (i.e. OUTFLOW-USE) and produce (i.e. INFLOW-PRODUCE) relations. In Hruby's reference model, use and consume are represented as if they were synonyms, whereas use relations represent how events temporary occupy resources without affecting their ability to participate in subsequent decrement events, while consume relations indicate that a resource is no longer available for subsequent events. This representation of the use relationship motivates the modeler to hide the actual resources that are consumed and produced (i.e. rights and abilities) during such a use event. In our reference model, this apparent synonymy has been accounted for by representing use relationships as part of the informative component (fig. 3.1) and consume relationships (i.e. INCREMENT EVENT-ECONOMIC RESOURCE) as part of the economic rationale component model (fig. 3.2).

Fig. 3.17 shows an exchange template in Hruby's articulation. It is fairly iden-

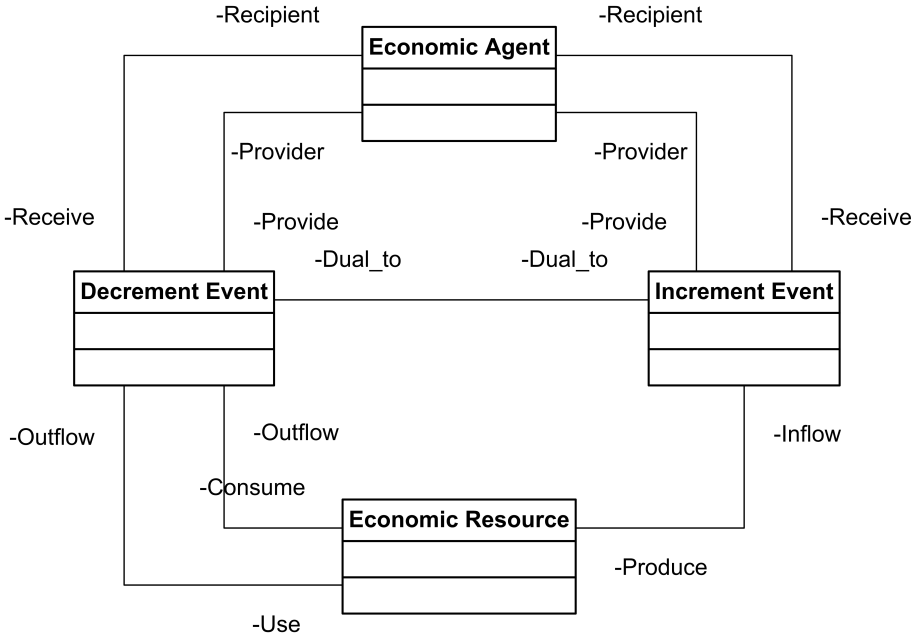

Figure 3.16: Hruby's REA reference model

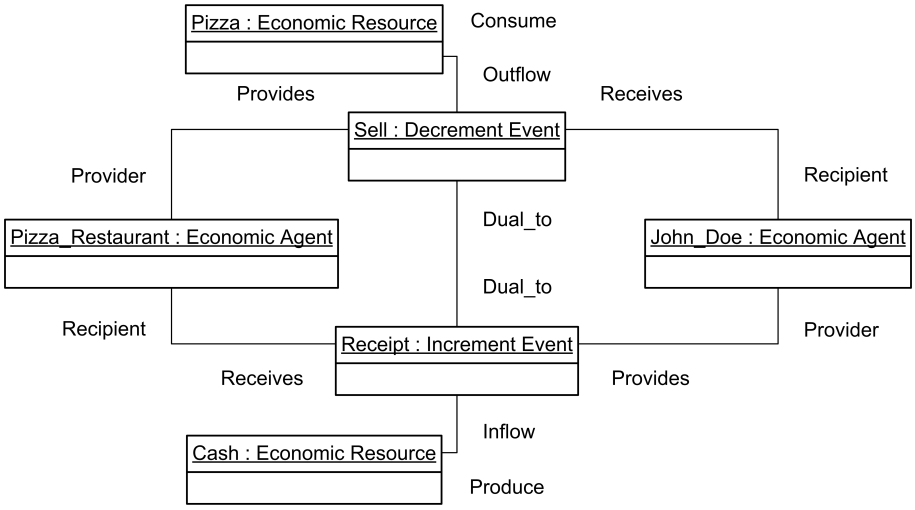

Figure 3.17: Hruby's REA exchange object model template

tical to McCarthy's articulation (fig. 3.14). The main difference is that the increment and decrement roles in McCarthy's model are replaced by specific classes. The inside and outside party roles on the participation relations in McCarthy's model have also been replaced by provider and recipient roles. It also lacks the discrimination between trading partners and event performers and by mixing trading-partner view semantics (i.e. increment and decrement) and independent view semantics (i.e. provide and receive) it also suffers from an implicit viewpoint that masks which party determines the increment and decrement classification of events.

Fig. 3.18 then displays a transformation object model that represents (part of) a production process. This kind of model cannot be produced using McCarthy's reference model, as McCarthy focuses on exchange processes only (and not also on conversion processes as Hruby's reference model does). The transformation is represented as an implicit exchange between process inputs (i.e. INGREDIENTS) and process outputs (i.e. PIZZA), where the employer (i.e. PIZZA_RESTAURANT) provides the inputs and receives the outputs from the employee (i.e. COOK) and the employee receives the inputs and provides the outputs to the employer. The transformation process itself is modeled as a duality between one or more decrement events that use or consume input resources and one or more increment events that produce output resources. The use of the exchange template to model transformation processes is problematic, since it represents the transformation process as if the executor (i.e. PIZZA COOK) gains economic control over the ingredi-

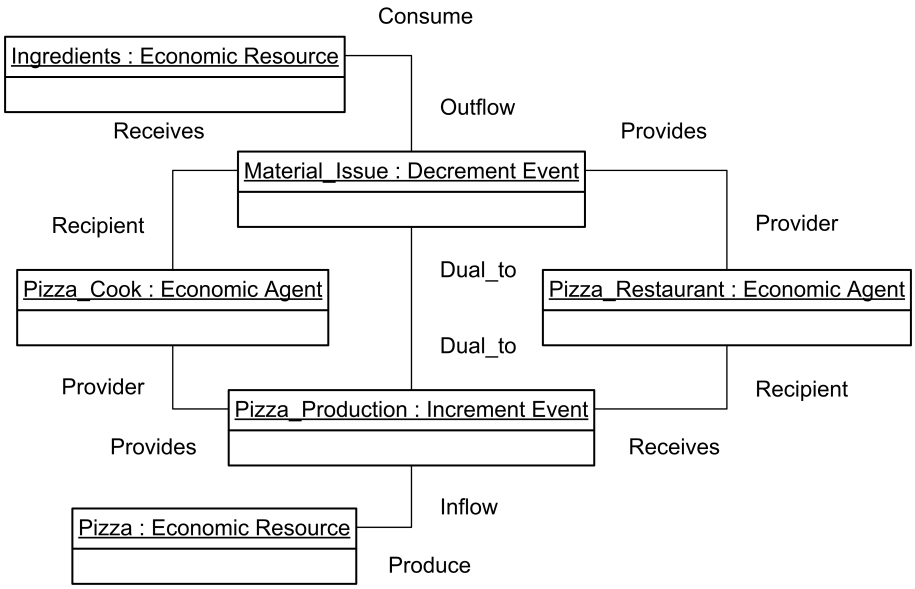

Figure 3.18: Hruby's REA conversion object model template

ents during the transformation process (i.e. pizza production). Therefore, using an exchange as a template for a transformation process does not provide a true and fair representation of reality. In our reference model, this shortcoming of Hruby's reference model is mitigated by discriminating between organizational units (e.g. PIZZA RESTAURANT) and agents (e.g. PIZZA COOK) and relating them differently to economic events (i.e. VIEW and PARTICIPATION association respectively). Additionally, the artificial decomposition of transformation events in input consuming and output producing events is overcome, using our reference model.

3.6.3 e3-Value Ontology

A business conceptualization that is also closely related to the REA ontology is specified by the e3-value ontology [Gor02], which was first developed to elicit the requirements for the creation of e-commerce information systems. The e3-value approach includes three viewpoints on e-commerce (i.e. the value, process and information systems viewpoint). The information systems viewpoint is related to e.g. the work of Batra and Sin [BS08] and Dunn et al. [DCH05], the process viewpoint is related to e.g. DEMO [Die06] and the value viewpoint is related to e.g. OeBTO [ISO07] and thus REA. The value viewpoint is used for value

modeling, i.e. the construction of business models that emphasize the creation, transfer and consumption of valuable objects in a value system [KDG+08]. The value viewpoint supports the design of new business constellations as, based on the models, the profitability of proposed constellations can be predicted for each of the partners involved.

Fig. 3.19 presents the core concepts of the e3-value ontology, modeling the pizza case that we used throughout this paper. Actors represent economically independent (legal) entities [Gor02] and market segments (e.g. CUSTOMERS and SUPPLIERS) are aggregates of actors. Actors correspond to organizational units in the REA reference information model. Economic agents, on the other hand, are not addressed since e3-value abstracts from the actual workflow of value activities. These market segments (e.g. CUSTOMERS) group individual customers (e.g. John Doe). Such a group can be modeled as an extension to the reference model [GM06].

These actors and market segments have value interfaces that consist of value offerings (e.g. offer pizza) that aggregate value ports. Value ports are used to show the value objects (e.g. PIZZA), which are similar to the resources in our reference information model resources, that actors request from (i.e. when the value port is an 'in'port) or provide to (i.e. when the value port is an 'out'port) their environment. These value ports can be seen as the increment and decrement classes in the reference information model. By connecting the 'in'value port of one value interface with the 'out'value port of another interface, a potential transfer of a value object is modeled. Hence, this connection, called a value exchange in e3-value, corresponds to a transfer event in the reference information model, so the 'in'and 'out'ports that are connected by this value exchange are the increment and decrement sides of the transfer event.

Value exchanges are grouped in value transactions as in the reference information model's exchange pattern (fig. 3.7). Value offerings group the equally directed (i.e. 'in'or 'out') value ports of a value interface. The value offering concept is used to represent goods or services that are requested or provided as a bundle. Finally, a value interface (e.g. PIZZA for MONEY) represents the economic reciprocity principle (i.e. adequate compensation for outgoing value objects) that is also incorporated in the duality class of the reference information model (i.e. buyer or seller side of exchange (fig. 3.5 and 3.6)). Just as a duality relates to the organizational unit to which the balancing economic reciprocity applies, a value interface relates to an actor (e.g. PIZZA RESTAURANT) or a market segment (e.g. CUSTOMERS, SUPPLIERS).

With the exception of the market segment construct and the value offering construct, all e3-value constructs and structures map onto constructs or concept models of the REA-based reference information model that we propose. In terms of this reference model, a value offering would be an aggregate of either increments

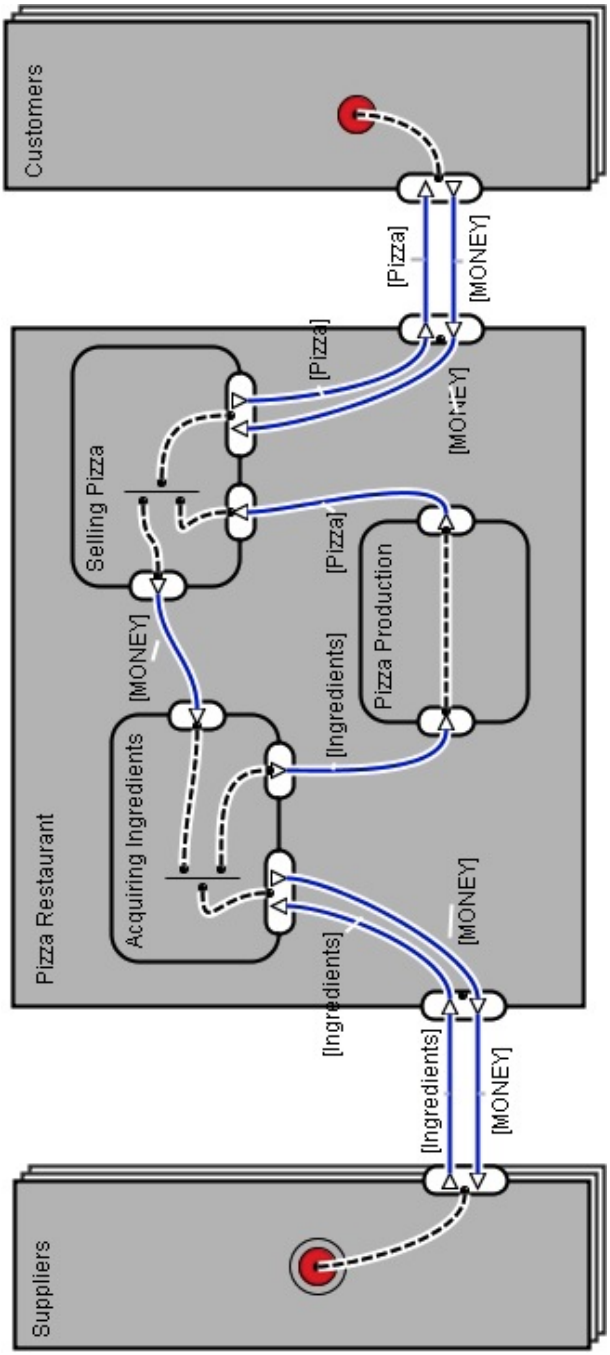

Figure 3.19: The e3-value ontology (source: [Gor02])

or decrements and a market segment would be an aggregate of organizational units sharing the same views. It should be noted that e3-value was designed for representing future business designs whereas the reference information model was designed for recording information (both historical and projected) about implemented business designs. Consequently, e3-value falls short on modeling planning, since e3-value models only potential (but not planned) exchanges of value objects. As the e3-value ontology has been developed to predict the profitability of innovative business designs in complex environments, it focuses on the interactions between organizational units (i.e. actors and market segments), exclusively taking the independent view, and falls short on describing internal business processes, which are considered out of scope of the value viewpoint [Gor02]. The reference information model presented in section 3.3, on the other hand, can account for this internal view of each of the parties involved in a business constellation, so specifies a more general business conceptualization than e3-value does.

3.6.4 SCOR

A popular business conceptualization for value systems is the Supply Chain Operations Reference (SCOR) model [Cou06] of which fig. 3.20 depicts the top level structure showing the management processes involved in supply chain management. In fig. 3.20, the organizational unit construct that is essential to the REA reference model is instantiated into five specifically named units (i.e. SUPPLIER'S SUPPLIER, SUPPLIER, YOUR COMPANY, CUSTOMER, CUSTOMER'S CUSTOMER). The vertical lines between these units'captions explicitly represent their mutual boundaries.

The source and deliver processes correspond to the buyer (fig. 3.5) and seller (fig. 3.6) sides of the REA reference information model exchange pattern, while their overlap (i.e. source of buyer and deliver of seller) signifies a potential integration into an independent-view exchange pattern (fig. 3.7). The make process, on the other hand, groups a chain of conversions, represented by chains of transformation event patterns (fig. 3.3), that connect an organizational unit's inputs to its outputs. The inputs to the make process are connected to the increment side of 'source'or buyer models, whereas the outputs of the make process are connected to the decrement side of 'deliver'or seller models. The return processes then correspond to the buyer and seller side of an exchange pattern that returns a received resource to settle the seller's positive claim and the buyer's negative claim. Finally, the plan process groups the commitments that are included in contracts and schedules, which coordinate the source, make, deliver and return processes.

As the SCOR reference model focuses on the implementation of supply chains, it provides a detailed categorization of the REA reference information model templates for the events, dualities, reciprocities and commitments that mediate re-

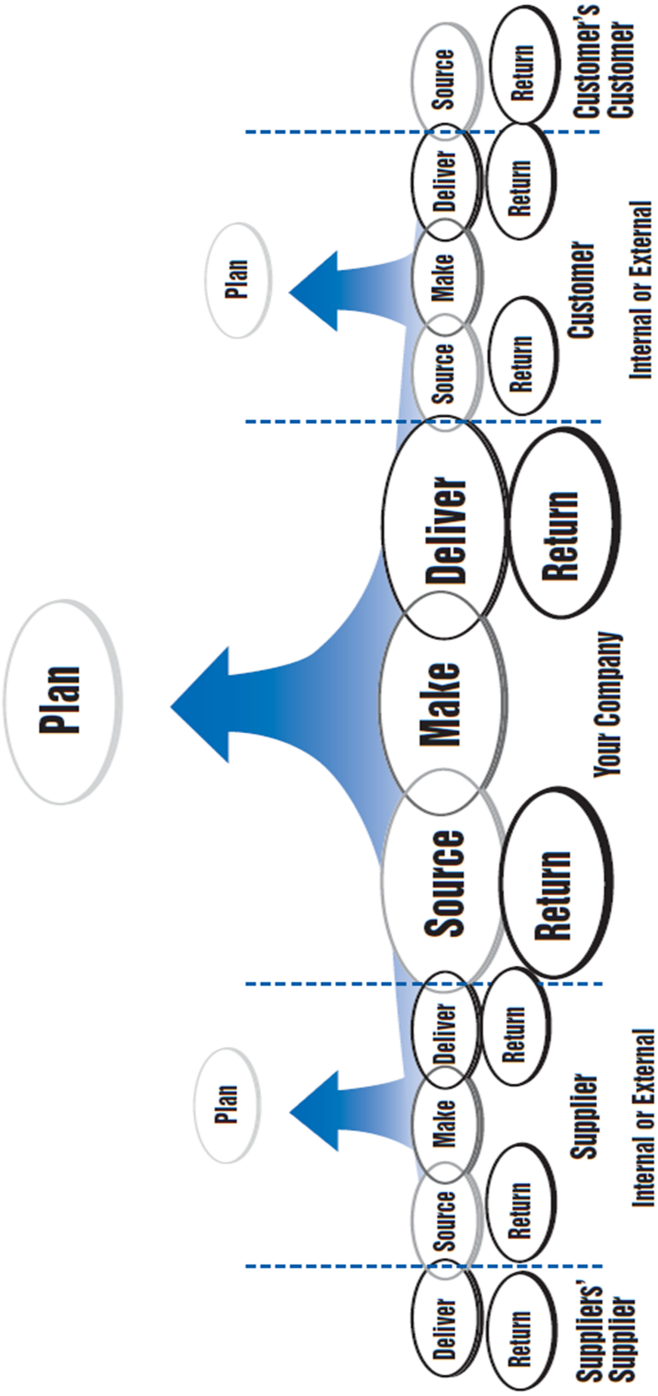

Figure 3.20: The top-level SCOR model (source: [Cou06])

source in- and outflows. Hence, the entire SCOR conceptualization can be articulated as a business modeling pattern catalogue that provides an architecture of semantic variations to the concept models specified in this paper, refining REA's controlled vocabulary into SCOR's controlled vocabulary. The REA reference information model, its concept models and controlled vocabulary could thus support a formalization of SCOR such that SCOR-based supply chain systems can more easily be integrated with other kinds of information systems based on the REA reference information model.

3.7 Conclusion and Future Research

This paper presented a reference information model for the business domain that can be used for the modeling of value chains and the value systems they are part of. The conceptual basis for this reference model is the REA ontology that provides a controlled vocabulary for business. The paper also presented business-specific concept models (in the form of archetypal object models or object model templates) that are embedded in the reference information model. Via these concept models it was demonstrated that the proposed reference information model enables taking both an independent view and a trading-partner view on business reality. This is undoubtedly the most distinctive feature of the proposed reference model because it allows modelers to construct business models that provide a basis for developing information systems for each enterprise taking part in a value system and for supporting the creation of system interoperability and information sharing amongst these business partners. For example, by integrating the features of McCarthy's and Hruby's reference models, our reference model enables the representation of conversions, transfers and exchanges via a single reference model, which eliminates the need for many model integration efforts.

The introduction of the REA organizational unit concept as business semantics viewpoint determining entity is a key feature of our reference information model. Where previously, the perspective on business reality of each enterprise was represented in a separate model, the views of different enterprises that are part of a value system (like a supply chain) can now be jointly represented in a single model via the organizational unit concept and its relations with resources, events and agents. This explicit representation of enterprise viewpoints allows for a central administration of transactions between and transformations within enterprises. Apart from the central administration of data generated by multiple enterprises, data interoperability is also expected to be facilitated when the integrated enterprises use the same reference model and agreement is reached about a minimal set of attributes (e.g. identifiers). The explicit enterprise scope also eases strategic analysis in the light of vertical integration (i.e. what would the new enterprise look like; when two supply chain partners (e.g. customer and supplier) merge). Due to the ex-

plicitly modeled enterprise scope, transfer events between the trading partners to be merged can be retrieved easily and reclassified as non-profit yielding internal transfer in case supply chain partners need to be consolidated.

The unification of increment and decrement events into a single economic event that is perceived differently by different organizational units can also help improve product traceability by identifying the event chains (i.e. transfer and conversion events) that lead to the products, irrespective of the number of enterprises (and enterprise information systems) in which products and their constituents have their origin. Such product tracing infrastructure might support product authentication in the battle on counterfeit and other supply chain intrusions (e.g. food safety scandals). It may also help to trace the origin of money (e.g. drugs money) in the battle against money laundering. As is demonstrated by Bechini et al. [BCMT08] who propose a structure similar to the one presented above, for tracking (i.e. looking for a resource's (potential) destination) and tracing (i.e. looking for a resource's origin) resources, using ebXML [ebX]. Although they provide a valuable approach for managing flows of goods, they do not enable the tracking and tracing of the money flows that remunerate them [Fos22]. This enabling of money tracking and tracing, together with product tracking and tracing, is another contribution of the reference information model in the design of inter-organisational information systems.

The work presented here is only a first step towards developing a business modeling approach that can accommodate the shift in focus of business modeling from the enterprise to the supply chain. To make the proposed reference information model usable, the concept models presented here need to be elaborated into a catalogue of business modeling patterns that are documented using modeling templates at the appropriate level of abstraction (e.g. as UML class diagram fragments instead of object model fragments). A second limitation, though the result of a deliberate choice, is that the reference information model abstracts from application specific inferences like the sequencing of events or other process control flow aspects that are, for instance, key to workflow modeling[5].

In the future we will further explore the abilities of REA as a language for business modeling and use the integrated REA reference model proposed in this paper to create strategy tools (e.g. for determining the opportunity of vertical integration, simulating the return of business process redesigns), which elaborate on the value viewpoint represented by the e3-value ontology. The REA business pattern language that refines the concept models presented in this paper will be diversified into a variety of business modeling patterns (e.g. by incorporating the entire SCOR reference model). The basic simulation models (and part of their functionality, including control flow) for the creation of such strategy tools were

[5]Note that the Workflow System conceptual model [vdAvH02] does not include control flow aspects.

presented in chapter 5 and 6 .

3
Summary

Objectives

Allowing each enterprise taking part in a value system to develop its own information system and at the same time support the creation of system interoperability and information sharing amongst business partners in the value system.

Approach

Grounding the observer-independent model in the mature Resource-Event-Agent (REA)

Findings

Information of different supply chain partners can be represented such that it can be integrated across enterprise boundaries, revealing the inadequacy of many academic and practitioner-defined conceptual models.

Contribution

This paper presents a general way of modeling business phenomena that is observer-independent and can be used for the modeling of value chains and value systems they are part of.

Practical Implications

This proposed reference information model presents a first step in developing a business modeling approach that can be used for both value chain and value system modeling and that provides a basis for both information systems development and integration.

4

Tracking and Tracing Future, Present, and Past Product and Money Flows

Abstract

This paper presents a reference model for the registration of economic data (e.g., transactions between business partners) that enables the tracking and tracing of product and money flows in the registered data. The model is grounded in the REA ontology, which has its origin in accounting and provides the conceptual foundation for the ISO open-edi transaction standard. The use of the reference model is illustrated with an example database that demonstrates the different usage scenarios covered by the model. These scenarios include tracking and tracing the composition of products (e.g., the origin of their parts) as well as the events that constitute the history of a product (e.g., logistic and production processes, transactions between trading partners), which is useful for supply chain authentication and quality assurance (e.g., for food safety). Other scenarios that are illustrated address the pairing of product and money flows, typical for business transactions and useful for monitoring the legal and illegal money flows associated with product flows (e.g., for eliminating terrorism funding with counterfeit), and the identification of future product and money flows, which is especially useful for supply chain management. Although many reference models for traceability, business transactions

and planning exist, there is currently no model that integrates all these features and leverages traceability to enable the monitoring of business transactions and process sequences.

Keywords

Reference Model, Traceability, Resource-Event-Agent Ontology, Money Flow, Product Flow, Business Transaction, Supply Chain Management

4.1 Introduction

Tracking and tracing are important notions in supply chain management, where tracking is defined as following a product's path through the supply chain from supplier to customer, and tracing as identifying a product's origin. [BCMT08] When product quality is capital, product traceability can discourage free-rider behavior (e.g., providing substandard products). [PS08] Therefore, product tracking and tracing can be found in the pharmaceutical, automotive, aircraft industry and the agricultural and food sector. [WC98] For example, in the aviation sector, aircraft parts are marked such that their lifecycle can be monitored carefully. [Kriar] Among these high-stake industries, the agricultural and food sector has been most visibly attributing attention to product tracking and tracing. Primarily because in the past, food borne (e.g., BSE [CM98]), contagious diseases in livestock (e.g., bovine tuberculosis [GMB+05], foot and mouth [FMM06]), and food safety concerns for customers and their pets [GVF07, MSF+05] affected the credibility of food industry safety schemes. Additionally, potential bioterrorism raised interest in monitoring food chains. [HPS09] Furthermore, retailers have found that commercial advantage can be gained from certain aspects of source verification, which enables the marketing of raw materials (e.g., appellation d'origine controlee, prosciutto di Parma). [Moe98, Pet01]

However, the food chain is not the only supply chain that can profit from cradle-to-grave supply chain monitoring. [Wel09] First, because important stakeholder groups hold companies responsible for social impacts (e.g., child labor, corruption, discrimination) and environmental impacts (e.g., pollution [Leo70]) in their product chain (e.g., batteries [GST+08], window frames [KvdB97]). [HDJ08] Second, because supply chain intrusions such as counterfeit negatively affect our economy. [Dek07,Low06] Such cradle-to-grave, conscious- or eco-design [HJA05, ZKLH97] and anti-counterfeit approaches to supply chain monitoring [SB07] could also benefit from electronic data interchange, registering the future paths of products (e.g., disposal or recycling [CP96]). Additionally, the registration of the money flows that remunerate product flows could also help to further increase the application of tracking and tracing (e.g., to prevent criminal and terrorist organizations from financing their activities with counterfeit).

Tracking and tracing has been implemented in various ways, using a range of technologies. Some techniques are limited to the identification of a product's origin (e.g., tracing the geographic origin of honey [SÜC+10] or beef [BSL+10]), others also identify the product's path. Some of these approaches are limited to in-house traceability in production plants (e.g., gozinto graphs [JVvDB03]). Other approaches span parts of or whole supply chains from raw material to consumer, including production, transportation, packing, distribution and processing. [Moe98, RGSR10] Many of these implementations use database and internet tech-

nology for monitoring transport processes (e.g., bovine traceability [Hou01], international animal traceability [MW01]), production processes (e.g., meat processing [MSF+05, MSLF02]), and entire supply chain management systems (e.g., location tracking [GJH+10]). Also a range of technologies (e.g., RFID [JCHC+05]) has been used to tag individuals products and batches (e.g., for traceability of aquatic animals [HHBL01]).

Next to the commercial advantage that can be gained from source verification (e.g., from marketing raw materials [Moe98] and quality assurance [LMR98, VV98]), commercial advantage can also be created by tracking and managing business transactions with customers and suppliers (e.g., for customer relationship management [GV05]) and managing supply chain designs (e.g., global supply chain optimization [BG10]). Apart from the business intelligence that can be created by monitoring transactions with customers and suppliers, also business process intelligence, which is created by monitoring the own production process, can generate competitive advantage. [GCC+04] McCormack [McC01] shows that when business processes are designed to support the overall supply chain, the overall performance of an organization improves. However, the information to implement such supply chain supporting business processes is scattered over various information systems (e.g., in-house tracking and tracing systems and tracking and tracing systems for the supply chain).

Therefore, what is needed is a tracking and tracing reference model that can improve the information flow with partners inside and outside the enterprise from both the operational and planning perspective. [Rab03] Such an improved information flow could be achieved through enterprise system consolidation, providing a unified view of business processes and functions performed by partners involved. [MJB06] Mitigating the risk of under- or over-specifying such as unified view (e.g., creating a solution that can only track car parts, creating a solution that also tracks address changes of employees) we derive the reference model from an existing conceptual model for intra- and inter-enterprise systems, which already captures an (implicit) consensus regarding the representation of the domain and has established soundness. [KS06]

This conceptual model is REA [McC82], which provides the scientific basis for the ISO-standardized open-edi business transaction ontology (OeBTO). [ISO07] Additionally, the REA ontology [GM02], which was originally developed as a generalized accounting framework based on the ideas of semantic data modeling [Che76] in which accountants and non-accountants share data about the same set of business phenomena [McC82], has been used for modeling production processes [Hru06], supply chain management and e-collaboration systems [HM00], enterprise information systems [BS08, DCH05], and management information systems [CS08]. Moreover, previous research has shown that REA can support the integration of business processes across enterprise boundaries. [GLP08] There-

fore, REA cannot be considered an accounting-only ontology. However, because of its accounting roots [GM86, HDC99, McC79], REA incorporates the accounting discipline's more than 500 years of practical experience in recording business transactions. REA is, to the best of our knowledge, the only conceptual model that supports at the same time the registration of past, current and future (e.g., accounts receivable) money flows, next to the registration of product flows, and it does so for flows both within and between enterprises.

Next to building a reference model for tracking and tracing, this paper presents a prototype application that is based on the model and that is used, in a Design Science tradition [HMJR04], to evaluate the model. Since the reference model presents a new kind of data model, it cannot be compared with existing models used for the same purposes, nor can existing data logs be used to prove its utility. As implementing an information system that supports both the overall supply chain and the individual business processes of the supply chain partners involves many challenges (e.g., the protection of strategic information and privacy [GAH01, JCHC+05, JCJ08, LB09a, SXZ08]) which would distract the attention from the contribution this paper aims to make, descriptive scenarios that abstract from these challenges are used to evaluate the reference model. These scenarios, which are integrated in an example case , demonstrate the utility of the reference model as implemented in the prototype application. [FFFvL97, MS95a] The descriptive scenarios suggest a reality check of the proposed reference model by providing a comprehensive and concise representation of the problem at a sufficient level of complexity. As our research artifact is novel, the descriptive scenario's provide the highest level of evaluation achievable at this stage of development. [HMJR04] The prototype application demonstrates that the proposed reference model can be used to represent both production processes and transactions between trading partners, while abstracting from issues such as privacy protection.

Section 4.2 presents the new reference model for tracking and tracing and explains how it was developed from REA. Section 4.3 presents the prototype application and section 4.4 compares our proposal with related research and discusses its differences with the original REA model. Finally, Section 4.5 presents conclusions, limitations and directions for future research.

4.2 Reference Model for Tracking and Tracing

The conceptual model that we propose for recordings inter- and intra-enterprise phenomena is shown in Figure 4.1, where it is represented as a class diagram. Many of the concepts and relations in the model are taken from the REA ontology. In this section, we present these concepts and relations and explain how they were used to build a reference model for tracking and tracing future, past, and present money and product flows.

The economic agent class is used for representing natural persons that act on behalf of legal persons, which are represented themselves using the organizational unit class. The concept of organizational unit refers to the passive role of a person as an owner or possessor of economic resources (confer infra). This means that organizational units have economic control over resources (i.e., ownership of the right to derive economic benefit from a resource), which entails the discretionary power to use or dispose of these resources via economic events (confer infra) in a legal way. The economic agent construct, on the other hand, represents the active role of a person as a performer of economic events. Organizational units represent the entities that experience the effect of events, whereas agents represent the entities that engage in events (e.g., an employee performs an event that affects his employer's resources). So agents may have access to resources of which they are not the owner (i.e., having custody but not economic control over the resources), which means that in that case the agents act on behalf of organizational units. For example, an employee is an agent for its employer (i.e., the employee performs tasks from which the employer reaps the full benefits).

The economic resource class represents objects (e.g., rights, goods and services) that are scarce, have utility and are under the control of an organizational unit (e.g., enterprise, household) [Iji75, McC82]. The scarceness indicates that not every organizational unit can control such resources at a certain point in time and indicates that for some organizational units trade is required to gain control over particular resources. The utility motivates why certain organizational units want to gain control over particular resources. The economic event class represents the events (e.g., produce, exchange, consume, distribute) that affect economic resources in the sense that they result in changes (i.e., increases and decreases) of resource stocks. [Yu76] The recognition of both sides of an economic event (i.e., increment event for a resource increase and decrement event for a resource decrease) is a unique feature of the reference model for tracking and tracing presented here, which acknowledges that a single event (e.g., a product shipment) can be perceived both as an increment event (i.e., receive product) and a decrement event (i.e., dispatch product) by different trading partners (e.g., increment for the buyer and decrement for the seller). As will be explained further on in the paper, this two-sided view of economic events is a key element in our solution for modeling inter- and intra-enterprise phenomena that enables tracking and tracing.

Not explicitly present in the REA ontology is the transaction view class, which we use to aggregate event perceptions in order to satisfy the REA ontology axiom[1] that requires every increment event to be eventually paired with one or more decrement events, and vice versa, to represent equitable trades (e.g., receiving a good is

[1]The REA ontology axioms stem from basic laws of business (e.g., economic reciprocity) and describe the invariant conditions of the business domain as well as prescribe the permissible range of variation in conceptual models for that business domain. [FM07]

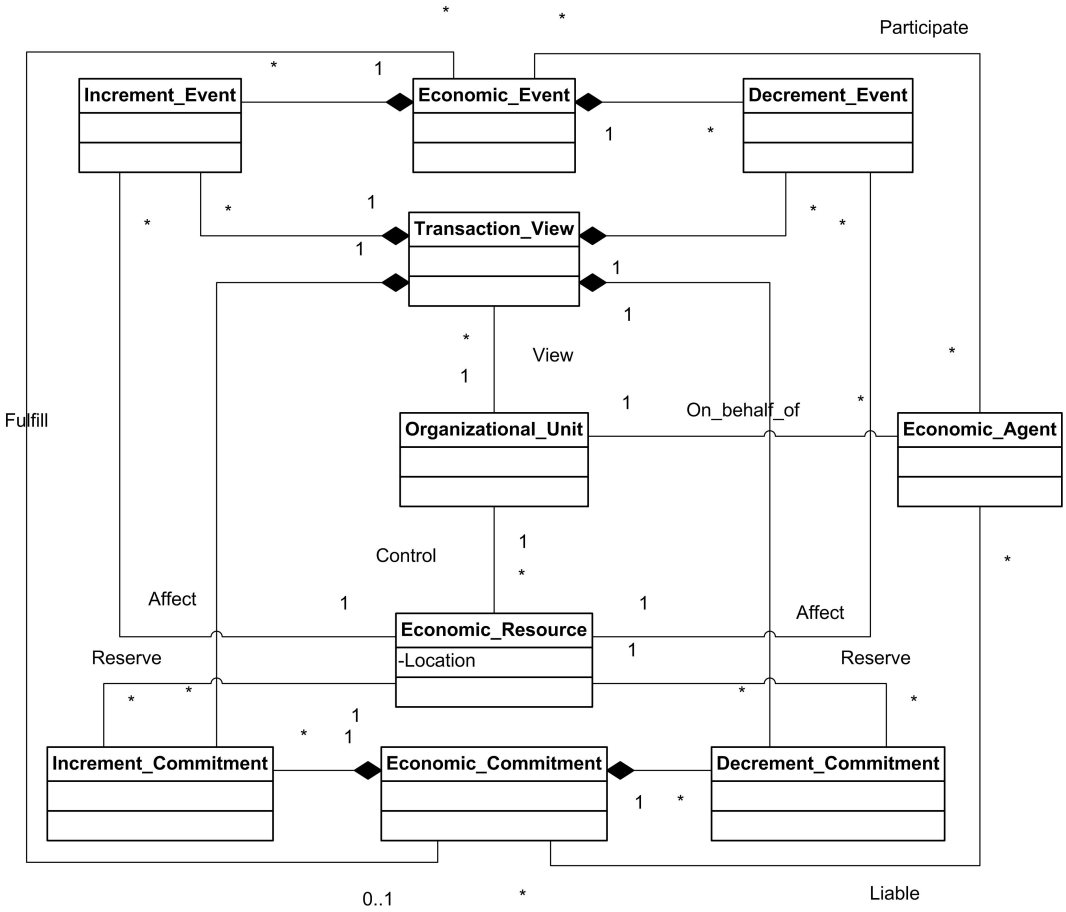

Figure 4.1: Tracking and tracing conceptual data model structure

balanced with a payment). [GM04, Iji75] For instance, in market transactions, this economic reciprocity dictates that when a company sells products to a customer (i.e., an economic event that decreases the value of the company's inventory of products), a requiting event like a payment or delivery of equally valued goods (e.g., as in barter trade) by the customer must follow, meaning that there is a dual economic event that balances the decrease in value caused by the sale.

Next to economic events, the REA ontology addresses commitments, which are represented by the economic commitment class and represent the promise to perform economic events in the future. Consequently, the conceptual model also addresses future economic events, next to past and current events. Since commitments represent future events the commitment structure replicates the event structure. Like an event, a single commitment (e.g., a buyer promises to pay a seller within 30 days after goods delivery) can be perceived as an increment by one trading partner (e.g., by the seller as it adds to accounts receivable) and a decrement by another trading partner (e.g., by the buyer as it adds to accounts payable). Like agents can participate in economic events, they can also be liable for commitments. Eventually, a commitment will be fulfilled by one or more events. A transaction view taken by an organizational unit may also aggregate reciprocal commitment perceptions (i.e., balancing increment and decrement commitments), just as it aggregates dual event perceptions.

Figure 4.2 shows an example where the model is used to represent the exchange of pizza for money by means of an object diagram that instantiates the class diagram in Figure 4.1. The example shows how the model can be used to record transactions between the organizational units Pizza Luigi and John. The pizza transfer is perceived as a decrement by Pizza Luigi and as an increment by John, while the transfer event is participated in by the agents Luigi and John (which can be seen as acting on behalf of himself). The remunerating money transfer, on the other hand, is perceived as an increment by Pizza Luigi and as a decrement by John. Like the pizza transfer, the money transfer event is participated in by Luigi and John. The transaction then connects the opposing transfers. From Pizza Luigi's point of view a pizza decrement is paired in duality with a money increment, while from John's point of view a pizza increment is paired in duality with a money decrement.

An interesting feature of the model in Figure 4.1 is that it cannot only be used to model transfers, but also transformations (e.g., production processes) and transportations (e.g., distribution processes). Figure 4.3, for instance, shows the process of baking pizza, which converts ingredients to pizza. The model represents the consumption of pizza ingredients (i.e., yeast, water, flour, hamburger meat and tomatoes) as decrement event objects that correspond to the stock decreases caused by the pizza production event as perceived by Pizza Luigi. The production of the pizza, on the other hand, is perceived as a stock increase by Pizza Luigi. Conse-

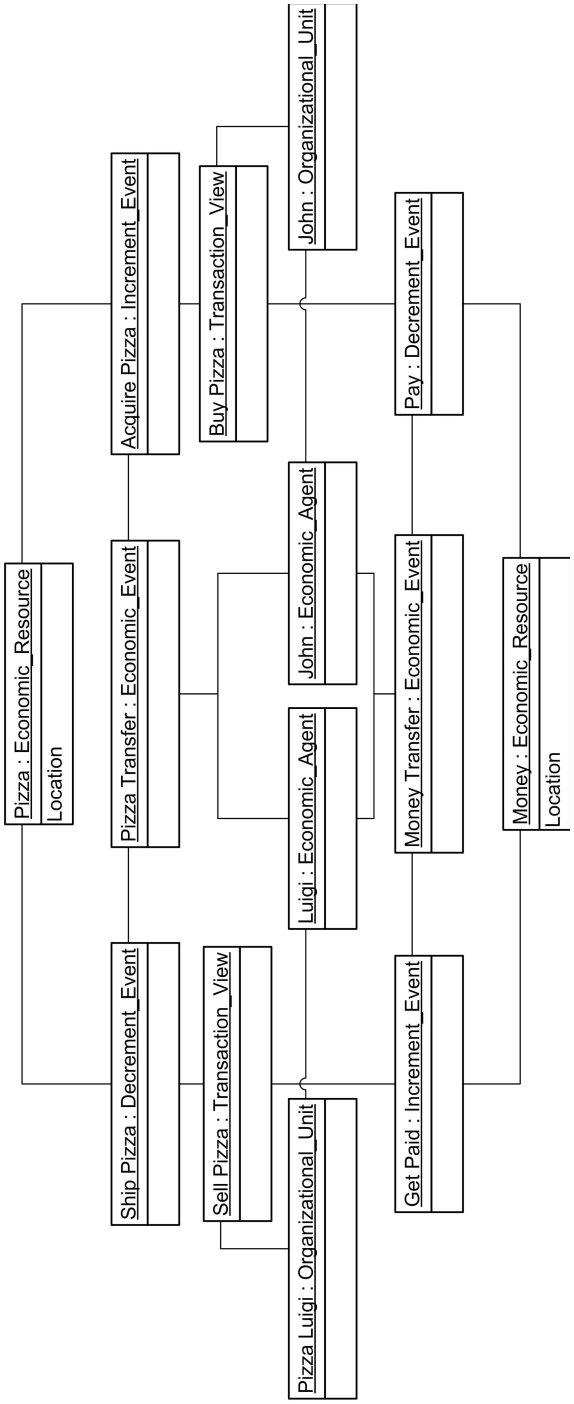

Figure 4.2: Exchange model with two mirrored transfer events

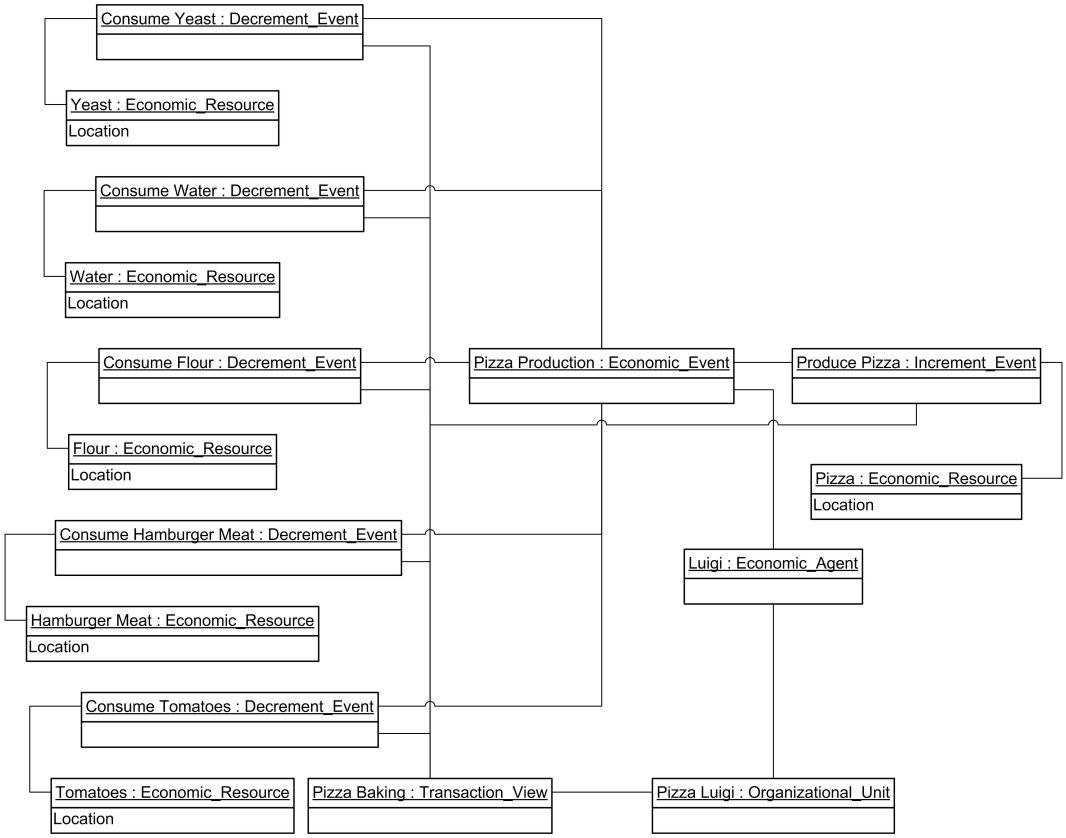

Figure 4.3: Conversion model with one transformation event

quently, the production is modeled as an increment event object. Figure 4.3 also shows that the pizza production event is performed by Luigi and that the pizza baking transaction causes both increases and decreases of Pizza Luigi's stocks. This means that a same economic event can also be perceived as both increment and decrement by the same organizational unit.

Subsequently, Figure 4.4 shows the transport of a pizza as part of its conversion process. The model represents the removal of the pizza from one location and the delivery at another location.

To complete the example, Figure 4.5 shows the fulfillment of a commitment to make bread. Figure 4.5 reveals that the commitment to make bread is fulfilled by two subsequent events (i.e., make dough, bake bread). The commitment to make bread is represented as the committed conversion of ingredients in bread. It is

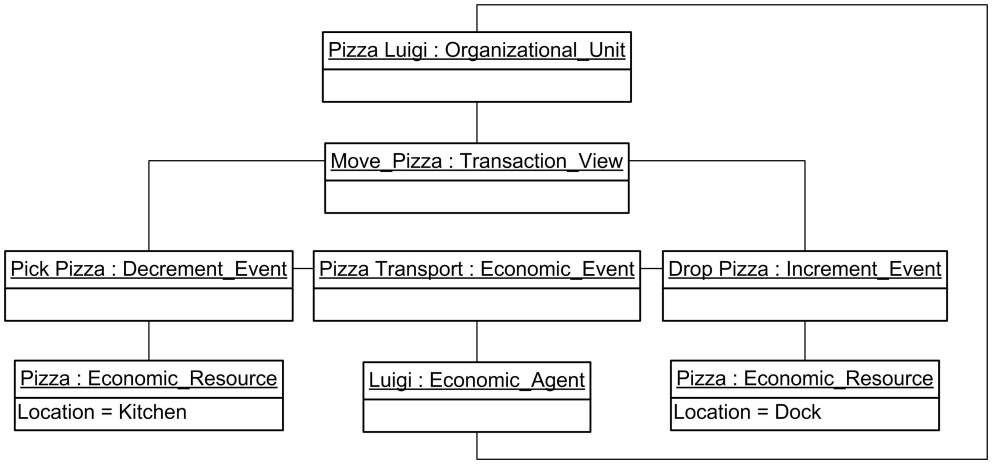

Figure 4.4: Conversion model with one transportation event

fulfilled by the process of converting ingredients into dough and dough into baked bread. Like the pizza production event in the other transformation template (Figure 4.3), the events and commitments belong to the same conversion cycle, which is represented as a transaction view.

The preceding object models illustrate how the model of Figure 4.1 can be used to register transactions between business partners, conversion processes and commitments. Next to representing the fulfillment of commitments, Figure 4.5 also shows subsequent events (i.e., make dough precedes bake bread). Subsequent events share the organizational unit that perceives them and the resource they affect. More precisely, the preceding event increases the resource stock, the following event decreases the same resource stock. In Figure 4.5 the make dough event is perceived as an increment to the dough stock in the kitchen by the baker Chet (i.e., produce dough), while the bake bread event is perceived as a decrement to the dough stock in the kitchen by baker Chet (i.e., consume dough).

As the exchange and conversion models allow us to identify the resource stocks that are affected by transfer, transformation and transportation events and subsequent events can be identified through their perception (i.e., increment or decrement) by the organizational units that control these resources, event chains can be constructed that span multiple organizational units. Such chains can then be used to track and trace product and money flows through transfers, transportations and transformations. The event notion enables the registration of past and current product and money flows, while the commitment notion enables the registration of future product and money flows. Finally, REA's fulfillment notion maps what

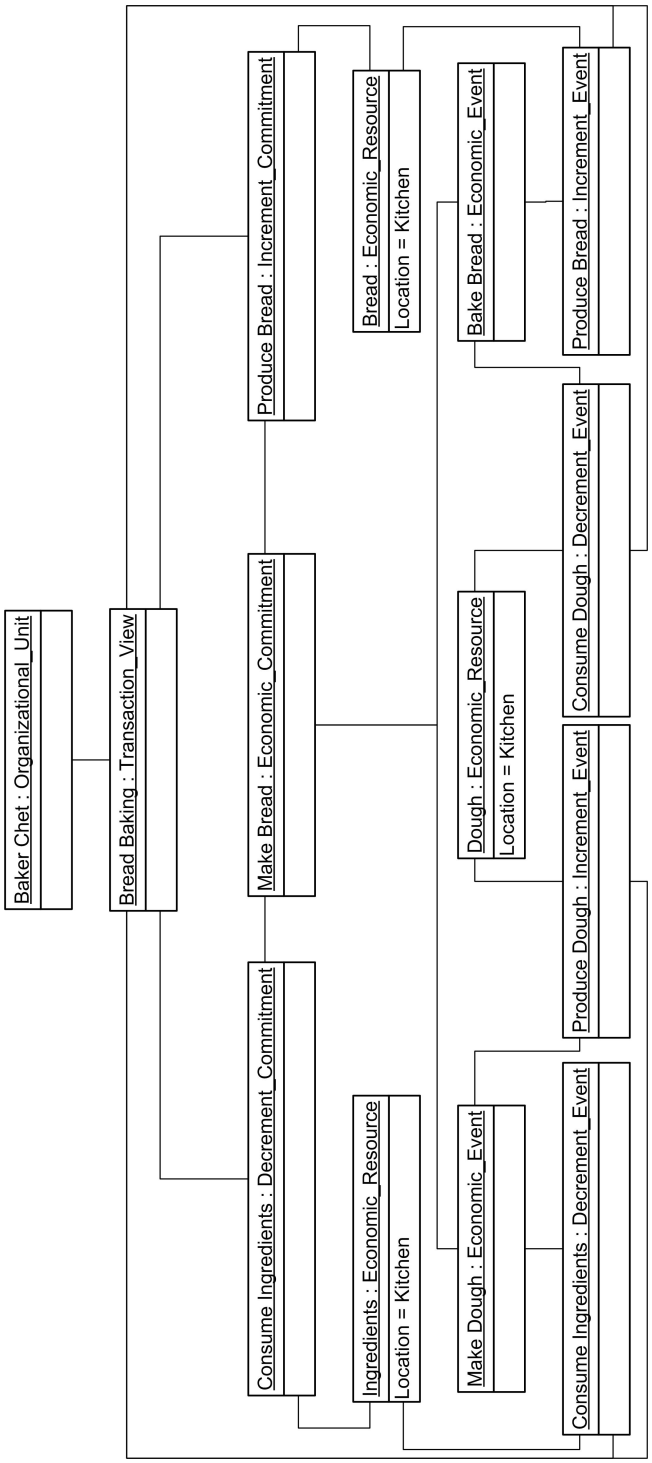

Figure 4.5: Bread baking commitment

happened in reality to what was originally committed.

4.3 Prototype Application: Baking Bread and Pizza, From Farm to Customer

In this section, we discuss a prototype application that illustrates the possibilities for tracking and tracing offered by the model presented in the previous section (Figure 4.1). The application concerns a food supply chain (Figure 4.6) that starts with cattle and corn farmers and ends with consumers that buy and consume pizza and bread. The supply chain also contains a flour mill, which transforms wheat into flour, a grinder, which transforms corn and grain into cattle feed, a butcher, which transforms cattle into hamburger meat, a grocer, which provides groceries to a baker and a pizza restaurant, a baker, which transforms flour and other ingredients into bread that is sold to customers, and a pizza restaurant, which converts flour, hamburger meat and other ingredients into pizza that is sold to customers. The data model of the application, constructed using our reference model, is shown in Figure 4.7. A Microsoft Access database that is based on this data model and is populated with example registrations is available on-line (`http://www.managementinformatics.ugent.be/Traceability.accdb`).

Figure 4.8 shows some example event summaries that were extracted from the application's database by joining and querying the Event, IncrementEvent, and DecrementEvent tables. The event data in the provider, from, and input columns originate in the DecrementEvent table (the foreign keys UnitName, Location, and ResourceId). The event data in the output, to, and recipient columns originate in the IncrementEvent table (the foreign keys ResourceId, Location, and UnitName).

With the data contained in an event summary, we can distinguish transfer, transformation, and transportation events. Transfers are events of which the increment side is perceived by one organizational unit (i.e., recipient) and the decrement side is perceived by another organizational unit (i.e., provider), while the resource that is affected by the event does not change form or substance (e.g., event25). [GM04] Transformations, on the other hand, are events of which the increment and decrement side is perceived by the same organizational unit, while the resources that are affected by the increment (i.e., output) and decrement side (i.e., input) differ (i.e., change form or substance) (e.g., event01). Finally, transportations are events of which the increment and decrement side is perceived by the same organizational unit and, while the resource that is affected changes location (i.e., 'from'is different from 'to') but not form or substance (e.g., event00). The event summaries thus indicate resource ownership changes with provider-recipient semantics, resource location changes with from-to semantics and resource form or substance changes with input-output semantics. Hence, the optional EventType at-

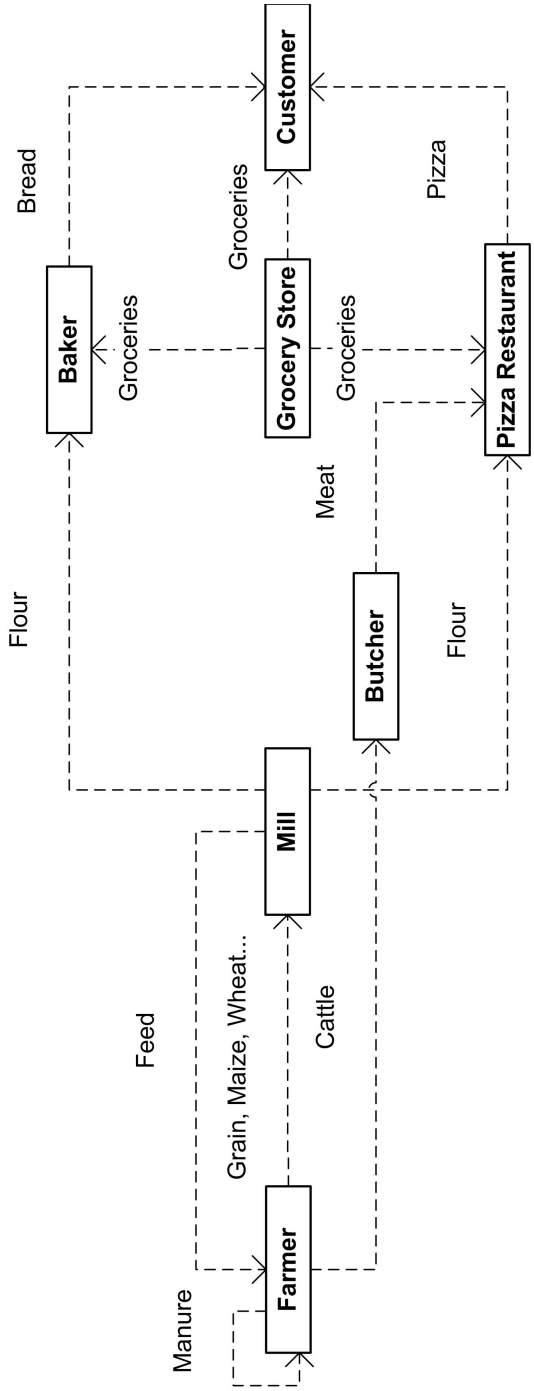

Figure 4.6: Example food supply chain

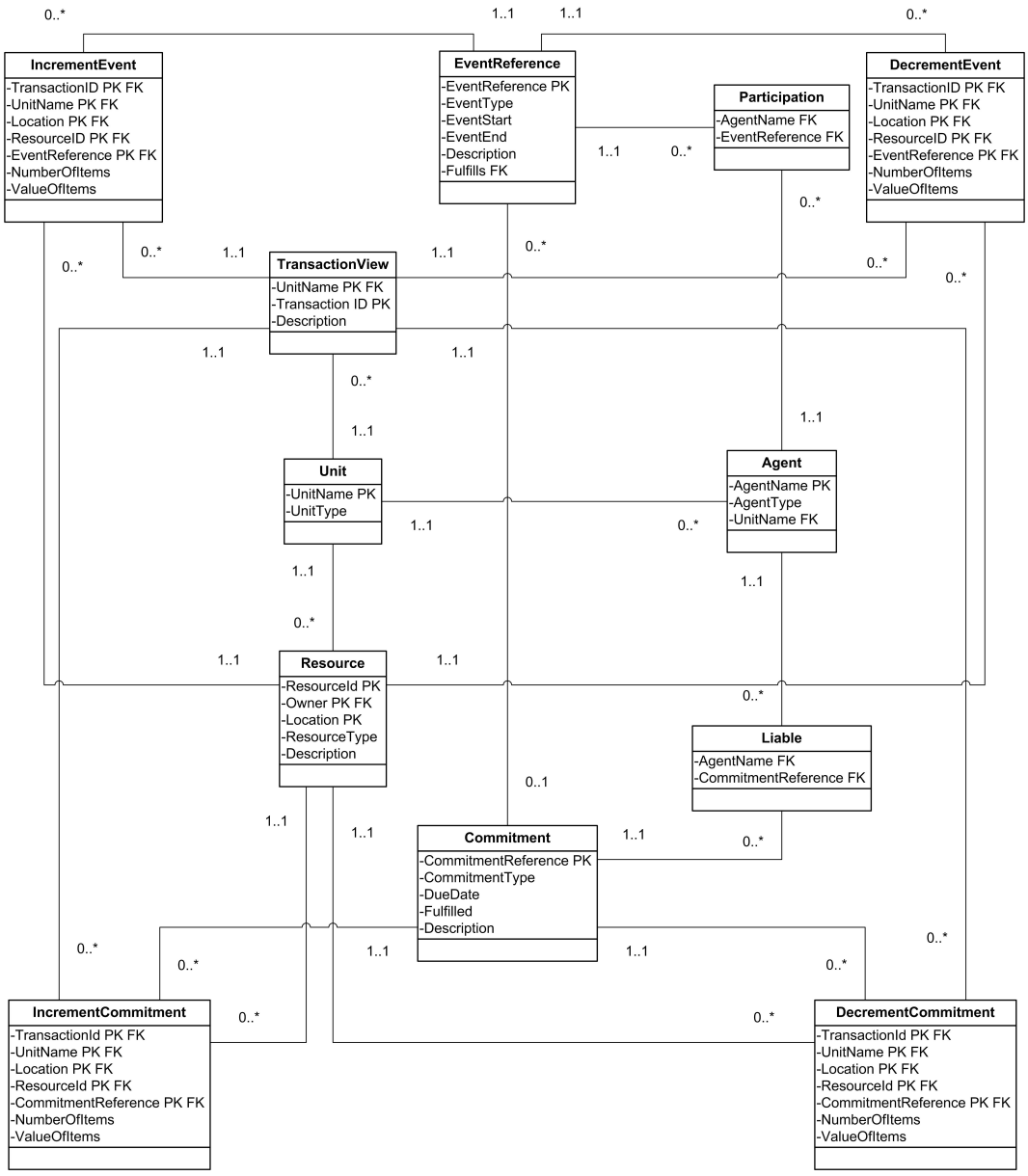

Figure 4.7: Application data model for traceability

EventRefer	Provider	From	Input	OutPut	To	Recipient	EventType
Event00	Grain Farmer	Dunghill	Manure	Manure	Field	Grain Farmer	Transportation
Event01	Grain Farmer	Field	Manure	Grain	Field	Grain Farmer	Transformation
Event09	Flour Mill	Warehouse	Wheat	Lot 1	Warehou	Flour Mill	Transformation
Event15	Butcher Pete	Cowshed	Daisy	Lot B	Fridge	Butcher Pete	Transformation
Event15	Butcher Pete	Cowshed	Daisy	Lot D	Warehou	Butcher Pete	Transformation
Event16	Baker Chet	Warehouse	Lot 1	Lot A1	Warehou	Baker Chet	Transformation
Event16	Baker Chet	Well	Water	Lot A1	Warehou	Baker Chet	Transformation
Event16	Baker Chet	Warehouse	Yeast	Lot A1	Warehou	Baker Chet	Transformation
Event17	Pizza Luigi	Warehouse	Yeast	Lot B1	Kitchen	Pizza Luigi	Transformation
Event17	Pizza Luigi	Warehouse	Lot 1	Lot B1	Kitchen	Pizza Luigi	Transformation
Event17	Pizza Luigi	Fridge	Lot A	Lot B1	Kitchen	Pizza Luigi	Transformation
Event17	Pizza Luigi	Warehouse	Tomatoes	Lot B1	Kitchen	Pizza Luigi	Transformation
Event17	Pizza Luigi	Well	Water	Lot B1	Kitchen	Pizza Luigi	Transformation
Event18	Tom	Kitchen	Lot A1	Utility	Home	Tom	Transformation
Event19	Dick	Kitchen	Lot A1	Utility	Home	Dick	Transformation
Event21	Harry	Kitchen	Lot B1	Utility	Home	Harry	Transformation
Event22	Cattle Farmer	Dunghill	Fresh Man	Fresh Man	Dunghill	Grain Farmer	Transfer
Event22M	Grain Farmer	Bank	Money	Money	Bank	Cattle Farmer	Transfer
Event25	Grain Farmer	Field	Grain	Grain	Field	Grinder's	Transfer
Event33	Flour Mill	Warehouse	Lot 1	Lot 1	Warehou	Baker Chet	Transfer
Event35	Flour Mill	Warehouse	Lot 1	Lot 1	Warehou	Pizza Luigi	Transfer
Event41	Baker Chet	Warehouse	Lot A1	Lot A1	Kitchen	Tom	Transfer
Event42	Baker Chet	Warehouse	Lot A1	Lot A1	Kitchen	Tom	Transfer
Event43	Baker Chet	Warehouse	Lot A1	Lot A1	Kitchen	Dick	Transfer
Event44	Pizza Luigi	Kitchen	Lot B1	Lot B1	Kitchen	Dick	Transfer
Event45	Pizza Luigi	Kitchen	Lot B1	Lot B1	Kitchen	Harry	Transfer

Figure 4.8: Event summaries

tribute of the Event entity type in the data model (Figure 4.7) is a derived attribute, which for the sake of clarity was added to the event summary.

Another database view that can be taken shows transaction summaries (Figure 4.9). This view is obtained by joining and querying the TransactionView, IncrementEvent, DecrementEvent, IncrementCommitment, and DecrementCommitment tables. The transaction identification and unit name are found in the TransactionView table. For each transaction view (i.e. unique combination of transaction identification and unit name), all related event or commitment records, in respectively the IncrementEvent, DecrementEvent, and IncrementCommitment, DecrementCommitment tables are selected. For each of these records, the event or commitment reference is included in the transaction summary, as well as the number and value of the items involved in the event or commitment. Also the affected resources are identified, as outflows for decrement events or commitments and as inflows for increment events and commitments.

If the transaction is an exchange, then the transaction is perceived by trading partners with opposing views. In Figure 4.9, we see that there are two views of transaction 22. For the cattle farmer, transaction 22 consists of a fresh manure outflow and a money inflow, where it consists of a fresh manure inflow and a money outflow for the grain farmer. Although these two transaction views share the same events (i.e., event22 and event22M), these events are perceived differently by the trading partners (i.e., the money inflow for the cattle farmer and the money outflow for the grain farmer are the same transfer event (i.e., event22M) and the fresh manure inflow for the grain farmer and the fresh manure outflow for the cattle farmer are also recognized as the same transfer event (i.e., event22)). If the transaction is a conversion, then there is only one transaction view, which shows the perception of the single organizational unit performing the conversion and the resource inflows and outflows that are involved in the conversion process. For example, transaction 17 shows that lot B1 is produced by consuming water, yeast, tomatoes, lot A and lot 1.

The event and transaction summaries provide the basic elements to construct product and money tracks and traces and to reveal their mutual dependency. The event summaries provide the information to construct product and money tracks and traces, whereas the transaction summaries show the dependency between product and money flows. Figure 4.10 is a view that shows the production history and origin of products, so can be used to construct resource traces. It shows, for instance, that lot BB originates in grain and maize, which both originate in manure. To construct a resource trace, as shown in Figure 4.10, we identify all inputs of the event(s) that produced the resource. For each of these inputs, we repeat the process and identify the inputs that lead to their production. This process is repeated until the desired length of the resource trace is achieved. To construct a resource track, we identify all outputs for which a particular resource was an input. For each of

Transaction	UnitName	EventRefere	Commitm	Outflow	Inflow	NumberO	ValueOfIt
Transaction17	Pizza Luigi	Event17			Lot B1	2	100
Transaction17	Pizza Luigi	Event17		Yeast		50	25
Transaction17	Pizza Luigi	Event17		Water		1	0
Transaction17	Pizza Luigi	Event17		Tomatoes		5	10
Transaction17	Pizza Luigi	Event17		Lot A		1	20
Transaction17	Pizza Luigi	Event17		Lot 1		2	4
Transaction22	Cattle Farmer	Event22		Fresh Manure		20	20
Transaction22	Cattle Farmer	Event22M			Money	0	20
Transaction22	Grain Farmer	Event22			Fresh Manure	20	20
Transaction22	Grain Farmer	Event22M		Money		0	20

Figure 4.9: Transaction summaries

TraceRoot	T1	T2	T3	T4	T5
Lot AA	Grain	Manure			
Lot AA	Maize	Manure			
Lot B	Daisy	Lot AA	Grain	Manure	
Lot B	Daisy	Lot AA	Maize	Manure	
Lot B	Daisy	Lot BB	Grain	Manure	
Lot B	Daisy	Lot BB	Maize	Manure	
Lot B1	Lot 1	Wheat	Manure		
Lot B1	Lot A	Bella	Lot AA	Grain	Manure
Lot B1	Lot A	Bella	Lot AA	Maize	Manure
Lot B1	Lot A	Bella	Lot BB	Grain	Manure
Lot B1	Lot A	Bella	Lot BB	Maize	Manure
Lot B1	Tomatoes				
Lot B1	Water				
Lot B1	Yeast				
Lot BB	Grain	Manure			
Lot BB	Maize	Manure			
Lot C	Bella	Lot AA	Grain	Manure	
Lot C	Bella	Lot AA	Maize	Manure	
Lot C	Bella	Lot BB	Grain	Manure	
Lot C	Bella	Lot BB	Maize	Manure	
Lot D	Daisy	Lot AA	Grain	Manure	
Lot D	Daisy	Lot AA	Maize	Manure	
Lot D	Daisy	Lot BB	Grain	Manure	
Lot D	Daisy	Lot BB	Maize	Manure	

Figure 4.10: Resource trace

these outputs we repeat this process and identify the outputs for which they were inputs. This process is repeated until the desired length of the resource track is achieved. As resources do not change form or substance during transfer and transportation events, we can abstract from these kinds of events for the construction of resource tracks and traces.

Furthermore, event tracks and traces, which show the sequence of events that lead to the creation of a product, can be constructed. Two sequential events are identified by the fact that they affect the same resource at the same location, controlled by the same organizational unit. The former event is recognized as an increment (e.g., produce, acquire) to the affected resource, the latter event is recognized as a decrement (e.g., consume, dispatch) to the affected resource. Since every event is perceived as increment and decrement by the same organizational

Start	C1	C2	C3	C4
Event07	Event22			
Event07	Event23			
Event07				
Event08				
Event08	Event22			
Event08	Event23			
Event09	Event33	Event16	Event42	Event18
Event09	Event33	Event16	Event41	Event18
Event09	Event33	Event16	Event43	Event19
Event09	Event35	Event17	Event45	Event21
Event09	Event35	Event17	Event44	
Event10	Event34			

Figure 4.11: Event track

unit (in case of transformations and transportations) or different organizational units (in case of transfers), preceding and following events can be identified and added to the event chain until it attains the desired length. Event traces identify the chain of events (i.e., transfer, transformation and transportations) that preceded a particular event, whereas event tracks identify the chain of events that followed a certain event. Figure 4.11 shows such an event track. It shows that event 11 is not followed by any event, where event 9 is followed by events 33, 35, 17, 16, 41, 42, 43, 44, 45, 18, 19 and 21. Events 9, 16, 17, 18, 19 and 21 represent transformation events (see the event summaries in Figure 4.8), where events 33, 35, 41, 42, 43, 44 and 45 represent transfer events (see the event summaries in Figure 4.8). Together, these events create 5 different paths that originate in event 9. Similar to event tracks and traces for product flows, event tracks and traces for money flows can be constructed. Since money does not change form or substance, the construction of resource tracks and traces for money is superfluous.

Event tracks allow producers to follow their products throughout the supply chain, which enables the discrimination of proper and improper product use. For example, the event track that starts with event 9 allows the flour mill to follow the use of its flour. Event 9, which represents the production of lot 1 (i.e., flour) from wheat by the flour mill, is followed by events 35 and 33. Event 33, which represents transferring part of lot 1 from the flour mill to Baker Chet (see the event summaries in Figure 4.8), is followed by event 16. Event 35, which represents transferring part of lot 1 from the flour mill to Pizza Luigi (see the event summaries in Figure 4.8), is followed by event 17. Event 16, which represents the production

of lot A1 (i.e., bread) by Baker Chet (see the event summaries in Figure 4.8), is followed by events 41, 42 and 43. Event 17, which represents the production of lot B1 (i.e., pizza) by Pizza Luigi (see the event summaries in Figure 4.8), is followed by events 44 and 45. Consequently, the flour from lot 1 has been used to produce pizza and bread. Events 41, 42 and 43 represent selling bread from lot A1 to Tom, Tom and Dick respectively (see the event summaries in Figure 4.8). Events 44 and 45 represent selling pizza from lot B1 to Dick and Harry respectively (see the event summaries in Figure 4.8). Events 18 and 19 then show the consumption of bread by Tom and Dick (see the event summaries in Figure 4.8), where event 21 shows the consumption of pizza by and Harry (see the event summaries in Figure 4.8). Since event 44 has not been followed by a consuming event, Dick did not consume his pizza yet. If, for example, Harry is diagnosed with a food borne illness, and the origin is traced back to the flour mill, Dick can be warned not to consume his pizza.

Where a flour contamination can also be retrieved and treated with resource traces and tracks, the more advanced event traceability provide more information about the operations that were performed on a product. Such information might be crucial when proper product treatment is crucial to guarantee product quality. For example, in a cold chain, food is guaranteed to preserve its quality when the chain is not broken (i.e., when the temperature never exceeds a certain level). Meat is a product that requires a cold chain to preserve its quality. The event summary of event15 in Figure 4.8 shows that lot D (i.e., steak, see the Resource table in the database) is stored in the warehouse instead of the fridge. Consequently, the cold chain is broken and the quality of lot D cannot be guaranteed. From the participation table in the database we can derive that Jim can be held accountable for this 'mistake'.

Since commitments mirror events, also commitment tracks and traces can be constructed. Two sequential commitments are identified by the fact that they relate to the same resource at the same location, controlled by the same organizational unit. The former commitment is recognized as a future resource increment (e.g., produce, acquire), the latter commitment is recognized as a future resource decrement (e.g., consume, dispatch). Since every commitment is perceived as increment and decrement by the same organizational unit (for transformation and transportation commitments) or different organizational units (for transfer commitments), preceding and following commitments can be identified and added to the commitment chain until it attains the desired length. Commitment traces identify the chain of commitments (i.e., transfer, transformation and transportations) that preceded a particular commitment, whereas commitment tracks identify the chain of commitments that followed a certain commitment.

Figure 4.12 shows an example of a commitment trace that allows supply chain partners to identify the critical path that precedes the fulfillment of commitments.

For example, commitment 16 can only be fulfilled when commitment 3 and 11 have been fulfilled. When these commitment tracks and traces are combined with the commitment summaries (Figure 4.13), which are constructed similarly to the event summaries (Figure 4.8), we can identify the economic resources and organizational units involved. For example, commitment 16 represents Butcher Pete's commitment to sell hamburger meat (i.e., lot H) to Pizza Luigi. Commitment 16 can only be fulfilled when commitment 3, which represents Butcher Pete's commitment to slaughter a cow (i.e., Betsy) and convert it into hamburger meat (i.e., lot H) and steak (i.e., lot G), has been fulfilled. However, commitment 3 can only be fulfilled on the condition that commitment 11, which is the transfer of Betsy from the cattle farmer to Butcher Pete, has been fulfilled. When Betsy is diagnosed with BSE, she will be removed from the food chain and commitment 11 cannot be fulfilled. Consequently, Pizza Luigi will not be able to collect lot H. To ensure Pizza Luigi's operations, alternative lots of hamburger meat will need to be identified. Such an alternative is lot F, which originates from Patsy and has not been committed to any transfer yet.

Similar to the event and commitment tracks and traces for product flows, presented above, event and commitment tracks and traces for money flows can be represented. With budgets as equivalents of lots, money can be partitioned to facilitate traceability and limit contamination (e.g., criminal money). Additionally, event tracks facilitate Paulian or revocatory action, which enables creditors to reclaim their goods or money from a third party that is a trading partner of their debtor. Furthermore, commitment tracks and traces for money flows can facilitate estimating the effect of a bankruptcy (e.g., Who are the creditors and debtors and how will they be affected?)

4.4 Related work

That REA is the foundation for the reference model for tracking and tracing does not mean that it is the reference model. In this section we clarify the differences between our model and REA and discuss the research contribution that we make to the academic field that studies conceptual modeling of economic phenomena, primarily with the aim of providing a (better) conceptual basis for the development of inter- and intra-enterprise systems. This section does not only compare the proposed model with REA, but also with the two existing tracking and tracing reference models, that we identified in a literature search.

4.4.1 Tracking and Tracing Models

Figure 4.14 shows a reference data model for traceability in manufacturing, which is designed to monitor product quality throughout the production processes per-

HistoryRoot	H1	H2	H3	H4
Commitment01				
Commitment02	Commitment10			
Commitment03	Commitment11			
Commitment04				
Commitment04	Commitment14			
Commitment04	Commitment15			
Commitment05				
Commitment05	Commitment16	Commitment03	Commitment11	
Commitment06	Commitment17			
Commitment07				
Commitment08				
Commitment09	Commitment01			
Commitment10				
Commitment11				
Commitment12				
Commitment13				
Commitment14				
Commitment15				
Commitment16	Commitment03	Commitment11		
Commitment17				
Commitment18	Commitment05			
Commitment18	Commitment05	Commitment16	Commitment03	Commitment11

Figure 4.12: Commitment trace

CommitmentRefere	Provider	From	Input	OutPut	To	Recipient	CommitmentType
Commitment01	Grinder's	Warehouse	Maize	Lot CC	Warehouse	Grinder's	Transformation
Commitment01	Grinder's	Field	Grain	Lot CC	Warehouse	Grinder's	Transformation
Commitment02	Butcher Pete	Cowshed	Patty	Lot E	Fridge	Butcher Pete	Transformation
Commitment02	Butcher Pete	Cowshed	Patty	Lot F	Fridge	Butcher Pete	Transformation
Commitment03	Butcher Pete	Cowshed	Betsy	Lot G	Fridge	Butcher Pete	Transformation
Commitment03	Butcher Pete	Cowshed	Betsy	Lot H	Fridge	Butcher Pete	Transformation
Commitment04	Pizza Luigi	Well	Water	Lot B2	Kitchen	Pizza Luigi	Transformation
Commitment04	Pizza Luigi	Warehouse	Yeast	Lot B2	Kitchen	Pizza Luigi	Transformation
Commitment04	Pizza Luigi	Warehouse	Tomatoes	Lot B2	Kitchen	Pizza Luigi	Transformation
Commitment04	Pizza Luigi	Fridge	Lot B	Lot B2	Kitchen	Pizza Luigi	Transformation
Commitment04	Pizza Luigi	Warehouse	Lot 1	Lot B2	Kitchen	Pizza Luigi	Transformation
Commitment05	Pizza Luigi	Fridge	Lot H	Lot B3	Kitchen	Pizza Luigi	Transformation
Commitment05	Pizza Luigi	Warehouse	Tomatoes	Lot B3	Kitchen	Pizza Luigi	Transformation
Commitment05	Pizza Luigi	Well	Water	Lot B3	Kitchen	Pizza Luigi	Transformation
Commitment05	Pizza Luigi	Warehouse	Yeast	Lot B3	Kitchen	Pizza Luigi	Transformation
Commitment05	Pizza Luigi	Warehouse	Lot 2	Lot B3	Kitchen	Pizza Luigi	Transformation
Commitment06	Tom	Kitchen	Lot A1	Utility	Home	Tom	Transformation
Commitment07	Harry	Kitchen	Lot B2	Utility	Home	Harry	Transformation
Commitment08	Grain Farmer	Warehouse	Wheat	Wheat	Warehouse	Flour Mill	Transfer
Commitment08M	Flour Mill	Bank	Money	Money	Bank	Grain Farmer	Transfer
Commitment09	Grinder's	Warehouse	Lot CC	Lot CC	Warehouse	Cattle Farmer	Transfer
Commitment09M	Cattle Farmer	Bank	Money	Money	Bank	Grinder's	Transfer
Commitment10	Cattle Farmer	Cowshed	Patty	Patty	Cowshed	Butcher Pete	Transfer
Commitment10M	Butcher Pete	Bank	Money	Money	Bank	Cattle Farmer	Transfer
Commitment11	Cattle Farmer	Cowshed	Betsy	Betsy	Cowshed	Butcher Pete	Transfer
Commitment11M	Butcher Pete	Bank	Money	Money	Bank	Cattle Farmer	Transfer
Commitment12	Flour Mill	Warehouse	Lot 2	Lot 2	Warehouse	Baker Chet	Transfer
Commitment12M	Baker Chet	Bank	Money	Money	Bank	Flour Mill	Transfer
Commitment13	Flour Mill	Warehouse	Lot 3	Lot 3	Warehouse	Baker Chet	Transfer
Commitment13M	Baker Chet	Bank	Money	Money	Bank	Flour Mill	Transfer
Commitment14	Flour Mill	Warehouse	Lot 1	Lot 1	Warehouse	Pizza Luigi	Tranfer
Commitment14M	Pizza Luigi	Bank	Money	Money	Bank	Flour Mill	Transfer
Commitment15	Butcher Pete	Fridge	Lot B	Lot B	Fridge	Pizza Luigi	Transfer
Commitment15M	Pizza Luigi	Bank	Money	Money	Bank	Butcher Pete	Transfer
Commitment16	Butcher Pete	Fridge	Lot H	Lot H	Fridge	Pizza Luigi	Transfer
Commitment16M	Pizza Luigi	Bank	Money	Money	Bank	Butcher Pete	Transfer
Commitment17	Baker Chet	Warehouse	Lot A1	Lot A1	Kitchen	Tom	Transfer
Commitment17M	Tom	Bank	Money	Money	Bank	Baker Chet	Transfer
Commitment18	Pizza Luigi	Kitchen	Lot B3	Lot B3	Kitchen	Harry	Transfer
Commitment18M	Harry	Bank	Money	Money	Bank	Pizza Luigi	Transfer

Figure 4.13: Commitment summaries

formed by a single organizational unit. [JVvDB03] It includes both the identification of the constituents of a product, lot or batch (e.g., via a bill of materials) and the recording of the manufacturing steps that led to the creation of this product, lot or batch (e.g., a bill of operations).

The material lot or batch and relation classes represent a multi-level bill of materials and lots. The material lot or batch class represents resources, where the relation class establishes the relations between resources produced (i.e., process outputs) and consumed (i.e., process inputs). These relations between process in- and outputs find their origin in actual operations, which are transformations events. Consequently, the implode and explode relations in Figure 4.14 can be considered the equivalent of the increment and decrement event classes in Figure 4.1, where the relation is represented by the economic event class.

Within the scope of production processes, the economic event class in Figure 4.1 is also equivalent with the actual operation class in Figure 4.14. In the manufacturing part of the reference model, the actual operation is attributed with operation variables and operation values, which provide additional information about the actual operations. The item and production order classes, on the other hand, represent the planning part of the model. Production orders are the equivalent of transformation commitments, where items are the resource bundles that are connected to the increment side of the commitment. However, the reference data model in Figure 4.14 is not able to reserve process inputs (i.e., connected to the decrement side of the commitment). Finally, the capacity unit class represents departments that are responsible for actual operations. In our reference model (Figure 4.1) this responsibility is attributed to economic agents and not to capacity units, which may also be locations inside an enterprise or large machines (e.g., steel press).

As the scope of the reference model proposed by Jansen-Vullers et al. [JVvDB03] is limited to the operations inside one enterprise, it cannot be used to represent transactions between trading partners as our reference model does. Consequently, it can only be used to monitor production processes and logistic flows inside one enterprise and cannot be used to monitor supply chains, which span multiple enterprises.

Bechini et al. [BCMT08] provide a traceability model (Figure 4.15) that supports product integration, division, alteration, transformation, movement, acquisition and providing events. Their product integrations, divisions, alterations and transformations describe configurations of the transformation events represented in Figure 4.3. Their product movements map to the transportations events in Figure 4.4 and their acquisition and providing events map to the transfer events that are part of the exchange model in Figure 4.2. Additionally, their lot and activity classes respectively map to the economic resource and economic event class in Figure 4.1. Furthermore, their responsible actor class maps to the organizational

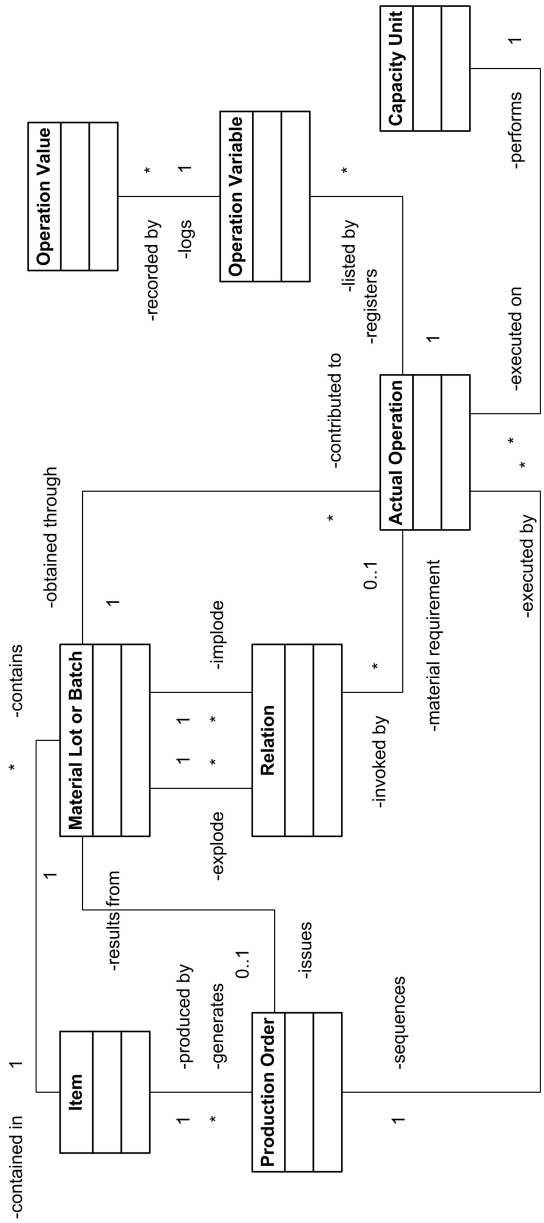

Figure 4.14: Reference data model for tracking and tracing [JVvDB03]

unit class in Figure 4.1.

Although the model proposed by Bechini et al. [BCMT08] allows representing transfer, transformation and transportation events (and in this sense differs from the model of Jansen-Vullers et al. (2003)), it lacks the ability to represent the money flows that remunerate product flows. Additionally, the model does not include future events and abstracts from the economic agents (e.g., truck driver) that participate in economic events or are liable for commitments. Moreover, the model does not acknowledge that an event can be perceived by more than one organizational unit as acquisition and providing events are attributed to only one responsible actor at a time. Consequently, a transfer event can only be perceived as an acquisition by the buyer or as a providing event by the seller, but not both simultaneously. In addition the model abstracts from the relation between the location (i.e., site class) of resources (i.e., lot class) and the events (i.e., activity class) that affect them (e.g., a transportation event is defined by the location of its inputs and the location of its outputs). More precisely, the model does not account for the relation between the location of an activity and the location of the lots that are involved in it. As there is no conceptual connection between the location of an event and the location of the resources that are involved in it, Bechini's traceability data model allows events to affect remote lots, which allows for omissions in the event chain (e.g., transports can be omitted consistently). As Bechini's model also abstracts from the economic rationale, it cannot account for trade. Finally, the model also abstract from future events, which prevents it from representing production orders and other commitments.

In conclusion, both Jansen-Vullers et al. [JVvDB03] and Bechini et al. [BCMT08] have proposed valuable models for product tracking and tracing, but these models are insufficiently complete for full-scale supply chain management as they, unlike our model, both abstract from money flows. Furthermore, Jansen-Vullers'model abstracts from inter-enterprise phenomena like transactions between and orders of trading partners, where Bechini's model abstracts from future events like sales orders and production orders.

4.4.2 REA Models

REA makes an explicit distinction between a trading-partner view and a so-called independent view on business assets and transactions. The former view (as in [GM02])) specifies a business conceptualization from the sole perspective of one particular party involved in business, called the 'trading partner'(e.g., an enterprise doing business in its role of customer, producer or supplier). The latter view, put forward in the ISO-standardized version of REA known as the Open-edi Business Transaction Ontology (OeBTO) [ISO07], looks at business from an independent observer perspective or 'helicopter'view (e.g., business seen as flows of goods,

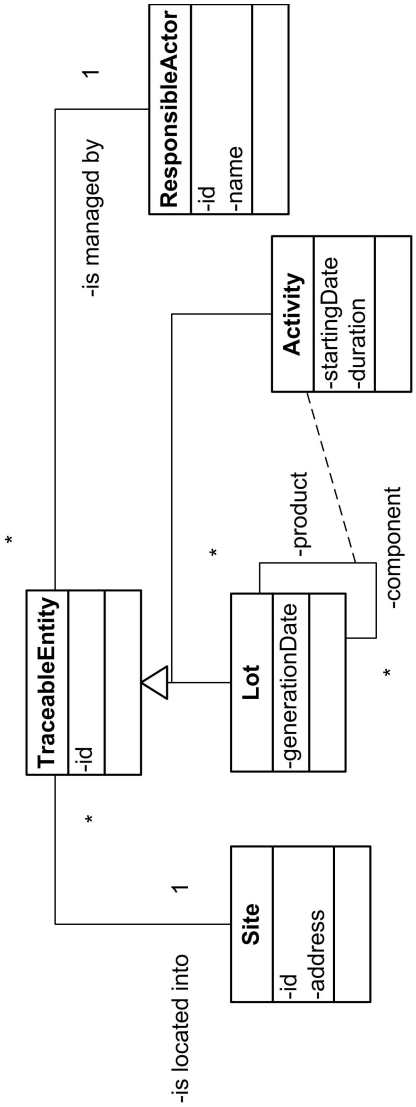

Figure 4.15: Traceability data model [BCMT08]

services and money between parties that are caused by events initiated by these parties). It is clear that both views, respectively focused on intra-enterprise phenomena (i.e., transformations and transportations) and inter-enterprise phenomena (i.e., transfers), are integrated in our conceptual model as it is used to track and trace products through the production processes of multiple supply chain partners.

Figure 4.16 shows a modernized version of McCarthy's foundational conceptual model [McC82] for the trading-partner view. To reply to Weber's critique [Web86] and comply with the analysis in [GM02], the n-ary 'control'relation has been decomposed in two separate participation (i.e., agent-event) relationships and the economic agent and economic unit have been merged in the economic agent class. In this conceptual model, subsequent activities in a production process can be identified by constructing chains of economic events for which the production (i.e., increment) of an economic resource has to precede its consumption (i.e., decrement). Similarly, bills of lots and materials can be constructed, identifying the consumed resources as inputs for the produced outputs.

Next to the representation of production processes, this conceptual model can also be used to represent transactions between trading partners. Within the scope of trade, the economic rationale underlying trade is given by the increment-decrement association (representing the duality relationship between an increment and a decrement event) and the inside-outside semantics of the participation relationships between economic events and economic agents. The Transaction-Inside_Party association reveals the agent whose view determines which events are increments and which are decrements. The Transaction-Outside_Party association relates the inside party's counterparty to the economic event. Where the conceptual model in Figure 4.1 can deal with the view of multiple (e.g., both) trading partners at the same time, McCarthy's foundational conceptual model can only deal with the view of one trading partner at a time consequently disabling the traceability of resources across enterprise boundaries.

Subsequently, Figure 4.17 shows McCarthy's conceptual model for the independent view as it appears in ISO's business transaction scenario standard [ISO07]. It should be noticed that this independent view replaces the 'increment'and 'decrement'notions by 'to'and 'from'semantics for the participation relationships. As a result, McCarthy's independent view does no longer contain increment and decrement semantics, which have to be derived from the To and From relationships. Namely, the To relationship relates the event to the agent that experiences an increment due to this event, the From relationship relates the event to the agent that experiences a decrement. As a result of this modeling choice, the duality relationship loses its increment and decrement semantics, which can no longer be represented explicitly. Consequently, the independent view enables tracking and tracing resource flows between trading partners, but loses the ability to monitor production processes inside these trading partners. As a result, supply chains that

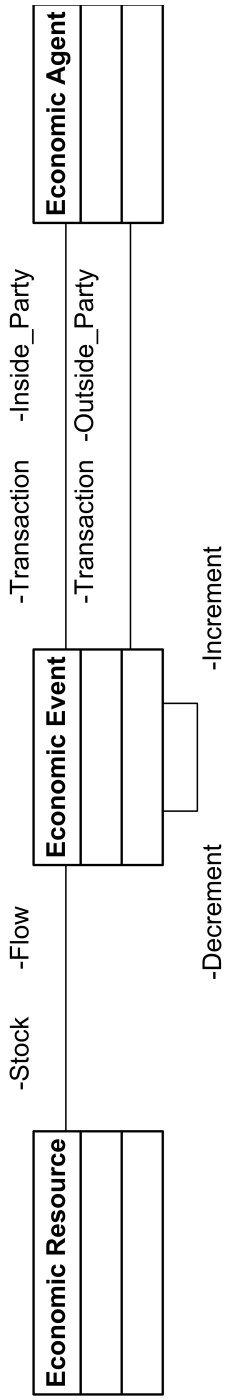

Figure 4.16: McCarthy's trading-partner view REA model (confer fig. 3.12)

contain conversion processes (e.g., assembly) cannot be monitored with REA's independent view model.

The REA ontology has also provided a basis for other conceptual models of business, like Hruby's model (Figure 4.18) [Hru06]. By explicitly discriminating increment and decrement events, it differs from McCarthy's trading-partner view model, which chooses to represent increment and decrement as roles of events and not as kinds of events. This articulation of the REA ontology has, alike our tracking and tracing model and McCarthy's trading-partner view, the advantage that it explicitly shows how resources are affected by events (i.e., the Inflow and Outflow relations), without having to rely on interpretation. The classification of events as increments or decrements depends on the viewpoint (e.g., seller, buyer) taken. This viewpoint should be unambiguous (confer infra) as Hruby's model explicitly takes the trading-partner view on economic phenomena. Nevertheless, the abstraction from view integration (i.e., trading-partner and independent) does impose considerable limitations to the use of Hruby's model for building tracking and tracing applications, in particular when it is used to integrate events performed by different supply chain partners into an event track or trace that spans an entire supply chain.

Hruby's model also incorporates provide (i.e., Provider-Provide) and receive (i.e., Recipient-Receive) relationships that connect agents with increment and decrement events. The provide (receive) relationship links the event to the agent that experiences a resource decrement (increment) due to the event, which makes Hruby's provide (receive) relationship equivalent with McCarthy's 'from'('to') relationship in the independent view. This indicates that the viewpoint that is taken is implicit and ambiguous in Hruby's model, as the viewpoint determining entity is the provider for a decrement event and the recipient for an increment event. However, the largest limitation of Hruby's conceptual model is that we cannot discriminate between dualities that construct events (e.g., transfer and transformation duality) and dualities that construct transactions (e.g., payment for purchase). This hampers the discrimination between product and money flows and the duality that balances them. Consequently, money is modeled as in input for acquisition processes and an output of sales in Hruby's model. As a result, product flows cannot be isolated from the money flows they mirror [Fos22] and pure product traceability is hampered.

Where McCarthy's trading-partner and independent views focus on the representation of market transactions, Hruby's model mainly focuses on operations performed by one enterprise at a time. Both views (i.e., trading-partner and independent) cannot be integrated with each other although they can be mapped and translated, since they both represent a different view on the same reality by hiding part of the semantics (i.e., the trading-partner view hides the To and From semantics, the independent view hides the increment and decrement semantics).

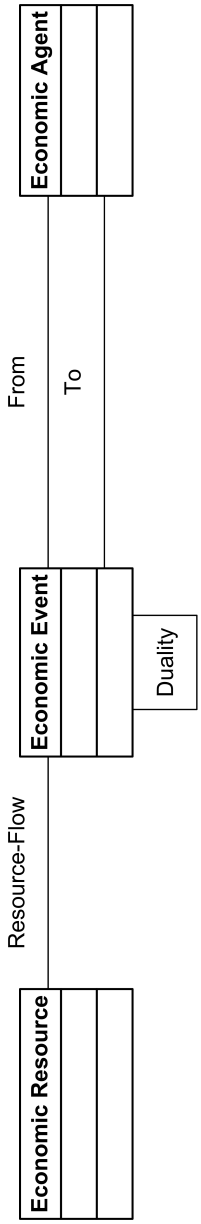

Figure 4.17: McCarthy's independent view REA model (confer fig. 3.13)

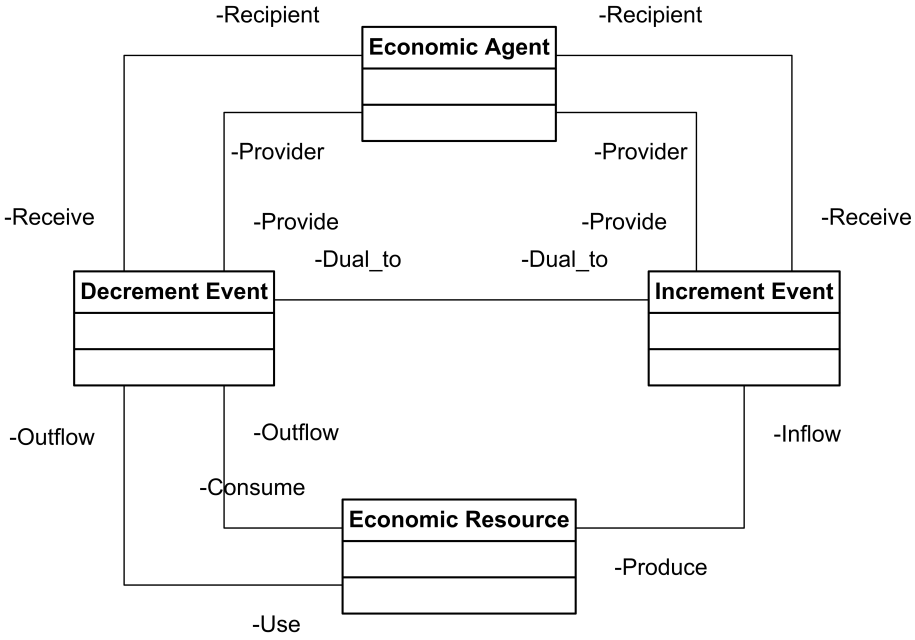

Figure 4.18: Hruby's REA reference model (confer fig. 3.16)

The model that we propose in this paper mitigates for this omission by representing the increment and decrement view on an event as separate classes. Additionally, it introduces the organizational unit concept that enables distinguishing between natural persons as participants in economic events and legal persons as entities that gain and lose economic control (e.g., ownership) over resources in the course of economic processes. As a result, our model enables the representation of 'from'('to') semantics by relating an event to an organizational unit through a decrement (increment) event. Additionally, these relations through the increment and decrement classes of our model conform to the inside view in McCarthy's trading-partner view as they represent which organizational unit experiences which economic event as an increment or decrement. The outside view can then be retrieved by finding the organizational unit that has the opposing view (i.e., increment or decrement) on the same economic event.

4.5 Conclusions, Limitations and Directions for Future Research

In the past, information systems focused on the enterprise as the core structure in business. The focus on the enterprise resulted in research on and the development of information structures for enterprise-wide management information systems, transaction-processing systems and accounting information systems. However, a continuously faster globalizing world economy and increasing cooperation among economic partners amplified the need to support the entire supply chain and not just individual players within it. This paper presented a conceptual model of inter- and intra-enterprise phenomena that can be used as a reference model for supporting applications of tracking and tracing for both enterprises and supply chains. This ability was illustrated with a prototype application for which we developed a data model and a database containing registrations that relate to both inter- and intra-enterprise phenomena. In particular, the example illustrates that the proposed model can be implemented such that it can be used to construct product tracks and traces from transaction data (i.e., transfers, transportations and transformations). Additionally, it illustrates that the registration of event and commitment perceptions enables us to trace and track the origin and (future) destination of product and money flows. The main contributions made by the model are the identification of the money flows that mirror the recorded product flows [Fos22] and the identification of the future paths of products in a supply chain. These contributions advance the current interpretation of tracking and tracing, which is limited to recording the past path and present location/existence of products though supply chains and abstracts from the registration of money flows and future product flows. [BCMT08]

The introduction of the organizational unit concept as business semantics viewpoint determining entity is a key feature of our model. Where previously, the perspective on business reality of each enterprise was represented in a separate model, the views of different enterprises that are part of a supply chain can now be jointly represented in a single model via the organizational unit concept and its relations with resources, events and agents. This explicit representation of enterprise viewpoints allows for a central administration of transactions between, transformations within enterprises and transports within and between supply chain partners. Apart from the central administration of data generated by multiple enterprises, the explicit representation of the scope of enterprises enables the construction of event chains that represent a product's history (i.e., including transfers, transportations and transformations, which construct exchanges and production processes). Another feature of the model that contributed to its ability to support product traceability irrespective of the number of enterprises (and enterprise information systems) in which products and their constituents have their origin, is the unification of increment and decrement events into a single economic event that is perceived differently by different organizational units.

The registration of money flows and the product flows they mirror may provide tools to impede money laundering by enabling tracing the (criminal) activities in which the money originates. The registration of the future paths of products may indicate the intended use of products, which would ease signaling improper use of products at an early stage. Additionally, such a registration of future paths of products might be useful in mitigating the effect of supply chain intrusions. For example, when a lot of products is destroyed or rejected, the effect on the remainder of the supply chain can be assessed and mitigating actions (e.g., delivery of alternative products) can be deployed in cooperation with the affected supply chain partners. Such product tracing infrastructure might also support product authentication (e.g., appellation d'origine controlee) in the battle on counterfeit and other supply chain intrusions (e.g., food safety scandals). In the prototype application section, we showed how event and transaction summaries can be transformed in product tracks and traces (e.g., product history, product composition) as well as event tracks and traces (e.g., for monitoring a cold chain).

Since the approach to track and trace both product and money flows both inside and between enterprises is novel to the best of our knowledge, the proposed model could not be benchmarked against similar reference models. Most comparable models or cases in literature focus on only one aspect (e.g., tracking products inside an enterprise, tracing product flows between supply chain partners, registering exchanges between trading partners). Consequently, no benchmark data or systems could be found. Therefore, we created a prototype application [FFFvL97] with the intended level of complexity, dealing with product and money flows between and within trading partners in the past, present and future.

In the future we would like to implement the proposed reference model for tracking and tracing to monitor real-world supply chains. However, the implementation of such a supply chain monitoring system requires the cooperation of each supply chain partner. Since supply chain partners have both parallel and opposing interests [BN96], access to 'strategic'information has to be limited. For example, for a supplier it is most probably undesirable to provide a customer access to data that allow the customer to estimate the actual production cost of a product and hence also the supplier's profit margin. However, in the case of vendor managed inventory (VMI) a customer may want to provide its supplier information about stock levels where competitors cannot have access to this information. Also information about an enterprise's cash supplies needs to be protected. Such 'strategic'and safety requirements considerably augment the complexity and hence cost of building a pilot application in a real-world setting. However, with the provided prototype we hope to have sufficiently illustrated the feasibility of such a real-world supply-chain monitoring system to motivate and fund the construction of a pilot application in a real-world supply chain.

4
Summary

Objectives

Demonstrating that the reference information model presented in chapter 3 can improve information flow with partners inside and outside the enterprise from both the operational and planning perspective.

Approach

Implementing the reference information model a prototype application (i.e. an example database that demonstrates the different usage scenarios covered by the model).

Findings

Although many reference models for traceability, business transactions and planning exist, no model existed that integrates all these features.

Contribution

Leveraging traceability to enable the monitoring of business transactions and process sequences.

Practical Implications

Supply chain authentication (e.g. for the marketing of raw materials) by tracking and tracing the composition of products as well as the events that constitute the history of a product and quality assurance (e.g. for food safety) by discouraging free-rider behavior.

Monitoring business transactions (e.g. for eliminating terrorism funding with counterfeit) and electronic data interchange by pairing product and money flows.

Supply chain management, through the identification of future product and money flows.

Part II

Simulation Models

5

Simulating Liquidity in Value and Supply Chains[1]

Abstract

This paper provides an ontology-based set of Petri-nets for simulating the effect of business process changes on an organisation's liquidity, and demonstrates that certain types of business process redesign can increase or reduce the amount of external funding that is required to prevent an organisation from defaulting on its debt. This debt defaulting may lead to proliferating liquidity constraints for subsequent supply chain partners. Consequently, this paper provides a proper toolkit for assessing and mitigating the propagation of liquidity constraints in supply chains. The paper uses the accounting-based Resource-Event-Agent ontology to create workflow patterns for modeling exchanges between supply chain partners and for the value chains that represent an organisation's internal processes. Both the exchange and internal processes continuously convert money into resources and vice versa. These models for money to resource and resource to money conversions are then used for constructing supply chain models for liquidity modeling and analysis.

[1] Laurier, W., Poels, G.: Simulating Liquidity in Value and Supply Chains. A. Albani et al. (Eds.): CIAO! 2009 and EOMAS 2009, Lecture Notes in Business Information Processing 34, pp. 40-54, 2009

Keywords

Resource-Event-Agent ontology, Petri-net, Simulation, Business Process Management, Workflow Model, Business Performance, Liquidity

5.1 Introduction

In recent years, business process and workflow management have concentrated on modeling, simulating and evaluating the physical and informational aspects of production and trade processes. [CKJJ04, DB07, SSR09, vdAvH02, VTM08] These processes are embedded in the enterprise's value creating activities that support the purpose of being profitable in the long term. This profitability is essentially a financial measure that is usually approximated in operational environments by efficiency and effectiveness measures such as cycle time, process time and production per time unit. Although these measures are generally believed to be connected to the profitability measure, the financial effect of changes in those operational measures is hard to simulate. What is needed, especially in a credit crunch during which cost thinking and consequently cost cutting soars, is a framework that allows us to simulate the effect of operational changes to an organisation's cash position and consequently its need for external funding (e.g. loans). This cash position is essential for an enterprise's continuity, and consequently its ability to reach its long-term goals, as insufficient cash supplies hamper the acquisition of inputs for the organisation's production processes and consequently also its ability to generate revenue from production process outputs.

In the systems design community, also the need *"to separate the stable ontological essence of an enterprise from the variable way in which it is realized and implemented"* [Die06] has been recognized, which has lead to the creation of various enterprise and business ontologies (e.g. enterprise ontology [Die06], e3-value [Gor02], REA [GM02], BMO [Ost04]) These ontologies describe the economic reality as a collection of constructs and axioms. The former create a controlled vocabulary for describing the business reality, whereas the latter represent *"fundamental truths that are always observed to be valid and for which there are no counterexamples or exceptions"* [Bah99].

As this paper intends to address the financial consequences of operational decisions within a stable framework that is sufficiently generic to be applied to different enterprises, the REA ontology was selected as the ontological basis for the presented simulation framework. The REA ontology was originally designed for sharing data concerning economic phenomena between accountants and non-accountants [McC82]. It has been applied in several other sub-domains of business (e.g. value chain [DCH05, Hru06] and supply chain modeling [HM00, ISO07]). In this paper, the REA ontology provides the conceptual basis for a set of Petri-net model construction patterns, which can be used for constructing business process simulation models. As Petri-nets provide an intuitive mathematically based formalization syntax for process representation that is widely used and supported by many tools for business process simulation they provide the preferred syntax for the models in this paper. Committing these simulation models to REA not only al-

lows checking their syntactic correctness, but also their semantic correctness (i.e. Do they represent proper business logic?). Syntax checking and simple simulation facilities are provided by the WoPeD[2] tool, which is used to create the exhibits in this paper, whereas more complex model simulations can be performed by the CPN tool[3].

Using the proposed set of REA-based Petri-net workflow model construction patterns, the paper proceeds by presenting generic workflow models for exchanges between supply chain partners and for the internal value chains that ensure an organization's continuity and sustainability. These exchange and value chain models are next combined into generic elements for constructing supply chain models that comply with the REA axiomatization. The paper then demonstrates that these supply chain models can be used to identify an organization's dependency on external funding and to analyze the effect of business process changes on the organization's liquidity (or cash position) and its need for external funding.

The Petri-net workflow model construction patterns are presented in section 5.2, together with the basics of the REA ontology. Sections 5.3 and 5.4 present the generic workflow models for exchanges and internal value chains. Then section 5.5 presents the integration of the latter two types of models into supply chain models and explains how these models can be analyzed to identify an organization's need for external funding. Section 5.6 presents an illustrative case in which our proposed modeling framework is applied to analyze the financial consequences of an intelligent workflow design in the invoicing process. Section 5.7 presents conclusions and future research.

5.2 Atomic REA Model Construction Patterns

This section presents three workflow model patterns in a Petri-net formalization that commit to the REA ontology and its axiomatization [GM04]. Committing to these axioms ensures modelers that the economic rationale is represented in their models. This rationale is what is generally abstracted from in conventional business process and workflow models, as the financial resources that are generated by these business processes and required for acquiring the inputs to these processes are often not shown. This omission helps designers to focus on the essential parts of the business process design but prevents them from assessing the financial consequences of process redesigns. These financial consequences of particular business process designs are the topic of this paper, as liquidity is a consequence of revenue generated from business process outputs and a prerequisite for acquiring the inputs to these processes.

[2] www.woped.org
[3] http://wiki.daimi.au.dk/cpntools/cpntools.wiki

The controlled REA vocabulary specifies four main concepts (i.e. resources, events, agents and economic units) [McC82]. Resources (e.g. goods and services) represent objects that are scarce, have utility and are under the control of economic units (i.e. legal or natural persons) that can use or dispose of them during events [Iji75, Yu76]. These events are initiated and controlled by agents (i.e. natural persons). As both economic units and agents represent persons, a more precise distinction needs to be made. This distinction specifies that agents (e.g. employee) act on behalf of economic units (e.g. employer, enterprise) [ISO07].

The REA ontology has three basic axioms [GM04]:

- REA axiom 1: *"At least one inflow event and one outflow event exist for each economic resource; conversely inflow and outflow events must affect identifiable resources."*
- REA axiom 2: *"All events effecting an outflow must be eventually paired in duality relationships with events effecting an inflow and vice-versa."*
- REA axiom 3: *"Each exchange needs an instance of both the inside and outside subsets."*

The first REA axiom specifies that each resource has an origin (i.e. inflow) and a purpose (i.e. outflow) in a production process (i.e. inputs and outputs) or an exchange (i.e. sales and purchases). Additionally, the first REA axiom requires economic events to affect resources. The second REA axiom defines the duality principle that materializes as claims (e.g. accounts receivable), which model decrements[4] (e.g. a shipment) that must be succeeded by increments[5] (e.g. receiving a payment), and as liabilities (e.g. accounts payable), which model increments (e.g. goods receipt) that must be succeeded by decrements (e.g. payment). The third REA axiom applies strictly to exchanges, stipulating that an exchange requires at least two parties (i.e. one taking the inside view (i.e. the enterprise that sells or buys) and one or more that are considered outside trading partners (e.g. customer, supplier) for this inside party).

As the models in the remainder of this paper are articulated in a workflow Petri-net formalization annotated with words from the REA ontology vocabulary, some essential differences between the REA and the workflow idiom need to be addressed. The most essential difference between the REA and workflow idiom is the use of the word 'resource'. In workflow models, the REA agents, in their role of 'performers of activities'are considered as resources because workflow resources represent both animate and inanimate means of production [vdAvH02]. In the REA idiom, resources represent solely inanimate means of production such as

[4] i.e. losing control over resources
[5] i.e. gaining control over resources

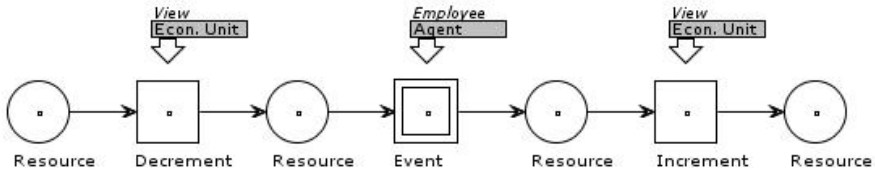

Figure 5.1: Event template

goods, services and money. We will stick to the REA interpretation in the remainder of this paper and thus clearly distinguish animate (i.e. agents) and inanimate (i.e. resources) means of production. Another difference between the REA and workflow vocabulary is the use of the word event. Events that are atomic from an economic point of view may represent entire operational processes in workflow models. Consequently, events as used in the REA vocabulary may represent complete workflow models with numerous activities or tasks that all contribute to the realization of the event.

The event template (fig. 5.1) shows the structure in which conventional workflow models need to be embedded to allow modeling the flows of value (i.e. a measure for the amount of money a resource can potentially generate) and money throughout supply and value chains. The double-edged box indicates where workflow models for value creating activities, such as logistic and production processes need to be inserted. According to the REA conceptualization, agents, who represent persons that act on behalf of economic units, participate in economic events [ISO07]. The economic units represent structures that can acquire, use and dispose of resources (e.g. organizations, enterprises, households). The decrement box (i.e. transition) shows that each (transfer or transformation) process requires inputs from the viewpoint of the economic unit that loses the control over these resources, whereas the increment box represents the outputs of such a process from the viewpoint of the economic unit that gains control over the converted or traded resources or resource bundles. The two economic units involved are different parties in case of trade and one and the same in case of conversions. The event template can be combined with itself to create entire supply chain models since every sink (i.e. right-hand side resource) of an event template equals the source (i.e. left-hand side resource) of a subsequent event template.

The duality templates (fig. 5.2 and 5.3) represent the creation of claims (fig. 5.2), which need to be settled by gaining control over new resources when economic units lose control over other resources, and liabilities (fig. 5.3), which need to be settled by losing control over resources when economic units gain control

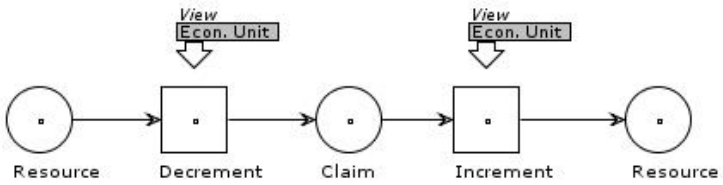

Figure 5.2: Claim duality template

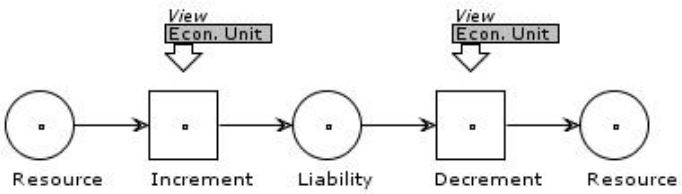

Figure 5.3: Liability duality template

over new resources. These dualities link oppositely directed resources flows. Conventionally, these resource flows represent goods or service flows that are paired with money flows. However, money flows can be paired with other money flows (e.g. loans) and goods and service flows can be paired with other goods and service flows (i.e. barter trade). In contrast to the event template, the increments and decrements in a duality template need to be related to the same economic unit as they represent this economic unit's acquisition and revenue cycles [DCH05] that model how economic units acquire and dispose of resources in exchanges.

5.3 Exchange Configurations

In this section, the potential configurations for exchanges are represented using the model construction patterns of the previous section. The potential for configuration is limited to two pattern variations as the claims and liabilities in exchange models need to mirror each other (i.e. a claim for a trade partner implies a liability for another trade partner and vice versa). The claims or liabilities that represent the revenue cycle of a seller are coupled with the claims or liabilities that represent the acquisition cycle of a buyer via transfer events that represent either the transfer of products (i.e. goods and services) or the transfer of money. These transfer events

are executed by agents that are accountable for them (e.g. a carrier).

Fig. 5.4 shows a conventional situation where sellers temporary pre-finance their sales, creating a claim (i.e. accounts receivable) that is a liability (i.e. accounts payable) for the buyer. Consequently the seller is exposed to default risk and requires the financial means (e.g. equity or long term capital) to fund these unpaid sales. The workflow model also reveals a logical sequence between the product and the money transfer (i.e. the product transfer precedes the money transfer). Consequently, the lifespan of the claim is determined by the duration of both transfers and the lifespan of the liability. Hence, reducing the term of payment and the duration of the transfers means reducing the need for external funding as the number of claims that needs to be financed is reduced.

Fig. 5.5 shows the opposite of fig. 5.4, meaning that in this exchange process model, buyers pre-finance their future purchases. As a result, sellers are liable to the buyers creating a liability for the seller (i.e. payments received on account of orders) and a claim for the buyer (i.e. orders in progress). This practice is rather exceptional, although it is applied in long-term projects (e.g. property development). In this situation, the buyer is exposed to default risk and requires sufficient financial means to fund these undelivered purchases. The according workflow model specifies that the money transfer precedes the product transfer, which makes the lifespan of the claim a sum of the duration of both transfer events and the lifespan of the liability. As in the previous workflow model, the duration of the claim, and consequently the amount of required funding, can be reduced speeding up the transfers and reducing the lifespan of the liability.

5.4 Value Chain Configurations

This section presents the four potential configurations of an economic unit's value chain, which contains an acquisition, a conversion, a revenue and a financing cycle [DCH05]. The acquisition cycle exchanges money for products, whereas the revenue cycle exchanges products for money. The conversion cycles consume products (i.e. raw materials) producing new products, while the money conversions in the financing cycle model the process of approving payments, which converts incoming money flows into outgoing money flows. Such money conversions include converting incoming money flows from loans and equity to outgoing money flows for acquiring resources. They also include converting incoming revenue to outgoing installments, dividends and money for resource acquisitions.

Both the acquisition and revenue cycles can be represented by claims or liabilities. In contrast to the exchange configurations in the preceding section, claims and liabilities do not need to mirror each other in value chain configurations since there are no opposing economic interests between the acquisition and the revenue cycle of an organization. The acquisition and revenue cycle of the same economic

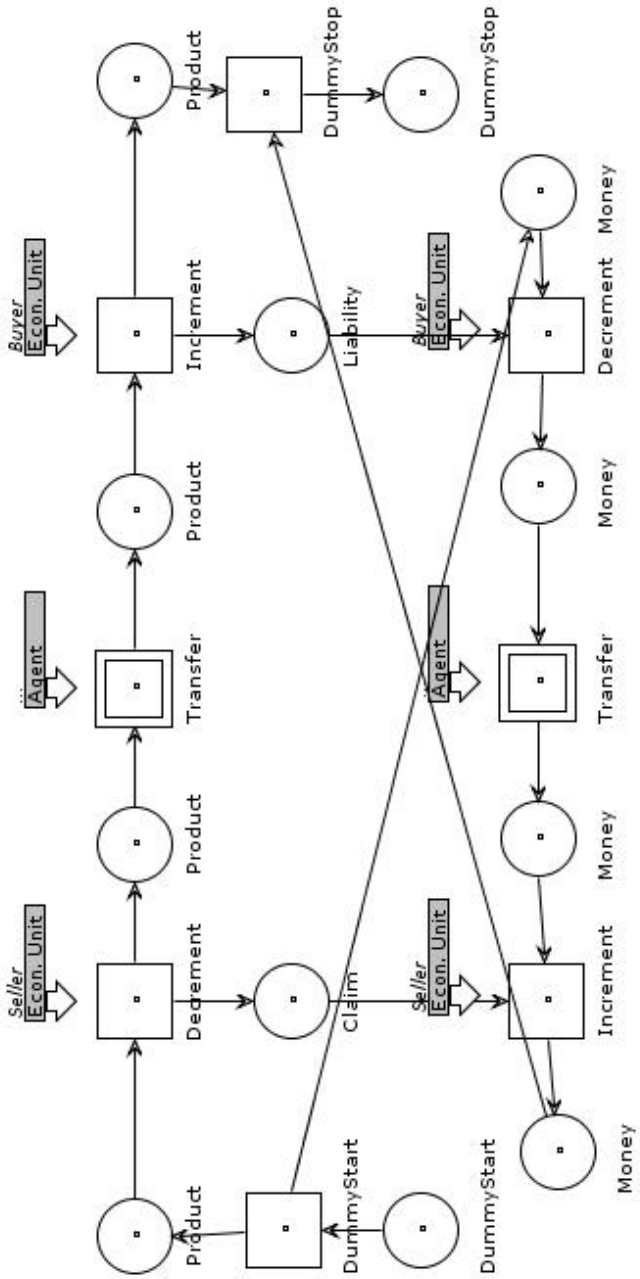

Figure 5.4: Buyer liable

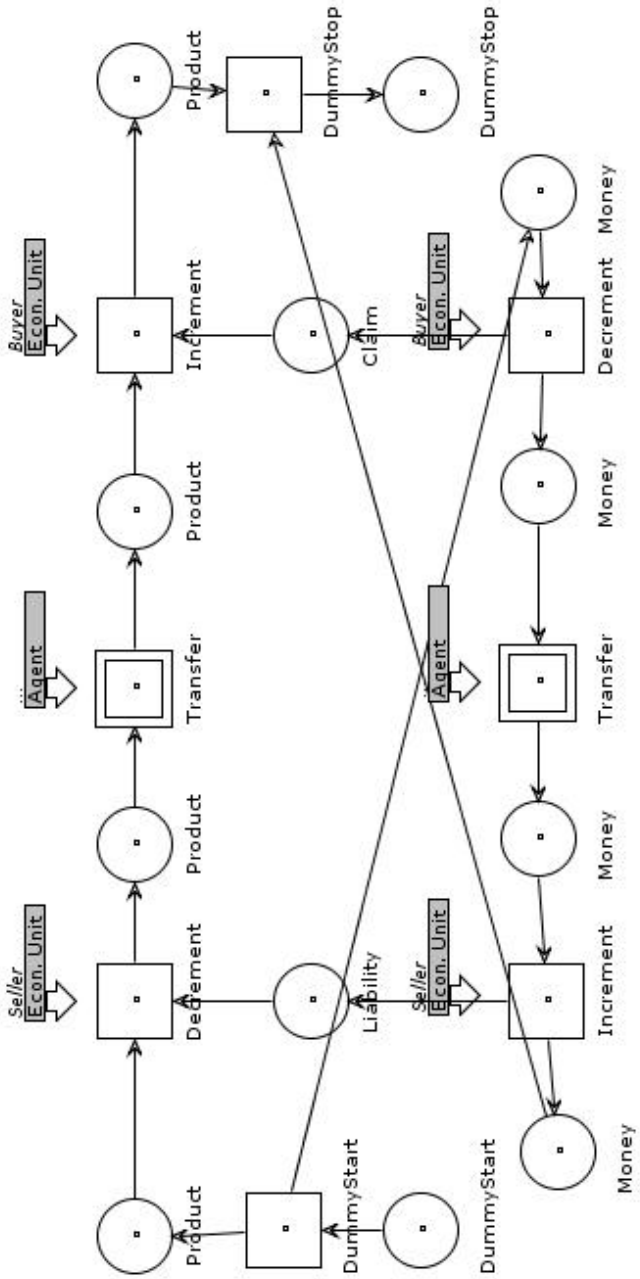

Figure 5.5: Seller liable

unit both try to maximize its profit, whereas the acquisition and revenue cycles of trading partners try to maximize the profit of their respective economic unit. As a result, the value chain pattern has four variations. As agents participate in the transfer events, they also participate in the conversion events in the value chain patterns (e.g. employees). For the reason of conciseness, the models in this section abstract from the increment and decrement sides of conversion events, which are hidden inside the double-edged conversion box. Consequently, the models represent a trading-partner view on business, which abstracts from the increments and decrements in an enterprise's internal processes.

In fig. 5.6, the organization's acquisition and revenue cycles create liabilities to the organization's suppliers and customers respectively. For organizations, this situation is ideal as customers pre-finance their own purchases and the suppliers finance their own sales. Consequently, organizations without long-term capital and equity could exist in theory. The workflow model reveals that the product (i.e. production process) and money (i.e. approving of payments) conversions are fully parallel but synchronized, which means that the lifespan of the liabilities is not directly determined by a conversion although the duration of the synchronization process (i.e. the entire workflow) is determined by the longest of the conversions.

Fig. 5.7, on the other hand, models a more conventional situation where suppliers pre-finance their sales (i.e. buyer liable). Consequently, the economic unit under review is liable to its supplier and has a claim on its customer. As a result, purchases provide (temporary) funding and sales require (temporary) funding. The workflow model for this value chain configuration reveals a logical sequence between the conversions (i.e. the product conversion needs to precede the money conversion). This sequence stipulates that the lifespan of the liability to the suppliers is determined by the duration of both conversion processes (i.e. production process and approving payments) and the lifespan of the claim on the customer.

Fig. 5.8 models the converse of fig. 5.7, representing customers that pre-finance their purchases (i.e. seller liable). As a result, the economic unit under review is liable to its customers and has a claim on its suppliers. Accordingly, sales provide temporary funding and purchases require funding. The workflow model for this value chain configuration shows a sequence of events which specifies that approving payments needs to precede the creation of value in production processes. Consequently, the lifespan of the liability is determined by the duration of the conversion events and the lifespan of the claim, as was also the case for the preceding value chain configuration.

Finally, fig. 5.9 shows the opposite of fig. 5.6, meaning that the economic unit under review pre-finances both its sales and purchases. Hence, both its suppliers and customers are liable to the economic unit under review. Consequently, both sales and purchases require external funding. This feature is also reflected in the workflow model as this workflow model suffers dead transitions [vdAvH02], re-

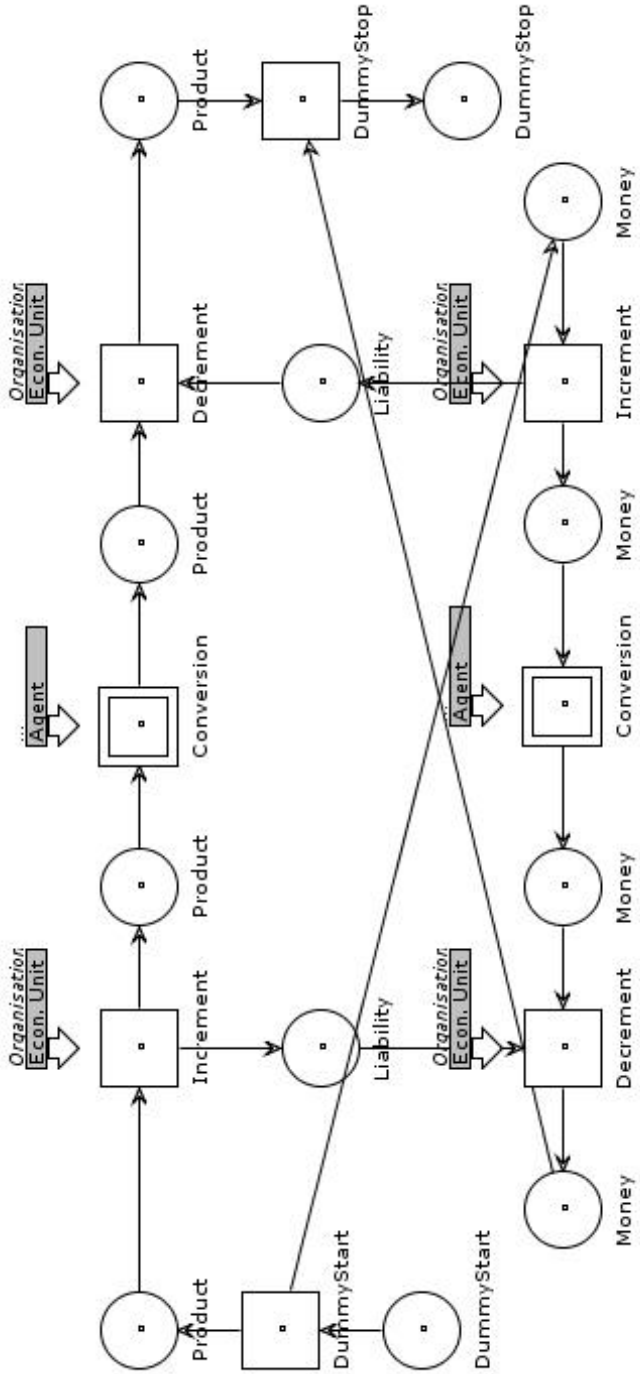

Figure 5.6: Economic unit liable to suppliers and customers

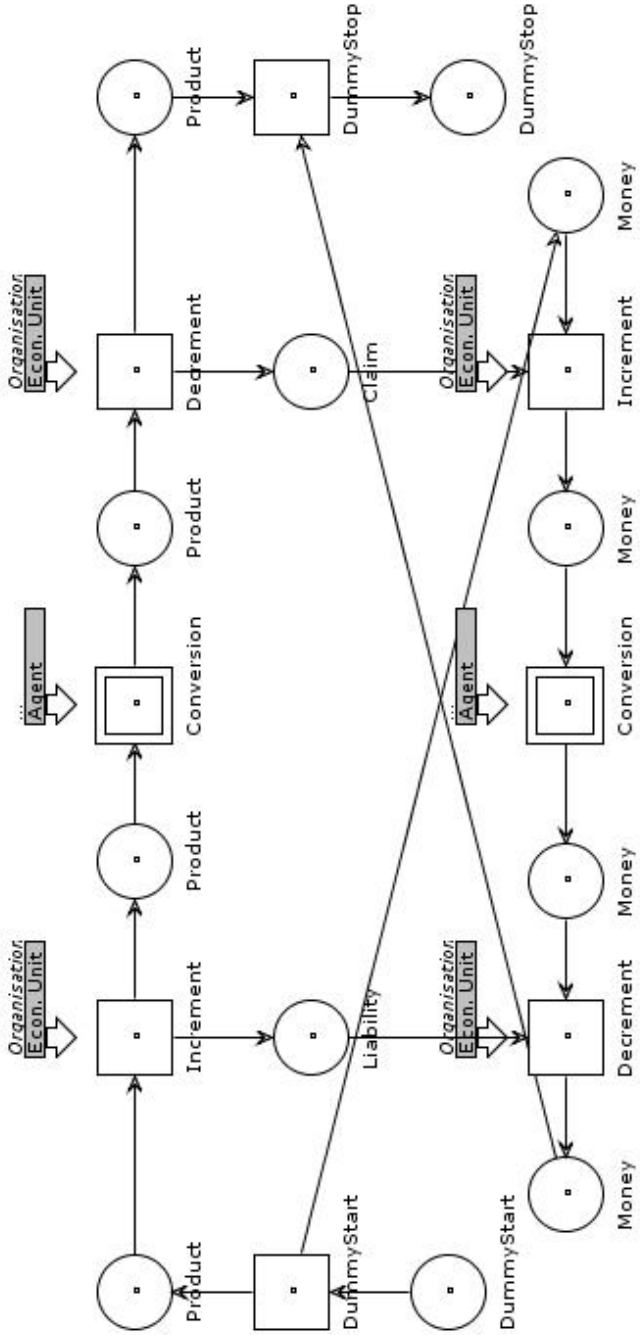

Figure 5.7: Economic unit liable to suppliers, claim on customers

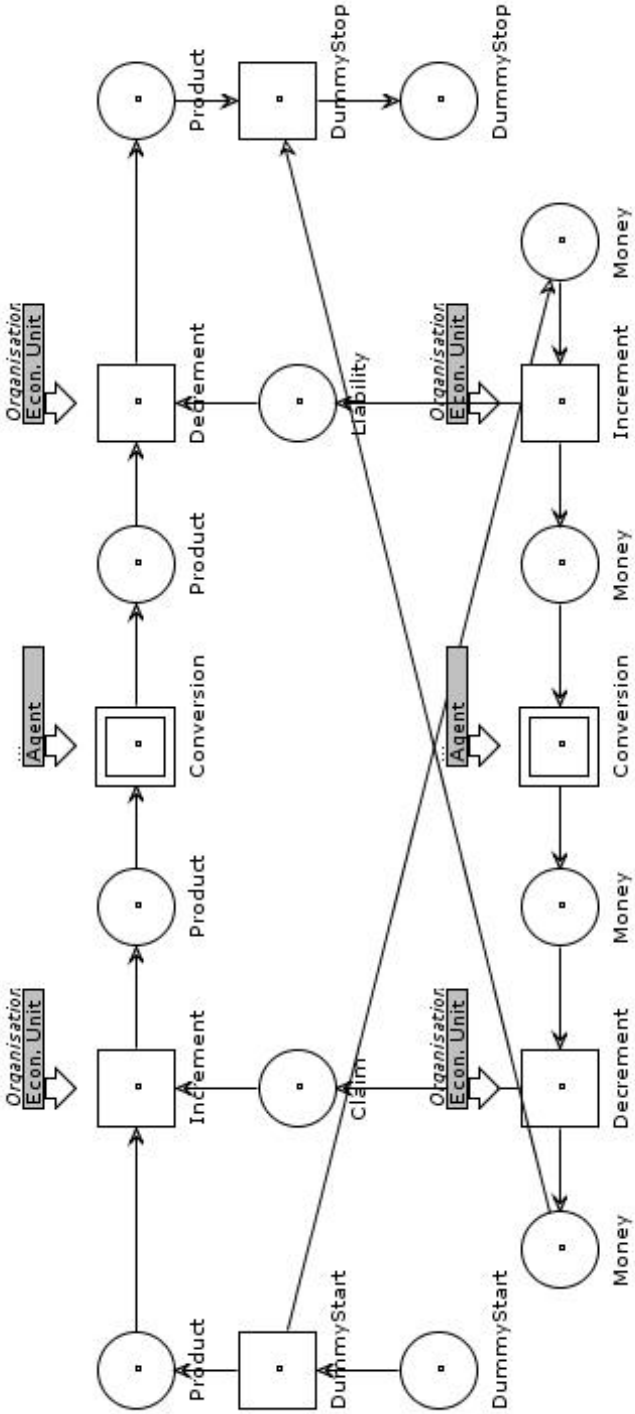

Figure 5.8: Economic unit claim on suppliers, liable to customers

flecting that this value chain configuration cannot be self-sustaining, which matches the requirement for external funding. This external funding would create additional money inflows that create liabilities owed to investors. In the workflow model this would mean breaking the loop into two mirrored value chain configuration with one claim and one liability each (i.e. fig. 5.7 and 5.8).

5.5 Supply Chain Pattern

In this section, the value chain and exchange configuration patterns are combined to create supply chain links that symbolize an economic unit and its interface (i.e. exchange) with a subsequent supply chain partner. In prior sections, no constraints have been imposed on the lifespan of dualities (i.e. claims and liabilities). Consequently, no distinction could be made between claims and liabilities that are ought to exist for a few seconds (e.g. in a cash sale) and claims and liabilities that should exist for several weeks, months or years (e.g. invoices, loans). To resolve this, the exchange pattern (fig. 5.4 and 5.5) has been extended with a business process that determines when liabilities are due (fig. 5.10). Such a business process is an example of a support activity in Porter's value chain [PM85]. Such an activity supports the economic processes in the economic events (fig. 5.1), which are containers for primary activities in Porter's value chain. According to the REA ontology, these primary activities are economic events and support activities consist of business events, which do not imply losing or gaining ownership over resources but are notwithstanding worth monitoring and controlling [ISO07]. The according workflow model (fig. 5.10) shows a business process (i.e. commitment) that determines the lifespan of the liability. This business process can be automated or executed by human agents. The process is coupled to the liability duality as valid liabilities can only be created from receipts and liabilities can only be due when all formalities have been completed (e.g. a payment for a purchase can only occur when the buyer received an invoice and the invoice is due). The template in fig. 5.10 is applicable to both exchange pattern variations in section 3; therefore it abstracts from 'money'and 'product'flows.

The first supply chain link (fig. 5.11) creates a structure in which organizations are liable to their customers who pre-finance their purchases. This pattern variation combines fig. 5.8 with 5.5, merging two rather exceptional configurations. The workflow model shows that the product delivery (i.e. product transfer) can be initiated (i.e. is 'enabled') as soon as the organization received the payment (i.e. money transfer) and the delivery is due. Next to the expected lifespan of the buyer's claim (i.e. the duration of both transfers and the claiming process), the workflow model also indicates the maximal lifespan (i.e. the duration of both transfers, both conversions and the supplier's claim on its supplier). The minimal lifespan is achieved when there is a stock of already converted products, which al-

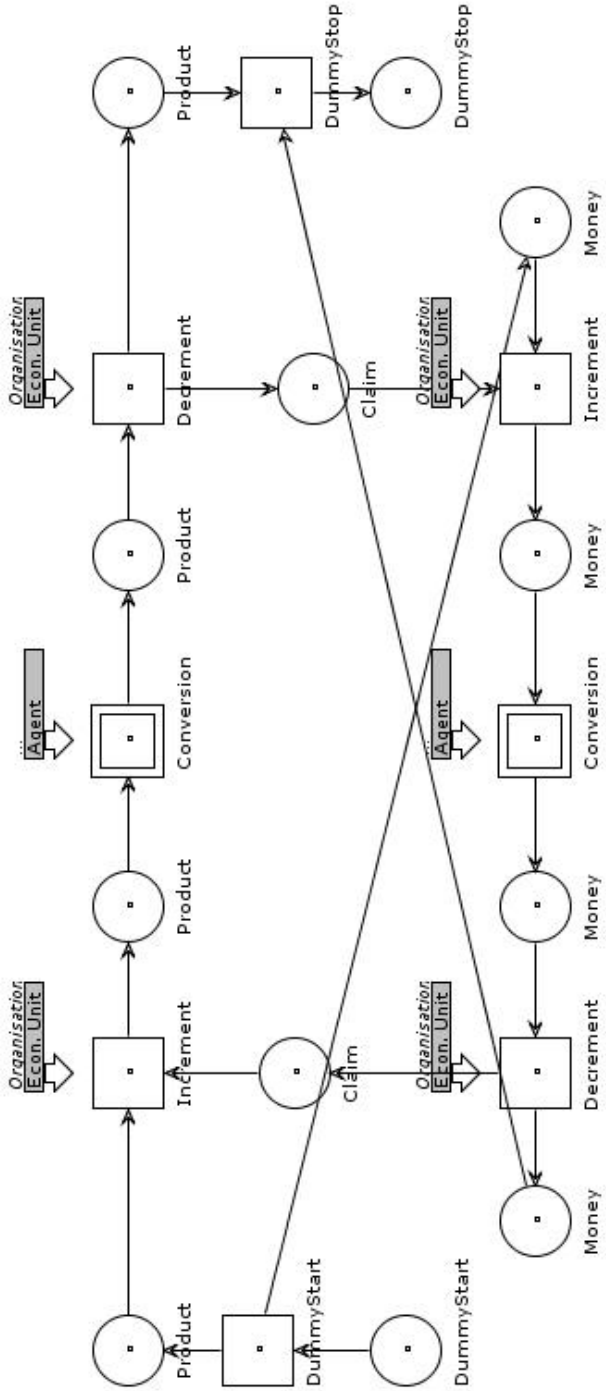

Figure 5.9: Economic unit claim on suppliers and customers

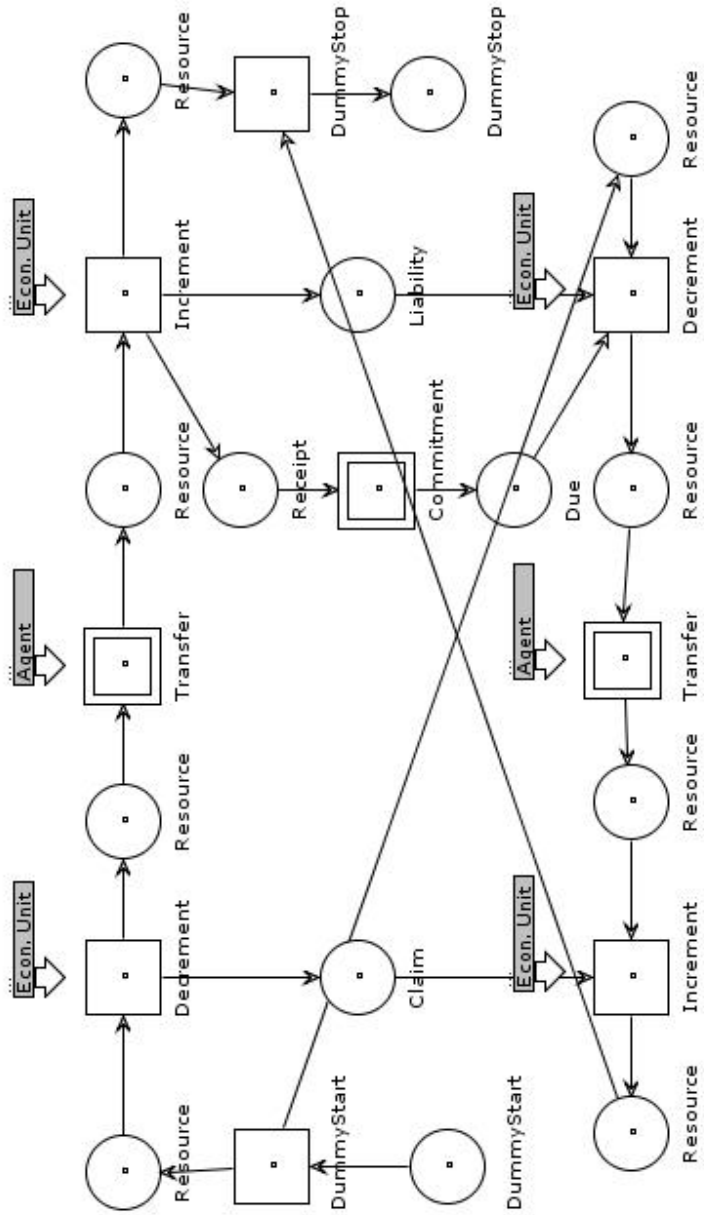

Figure 5.10: Duality lifecycle template

low us to reduce the lifespan of the liability to zero. The maximal lifespan occurs when the inputs to the conversions still need to be acquired. As both the right- and left-hand side of fig. 5.11 represent acquisition cycles that are implemented with claims, this supply chain pattern variation can be combined with itself to create entire pre-financed supply chains.

Whereas fig. 5.11 shows a supply chain link in which an organization is liable to its customers, fig. 5.12 models the more conventional situation where organizations have claims on their customers and the lifespan of the according liabilities is determined by the invoicing process. This pattern variation combines fig. 5.7 with 5.4. The workflow model reveals that product transfers can occur autonomously but money transfers occur only as a counter-value for product transfers when the according invoices are due. In this template, the lifespan of the liabilities is constrained by the invoicing process (i.e. the minimal and expected lifespan) and the conversions and transfers further up (i.e. right-hand side) the supply chain. The maximal lifespan of the liability on the left-hand side of fig. 5.12, is determined by the duration of both transfers and conversions and by the lifespan of the liability on the right-hand side of fig. 5.12. As with the preceding link (fig. 5.11), fig. 5.12 can be combined with itself to create entire supply chain models in which suppliers temporarily finance their sales.

Similar and more complex supply chains can be constructed applying the duality lifespan template to the variations of the exchange pattern shown in section 3, combining them with the value chain patterns in section 4.

5.6 Discussion

This section demonstrates and illustrates the effect of the duration of business processes on the amount of external funding required. This effect reveals the economic rationale for business process redesign [RLM05] and management [Sch08] as it is currently practiced. As an example, we take the conventional supply chain template where organizations are only liable to organizations upstream the supply chain (i.e. their suppliers) (fig. 5.12) and discuss the implications of different choices regarding business process configuration to the required amount of external funding. The presented theoretical example addresses an archetypal supermarket first and an archetypal retail store next. Both types of organization perform similar activities but the underlying business processes may be configured differently.

We assume that, due to its market power, the supermarket can obtain a 3 month (i.e. 90 days) pre-financing period from its suppliers. This pre-financing period is to be represented as a time trigger in a timed Petri-net representation of the invoicing process of the supermarket's supplier. This invoicing process is represented at the left-hand side of fig. 5.13. The product conversion process of the supermarket

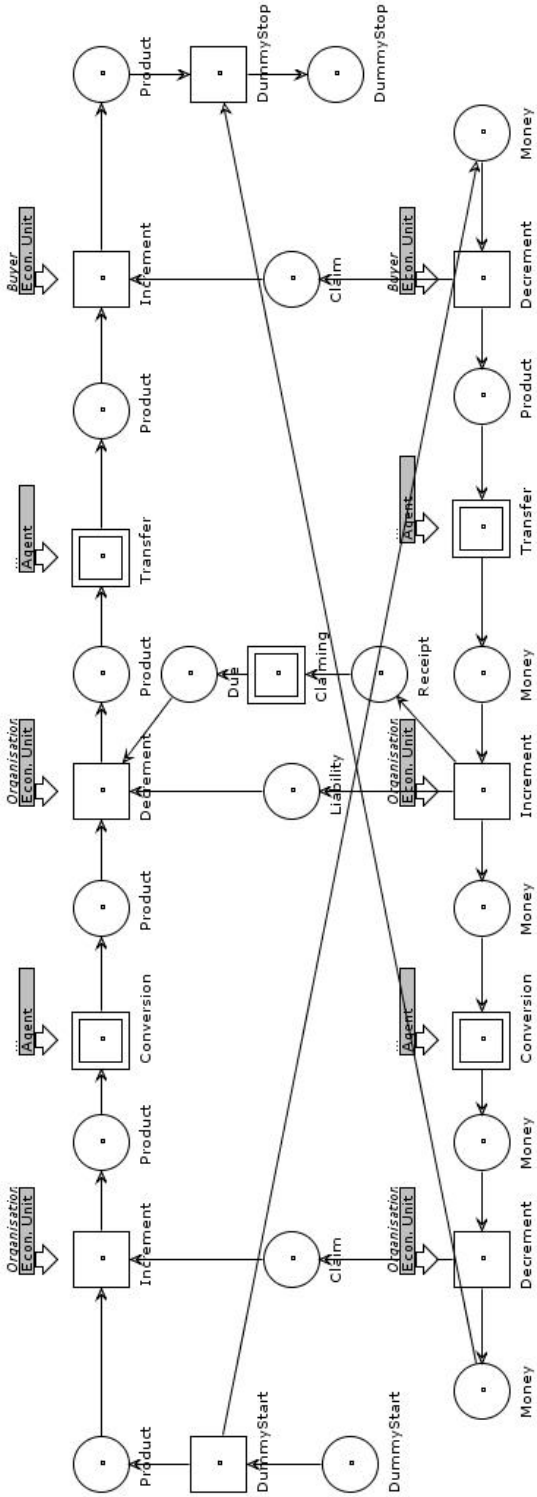

Figure 5.11: Supply chain link, claim on suppliers and liable to customers

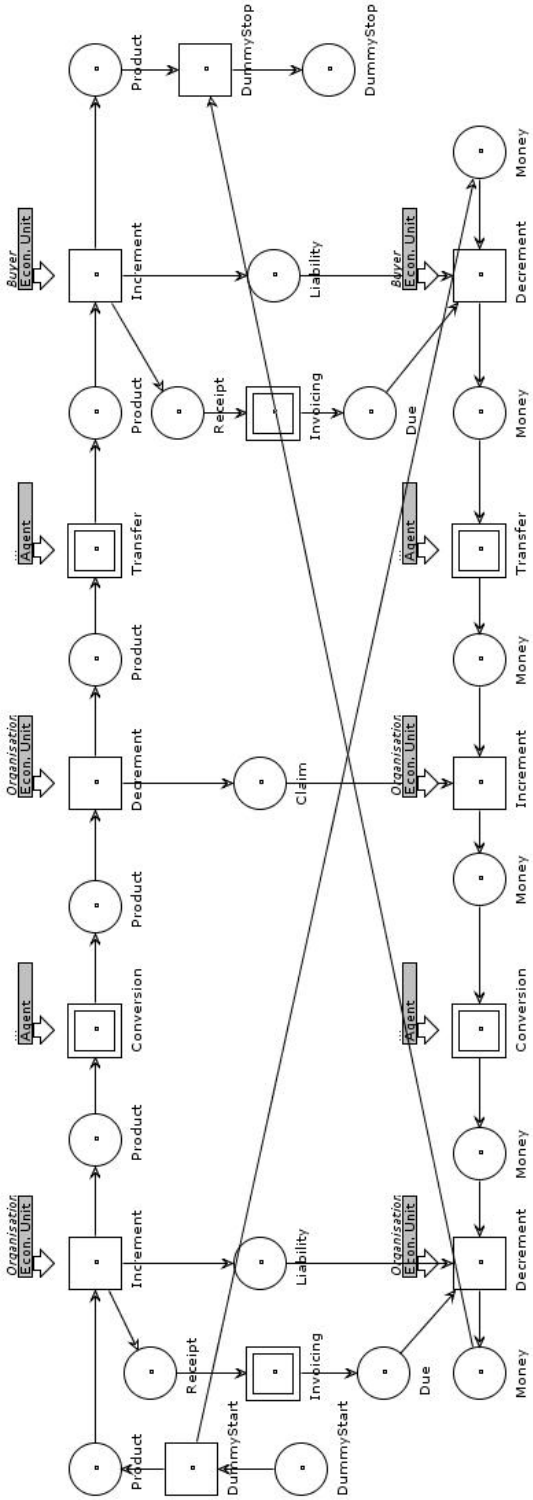

Figure 5.12: Supply chain link, claim on customers and liable to suppliers

is a complex logistic process in which products reside 2 months (i.e. 60 days) on average. Next, they have an average shelf-life of approximately one week (i.e. 5 days), which is represented as the product transfer process in fig. 5.13. Considering that most supermarket sales are cash sales, the duration of the right-hand side invoicing process is reduced to 0 days. Subsequently, it takes 2 days on average to put the cash received selling product on the supermarket's bank account, which is represented as the money transfer process. Once the money is on the bank account it can be spent again. However, it takes another week (i.e. 5 days) on average to approve payments (i.e. one meeting every two weeks). Consequently, abstracting from the profit margin, it takes the supermarket 72 days (i.e. 60 + 5 + 0 + 2 + 5) on average to have its money available for a subsequent business cycle. As the supermarket obtained a 90 day pre-financing period, it has an additional 18 days of funding it can use to fund other things. This means that the supermarket can use 5% (i.e. 18 days / 90 days * 3 months / 12 months) of its annual turnover to fund fixed assets.

The retail store, on the other hand, has no market power and can only obtain a 30 day pre-financing period from its suppliers. It takes the retailer one week (i.e. 5 days) on average to take the purchased products from its warehouse to the shelf. In contrast to the supermarket, the shelf-life of retail products is much longer e.g. 1? month (i.e. 50 days). As with the supermarket, most retail sales are cash sales which take 0 days to complete. Further, it takes 2 days on average to transfer the cash from the counter to the bank account. Then the retailer checks its accounts payable once a week (i.e. the money conversion takes 3 days on average). This results in a business process cycle that takes 60 days on average (i.e. 5 + 50 + 0 + 2 + 3) which is twice the period that has been pre-financed by the supplier. Consequently, the retailer continuously needs to fund one month of turnover with other financial means (e.g. loans, equity).

Both the retailer and the supermarket can enhance their liquidity (i.e. reducing the need for extra funding) by reducing the duration of the conversion and transfer processes. The duration of their invoicing process, which was set 0 in both examples, also indicates a potential risk for liquidity constraints further down the supply chain (i.e. customers and customer's customers) for economic units further up the supply chain (i.e. suppliers and supplier's suppliers) as liquidity constraints deadlock the money transfers that enable economic units to close their internal value chain. This (temporal) deadlock increases the need for external funding or imposes liquidity constraints if additional money is not found. Other potential solutions are shortening the pre-financing period for customers (which may induce new liquidity constraints for the customers) demanding longer pre-financing periods from suppliers (which may create liquidity constraints for them) or shortening the conversion and transfer processes in the business model, which is a motivation for business process redesign.

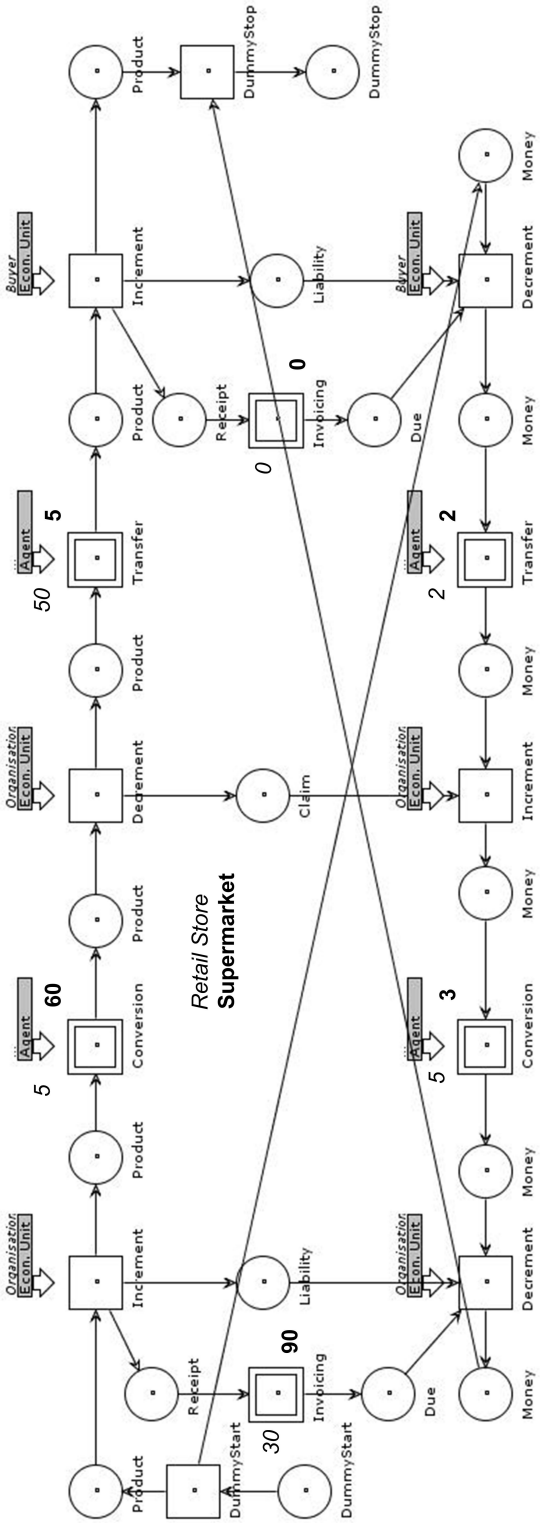

Figure 5.13: Exhibit: supermarket and retail store

Deadlock can also appear for the supply chain template in fig. 5.11 when the supplier does not deliver on time, as the customer cannot generate new financial means from products that it can sell to its own customers. The difference with the template in fig. 5.12 is that the liquidity constraints move further down (i.e. towards customers and customer's customers) the supply chain instead of further up the chain. For the symmetric value chain templates (fig. 5.6 and 5.9) the situation is also rather straightforward. For the double liability template (fig. 5.6) the liquidity is determined by the equilibrium between the opposing processes (i.e. product and money conversions) and their ability to generate resources. For the double claim (fig. 5.9) the situation is even simpler as this template is not able to generate its own means and needs to be funded completely.

5.7 Conclusion and Future Research

This paper presented an ontology-based modeling framework for evaluating the financial consequences of alternative business process design choices. This framework is especially useful for analyzing the amount of external funding required. First, the main constructs and axioms of the REA ontology, which provides a conceptual basis for the modeling framework, were introduced. Next, these constructs and axioms were used to create workflow model construction patterns (using a Petri-net formalization). These patterns were subsequently used to build modeling templates for the value creating processes of economic units and for exchanges between supply chain partners. The paper presented two alternative configurations for exchanges and four alternative configurations for value creating business processes. These modeling template variations can be combined to create supply chain modeling templates, two of which (i.e. the more typical cases) were shown in the paper.

For each of these exchange, value chain and supply chain configurations, the corresponding Petri-net modeling template was analyzed to assess its ability to generate its own financial resources. These analyses showed that requirements for external funding depend upon the lifespan of claims and liabilities and the duration of operational processes (conversions and transfers). Based on the analysis, suggestions were made to reduce the dependency on external funding. These suggestions can be implemented through the redesign and management of business processes. Among these redesign efforts, especially the redesigns that reduce value chain cycle time (i.e. the time between the acquisition of a resource and the moment a new similar resource can be acquired with the means generated by the former resource) were identified as useful for reducing an organization's dependency upon external funding since they reduce the amount of assets that is to be funded with liabilities (e.g. debt).

In the future, we intend to apply the templates presented here to detailed busi-

ness process designs instead of the highly abstract aggregate process models used in this paper. Consequently, claims and liability structures will be identified for each individual resource sold (e.g. product and services) or purchased (e.g. raw materials, labour, fixed assets). This will allow us to assess the consequences of individual business process redesigns to an organization's liquidity. These model simulations will involve modeling timed Petri-nets and simulating their behavior using the advanced simulation and monitoring capabilities provided by the CPN tool. In a later stage, these simulation efforts could evolve towards the development of a business process management software tool that supports managers in identifying and motivating priorities among potential business process redesigns by showing the financial impact of these redesigns.

5
Summary

Objectives

Addressing the financial consequences of operational decisions within a stable conceptual framework that is sufficiently generic to be applied to different enterprises.

Approach

Creating workflow patterns that are committed to the REA ontology for modeling exchanges between supply chain partners and for the value chains, which represent an organizationś internal processes.

Findings

Certain types of business process redesign can increase or reduce the amount of external funding that is required prevent an organization from defaulting on its debt.

Contribution

Simulation models using a Petri-net syntax that are committed to the REA allow being checked for both syntactic and semantic correctness.

Practical Implications

Providing a proper toolkit for simulating the effect of business process changes on an organizationś liquidity, and assessing and mitigating the propagation of liquidity constraints in supply chains.

6

A Simulation Model Articulation of the REA Ontology[1]

Abstract

This paper demonstrates how the REA enterprise ontology can be used to construct simulation models for business processes, value chains and collaboration spaces in supply chains. These models support various high-level and operational management simulation applications, e.g. the analysis of enterprise sustainability and day-to-day planning.First, the basic constructs of the REA ontology and the ExSpect modeling language for simulation are introduced. Second, collaboration space, value chain and business process models and their conceptual dependencies are shown, using the ExSpect language. Third, an exhibit demonstrates the use of value chain models in predicting the financial performance of an enterprise.

[1]Laurier, W., Poels, G.: A Simulation Model Articulation of the REA Ontology. R. Meersman, P. Herrero, and T. Dillon (Eds.): OTM 2009 Workshops, Lecture Notes in Computer Science 5872, pp. 554-563, 2009

Keywords

Resource-Event-Agent ontology, supply chain, enterprise, accounting model, simulation

6.1 Introduction

This paper demonstrates how an enterprise ontology can be useful in building various kinds of business simulation models. Such simulation models predict the behavior of newly designed business systems before they are implemented. In technological fields, simulation models are useful when the cost of error associated with R&D is high. For instance, when a new plane type is designed, its behavior is simulated before the actual plane is constructed as building a plane is expensive and time consuming and operating a potentially flawed experimental plane might imply loss of lives. Also in business, simulation models are useful for predicting the effects of business changes and redesign actions. For instance, enterprises that suffer liquidity constraints in an environment where external funding is scarce (e.g. unsteady banks and capital markets) may also encounter high costs of error since redesigns that worsen liquidity constraints may immediately imply the end of an enterprise's existence.

The enterprise ontology we look at is the Resource-Event-Agent (REA) ontology [DCH05, GM02, GM04, ISO07, McC03], which originates in a semantic accounting data model that was designed for a shared data environment in which accountants and non-accountants maintain information about the same economic phenomena [McC82]. Apart from accounting systems design, the REA ontology is applied in accounting education, model-driven systems design, supply-chain collaboration and knowledge representation [GLP08].

The accounting background of the REA ontology allows simulation model builders to inject an economic rationale into their simulation models, and in particular to integrate models showing financial flows with the more traditional models showing logistic flows and workflows [vdAvH02]. The explicit connection between logistic processes and financial processes increases the configurability and reusability of the simulation models. For instance, a business process simulation model for determining process capacity, cycle time and process time can easily be reconfigured for estimating the value of and cost of capital for the 'work in progress'that will appear as an asset on the balance sheet. Another advantage of REA ontology-based simulation models is that the combined effects of redesigning individual business processes and workflows can be simulated in enterprise or supply chain level models, which may help to identify synergies and interferences between the redesign actions taken on different processes or workflows.

The idea of using the REA ontology for simulation purposes comes from Church and Smith [CS08], who consider it the best-of-class supply chain and business process ontology for constructing business simulation models. Church and Smith merely explore the idea of ontology-based simulation models by using REA for modeling continuous flows in an enterprise using System Dynamics[2].

[2] `http://www.systemdynamics.org/`

The simulation of continuous flows solely supports high-level management applications (e.g. long-term capacity planning). We provide further evidence of the REA ontology's simulation modelling power by means of REA ontology-based models for discrete event simulation. which supports various kinds of operational management applications (e.g. day-to-day planning). This type of simulation is based on the identification of individual cases for which a lifecycle can be developed based upon case characteristics and on the use of decision algorithms for specific process stages (e.g. if a loan is rated above average it will be approved and granted). For the articulation of the discrete event simulation models, the ExSpect[3] simulation tool will be used.

Section 6.2 presents the constructs of the REA ontology and the simulation language of the ExSpect tool. Section 6.3 shows how discrete event business simulation models for enterprises that engage in supply chains, including models for business processes, value chains and collaboration spaces, are constructed using the REA ontology. Section 6.4 then compares four alternative value chain designs, on their financial performance (e.g. turnover, total liabilities). Section 6.5 ends the paper with conclusions and ideas for further research.

6.2 Overview of the REA Ontology and the ExSpect Simulation Language

6.2.1 The ExSpect Simulation Language

The ExSpect simulation language is a visual modeling language that is based on high-level (i.e. hierarchical, timed and colored) Petri-nets. Petri-nets model processes by means of tokens, places and transitions. The distribution of the tokens over the places in a Petri-net represents the state of the process. The transitions specify a strict precedence relation between the places a token can occupy during the execution of a process. Hence, transitions model events that change the state of the process. Colored Petri-nets allow tokens to have attributes describing them, which means that the state of a process is also determined by the values that the attributes of the tokens take. Hierarchical Petri-nets allow decomposition structures such that a Petri-net can be refined by other Petri-nets [ZB97]. In classic Petri-nets, all transitions represent atomic events, while in hierarchical Petri-nets transitions can represent complex events (i.e. events composed of other events). Finally, timed Petri-nets allow transitions to have a duration, which makes them useful for representing non-instantaneous events.

In colored Petri-nets, the attributes of a token can be used to give a stable identity to the individual that is represented by the token. In workflow modeling,

[3] `http://www.exspect.com/`

such an individual with a stable identity is called a case [vdAvH02] (e.g. a document that needs to be processed). When these identity bearing tokens reside in a place, they represent a lifecycle phase of such a case. Consequently, transitions define strict precedence relations between the lifecycle phases of a case or model a precedence relation between different cases.

6.2.2 The REA Ontology

A remarkable conceptual dichotomy in REA is the difference between economic phenomena and business phenomena. The former create or represent economic value (e.g. the acceptance of a sales order), while the latter are considered worth monitoring and controlling although they do not directly create or represent economic value (e.g. the registration of an accepted sales order). For the injection of economic rationale into business simulation models, this distinction between value bearing phenomena (i.e. economic) and value supporting phenomena (i.e. business) is primordial as it discriminates the economic rationale driven business logic from the operational implementation of the business logic. The economy/business dichotomy in the REA ontology relates closely to Porter's [PM85] discrimination of primary activities, which construct a scarce and valuable product, and support activities, which allow primary activities to take place. Primary activities directly contribute to the creation of economic value and are represented in the REA ontology as *Economic Events*. Support activities facilitate and enable this value creation and are represented in the REA ontology as *Business Events*.

Resources in the REA ontology are defined as goods, services or rights that have utility and are under the control of a legal or natural person. Economic resources can be distinguished from business resources because the former are inevitably scarce, which motivates their economic value [Iji75, ISO07, McC82]. Business resources support business activities without being scarce (e.g. information can be replicated/duplicated at very low cost). *Events* are occurrences in time that relate subsequent process states to each other. Economic events are discriminated from business events as the former involve gaining (i.e. increment) or losing (i.e. decrement) control over economic resources, in contrast to the latter [ISO07, McC82, Yu76]. *Agents* are natural persons that act on behalf of economic units (e.g. enterprises). Economic agents are accountable for, participate in or initiate economic events, whereas business agents are accountable for, participate in or initiate business events. *Units* determine the scope of the context in which economic activities take place. Economic units represent financially self-sustaining entities, while business units characterize non-self-supporting structures that are usually embedded in economic units. Table 6.1 lists some examples for each basic REA construct presented here.

	Economic	Business
Resource	e.g. goods, infrastructure services, machines, rights	e.g. documents, manuals schedules, signals
Event	e.g. payment, purchase consumption, production, sale	e.g. scheduling a production run sending an invoice
Agent	e.g. cashier, salesperson production line operator	e.g. accounting clerk production planner
Unit	e.g. enterprise, organisation	e.g. department, team

Table 6.1: Examples of basic REA Concepts

6.3 Economic Rationale Models

In this section, it is shown how the economic rationale that is expressed in the REA ontology, can be represented in collaboration space, value chain and business process simulation models. *Collaboration space* models encompass the supply chain view on economic phenomena and focus on (physical) resource flows between economic units. Such models are especially useful for representing and simulating logistic, service and money flows between supply chain partners. *Value chain* [DCH05, McC03] models represent supply chain partners as extended open systems [Gil69] that sustain their continuity by enforcing economic reciprocity. This kind of simulation model can be used for assessing the financial characteristics (e.g. assets, liabilities, turnover) of individual supply chain partners. *Business process* models show the actual underlying processes. These models allow analyzing operational process characteristics (e.g. bottlenecks, cycle time, capacity).

The collaboration space model in fig. 6.1 shows the interactions between three supply chain partners, which we refer to as the enterprise (i.e. the supply chain partner that is the focal point of the simulation model), the supplier of the enterprise and the customer of the enterprise. This collaboration space model shows both business (i.e. offer, order and invoice) and economic (i.e. product and money) resources that are transferred between these partners. The underlying business processes, which can be modeled by decomposing the Petri-net, specify how an offer is generated and transformed into an order; how this order triggers a product transfer and the creation of an invoice and how this invoice triggers a money transfer. Since the economic units (i.e. Supplier, Enterprise, Customer) in the model are modeled as complex transitions (i.e. yellow boxes) that cover more complex underlying processes, each of the resource transfers (e.g. ProductSupp) in the collaboration space is allowed to occur individually, unless the underlying business process specifies otherwise.

Next to business processes, also value chains, which introduce the economic rationale for participating in a supply chain, determine the behavior of economic units (e.g. illiquidity may force an economic unit to halt its business processes).

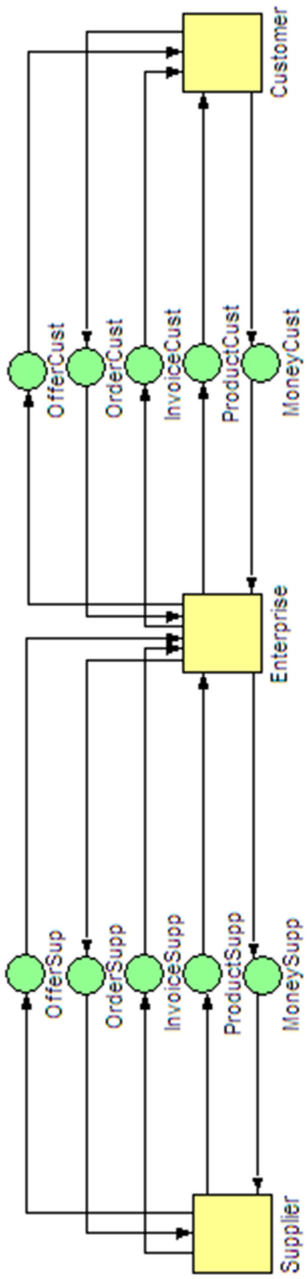

Figure 6.1: Collaboration space model for an enterprise with one supplier and one customer

Fig. 6.2 shows a value chain model for the enterprise that was the focal point in the collaboration space model of fig. 6.1. This value chain model presents a generic and extensible template for enterprises participating in supply chains. This value chain model consists of four cycles [DCH05, McC03]. The acquisition cycle, which is shown at the left-hand side of fig. 6.2, models the interface with suppliers via a complex transition (called Acquisition_Cycle), representing a duality between the Product_Inflow transition and the Money_Outflow transition. This duality reflects an axiom of the REA ontology that states that *"All events effecting an outflow must be eventually paired in duality relationships with events effecting an inflow and vice-versa."* [GM04] This means that economic rationale requires adequate compensation for the economic resources that leave an economic unit (i.e. the supplier's products must be paid for by the enterprise) and dictates the existence of requiting economic events for economic events that create claims and liabilities. Analogously, the revenue cycle, which is represented as the complex transition Revenue_Cycle at the right-hand side of fig. 6.2, shows the balance between the value of goods sold and the revenues that remunerate them. Similarly, the conversion cycle, which is represented as the complex transition Conversion_Cycle at the upper side of fig. 6.2, models the business unit in which the value creating transformation processes take place. Finally, the financing cycle, which is represented as the complex transition Finance_Cycle at lower side of fig. 6.2, models the business unit in which the process of approving payments and controlling budgets takes place. The template in fig. 6.2 can be extended by adding more acquisition, conversion, revenue financing cycles and transfers (e.g. the payroll cycle [DCH05] that exchanges labour and money (i.e. wages) between the enterprise and its employees).

Next to the value chain (i.e. an aggregate of acquisition, conversion, revenue and financing cycles), which shows that the balance between incoming and outgoing money and product flows is essential for the sustainability of an economic unit's economic activities, the value chain model in fig. 6.2 is equipped with transfer events that represent the parts of an exchange process that an economic unit is accountable for. Fig. 6.2 assumes that the enterprise is accountable for both transfers that relate to its revenue cycle (i.e. product outflow and money inflow). The complete transfer economic event is modeled by connecting the decrement side of a transfer event under control of one economic unit (e.g. the TransferProduct transition between the OutflowProductO place and the ProductCust place on the right-hand side of fig. 6.2) to its increment side, which is under the control of a subsequent economic unit (and which would be shown, e.g. as the TransferProduct transition between the ProductSupp place and the InputProduct place, in the value chain model of that economic unit). Together, both transfer events form an economic event template [LP09], which starts with an economic resource decrement (i.e. outflow) and ends with an economic resource increment (i.e. inflow), that al-

lows integrating the value chain models of different supply chain partner into one simulation model that can subsequently be shown in a reduced form (i.e. going one level up in the Petri-net decomposition hierarchy) as a collaboration space model.

Fig. 6.3 depicts a simple business process which is part of the 'Conversion Cycle' business unit in fig. 6.2 . It models a transformation economic event, which starts with an economic resource decrement and ends with an economic resource increment, that is triggered by an order (i.e. a business resource) and gives rise to the creation of an invoice (i.e. another business resource). The business events that are part of the economic event can be represented by further decomposing the TransformationProduct transition. Similarly, the business events that are part of the 'TransferProduct' and 'TransferMoney' economic events in fig. 6.2 can be represented in business process models that underlie these transfer economic events.

6.4 Exhibit: alternative Value Chain Designs

This section applies the templates presented in the preceding section, to evaluate four alternative theoretical business designs, on their financial performance (e.g. turnover, debt, value of work in progress). The initial value chain model (i.e. Design 0) has an acquisition cycle in which the supplier finances its sales to the enterprise for a 30 day period (i.e. term of payment), a conversion cycle that takes 50 days, a revenue cycle that offers a 30 day payment term to the customer, a product transfer that takes 3 days, a money transfer that takes 2 days and a financing cycle that takes 5 days. The first redesign (i.e. Design 1) halves the duration of the conversion cycle (i.e. 50 to 25 days). The second redesign (i.e. Design 2) reduces the term of payment that is offered to the customer with 25 days (i.e. 30 to 5 days) and the third redesign shows the effect of a supplier that reduces the term of payment (i.e. 30 to 5 days) for the enterprise.

To abstract from influences other than value chain design, all underlying business processes have a standard deviation of 0 days for their duration (i.e. service time). The elimination of business process service time volatility prevents the redesign effect from being masked by the aggregate volatility of all service times in the value chain. The simulation models for the alternative designs also abstract from price volatility, debt default and profit margins. Eliminating price volatility prevents the financial values (e.g. liabilities, turnover) in the models from being distorted by price fluctuations. Debt default would increase the need for external funding, as default implies that customers do not provide the required financial means to compensate the assets that were transferred to them. Finally, profit margins reduce the need for external funding as they imply that the customers provide more financial means to the enterprise then required for replacing the assets that were transferred to these customers. As a result of these abstractions, the remain-

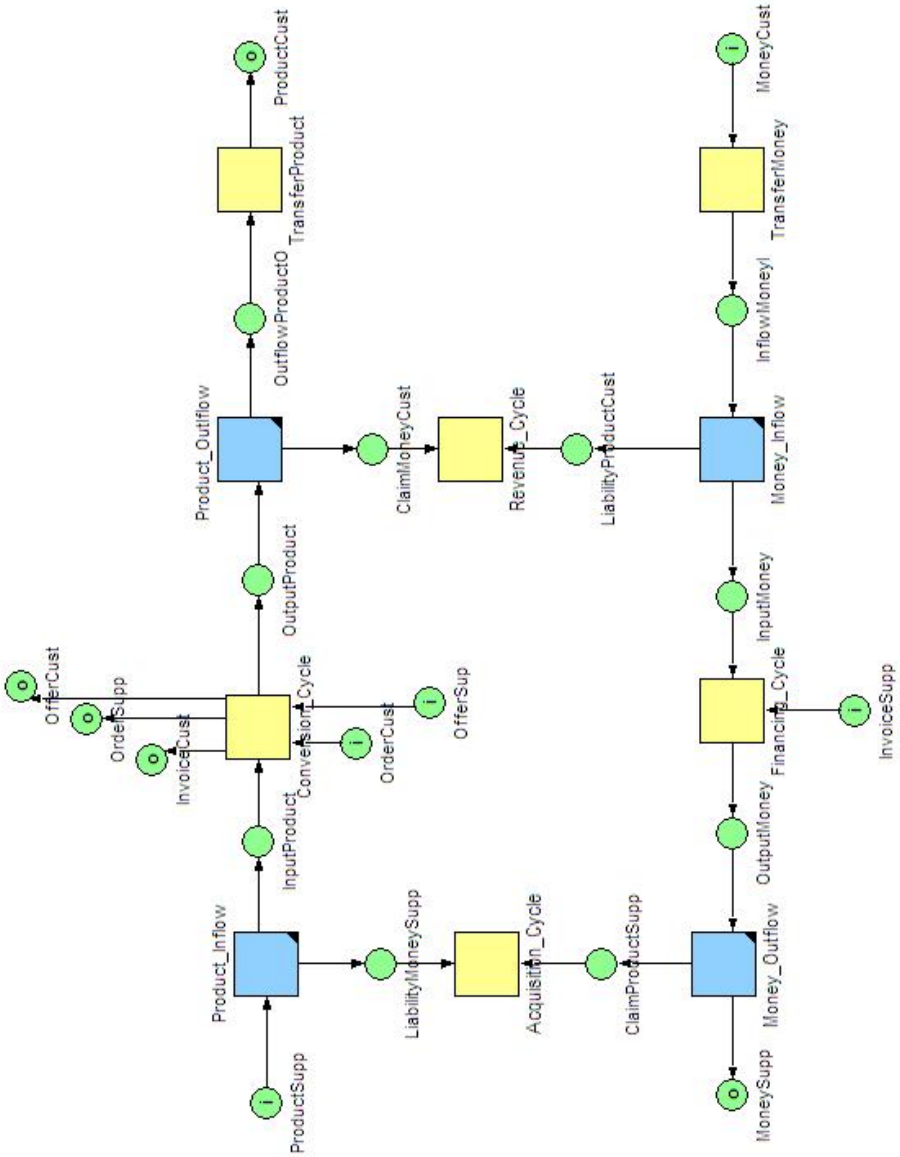

Figure 6.2: Value chain model for the enterprise in the collaboration space model for fig. 6.1

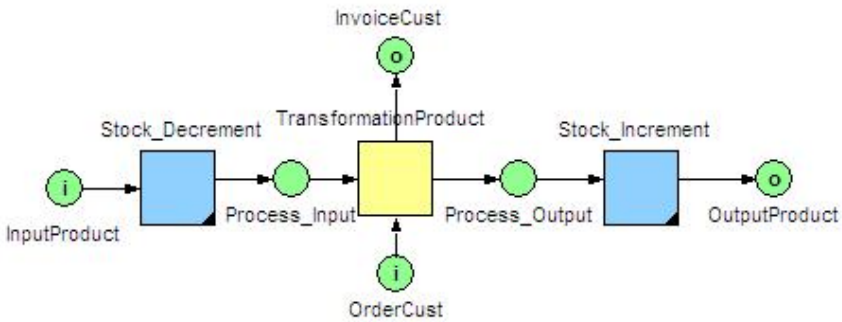

Figure 6.3: Business process model for the conversion cycle of the enterprise in the value chain model of fig. 6.2

ing variables in the simulation models are the value chain design and the inter-arrival time between the product deliveries in the supply driven value chain (i.e. assuming product supply is the trigger for all other economic events in the value chain model).

Table 6.2 shows simulation results that originate in a set of 10 simulation runs with 300+ cases each created by a token generator that varies the inter-arrival time randomly and assigns a value of 1, representing € 1 000 000, to each individual token. The original design (i.e. Design 0) has a value chain cycle time[4] of 90 days (i.e. 50 + 3 + 30 + 2 + 5). The liabilities equal the value of all assets in the value chain (i.e. work in progress, accounts receivable and cash) for which no requiting money outflow has occurred. The need for external funding (i.e. NEF) then represents the liabilities that is not financed by supply chain partners (mostly the supplier, through terms of payment) and for which other funding (e.g. equity, bank loans) needs to be found. Turnover then aggregates the value of all tokens that flow through the value chain during one simulated year (i.e. 360 days). The receivables aggregate the value of all tokens that are occupied by the revenue cycle (during 30 days), work in progress (i.e. WIP) aggregates the value of all tokens that are occupied by the conversion cycle (during 50 days) and cash aggregates the value of all tokens in the financing cycle (during 5 days).

The first redesign results in a reduction of the value chain cycle time by 25 days. Since the underlying business processes have no variance, the cycle time's standard deviation is 0. This conversion cycle service time reduction decreases the average value of the liabilities by the same percentage as the value chain cycle time

[4]The time between the acquisition of a raw material and the moment the money that is generated by these raw materials, after conversion and sale, is available for a subsequent acquisition of raw materials.

	Design 0		Design 1		Design 2		Design 3	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Cycle Time*	90	0	65	0	65	0	90	0
Liabilities**	22,4	4,5	16,4	3,8	16,9	3,9	22,4	4,5
NEF**	15,1	3,7	9,0	2,8	9,4	2,9	21,2	4,4
Turnover**	89,5	/	89,9	/	92,7	/	89,5	/
Receivables**	9,0	2,9	9,0	2,8	3,0	1,6	9,0	2,9
WIP**	12,7	3,5	6,6	2,4	13,1	3,5	12,7	3,5
Cash**	1,7	1,2	1,7	1,2	1,8	1,2	1,7	1,2

Table 6.2: Simulation results for alternative value chain designs (: in days; ** in € 1 000 000)*

is reduced (i.e. 28%), while the average need for external funding is reduced by 40%. Both these reductions originate in a 50% reduction of the work in progress (i.e. WIP), which represents the value of the assets that are involved in conversion processes. This reduction is the result of a reduced conversion cycle service time and a stable product supply and flow. The second redesign also reduces the value chain cycle time by 25 days and the average liabilities with 28% and the need for external funding with 40% due to a 67% decrease in the value of the accounts receivable, which originates in a shorter term of payment and a stable revenue flow. Finally, redesign 3 shows the effect of a reduced term of payment on an enterprise's need for external funding, as the need for external funding increases with 40% while the total value of the enterprise's liabilities remains unchanged.

This exhibit motivates business process redesign efforts [RLM05] as it shows that shorter service times in the conversion cycle reduce an enterprise's liabilities and need for external funding, without burdening its customers with an extra need for external funding which may result in a higher cost of capital or increased illiquidity and insolvency risk. This change in the need for external funding for both the seller and the buyer reveals an interference between the seller's revenue cycle and the buyer's acquisition cycle through the shared variable 'term of payment'. Also aggregate effects of business model changes could be simulated if multiple redesigns are incorporated in a single simulation model run. Through the shared variable 'term of payment', also the propagation of liquidity constrains throughout entire collaboration spaces could be simulated (e.g. liquidity constraints for a customer may evoke delayed payments to its supplier, which increases this supplier's need for external funding, which may lead to delayed payments of this supplier to its own suppliers etc., see e.g. [LP09])

In a day-to-day planning approach, value chain models as presented here could be used to predict the prolonged term of payment for each individual invoice in the waiting line, based upon a priority algorithm (e.g. first-in-first out, largest debt) and a prediction of the expected incoming money flows, based on the due date of

individual claims on customers or predicted sales and terms of payment offered to customers.

6.5 Conclusion and Future Research

This paper showed that the REA enterprise ontology can be used to guide the construction of business simulation models, including models for simulating collaboration spaces between supply chain partners, value chains and business processes. The main advantage of using the REA ontology to construct business simulation models is that the economic rationale expressed by this ontology facilitates the integration of financial flows in simulation models that are traditionally focused on logistics flows of goods and services, information flows and workflows. The resulting models can support assessing the financial consequences of specific business process and supply chain configurations. For example, the effect of a customer that delays its payments or a supplier that shortens or lengthens its term of payment on an enterprise's cash position can be assessed.

The simulation language used for REA ontology-based business simulation models is the one of the ExSpect tool and is based on the high-level Petri-net formalism. The use of hierarchical Petri-nets allows constructing decomposition hierarchies of discrete event simulation models, such that simulation models at different levels of aggregation (i.e. business process, value chain, collaboration space) can easily be integrated. The use of colored Petri-nets was key to injecting economic rationale into simulation models built using the REA ontology and the ExSpect tool, as these colored Petri-nets allow the attribution of economic value to tokens representing economic resources. Finally, timed Petri-nets allow for the representation of actual business process characteristics (e.g. service time).

Since the simulation exhibit presented in this paper abstracts from many natural business variables (e.g. service time variability) and economic variables (e.g. price fluctuations), we intend to elaborate more realistic business cases that demonstrate the use of discrete event simulation models in conceiving and evaluating (alternative) business designs.

6
Summary

Objectives

Providing support for high-level and operational management simulation applications.

Approach

First, the interactions between trading partners are modeled in collaboration space models that instantiate the ExSpect modeling language and commit to the REA ontology. Second, the rationale of each trading partner's internal processes is modeled in value chain models that also instantiate the ExSpect modeling language and commit to the REA ontology. Third, individual business processes in a value chain are modeled with the ExSpect language.

Findings

The explicit connection between logistic and financial processes increases the configurability and reusability of simulation models that commit to the REA ontology.

Contribution

Allowing simulation model builders to inject an economic rationale into their simulation models, and in particular integrating financial flow models with more traditional logistic flow and workflow models.

Practical Implications

Predicting the financial performance of an enterprise, based on both operational and financial parameters.

7
Conclusion

In this dissertation, we have developed a new modeling approach for developing and integrating conceptual models for the business domain by articulating models as ontological instantiations [AK03] of the REA ontology [GM02]. In addition, we have demonstrated that the REA ontology is not only applicable for conceptual data modeling, but also for the conception and integration of business simulation models.

First, we have introduced an ontology-based pattern representation technique that generalizes REAs structuring orientation [DM97] for other ontologies to facilitate the interpretation of data modeling patterns. The first paper of the data model section demonstrated the use of the structuring orientation for the interpretation of well-known conceptual data modeling patterns and the abstraction mechanisms they contain. This structuring orientation visualizes pattern overlap and pattern integration possibilities. Since the generalized structuring orientation enables the use of ontologies other than REA, different ontologies can be used to structure patterns. Consequently, the generalized structuring orientation enables the development of tool support for pattern selection, based on the ontologies that are selected to structure patterns, and pattern integration, using the visual interpretation support that is provided by the structuring orientation. Where the illustration in the paper is limited to the core constructs of the REA and UFO ontologies, we have demonstrated the feasibility of pattern structuring. In the future we hope to demonstrate the utility of a multi-dimensional structuring orientation, as opposed to the structuring orientation of the REA ontology whose benefits for model understanding have been demonstrated before [PMGP07]. Such a multi-dimensional

structuring orientation requires the structuring of larger pattern catalogues and pattern languages, using multiple ontologies as structuring dimensions, and potentially also various modeling languages. For example, the presented framework can be applied to conceptual data modeling patterns from sources other than Batra [Bat05] (e.g., Fowler [Fow97]). To account for the different domains covered by these patterns, other domain ontologies will need to be added to the framework. Furthermore, analysis patterns are not limited to conceptual data modeling patterns (e.g., UML sequence diagrams based on the REA ontology have been presented [BS08]). Therefore, structuring other types of conceptual models (e.g., Harel state charts, event diagrams) also merits investigation.

Second, we have introduced an information reference model that is an instantiation of the REA ontology. However, it is not a traditional REA ontology instantiation as it integrates the independent and trading partner view, for which distinct models have been articulated hitherto. Where the independent view observes inter-enterprise events from the perspective of a person that is not involved in these events, the trading partner view regards inter- and intra-enterprise events from the perspective of a party taking part in those events. [ISO07] To account for both the trading partner and independent view, our reference information model represents the opposing view of trading partners explicitly. The explicit representation of a trading partner's view on economic events enables the identification of events across enterprise boundaries through the incorporation of increment and decrement semantics and to (i.e., perceived as increment by) and from (i.e., perceived as decrement by) semantics in a single data model. This single data model enables for a central administration of data generated by multiple enterprises. Additionally, the flexibility of the reference information model is expected to support data integration and interoperability. Integration can be achieved when partners use the reference information model as a conceptual basis for their information systems and agreement is reached about a minimal set of attributes (e.g., identifiers). Interoperability can be realized when the reference information model is used as a conceptual foundation for a mediator between individual enterprise information systems in a multi-database environment [WM04]. Such integration is expected to facilitate the creation of consolidated annual accounts through an explicit representation of transactions (i.e., from-to) between consolidation partners (e.g., subsidiaries).

Third, the REA reference information model can be implemented to achieve traceability for product and money flows. Where the most advanced traceability approaches are limited to product flows [BCMT08], our reference information model is demonstrated to track and trace both the product flows and the money flows that remunerate them, integrating them through the business transactions that comply with the economic rationale. Furthermore, the reference information models ability to construct tracks and traces of commitments enables monitoring

critical paths in supply chains (e.g., which delays in the acquisition and production processes of my supply chain partners are acceptable without delaying my processes?). Such monitoring of critical paths in supply chains could also support the elaboration of mitigation scenarios (e.g., what alternatives do I have to prevent delay in my processes when a delivery is delayed or cancelled?). Next to traditional supply chain management, the reference information model can provide the basis for estimating the (potential) financial consequences of such delays and cancellations (e.g., delayed or reduced revenue). Moreover, such a traceability infrastructure can support marketing raw materials (e.g., prosciutto di Parma, appellation d'origine controlee, local produce).

As the REA ontology finds its origin in accounting, the reference information model could also be used to generate annual accounts. Combining traceability with accounting, the reference information model could also provide support for countering organized crime. For example, terrorists are alleged to fund their activities with counterfeit. [Dek07, Low06] Counterfeit can be fought through product authentication (i.e., traceability) and identifying the destination of the generated money flows is possible when traceability for money flows is combined with accounting. Since the implementation of a full-scale tracking and tracing application that also incorporates accounting involves the protection of strategic information (e.g., profit margins) and the enforcement of privacy policies (e.g., purchases of a natural person), the proof of concept presented in the third paper of the data modeling section is limited to an illustration of the model's expressive power for tracking and tracing at data level. This approach abstracts from the model's accounting potential and the prevention of unauthorized access to strategic or privacy related information. In the future we will demonstrate the reference information model's accounting potential, illustrating consolidation accounting. Subsequently, we will address the data protection issues that are required to implement a pilot application.

Fourth, REA-based workflow models were presented that allow for simulating the effect of business (process) model changes (e.g., reduced cycle time, changed terms of payment) on an enterprise's liquidity. Since the REA ontology incorporates separate independent and a trading partner views (and we have shown how to integrate these views into one model in the data model section of the dissertation), transactions between trading partners as well as value chains of individual trading partners can be modeled. Additionally, it is shown how these transaction and value chain models can be integrated to construct workflow models that represent product and money flows through supply chains. These models can be used to analyze the self-funding potential of supply chain configurations (i.e., To what extent can money flows in the supply chain finance the product flows they remunerate?). Such simulation models can be used to analyze potential proliferation of liquidity constraints in supply chains (e.g., Will a supplier fail if his customer

fails to settle a due payment?). Furthermore, such simulation models can be used to evaluate alternative solution designs (e.g., Will longer terms of payment bring relief or cause more trouble?) Where the presented paper illustrates how such simulation models can be constructed, we would like to incorporate such simulation models in decision support tools for managers in the future. Such an integration should provide them with more insight (e.g., What is the actual term of payment?) for the prevention or mitigation of liquidity constraints.

Fifth, the knowledge gained through developing REA-based workflow models was used to construct layered business simulation models. To enhance the expressive power of the simulations models the Expect tools library for logistic simulations was adapted such that it also accounted for the value of the simulated discrete product flows and the balancing money flows. The upper-layer of the simulation model hierarchy addresses the collaboration space [ISO07] in which trading partners interact with each other, representing supply chains and value networks from the independent perspective (i.e., from the perspective of an observer that is not taking part in any transaction). The middle-layer of the simulation model hierarchy represents the value chain of one trading partner. Such a value chain represents a trading partner's acquisition, revenue and manufacturing and financing processes from the perspective of that trading partner. [DCH05, McC03] Finally, the bottom-layer of the simulation model hierarchy addresses traditional business process models, which represent operational and administrative processes (e.g., manufacturing processes [BDBR03]). Together, these three modeling layers enable the modeling of entire value systems, which incorporate both the processes inside each trading partner as the interactions between trading partners [PM85], through the integration of business process models in value chain models and value chain models in supply chain models.

As the REA ontology abstracts from business processes, the bottom layer is only addressed marginally (i.e., sufficiently to illustrate the integration mechanism) in the dissertation. However, more traditional business process modeling approaches [AS04, Gia01] focus on business process representation. Integration of value system models [PM85] and traditional business process models in an integrated business modeling method is a logical next step in the development process. Such an integration results in the articulation of models that incorporate financial parameters (e.g., turnover) as well as operational parameters (e.g., cycle time), as has been demonstrated in the dissertation. As the REA ontology accounts for both the independent and trading partner view (though articulated in separate models), the modeling of entire value systems is expected to be supported. Although the three layers can be integrated, they can also be modeled independently (e.g., a collaboration space can be modeled abstracting from the underlying value chains, business processes can be modeled without considering the value systems they are part of). As mentioned above, we would like to incorporate such layered business

(process) simulation in a management tool.

In general, we have demonstrated that ontologies can serve as a connector technology [Lin04a] that improves the integration capabilities of conceptual models. More precisely, we have developed a new ontology-based conceptual modeling approach, which finds it origin in the REA ontology, for both data and simulation models representing phenomena of interest in the domain of business. First, we showed how a domain (i.e., vertical) and upper-level (i.e., horizontal) ontology [Lin04b] can support model development and integration through patterns. Second, we have developed a reference information model that organizes enterprise information such that interoperability and integration of inter- and intra-enterprise information systems is facilitated. Third, we have demonstrated how the reference information model can provide the conceptual basis for the development of an innovative tracking an tracing application, which extends the current scope of traceability towards the tracking and tracing of committed product and money flows through entire value systems. Fourth, we showed that the REA ontology can be instantiated to build and integrate various kinds of business simulation models.

This dissertation positioned the REA ontology as a conceptual foundation for both conceptual data models and business simulation models. Since the use of REA for conceptual data modeling represents the major part of REA's momentum, our future efforts will mostly concentrate on gaining momentum in simulation modeling. Eventually, we would also like to use REA for business intelligence (e.g., analyzing data logs in terms of financial parameters) by combining its ability to represent value systems, as has been demonstrated for simulation models in chapter 5 and 6, with its role as a conceptual foundation for data models, as has been demonstrated in chapter 2 to 4. This approach should not only provide business people with robust data models, but also with decision support tools that use these data models to provide them with the information they need (i.e., financial and/or operational) for evaluation (e.g., analyzing data logs) and prognosis (e.g., generating data for simulating future business (process) lay-outs).

Bibliography

[20505] ISO DTR 20514. Health informatics - electronic health record: definition, scope and context. *International Organization for Standardization*, —2005—.

[ABS05] Solomon Antony, Dinesh Batra, and Radhika Santhanam. The use of a knowledge-based system in conceptual data modeling. *Decision Support Systems*, 41(1):176–188, —2005—.

[AG08] Samuil Angelov and Paul Grefen. An e-contracting reference architecture. *Journal of Systems and Software*, 81(11):1816–1844, —2008—. 0164-1212 doi: DOI: 10.1016/j.jss.2008.02.023.

[AK03] Colin Atkinson and Thomas Kuhne. Model-driven development: a metamodeling foundation. *Software, IEEE*, 20(5):36–41, —2003—. 0740-7459.

[AS04] Ruth Sara Aguilar-Saven. Business process modelling: Review and framework. *International Journal of Production Economics*, 90(2):129–149, —2004—.

[ASM96] ASMA. Asma product database, release 7, —1996—.

[Bah99] Ali Bahrami. *Object oriented systems development*. Irwin/McGraw-Hill, Boston, Mass. ; London, —1999—.

[Bat05] Dinesh Batra. Conceptual data modeling patterns: Representation and validation. *Journal of Database Management*, 16(2):84–106, —2005—.

[BCMT08] Alessio Bechini, Mario G. C. A. Cimino, Francesco Marcelloni, and Andrea Tomasi. Patterns and technologies for enabling supply chain traceability through collaborative e-business. *Information and Software Technology*, 50:342–359, —2008—.

[BDBR03] K. D. Barber, F. W. Dewhurst, R. L. D. H. Burns, and J. B. B. Rogers. Business-process modelling and simulation for manufac-

turing management: A practical way forward. *Business Process Management*, 9:527–542, —2003—.

[Ber08] Robert J. Bernabucci. Supply chain gains from integration. *Financial Executive*, 24(3):46–48, —2008—.

[BG06] Perakath Benjamin and Michael Graul. A framework for adaptive modeling and ontology-driven simulation (famos). In *Enabling Technologies for Simulation Science X*, volume 6227, pages 622705–11, Orlando (Kissimmee), FL, USA, —2006—. SPIE.

[BG10] Matt Bassett and Leslie Gardner. Optimizing the design of global supply chains at dow agrosciences. *Computers & Chemical Engineering*, 34(2):254–265, —2010—.

[BGHS+05] Paolo Bresciani, Paolo Giorgini, Brian Henderson-Sellers, Graham Low, Michael Winikoff, Giancarlo Guizzardi, and Gerd Wagner. Towards ontological foundations for agent modelling concepts using the unified fundational ontology (ufo). In *Agent-Oriented Information Systems II*, volume 3508 of *Lecture Notes in Computer Science*, pages 110–124. Springer Berlin / Heidelberg, —2005—.

[BHSLPLB03] Franck Barbier, Brian Henderson-Sellers, Annig Le Parc-Lacayrelle, and Jean-Michel Bruel. Formalization of the whole-part relationship in the unified modeling language. *IEEE Transactions on Software Engineering*, 29(5):459–470, —2003—.

[BK91] Rajiv D. Banker and Robert J. Kauffman. Reuse and productivity in integrated computer- aided software engineering: An empirical study, —1991—.

[BN96] Adam Brandenburger and Barry Nalebuff. *Co-opetition*. Doubleday, New York, 1st edition, —1996—.

[Bor97] Willem Nico Borst. *Construction of engineering ontologies for knowledge sharing an reuse*. Construction of engineering ontologies for knowledge sharing an reuse, Universiteit Twente, —1997—.

[BPM06] Perakath Benjamin, Mukul Patki, and Richard Mayer. Using ontologies for simulation modeling, —2006—. Proceedings of the 38th conference on Winter simulation Monterey, California.

[Bro83] Michael L. Brodie. Association: A database abstraction for semantic modelling, —1983—. Proceedings of the Second International Conference on the Entity-Relationship Approach to Information Modeling and Analysis.

[BS08] Dinesh Batra and Thant Sin. The ready model: Patterns of dynamic behavior in rea-based accounting applications. *Information Systems Management*, 25:200–210, —2008—.

[BSL+10] Yeon-Sik Bong, Woo-Jin Shin, A-Reum Lee, Young-Soo Kim, Kangjoo Kim, and Kwang-Sik Lee. Tracing the geographical origin of beefs being circulated in korean markets based on stable isotopes. *Rapid Communications in Mass Spectrometry*, 24(1):155–159, —2010—.

[Bus03] Christoph Bussler. *B2B-integration*. Springer, New York ; London, —2003—.

[BW04] Dinesh Batra and Nicole A. Wishart. Comparing a rule-based approach with a pattern-based approach at different levels of complexity of conceptual data modelling tasks. *International Journal of Human-Computer Studies*, 61(4):397–419, —2004—.

[CDA99] Silvana Castano and Valeria De Antonellis. A discovery-based approach to database ontology design. *Distributed and Parallel Databases*, 7(1):67–98, —1999—.

[CFMK07] Dayana Cope, Mohamed Sam Fayez, Mansooreh Mollaghasemi, and Assem Kaylani. Supply chain simulation modeling made easy: an innovative approach, —2007—. 1351879 1887-1896.

[Che76] Peter Chen. The entity-relationship model: toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, —1976—.

[CKJJ04] Hyunbo Cho, Boonserm Kulvatunyou, Hanil Jeong, and Albert Jones. Using business process specifications and agents to integrate a scenario-driven supply chain. *International Journal of Computer Integrated Manufacturing*, 17:546–560, —2004—.

[CM98] Ray Calder and Pamela Marr. A beef producer initiative in traceability: Scottish borders tag. *Supply Chain Management*, 3(3):123–123, —1998—.

[CMO95] Jia-Lin Chen, Dennis McLeod, and Daniel O'Leary. Domain-knowledge-guided schema evolution for accounting database systems. *Expert Systems with Applications*, 9(4):491–501, —1995—.

[CNM97] Peter Coad, David North, and Mark Mayfield. *Object models : strategies, patterns, and applications*. Yourdon Press computing series. Yourdon Press, Upper Saddle River, N.J., 2nd edition, —1997—.

[Cou06] Supply-Chain Council. Supply-chain operations reference-model, —2006—. http://www.supply-chain.org/.

[CP96] Amelia L. Craighill and Jane C. Powell. Lifecycle assessment and economic evaluation of recycling: A case study. *Resources, Conservation and Recycling*, 17(2):75–96, —1996—.

[CS95] Alistair E. Campbell and Stuart C. Shapiro. Ontological mediation: An overview. *IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing*, —1995—.

[CS08] Kim Church and Rod Smith. Rea ontology-based simulation models for enterprise strategic planning. *Journal of Information Systems*, 22(2):301–329, —2008—.

[Dav08] Rob Davis. *Aris design platform : advanced process modelling and administration*. Springer, New York, —2008—.

[DB07] Rob Davis and Eric Brabänder. *ARIS design platform : getting started with BPM*. Springer, New York ; London, —2007—.

[DCH05] Cheryl L. Dunn, J. Owen Cherrington, and Anita Sawyer Hollander. *Enterprise information systems : a pattern-based approach*. McGraw-Hill/Irwin, Boston, 3rd edition, —2005—.

[DDRW06] Jin Dong, Hongwei Ding, Changrui Ren, and Wei Wang. Ibm smartscor - a scor based supply chain transformation platform through simulation and optimization techniques, —2006—. 1218234 650-659.

[Dek07] Donald Dekieffer. Terrorist links to commercial fraud in the united states. *Business Credit*, 109(1):24–25, —2007—.

[Die06] Jan L. G. Dietz. *Enterprise ontology : theory and methodology*. Springer, Berlin ; New York, —2006—.

[DM97] Cheryl L. Dunn and William E. McCarthy. The rea accounting model: Intellectual heritage and prospects for progress. *Journal of Information Systems*, 11(1):31–51, —1997—.

[Dou07] Guy Doumeingts. *Enterprise interoperability : new challenges and approaches*. Springer, Berlin ; London, —2007—.

[DS95] Guido Dedene and Monique Snoeck. Formal deadlock elimination in an object oriented conceptual schema. *Data & Knowledge Engineering*, 15(1):1–30, —1995—.

[ebX] ebXML. enabling a global electronic market. http://www.ebxml.org/.

[EP00] Hans-Erik Eriksson and Magnus Penker. *Business modeling with UML : business patterns at work*. John Wiley & Sons, New York, —2000—.

[Eur] United Nations Economic Commission for Europe. United nations directories for electronic data interchange for administration, commerce and transport. http://www.unece.org/trade/untdid/welcome.htm.

[EW05] Joerg Evermann and Yair Wand. Toward formalizing domain modeling semantics in language syntax. *Software Engineering, IEEE Transactions on*, 31(1):21–37, —2005—. 0098-5589.

[FFFvL97] Martin S. Feather, Stephen Fickas, Anthony Finkelstein, and Axel van Lamsweerde. Requirements and specification exemplars. *Automated Software Engineering*, 4(4):419–438, —1997—.

[FL03] Peter Fettke and Peter Loos. Ontological evaluation of reference models using the bunge-wand-weber model. In *Americas Conference on Information Systems*, Tampa, Florida, —2003—.

[FM07] Frederico Fonseca and James Martin. Learning the differences between ontologies and conceptual schemas through ontology-driven information systems. *Journal of the Association for Information Systems*, 8:129–142, —2007—.

[FMM06] Dimitris Folinas, Ioannis Manikas, and Basil Manos. Traceability data management for food chains. *British Food Journal*, 108(8):622–633, —2006—.

[Fos22] William T. Foster. The circuit flow of money. *American Economic Review*, 12(3):460, —1922—.

[Fow97] Martin Fowler. *Analysis patterns : reusable object models*. The Addison-Wesley series in object-oriented software engineering. Addison Wesley, Menlo Park, Calif., —1997—.

[FRM05] Mohamed Fayez, Luis Rabelo, and Mansooreh Mollaghasemi. Ontologies for supply chain simulation modeling, —2005—. 1163120 2364-2370.

[GAH01] Cesar Garita, Hamideh Afsarmanesh, and Louis O. Hertzberger. The prodnet cooperative information management for industrial virtual enterprises. *Journal of Intelligent Manufacturing*, 12(2):151–170, —2001—.

[Gar92] Gartner. Sometimes you gotta break the rules, November 23 —1992—.

[GCC+04] Daniela Grigori, Fabio Casati, Malu Castellanos, Umeshwar Dayal, Mehmet Sayal, and Ming-Chien Shan. Business process intelligence. *Computers in Industry*, 53(3):321–343, —2004—.

[GCG+99] H. M. J. Goldschmidt, M. J. T. Cox, R. J. E. Grouls, W. A. J. H. Van de Laar, and G. G. Van Merode. Design and positioning of a reference information model for clinical laboratories. *Laboratory Automation & Information Management*, 34(1):1–6, —1999—. 1381-141X.

[GG95] Nicola Guarino and Pierdaniele Giaretta. Ontologies and knowledge bases: towards a terminological clarification. In N. Mars, editor, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, page 314. IOS Press, —1995—.

[GHJV95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns : elements of reusable object-oriented software*. Addison-Wesley professional computing series. Addison-Wesley, Reading, Mass., —1995—.

[Gia99] Ronald E. Giachetti. A standard manufacturing information model to support design for manufacturing in virtual enterprises. *Journal of Intelligent Manufacturing*, 10(1):49–60, —1999—.

[Gia01] George M. Giaglis. A taxonomy of business process modeling and information systems modeling techniques. *International Journal of Flexible Manufacturing Systems*, 13(2):209–228, —2001—.

[Gil69] Glenn Gilman. The manager and the systems concept. *Business Horizons*, 12(4):19–28, —1969—.

[GJH+10] Hector Gonzalez, Han Jiawei, Cheng Hong, Li Xiaolei, Diego Klabjan, and Wu Tianyi. Modeling massive rfid data sets: A gateway-based movement graph approach. *Knowledge and Data Engineering, IEEE Transactions on*, 22(1):90–104, —2010—.

[GLP08] Frederik Gailly, Wim Laurier, and Geert Poels. Positioning and formalizing the rea enterprise ontology. *Journal of Information Systems*, 22(2):219–248, —2008—.

[GM86] Graham Gal and William E. McCarthy. Operation of a relational accounting system. *Advances in Accounting*, 3:83–112, —1986—.

[GM00] Guido L. Geerts and William E. McCarthy. Augmented intensional reasoning in knowledge-based accounting systems. *Journal of Information Systems*, 14(2):127, —2000—.

[GM02] Guido L. Geerts and William E. McCarthy. An ontological analysis of the economic primitives of the extended-rea enterprise information architecture. *International Journal of Accounting Information Systems*, 3(1):1–16, —2002—.

[GM04] Guido L. Geerts and William E. McCarthy. The ontological foundation of rea enterprise information systems, —2004—.

[GM06] Guido L. Geerts and William E. McCarthy. Policy-level specifications in rea enterprise information systems. *Journal of Information Systems*, 20(2):37–63, —2006—.

[GMB+05] Marius Gilbert, Andy Mitchell, David Bourn, Joshua Mawdsley, Richard Clifton-Hadley, and William Wint. Cattle movements and bovine tuberculosis in great britain. *Nature*, 435:491–496, —2005—.

[Gor02] Jaap Gordijn. *Value-based requirements Engineering: Exploring innovative e-commerce ideas*. PhD thesis, Free University of Amsterdam, —2002—.

[GP07] Frederik Gailly and Geert Poels. Ontology-driven business modelling: Improving the conceptual representation of the rea-ontology. *26th International Conference on Conceptual Modeling (ER 2007)*, Lecture Notes in Computer Science 4801:407–422, —2007—.

[GR00] Peter Green and Michael Rosemann. Integrated process modeling: An ontological evaluation. *Information Systems*, 25(2):73–87, —2000—.

[GRIM07] Peter Green, Michael Rosemann, Marta Indulska, and Chris Manning. Candidate interoperability standards: An ontological overlap analysis. *Data & Knowledge Engineering*, 62(2):274–291, —2007—.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, —1993—.

[GST+08] A. M. Genaidy, R. Sequeira, T. Tolaymat, J. Kohler, and M. Rinder. An exploratory study of lead recovery in lead-acid battery lifecycle in us market: An evidence-based approach. *Science of The Total Environment*, 407(1):7–22, —2008—.

[Gua98] Nicola Guarino. Formal ontology and information systems. In *Proceedings of FOIS'98*, pages 3–15, Trento, Italy, —1998—. IOS Press.

[Gui07] Giancarlo Guizzardi. On ontology, ontologies, conceptualizations, modeling languages, and (meta)models. In O. Vasilecas, J. Eder, and A. Caplinskas, editors, *7th International Baltic Conference on Databases and Information Systems*, volume 155 of *Frontiers in Artificial Intelligence and Applications*, pages 18–39, Amsterdam, —2007—. I O S Press.

[GV05] Guy H. Gessner and Linda Volonino. Quick response improves returns on business intelligence investments. *Information Systems Management*, 22(3):66–74, —2005—.

[GVF07] Guy H. Gessner, Linda Volonino, and Lynn A. Fish. One-up, one-back erm in the food supply chain. *Information Systems Management*, 24:213–222, —2007—.

[GW05] Giancarlo Guizzardi and Gerd Wagner. Some applications of a unified foundational ontology in business modeling. In Peter Green and Michael Rosemann, editors, *Business Systems Analysis with Ontologies*, pages xv, 379 p. Idea group publishing, Hershey PA, —2005—.

[HDC99] Anita Sawyer Hollander, Eric L. Denna, and J. Owen Cherrington. *Accounting, information technology, and business solutions*. Irwin/McGraw-Hill, Boston, MA, 2nd ed. edition, —1999—.

[HDJ08] Michael Zwicky Hauschild, Louise Camilla Dreyer, and Andreas Jørgensen. Assessing social impacts in a life cycle perspective–lessons learned. *CIRP Annals - Manufacturing Technology*, 57(1):21–24, —2008—.

[HHBL01] T. Hastein, B.J. Hill, F. Berthe, and D.V. Lightner. Traceability of aquatic animals. *revue scientifique et technique (International Office of Epizootics)*, 20(2):564–583, —2001—.

[HHKV08] Frantisek Hunka, Miroslav Hucka, Josef Kasik, and Dominik Vymetal. Could a resource be simultaneously a schedule according to the rea ontology?, 2008-06-15 —2008—. 3rd REA technology workshop, Montpellier, France.

[HJA05] Michael Zwicky Hauschild, Jack Jeswiet, and Leo Alting. From life cycle assessment to sustainable production: Status and perspectives. *CIRP Annals - Manufacturing Technology*, 54(2):1–21, —2005—.

[HM00] Robert Haugen and William E. McCarthy. Rea, a semantic model for internet supply chain collaboration, —2000—. http://www.jeffsutherland.org/oopsla2000/mccarthy/mccarthy.htm.

[HMJR04] Alan R. Hevner, Salvatore T. March, Park Jinsoo, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, —2004—.

[Hou01] R. Houston. A computerised database system for bovine traceability. *Revue scientifique et technique (International Office of Epizootics)*, 20(2):652–661, —2001—.

[HPS09] Emma Hartnett, Gregory M. Paoli, and Donald W. Schaffner. Modeling the public health system response to a terrorist event in the food supply. *Risk Analysis*, 29(11):1506–1520, —2009—.

[Hru06] Pavel Hruby. *Model-driven design using business patterns*. Springer, Berlin, —2006—.

[HS06a] Irit Hadar and Pnina Soffer. Variations in conceptual modeling: Classification and ontological analysis. *Journal of the Association for Information Systems*, 7:568–592, —2006—.

[HS06b] Håkan Håkansson and Ivan Snehota. No business is an island: The network concept of business strategy. *Scandinavian Journal of Management*, 22(3):256–270, —2006—. 0956-5221.

[HS06c] Claudia Hess and Christoph Schlieder. Ontology-based verification of core model conformity in conceptual modeling. *Computers Environment and Urban Systems*, 30(5):543–561, —2006—.

[IEE90] IEEE. Ieee std 610.12-1990(r2002). *IEEE Standard Glossary of Software Engineering Terminology*, —1990—.

[Iji75] Yuji Ijiri. *Theory of accounting measurement*. Studies in accounting research. American Accounting Association, Sarasota, Fla., —1975—.

[ISO07] ISO/IEC. Information technology - business operational view part 4: Business transaction scenario - accounting and economic ontology, —2007—.

[JCHC+05] Peter Jones, Colin Clarke-Hill, Daphne Comfort, David Hillier, and Peter Shears. Radio frequency identification and food retailing in the uk. *British Food Journal*, 107(6):356–360, —2005—.

[JCJ08] Hosang Jung, Frank Chen, and Bongju Jeong. Decentralized supply chain planning framework for third party logistics partnership. *Computers & Industrial Engineering*, 55(2):348–364, —2008—.

[JK09] Kurt Jensen and Lars M. Kristensen. *Coloured petri nets : modeling and validation of concurrent systems*. Springer, Dordrecht ; London, —2009—.

[JVvDB03] Monique H. Jansen-Vullers, Christian A. van Dorp, and Adriaan J. M. Beulens. Managing traceability information in manufacture. *International Journal of Information Management*, 23(5):395–413, —2003—. 0268-4012.

[KC05] Ahmad Kayed and Robert M. Colomb. Using bww model to evaluate building ontologies in cgs formalism. *Information Systems*, 30(5):379–398, —2005—.

[KDG+08] Koen Kok, Zsofia Derzsi, Jaap Gordijn, Maarten Hommelberg, Cor Warmer, René Kamphuis, and Hans Akkermans. Agent-based electricity balancing with distributed energy resources, a multiperspective case study, —2008—.

[KFS07] Henry M. Kim, Mark S. Fox, and Arijit Sengupta. How to build enterprise data models to achieve compliance to standards or regulatory requirements (and share data). *Journal of the Association for Information Systems*, 8(2):105–128, —2007—.

[Kriar] Ken Krizner. Boeing marks the part. *Frontline Solutions*, 1:11, Year.

[KS06] Viswanathan Kodaganallur and Shim Sung. Analysis patterns: A taxonomy and its implications. *Information Systems Management*, 23:52–61, —2006—.

[KvdB97] Patricia P. A. A. H. Kandelaars and Jeroen C. J. M. van den Bergh. Dynamic analysis of materials-product chains: An application to window frames. *Ecological Economics*, 22(1):41–61, —1997—.

[LB09a] Alex X. Liu and LeRoy A. Bailey. Pap: A privacy and authentication protocol for passive rfid tags. *Computer Communications*, 32(7-10):1194–1199, —2009—.

[LB09b] Diego M. Lopez and Bernd G. M. E. Blobel. A development framework for semantically interoperable health information systems. *International Journal of Medical Informatics*, 78(2):83–103, —2009—. 1386-5056.

[Leo70] Wassily Leontief. Environmental repercussions and the economic structure: An input-output approach. *Review of Economics & Statistics*, 52(3):262, —1970—.

[Lin04a] David S. Linthicum. Leveraging ontologies: The intersection of data integration and business intelligence, part 1. *DM Review*, 14(6):41–43, —2004—.

[Lin04b] David S. Linthicum. Leveraging ontologies: The intersection of data integration and business intelligence, part 2, 29/04/2010 —2004—. http://www.information-management.com/issues/20040701/1006080-1.html?pg=1.

[LM01] Ora Lassila and Deborah L. McGuinness. The role of frame-based representation on the semantic web. *Nokia Research Center*, —2001—.

[LMR98] Philip Leat, Pamela Marr, and Ceri Ritchie. Quality assurance and traceability - the scottish agri-food industry's quest for competitive advantage. *Supply Chain Management*, 3(3):115–115, —1998—.

[Low06] Peter Lowe. Counterfeiting: links to organised crime and terrorist funding. *Journal of Financial Crime*, 13(2):255–257, —2006—.

[LP09] Wim Laurier and Geert Poels. Simulating liquidity in value and supply chains. In A. Albani, J. Barjis, and Jan L. G. Dietz, editors, *CIAO! 2009 and EOMAS 2009*, volume 34 of *Lecture Notes in Business Information Processing*, pages 40–54, Amsterdam, —2009—. Springer-Verlag.

[LS87] Jill H. Larkin and Herbert A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65–100, —1987—.

[McC] William E. McCarthy. An rea model of an economic exchange. https://www.msu.edu/user/mccarth4/cookie–elmo–basic%20REA.ppt.

[McC79] William E. McCarthy. An entity-relationship view of accounting models. *Accounting Review*, 54(4):667, —1979—.

[McC82] William E. McCarthy. The rea accounting model: A generalized framework for accounting systems in a shared data environment. *Accounting Review*, 57(3):554–578, —1982—.

[McC01] Kevin McCormack. Business process orientation: Do you have it? *Quality Progress*, 34(1), —2001—.

[McC03] William E. McCarthy. The rea modeling approach to teaching accounting information systems. *Issues in Accounting Education*, 18(4):427–441, —2003—.

[MJB06] Sanjay Mahato, Anoop Jain, and V. Balasubramanian. Enterprise systems consolidation. *Information Systems Management*, 23:7–19, —2006—.

[Moe98] T. Moe. Perspectives on traceability in food manufacture. *Trends in Food Science & Technology*, 9(5):211–214, —1998—.

[MPS95] Renate Motschnig-Pitrik and Veda C. Storey. Modelling of set membership: The notion and the issues. *Data & Knowledge Engineering*, 16:147–185, —1995—.

[MS95a] Salvatore T. March and Gerald F. Smith. Design and natural science research on information technology. *Decision Support Systems*, 15(4):251–266, —1995—.

[MS95b] Daniel. L. Moody and Graeme C. Simsion. Justifying investment in information resource management. *Australian Journal of Information Systems*, 3(1):25–37, —1995—.

[MS03] Daniel L. Moody and Graeme G. Shanks. Improving the quality of data models: empirical validation of a quality management framework. *Information Systems*, 28(6):619–650, —2003—.

[MSF+05] A. Mousavi, M. Sarhadi, S. Fawcett, S. Bowles, and M. York. Tracking and traceability solution using a novel material handling system. *Innovative Food Science & Emerging Technologies*, 6(1):91–105, —2005—.

[MSLF02] A. Mousavi, M. Sarhadi, A. Lenk, and S. Fawcett. Tracking and traceability in the meat processing industry: A solution. *British Food Journal*, 104(1):7–19, —2002—.

[MW01] J. McGrann and H. Wiseman. Animal traceability across national frontiers in the european union. *Revue scientifique et technique (International Office of Epizootics)*, 20(2):406–412, —2001—.

[Ode98] James J. Odell. *Advanced object-oriented analysis and design using UML*. SIGS reference library ; 12. Cambridge University Press ; New York : SIGS, Cambridge, —1998—.

[OHS05] Andreas L. Opdahl and Brian Henderson-Sellers. A unified modelling language without referential redundancy. *Data & Knowledge Engineering*, 55(3):277–300, —2005—.

[OHSB01] Andreas L. Opdahl, Brian Henderson-Sellers, and Franck Barbier. Ontological analysis of whole-part relationships in oo-models. *Information and Software Technology*, 43(6):387–399, —2001—.

[O'L04] Daniel E. O'Leary. On the relationship between rea and sap. *International Journal of Accounting Information Systems*, 5(1):65–81, —2004—.

[Oli07] Antoni Olivé. *Conceptual modeling of information systems*. Springer, Berlin ; New York, —2007—.

[OMG08] OMG. Business motivation model, —2008—. www.omg.org.

[OMG09] OMG. Omg unified modeling language (omg uml) infrastructure (version 2.2), —2009—. www.omg.org.

[OPT05] Alexander Osterwalder, Yves Pigneur, and Christopher L. Tucci. Clarifying business models: Origins, present and future of the concept. *Communications of AIS*, 16(2005):1–25, —2005—.

[Ost04] Alexander Osterwalder. *The Business Model Ontology - a proposition in a design science approach*. PhD thesis, University of Lausanne, —2004—.

[Pet01] R.G. Pettitt. Traceability in the food animal industry and supermarket chains. *Revue scientifique et technique (International Office of Epizootics)*, 20(2):584–597, —2001—.

[PM85] Michael E. Porter and Victor E. Millar. How information gives you competitive advantage. *Harvard Business Review*, 63(4):149–160, —1985—.

[PMGP07] Geert Poels, Ann Maes, Frederik Gailly, and Roland Paemeleire. The pragmatic quality of resources-events-agents diagrams: an experimental evaluation. *Information Systems Journal*, —2007—. 10.1111/j.1365-2575.2007.00253.x.

[PR04] Jinsoo Park and Sudha Ram. Information systems interoperability: What lies beneath? *ACM Transactions on Information Systems*, 22(4):595–632, —2004—.

[PS08] Sebastien Pouliot and Daniel A. Sumner. Traceability, liability, and incentives for food safety and quality. *American Journal of Agricultural Economics*, 90:15–27, —2008—.

[Rab03] Steve Rabin. The real-time enterprise, the real-time supply chain. *Information Systems Management*, 20:58, —2003—.

[RBS09] Iris Reinhartz-Berger and Arnon Sturm. Utilizing domain models for application design and validation. *Information and Software Technology*, 51(8):1275–1289, —2009—.

[RGSR10] L. Ruiz-Garcia, G. Steinberger, and M. Rothmund. A model and prototype implementation for tracking and tracing agricultural batch products along the food chain. *Food Control*, 21(2):112–121, —2010—.

[RJB+08] Jolita Ralyté, Manfred A. Jeusfeld, Per Backlund, Harald Kühn, and Nicolas Arni-Bloch. A knowledge-based approach to manage information systems interoperability. *Information Systems*, 33(7-8):754–784, —2008—.

[RLM05] Hajo A. Reijers and Selma Liman Mansar. Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. *Omega*, 33(4):283–306, —2005—.

[RLvdA03] Hajo A. Reijers, Selma Limam, and Wil M. P. van der Aalst. Product-based workflow design. *J. Manage. Inf. Syst.*, 20(1):229–262, —2003—. 1289808.

[SB07] Bruno Salgues and Kalyan Bollampally. Usage of modern ict in the product/component distribution system, 20-22 June —2007—.

[Sch98] August-Wilhelm Scheer. *Business process engineering : reference models for industrial enterprises*. Springer, Berlin ; New York, study edition, —1998—.

[Sch08] Richard Schonberger. *Best practices in lean six sigma process improvement : a deeper look*. John Wiley & Sons, Hoboken, N.J., —2008—.

[SD96] Monique Snoeck and Guido Dedene. Generalization/specialization and role in object oriented conceptual modeling. *Data & Knowledge Engineering*, 19(2):171–195, —1996—.

[SD98] Monique Snoeck and Guido Dedene. Existence dependency: The key to semantic integrity between structural and behavioral aspects of object types. *IEEE Transactions on Software Engineering*, 24(4):233–251, —1998—. 0098-5589.

[SH07] Pnina Soffer and Irit Hadar. Applying ontology-based rules to conceptual modeling: a reflection on modeling decision making. *European Journal of Information Systems*, 16(5):599–611, —2007—.

[Sha97] Graeme Shanks. Conceptual data modelling: an empirical study of expert and novice data modellers. *Australian Journal of Information Systems*, 4(2):63–73, —1997—.

[SIG97] Len Silverston, William H. Inmon, and Kent Graziano. *The data model resource book : a library of logical data models and data warehouse designs*. Wiley, New York, —1997—.

[Sim96] Herbert A. Simon. *The sciences of the artificial*. MIT Press, Cambridge, Mass., 3rd edition, —1996—.

[Sim00] Graeme C. Simsion. *Data Modeling Essentials: A Comprehensive Guide to Data Analysis, Design and Innovation*. Coriolis Group, 2nd edition, —2000—.

[SL02] Kitae Shin and Choon Seong Leem. A reference system for internet based inter-enterprise electronic commerce. *Journal of Systems and Software*, 60(3):195–209, —2002—.

[SLG+04] Monique Snoeck, Wilfried Lemahieu, Frank Goethals, Guido Dedene, and Jacques Vandenbulcke. Events as atomic contracts for component integration. *Data & Knowledge Engineering*, 51(1):81–107, —2004—.

[SLM06] Gregory A. Silver, Lee W. Lacy, and John A. Miller. Ontology based representations of simulation models following the process interaction world view, —2006—. 1218323 1168-1176.

[Sno99] Monique Snoeck. *Object-oriented enterprise modelling with MERODE*. Leuven University Press, Leuven, —1999—.

[SP00] Monique Snoeck and Geert Poels. Improving the reuse possibilities of the behavioral aspects of object-oriented domain models. In *19th International Conference on Conceptual Modeling — ER 2000*, volume 1920 of *Lecture Notes in Computer Science*, pages 423–439. Springer, —2000—.

[SS06] Vijayan Sugumaran and Veda C. Storey. The role of domain ontologies in database design: An ontology management and conceptual modeling environment. *ACM Transactions on Database Systems*, 31:1064–1094, —2006—.

[SSR09] D. Stefanovic, N. Stefanovic, and B. Radenkovic. Supply network modelling and simulation methodology. *Simulation Modelling Practice and Theory*, 17(4):743–766, —2009—.

[STR95] Veda C. Storey, Cheryl Bagley Thompson, and Sudha Ram. Understanding database design expertise. *Data & Knowledge Engineering*, 16(2):97–124, —1995—.

[STW03] Graeme Shanks, Elizabeth Tansley, and Ron Weber. Using ontology to validate conceptual models. *Communications of the ACM*, 46(10):85–89, —2003—.

[SÜC+10] I. Stanimirova, B. Üstün, T. Cajka, K. Riddelova, J. Hajslova, L. M. C. Buydens, and B. Walczak. Tracing the geographical origin of honeys based on volatile compounds profiles assessment using pattern recognition techniques. *Food Chemistry*, 118(1):171–176, —2010—.

[SXZ08] Q. Z. Sheng, Li Xue, and S. Zeadally. Enabling next-generation rfid applications: Solutions and challenges. *Computer*, 41(9):21–28, —2008—.

[TK74] Amos Tversky and Daniel Kahneman. Judgment under uncertainty - heuristics and biases. *Science*, 185(4157):1124–1131, —1974—.

[TRK90] Cheryl Bagley Thompson, Scheila A. Ryan, and Harriet Kitzman. Expertise: The basis for expert system development. *Advances in Nursing Science*, 13(2):1–10, —1990—.

[UKMZ98] Mike Ushold, Martin King, Stuart Moralee, and Yannis Zorgios. The enterprise ontology. *The Knowledge Engineering Review: Special Issue on Putting Ontologies to Use*, 13(1):31–89, —1998—.

[vdAvH02] Wil M. P. van der Aalst and Kees Max van Hee. *Workflow management : models, methods and systems*. Cooperative information systems. MIT Press, Cambridge, Mass., —2002—.

[Ver05] Jan Verelst. The influence of the level of abstraction on the evolvability of conceptual models of information systems. *Empirical Software Engineering*, 10(4):467–494, —2005—.

[VL04] Stephen L. Vargo and Robert F. Lusch. Evolving to a new dominant logic for marketing. *Journal of Marketing*, 68:1–17, —2004—.

[VLC08] Geert Vanderhulst, Kris Luyten, and Karin Coninx. Rewire: Creating interactive pervasive systems that cope with changing environments by rewiring. In *4th International Conference on Inteligent Environments*, pages 1–8, Seattle, WA, —2008—. IEEE.

[VTM08] Kostas Vergidis, Ashutosh Tiwari, and Basim Majeed. Business process analysis and optimization: Beyond reengineering. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(1):69–82, —2008—.

[VV98] Jacques Viaene and Wim Verbeke. Traceability as a key instrument towards supply chain and quality management in the belgian poultry meat chain. *Supply Chain Management*, 3(3):139–139, —1998—.

[W3C] W3C. Web of services. http://www.w3.org/standards/webofservices/.

[Wan96] Yair Wand. Ontology as a foundation for meta-modelling and method engineering. *Information and Software Technology*, 38(4):281–287, —1996—.

[WC98] T. P. Wilson and W. R. Clarke. Food safety and traceability in the agricultural supply chain: using the internet to deliver traceability. *Supply Chain Management*, 3(3):127–127, —1998—.

[Web86] Ron Weber. Data models research in accounting: An evaluation of wholesale distribution software. *Accounting Review*, 61(3):498, —1986—.

[Wel09] Jerry Welcome. How to manage a cradle-to-grave supply chain. *Material Handling Management*, 64(10):12–12, —2009—.

[WM04] T. W. Wang and K. E. Murphy. Semantic heterogeneity in multi-database systems: A review and a proposed meta-data structure. *Journal of Database Management*, 15(4):71–87, —2004—.

[WMPW95] Yair Wand, David E. Monarchi, Jeffrey Parsons, and Carson C. Woo. Theoretical foundations for conceptual modelling in information systems development. *Decision Support Systems*, 15(4):285–304, —1995—.

[WNP03] W. Whitlock, K. Nelson, and R Papp. *Modern database management casebook (6th ed.)*. Pearson Education, Upper Saddle River, NJ, —2003—.

[WPS09] Tobias Walter, Fernando Silva Parreiras, and Steffen Staab. Ontodsl: An ontology-based framework for domain-specific languages, —2009—. 1691363 408-422.

[Yu76] Shih Cheng Yu. *The structure of accounting theory*. University Presses of Florida, Gainesville, —1976—.

[Zac87] John A. Zachman. A framework for information systems architecture. *IBM systems journal*, 26(3):276–292, —1987—.

[ZB97] W. M. Zuberek and I. Bluemke. Hierarchies of place/transition refinements in petri nets. In *5th IEEE International Conference on Emerging Technologies and Factory Automation*, pages 355–360, Kauai, Hawaii, —1997—. IEEE.

[ZBK09] Bahman Zamani, Greg Butler, and Sahar Kayhani. Tool support for pattern selection and use. *Electronic Notes in Theoretical Computer Science*, 233:127–142, —2009—.

[ZKLH97] Hong C. Zhang, Tsai C. Kuo, Huitian Lu, and Samuel H. Huang. Environmentally conscious design and manufacturing: A state-of-the-art survey. *Journal of Manufacturing Systems*, 16(5):352–371, —1997—.

A

Appendix chapter 2

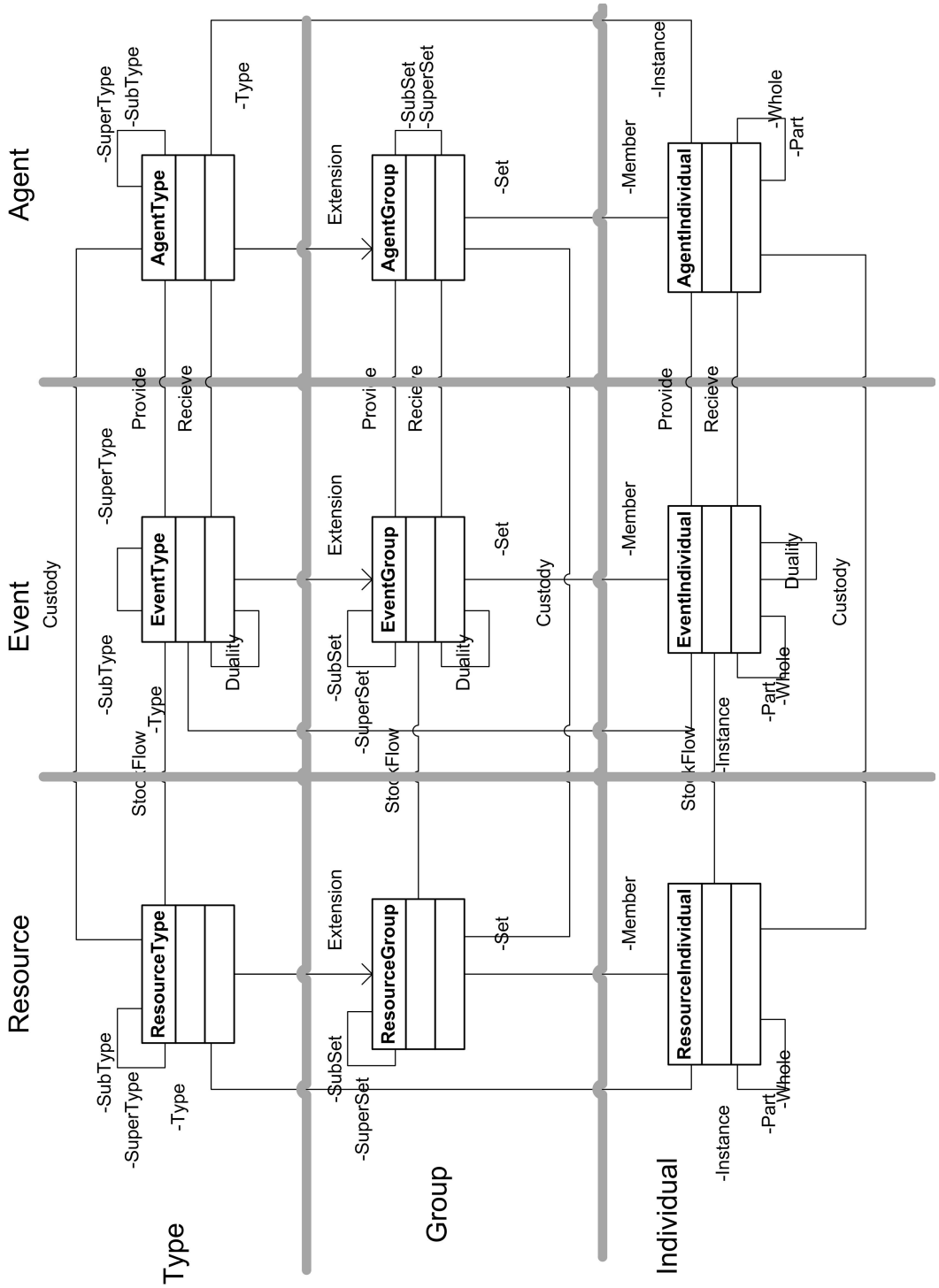

Figure A.1: Fully developed combination of the enterprise (fig. 2.2) and abstraction (fig. 2.3) pattern

B

Appendix chapter 3

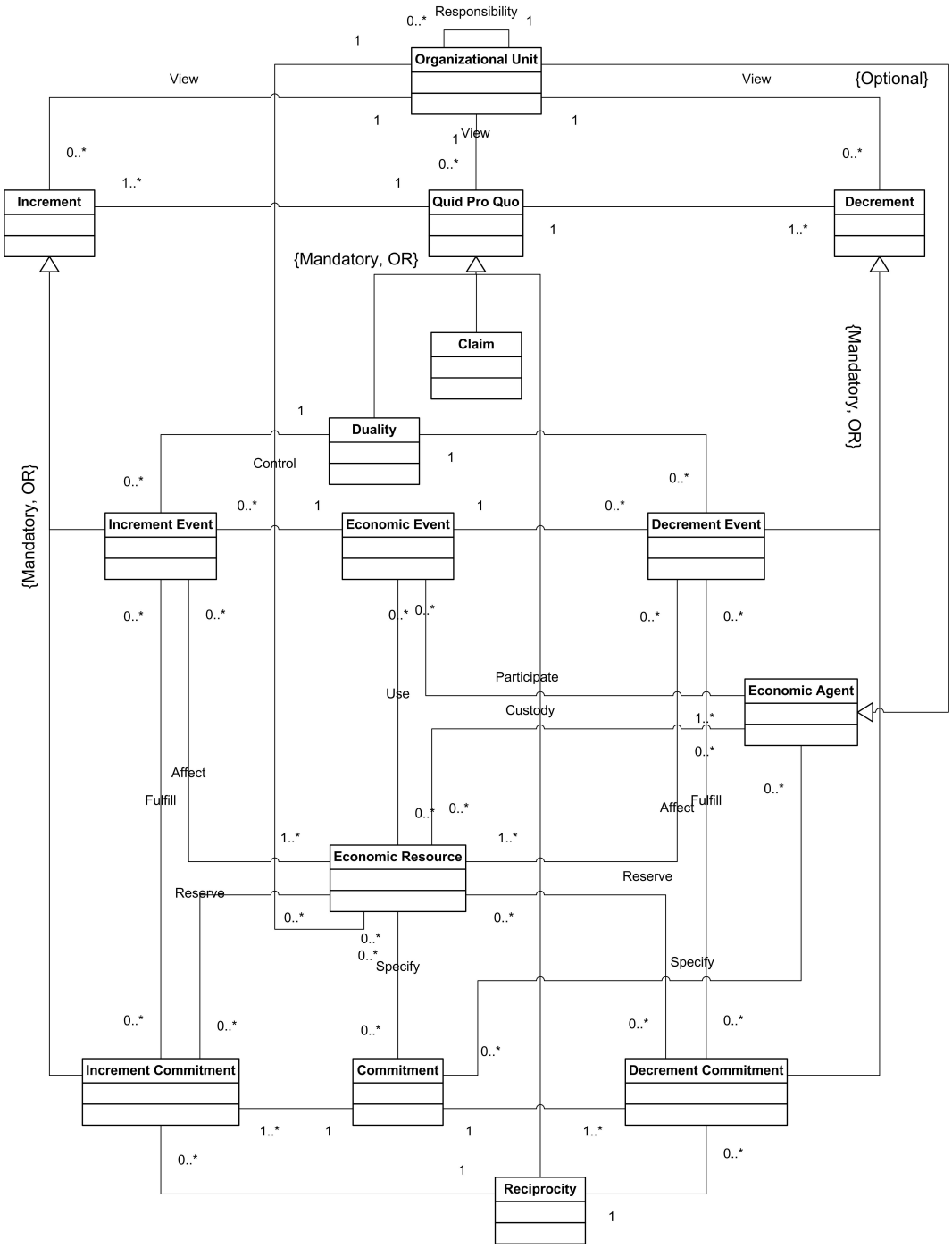

Figure B.1: Fully developed combination of the informative (fig. 3.1) and economic rationale component (fig. 3.2) of the reference information model