

**Isabelle Comyn-Wattiau  
Katsumi Tanaka  
Il-Yeol Song  
Shuichiro Yamamoto  
Motoshi Saeki (Eds.)**

**LNCS 9974**

# **Conceptual Modeling**

**35th International Conference, ER 2016  
Gifu, Japan, November 14–17, 2016  
Proceedings**

 **Springer**

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, Lancaster, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Zurich, Switzerland*

John C. Mitchell

*Stanford University, Stanford, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*TU Dortmund University, Dortmund, Germany*

Demetri Terzopoulos

*University of California, Los Angeles, CA, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Gerhard Weikum

*Max Planck Institute for Informatics, Saarbrücken, Germany*

More information about this series at <http://www.springer.com/series/7409>

Isabelle Comyn-Wattiau · Katsumi Tanaka  
Il-Yeol Song · Shuichiro Yamamoto  
Motoshi Saeki (Eds.)

# Conceptual Modeling

35th International Conference, ER 2016  
Gifu, Japan, November 14–17, 2016  
Proceedings

*Editors*

Isabelle Comyn-Wattiau  
CEDRIC-CNAM and ESSEC Business  
School  
Cergy-Pontoise  
France

Katsumi Tanaka  
Kyoto University  
Kyoto  
Japan

Il-Yeol Song  
Drexel University  
Philadelphia, PA  
USA

Shuichiro Yamamoto  
Nagoya University  
Nagoya  
Japan

Motoshi Saeki  
Tokyo Institute of Technology  
Tokyo  
Japan

ISSN 0302-9743                      ISSN 1611-3349 (electronic)  
Lecture Notes in Computer Science  
ISBN 978-3-319-46396-4              ISBN 978-3-319-46397-1 (eBook)  
DOI 10.1007/978-3-319-46397-1

Library of Congress Control Number: 2015950889

LNCS Sublibrary: SL3 – Information Systems and Applications, incl. Internet/Web, and HCI

© Springer International Publishing AG 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature  
The registered company is Springer International Publishing AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

This volume contains a collection of research papers that constitute the technical program of the 35th International Conference on Conceptual Modeling (ER 2016), held in Gifu, Japan, during November 14–17, 2016. Chen’s seminal work on the entity relationship (ER) model coincided with the emergence of conceptual modeling as a distinct field. One of the unique and valuable dimensions of the ER conference series is the way it brings researchers and practitioners from around the world to discuss ways to raise new challenges in conceptual modeling. Also known as the “Entity Relationship” or “ER” conference, this conference series has been held at an interesting variety of locations, rotating in successive years between Europe, Asia, and the Americas, attracting a diverse international community of scholars. Conceptual modeling is a process aiming at abstracting some aspects of the real world and representing them in the form of a model that can be used for understanding and communication; conceptual models typically are used in the development of computer-based information systems. The technical program for ER 2016 included papers addressing a number of current and emerging topics in conceptual modeling. In response to the call for papers, we received 113 abstracts and 89 full papers. The Program Committee provided at least three reviews for each paper, and on the basis of these reviews we selected 23 full papers (an acceptance rate of 25.84 %) and 18 short papers (a combined acceptance rate of 46.60 %). These papers can be grouped into several topical areas, including modeling and executing business processes, semantic annotations, conceptual modeling guidance, ontologies, business process management and modeling, requirements engineering, goal modeling, schema mapping, and applications of conceptual modeling. We express gratitude to all who helped make ER 2016 a success. It required the significant efforts of many people to make this conference possible. We thank the 93 Program Committee members along with the numerous external reviewers who reviewed and discussed the submitted manuscripts. These reviewers represent 26 different countries, which serves to bring a broad set of perspectives to the research arena. We especially thank the authors who took the time to carefully write up the results of their research and submit papers for consideration. The quality of these papers is a tribute to the authors and also to the reviewers who have guided any necessary improvement. Last but not least, we are greatly indebted to the three keynotes speakers: Prof. Tok Wang Ling of National University of Singapore, Prof. Oscar Pastor of Polytechnic University of Valencia, Spain, and Prof. Hideaki Takeda of National Institute of Informatics, Japan for accepting our invitation to address this conference.

July 2016

Isabelle Comyn-Wattiau  
Katsumi Tanaka  
Il-Yeol Song  
Shuichiro Yamamoto  
Motoshi Saeki

# Organizing Committee

## Honorary Chair

Kiyoshi Agusa                      Nanzan University, Japan

## Conference Co-chairs

Shuichiro Yamamoto              Nagoya University, Japan  
Motoshi Saeki                      Tokyo Institute of Technology, Japan

## Program Committee Co-chairs

Isabelle Comyn-wattiau            CEDRIC-CNAM and ESSEC Business School, France  
Katsumi Tanaka                    Kyoto University, Japan  
Il-Yeol Song                        Drexel University, USA

## Workshop Co-chairs

Juan Trujillo                        University of Alicante, Spain  
Sebastian Link                      University of Auckland, New Zealand

## Tutorial Co-chairs

Atsushi Ohnishi                    Ritsumeikan University, Japan  
Panos Vassiliadis                University of Ioannina, Greece

## Panel Co-chairs

Sudha Ram                         University of Arizona, USA  
Esteban Zimányi                 Université Libre de Bruxelles, Belgium

## Tool Demonstration and Poster Co-chairs

Aditya Ghose                      University of Wollongong, Australia  
Takashi Kobayashi               Tokyo Institute of Technology, Japan

## PhD Symposium Co-chairs

Tsuneo Ajisaka                    Wakayama University, Japan  
Carson Woo                        The University of British Columbia, Canada

## **Symposium on Conceptual Modeling Education (SCME 2016) Co-chairs**

Karen Davis University of Cincinnati, USA  
Xavier Franch Universitat Politècnica de Catalunya, Spain

### **Treasurer**

Takako Nakatani The Open University of Japan, Japan

### **Local Organizing Co-chairs**

Shuji Morisaki Nagoya University, Japan  
Atsushi Yoshida Nanzan University, Japan

### **Liasons to IPSJ**

Isamu Hasegawa Square Enix, Japan

### **Publicity Chair and Web Master**

Shinpei Hayashi Tokyo Institute of Technology, Japan

### **Student Volunteer Co-chairs**

Noritoshi Atsumi Kyoto University, Japan  
Hiroaki Kuwabara Nanzan University, Japan

### **Liaison to Steering Committee**

Sudha Ram University of Arizona, USA

### **Advisor**

Mikio Aoyama Nanzan University, Japan

### **Program Committee**

Jacky Akoka CNAM and TEM, France  
Yuan An Drexel University, USA  
Joao Araujo Universidade Nova de Lisboa, Portugal  
Zhifeng Bao University of Tasmania, Australia  
Ladjel Bellatreche ENSMA, France  
Sandro Bimonte IRSTEA, France  
Mokrane Bouzeghoub UVSQ/CNRS, France  
Shawn Bowers Gonzaga University, USA

Stephane Bressan	National University of Singapore, Singapore
Stefano Ceri	Politecnico di Milano, Italy
Roger Chiang	AIS, USA
Dickson K.W. Chiu	The University of Hong Kong, SAR China
Byron Choi	Hong Kong Baptist University, SAR China
Alfredo Cuzzocrea	University of Trieste, Italy
Fabiano Dalpiaz	Utrecht University, The Netherlands
Karen Davis	University of Cincinnati, USA
Valeria De Antonellis	University of Brescia, Italy
Sergio De Cesare	Brunel University, UK
José Palazzo M. de Oliveira	Federal University of Rio Grande do Sul, Brazil
Lois Delcambre	Portland State University, USA
Gill Dobbie	University of Auckland, New Zealand
Johann Eder	Alpen Adria Universität Klagenfurt, Austria
Xavier Franch	Universitat Politècnica de Catalunya, Spain
Avigdor Gal	Technion, Israel
Sepideh Ghanavati	Luxembourg Institute of Science and Technology, Luxembourg
Aditya Ghose	University of Wollongong, Australia
Paolo Giorgini	University of Trento, Italy
Georg Grossmann	University of South Australia, Australia
Giancarlo Guizzardi	Federal University of Espirito Santo, Brazil
Renata Guizzardi	Universidade Federal do Espirito Santo, Brazil
Arantza Illarramendi	Basque Country University, Spain
Matthias Jarke	RWTH Aachen University, Germany
Manfred Jeusfeld	University of Skövde, Sweden
Ivan Jureta	University of Namur, Belgium
Dimitris Karagiannis	University of Vienna, Austria
Kamalakar Karlapalem	CDE, IIIT Hyderabad, India
David Kensch	RWTH Aachen University, Germany
Dik Lee	Hong Kong University of Science and Technology, SAR China
Mong Li Lee	National University of Singapore, Singapore
Julio Cesar Leite	PUC-Rio, Brazil
Guoliang Li	Tsinghua University, China
Stephen Liddle	Brigham Young University, USA
Tok Wang Ling	National University of Singapore, Singapore
Sebastian Link	The University of Auckland, New Zealand
Pericles Loucopoulos	The University of Manchester, UK
Hui Ma	Victoria University of Wellington, New Zealand
Wolfgang Maass	Saarland University, Germany
Heinrich C. Mayr	Alpen-Adria-Universität Klagenfurt, Austria
Haralambos Mouratidis	University of Brighton, UK
John Mylopoulos	University of Toronto, Canada
Wilfred Ng	Hong Kong University of Science and Technology, SAR China

Antoni Olivé	Universitat Politècnica de Catalunya, Spain
Andreas L. Opdahl	University of Bergen, Norway
Jinsoo Park	Seoul National University, Korea
Jeffrey Parsons	Memorial University of Newfoundland, Canada
Chris Partridge	Brunel University, BORO Solutions, UK
Oscar Pastor Lopez	Universitat Politècnica de Valencia, Spain
Barbara Pernici	Politecnico di Milano, Italy
Geert Poels	Ghent University, Belgium
Henderik Proper	Public Research Centre Henri Tudor, Luxembourg
Christoph Quix	Fraunhofer FIT, Germany
Jolita Ralyté	University of Geneva, Switzerland
Sudha Ram	University of Arizona, USA
Iris Reinhartz-Berger	University of Haifa, Israel
Stefano Rizzi	University of Bologna, Italy
Colette Rolland	Université Paris 1 Panthéon-Sorbonne, France
Antonio Ruiz-Cortés	University of Seville, Spain
Bernhard Rumpe	RWTH Aachen University, Germany
Keun Ho Ryu	Chungbuk National University, Korea
Sourav S. Bhowmick	Nanyang Technological University, Singapore
Mehrdad Sabetzadeh	University of Luxembourg, Luxembourg
Camille Salines	Université Paris 1 Panthéon-Sorbonne, France
Peretz Shoval	Ben-Gurion University, Israel
Pnina Soffer	University of Haifa, Israel
Veda Storey	Georgia State University, USA
Arnon Sturm	Ben-Gurion University, Israel
David Taniar	Monash University, Australia
Ernest Teniente	Universitat Politècnica de Catalunya, Spain
James Terwilliger	Microsoft Corporation, USA
Bernhard Thalheim	Christian Albrechts University Kiel, Germany
Panos Vassiliadis	University of Ioannina, Greece
Gerd Wagner	Brandenburg University of Technology at Cottbus, Germany
Barbara Weber	University of Innsbruck, Austria
Roel Wieringa	University of Twente, The Netherlands
Carson Woo	University of British Columbia, Canada
Huayu Wu	Institute for Infocomm Research, Singapore
Hwan-Seung Yong	Ewha Womans University, Korea
Eric Yu	University of Toronto, Canada

## **Additional Reviewers**

Fatma Başak Aydemir	Amador Durán Toro	Diana Marosin
Kamel Barkaoui	Christophe Feltus	Michele Melchiori
Nabila Berkani	Jesús García Galán	Yassine Ouhammou
Vincent Bertram	Mohamad Gharib	Elda Paja
Devis Bianchini	Oliver Kautz	Mattia Salnitri
Dominik Bork	Lu Li	Arik Senderovich
Andrea Burattin	Mingzhao Li	Nikolaos Tantouris
Brice Chardin	Achim Lindt	Marc van Zee
Adela Del Río Ortega	Garm Lucassen	Peng Wang

## **Organized by**

- Special Interest Group on Software Engineering, Information Processing Society of Japan
- The ER Institute (ER Steering Committee)

## **In Cooperation with**

- The Database Society of Japan
- Special Interest Group on Database Systems, Information Processing Society of Japan
- Information and Systems Society and Technical Committee on Data Engineering, The Institute of Electronics, Information and Communication Engineers
- IEEE Computer Society Japan Chapter
- ACM SIGMOD Japan
- Software Engineers Association

**Industrial Sponsors**



# Contents

## Keynotes

Improving the Correctness of Some Database Research Using ORA-Semantics . . . . .	3
<i>Tok Wang Ling, Zhong Zeng, Mong Li Lee, and Thuy Ngoc Le</i>	
Conceptual Modeling of Life: Beyond the Homo Sapiens . . . . .	18
<i>Oscar Pastor</i>	

## Analytics and Conceptual Modeling

A Conceptual Modeling Framework for Business Analytics . . . . .	35
<i>Soroosh Nalchigar, Eric Yu, and Rajgopal Ramani</i>	
NOSQL Design for Analytical Workloads: Variability Matters . . . . .	50
<i>Victor Herrero, Alberto Abelló, and Oscar Romero</i>	
Translating Bayesian Networks into Entity Relationship Models . . . . .	65
<i>Frank Rosner and Alexander Hinneburg</i>	
Key Performance Indicator Elicitation and Selection Through Conceptual Modelling. . . . .	73
<i>Alejandro Maté, Juan Trujillo, and John Mylopoulos</i>	

## Conceptual Modeling and Ontologies

Insights on the Use and Application of Ontology and Conceptual Modeling Languages in Ontology-Driven Conceptual Modeling . . . . .	83
<i>Michael Verdonck and Frederik Gailly</i>	
An Ontological Approach for Identifying Software Variants: Specialization and Template Instantiation . . . . .	98
<i>Iris Reinhartz-Berger, Anna Zamansky, and Yair Wand</i>	
The Role of Ontology Design Patterns in Linked Data Projects. . . . .	113
<i>Valentina Presutti, Giorgia Lodi, Andrea Nuzzolese, Aldo Gangemi, Silvio Peroni, and Luigi Asprino</i>	
Bridging the IT and OT Worlds Using an Extensible Modeling Language . . .	122
<i>Paola Lara, Mario Sánchez, and Jorge Villalobos</i>	

**Requirements Engineering**

Possibilistic Cardinality Constraints and Functional Dependencies . . . . .	133
<i>Tania K. Roblot and Sebastian Link</i>	
Exploring Views for Goal-Oriented Requirements Comprehension . . . . .	149
<i>Lyrene Silva, Ana Moreira, João Araújo, Catarina Gralha, Miguel Goulão, and Vasco Amaral</i>	
Keys with Probabilistic Intervals . . . . .	164
<i>Pieta Brown, Jeeva Ganesan, Henning Köhler, and Sebastian Link</i>	

**Advanced Conceptual Modeling**

On Referring Expressions in Information Systems Derived from Conceptual Modelling . . . . .	183
<i>Alexander Borgida, David Toman, and Grant Weddell</i>	
DeepTelos: Multi-level Modeling with Most General Instances . . . . .	198
<i>Manfred A. Jeusfeld and Bernd Neumayr</i>	
Pragmatic Quality Assessment for Automatically Extracted Data . . . . .	212
<i>Scott N. Woodfield, Deryle W. Lonsdale, Stephen W. Liddle, Tae Woo Kim, David W. Embley, and Christopher Almqvist</i>	
UnifiedOCL: Achieving System-Wide Constraint Representations . . . . .	221
<i>David Weber, Jakub Szymanek, and Moira C. Norrie</i>	

**Semantic Annotations**

Building Large Models of Law with NómoST . . . . .	233
<i>N. Zeni, E.A. Seid, P. Engiel, S. Ingolfo, and J. Mylopoulos</i>	
An Efficient and Simple Graph Model for Scientific Article Cold Start Recommendation . . . . .	248
<i>Tengyuan Cai, Hongrong Cheng, Jiaqing Luo, and Shijie Zhou</i>	
Keyword Queries over the Deep Web . . . . .	260
<i>Andrea Cali, Davide Martinenghi, and Riccardo Torlone</i>	
Sensor Observation Service Semantic Mediation: Generic Wrappers for In-Situ and Remote Devices . . . . .	269
<i>Manuel A. Regueiro, José R.R. Viqueira, Christoph Stasch, and José A. Taboada</i>	

**Modeling and Executing Business Processes**

Probabilistic Evaluation of Process Model Matching Techniques . . . . . 279  
*Elena Kuss, Henrik Leopold, Han van der Aa, Heiner Stuckenschmidt,  
and Hajo A. Reijers*

Context-Aware Workflow Execution Engine for E-Contract Enactment . . . . . 293  
*Himanshu Jain, P. Radha Krishna, and Kamalakar Karlapalem*

Annotating and Mining for Effects of Processes . . . . . 302  
*Suman Roy, Metta Santiputri, and Aditya Ghose*

**Business Process Management and Modeling**

Automated Discovery of Structured Process Models: Discover Structured  
vs. Discover and Structure . . . . . 313  
*Adriano Augusto, Raffaele Conforti, Marlon Dumas, Marcello La Rosa,  
and Giorgio Bruno*

Detecting Drift from Event Streams of Unpredictable Business Processes. . . . . 330  
*Alireza Ostovar, Abderrahmane Maaradji, Marcello La Rosa,  
Arthur H.M. ter Hofstede, and Boudewijn F.V. van Dongen*

Modeling Structured and Unstructured Processes: An Empirical Evaluation. . . . . 347  
*Evellin Cardoso, Katsiaryna Labunets, Fabiano Dalpiaz,  
John Mylopoulos, and Paolo Giorgini*

**Applications and Experiments of Conceptual Modeling**

MetaScience: An Holistic Approach for Research Modeling . . . . . 365  
*Valerio Cosentino, Javier Luis Cánovas Izquierdo, and Jordi Cabot*

Comparison and Synergy Between Fact-Oriented and Relation  
Extraction for Domain Model Generation in Regulatory Compliance . . . . . 381  
*Sagar Sunkle, Deepali Kholkar, and Vinay Kulkarni*

Development of a Modeling Language for Capability Driven Development:  
Experiences from Meta-modeling . . . . . 396  
*Janis Stirna and Jelena Zdravkovic*

Applying Conceptual Modeling to Better Understand the Human Genome . . . . . 404  
*José F. Reyes Román, Óscar Pastor, Juan Carlos Casamayor,  
and Francisco Valverde*

**Schema Mapping**

Data Analytics: From Conceptual Modelling to Logical Representation . . . . . 415  
*Qing Wang and Minjian Liu*

UMLtoGraphDB: Mapping Conceptual Schemas to Graph Databases . . . . . 430  
*Gwendal Daniel, Gerson Sunyé, and Jordi Cabot*

Facilitating Data-Metadata Transformation by Domain Specialists  
in a Web-Based Information System Using Simple Correspondences . . . . . 445  
*Scott Britell, Lois M.L. Delcambre, and Paolo Atzeni*

**Conceptual Modeling Guidance**

Visualizing User Story Requirements at Multiple Granularity Levels  
via Semantic Relatedness . . . . . 463  
*Garm Lucassen, Fabiano Dalpiaz, Jan Martijn E.M. van der Werf,  
and Sjaak Brinkkemper*

User Progress Modelling in Counselling Systems: An Application  
to an Adaptive Virtual Coach . . . . . 479  
*Nuria Medina-Medina, Zoraida Callejas, Kawtar Benghazi,  
and Manuel Noguera*

Stepwise Refinement of Software Development Problem Analysis. . . . . 488  
*Tsutomu Kobayashi, Fuyuki Ishikawa, and Shinichi Honiden*

Tailoring User Interfaces to Include Gesture-Based Interaction with gestUI. . . . 496  
*Otto Parra, Sergio España, and Oscar Pastor*

Unlocking Visual Understanding: Towards Effective Keys for Diagrams . . . . . 505  
*Nicolas Genon, Gilles Perrouin, Xavier Le Pallec, and Patrick Heymans*

**Goal Modeling**

MEMO GoalML: A Context-Enriched Modeling Language to Support  
Reflective Organizational Goal Planning and Decision Processes. . . . . 515  
*Alexander Bock and Ulrich Frank*

Can Goal Reasoning Techniques Be Used for Strategic Decision-Making? . . . . 530  
*Elda Paja, Alejandro Maté, Carson Woo, and John Mylopoulos*

Requirements Evolution and Evolution Requirements  
with Constrained Goal Models . . . . . 544  
*Chi Mai Nguyen, Roberto Sebastiani, Paolo Giorgini,  
and John Mylopoulos*

RationalGRL: A Framework for Rationalizing Goal Models  
Using Argument Diagrams . . . . . 553  
*Marc van Zee, Diana Marosin, Floris Bex, and Sepideh Ghanavati*

**Author Index** . . . . . 561

# Annotating and Mining for Effects of Processes

Suman Roy<sup>1</sup>(✉), Metta Santiputri<sup>2</sup>, and Aditya Ghose<sup>2</sup>

<sup>1</sup> Infosys Ltd., # 44 Electronics City, Hosur Road, Bangalore 560 100, India

Sunam.Roy@infosys.com

<sup>2</sup> School of Computer Science and Software Engineering, University of Wollongong,  
Wollongong, NSW 2500, Australia

ms804@uowmail.edu.au, aditya@uow.edu.au

**Abstract.** We provide a novel explicit annotation of a process model by way of accumulating effects of individual tasks specified by analysts using belief bases and computing the accumulated effect up to the point of execution of the process model in an automated manner. This technique permits the analyst to specify immediate effect annotations in a practitioner-accessible simple propositional logic formulas and generates a sequence of tasks along with cumulative effects, called effect logs. Further we propose and solve an effect mining problem, that is, given an effect log discover the process model with effect annotations of individual tasks which is close to the original annotated process model.

**Keywords:** Business process modeling · Semantic annotation · Effects · Belief bases · Annotated processes · Effect logs · Effect mining

## 1 Introduction

In this work we describe a technique for an explicit semantic annotation of process models. We require practitioners to provide a description of the immediate effects of each task. An effect of a task becomes true when the latter gets executed. These effects are propagated across different nodes (tasks and gateways) in the process model and then accumulated in a context-sensitive manner automatically, such that the cumulative effect annotation associated with any task in a BPMN process model<sup>1</sup> would describe the effects achieved by the process as if the process were executed up to that point [3]. The cumulative effect description of a task might be non-deterministic, this non-determinism may be caused by the execution of parallel gateways resulting in interleaving of paths in the process model, and belief update leading to multiple alternative means of resolving inconsistencies generated by the “undoing” of effects. Our effect annotation borrows ideas from Semantic Business Process Validation (SBPV) which

---

S. Roy—This work was done when the author visited University of Wollongong during July–Dec’14 to work on Infosys-CRC funded project on data-driven process discovery.

<sup>1</sup> Process models captured using industry standard notation BPMN.

combines concepts from the workflow [6, 8, 13] and AI action and changes [4, 15]. While we adopt the token semantics from workflow literature [14, 15] for determining execution traces of processes we use belief operator from AI [5, 7] for updating the effect and their accumulation on enabling of nodes. By this way we are able to compute an effect trace of the process which is a sequence of pairs consisting of task and the cumulative effect at this node.

Next we consider an inverse problem: given an effect log, *i.e.*, a set of effect traces, how one can determine the original effects at the tasks? We call this effect mining problem, *aka* process mining problem [11, 12]. Although process designers take utmost care in designing process models there is no guarantee that they indeed reflect the correct models under consideration. The goal of process discovery is to derive some sort of model that describes the process as accurately as possible. Similarly, the goal of effect discovery is to generate a semantically annotated process which is close to the original semantic description of the process model. As one solution approach for this effect mining problem we propose a modification of process discovery algorithm, *viz.*,  $\alpha$ -mining problem to find out a process model along with the effects associated with each task.

**Related Work.** There is a rich body of work on semantic annotation for web services. In one of them Weber *et al.* proposed an approach, Semantic Business Process Validation (SBPV) [15], in which axiomatic task descriptions are annotated and propagated across process models. The SBPV approach requires the user to completely specify pre-conditions and post-conditions that are context sensitive. In another approach Hinge *et al.* [3] proposed a technique for obtaining semantic effect descriptions of BPMN process models, without requiring the analyst to express excessive formal specification. Using an approach similar to SBPV in [4] Hoffman *et al.* proposed a framework of annotating processes for capturing the semantics of task execution in which compliance is checked against a set of constraints imposed on process states. Motivated by SBPV [4, 15] and the annotation method suggested in [3] we lift ideas from AI and use belief bases to annotate the tasks with effects, propagate these effects across the nodes automatically in a context sensitive manner, and compute the accumulated effects of each task up to the execution point.

There is not much work on effect mining we could find. In [10] Santiputri *et al.* present a data-driven approach for mining semantically annotated business processes. The authors assume event logs in execution histories of business processes describing both task execution events (found in process logs) and state update events (recorded in effect logs) at disposal and mine for immediate effect annotations for each task in the process model to be edited and refined by analysts. In our work we only consider effect logs which is a collection of pairs consisting of tasks and accumulated effects, and discover the process model with effect annotations for each task. While in the previous work the authors use a variant of sequence mining algorithms for effect mining we use a modified process mining algorithm for effect discovery.

The paper is organized as follows. We introduce a semantic annotation of process models in Sect. 2 along with an example of effect annotation. The problem of effect mining is defined in Sect. 3. Finally we conclude in Sect. 4.

## 2 A Semantic Annotation of Processes

We introduce an annotation framework for business processes where tasks are annotated with effects. An effect of a task is some fact which materializes when the task is executed. It is captured by analysts providing a description of the immediate effects of each process task, i.e., a context independent specification of the functionality of each task. For the sake of easier readability we first introduce a simple process graph without annotation. We use a formalism of a business process which bears close resemblance with those described in [1, 4, 15] and that of work-flows [8].

A *BPM process* is a graph (also called a process model graph)  $\mathbf{P} = (\mathcal{N}, \mathcal{F})$  where  $\mathcal{N}$  is a finite set of nodes which is partitioned into the set of tasks  $\mathcal{T}$ , the set of gateways  $\mathcal{G}$ , and the set of events  $\mathcal{E}$ , i.e.,  $\mathcal{N} \hat{=} \mathcal{T} \uplus \mathcal{G} \uplus \mathcal{E}$ ;  $\mathcal{G}$  can be further partitioned into disjoint sets of decision merges,  $\mathcal{G}_M$  ( $\mathcal{G}_M^{and}$  (synchronizer) and  $\mathcal{G}_M^{xor}$  (merge)) and decision splits,  $\mathcal{G}_S$  ( $\mathcal{G}_S^{and}$  (fork) and  $\mathcal{G}_S^{xor}$  (choice)); a set  $\mathcal{E}$  of events which is a disjoint union of two sets of events  $\mathcal{E}_s$  and  $\mathcal{E}_f$ , where  $\mathcal{E}_s$  is the set of start events with no incoming edges,  $\mathcal{E}_f$  is the set of end events with no outgoing edges; and  $\mathcal{F} \subseteq (\mathcal{N} \setminus \mathcal{E} \times \mathcal{N} \setminus \mathcal{E}) \cup (\mathcal{E}_s \times \mathcal{N} \setminus \mathcal{E}) \cup (\mathcal{N} \setminus \mathcal{E} \times \mathcal{E}_f)$  corresponds to sequence flows connecting tasks with tasks, tasks with gateways, gateways with tasks, start nodes with tasks and tasks with end nodes.

Let  $in(n)$  ( $out(n)$ ) be the set of incoming (outgoing) edges to (out of) node  $n \in \mathcal{N}$ . We impose the following conditions:  $\forall n_s \in \mathcal{E}_s, |in(n_s)| = 0$ , and  $|out(n_s)| = 1$ ;  $\forall n_f \in \mathcal{E}_f, |in(n_f)| = 1$ , and  $|out(n_f)| = 0$ ; for every  $n \in \mathcal{T}, |in(n)| = |out(n)| = 1$ ; for every  $n \in \mathcal{G}_S, |in(n)| = 1$  and  $|out(n)| > 1$ ; for every  $n \in \mathcal{G}_M, |in(n)| > 1$  and  $|out(n)| = 1$ ; any outgoing edge  $out(n)$  from a fork node  $n \in \mathcal{G}_S^{and}$  will have a task node  $t$  appearing immediately after  $n$ :  $(n, t) \in \mathcal{F}$ , and every node is on a path from some start node to some end node. If these conditions hold then we say the business process to be *well-formed*. We shall consider only well-formed business processes henceforth.

Let us now specify the semantics of control elements of a business process. Given a process  $\mathbf{P} = (\mathcal{N}, \mathcal{F})$ , a *state* of  $\mathbf{P}$  is a marking  $\mu : \mathcal{F} \rightarrow \mathbb{N}$ , also called a token mapping. The number of tokens may change during the execution of the process, when the transitions are enabled. A state  $\mu'$  is reached from state  $\mu$  via node  $n$ , written as  $\mu \xrightarrow{n} \mu'$ , when  $n$  can be a task, AND-split, AND-join or XOR-split or XOR-join, for details see [9, 15].

The initial state is given by a marking  $\mu_0$  where  $\mu_0(e_s) = 1$ , for all  $e_s \in \mathcal{E}_s$ , and  $\mu_0(e) = 0$  for all other edges  $e$ . A node  $n$  is said to be *activated* in a state  $\mu$  if there exists state  $\mu'$  such that  $\mu \xrightarrow{n} \mu'$ . A state  $\mu'$  is reachable from a state  $\mu$ , denoted as  $\mu \xrightarrow{*} \mu'$  if there exists a (possibly finite) path,  $\rho : n_s, n_1, \dots, n_f \in (\mathcal{N})$  and a finite sequence of markings  $\mu_1, \dots, \mu_k$  such that  $\mu$  is activated in  $n_s$  and  $\mu \xrightarrow{n_s} \mu_1 \xrightarrow{n_1} \dots \xrightarrow{n_i} \mu_k$  and  $\mu' = \mu_k$ .

For a path  $\rho$  we denote the projection of  $\rho$  on set of tasks  $\mathcal{T}$  as  $\rho|_{\mathcal{T}}$ . A *complete trace* (also called *trace*) of a business process  $\mathbf{P}$  is a sequence of tasks  $\tau = t_1, \dots, t_l$ , ( $t_i \in \mathcal{T}$ ,  $1 \leq i \leq l$ ) such there is a path  $\rho = n_s, n_1, \dots, n_f \in \mathcal{N}$  in  $\mathbf{P}$  and  $\tau = \rho|_{\mathcal{T}}$ . The set of all traces of a process  $\mathbf{P}$  is denoted as  $\mathbb{T}_{\mathbf{P}}$ . For the remainder of the paper we assume our process to be sound [2, 15].

For annotating processes with effects we shall use logical propositions and assume the existence of a countable set  $\mathcal{P}$  of propositions. The set of all literals over  $\mathcal{P}$  is denoted as  $\mathcal{L}_{\mathcal{P}}$ . A belief base  $\mathcal{B}$  is a conjunction of literals in  $\mathcal{P}$  which is logically consistent. It can be written using set-theoretic notation. For example, if a belief base  $B = a \wedge \neg b \wedge c$ , where  $a, b, c \in \mathcal{P}$  then  $B = \{a, \neg b, c\}$ . A theory  $\mathbf{T}$  over  $\mathcal{P}$  can be taken to be any propositional theory. A knowledge base  $\mathcal{K}$  is a pair  $(\mathcal{P}, \mathbf{T})$ , where  $\mathbf{T}$  can be assumed to consist of rules and facts. Wlog we can assume  $\mathbf{T}$  to be the conjunction of those rules and facts.

We define an annotated process model/graph as  $\mathbb{G}_{\mathbf{P}} = (\mathcal{N}, \mathcal{F}, \mathcal{K}, \mathcal{A})^2$ , where  $\mathbf{P} = (\mathcal{N}, \mathcal{F})$  is the underlying process model as before,  $\mathcal{K} = (\mathcal{P}, \mathbf{T})$  is the underlying knowledge base annotation,  $\mathcal{A}$  is a partial function mapping  $n \in \mathcal{T}$  to  $\text{eff}(n) \subseteq \mathcal{B}(\mathcal{P})$ , and mapping  $e \in \text{out}(n)$  for  $n \in \mathcal{G}_S^{\text{xor}}$  to  $(\text{con}(e), \text{pos}(e))$  where  $\text{con}(e) \in \mathcal{L}_{\mathcal{P}}$  and  $\text{pos}(e) \in \{1, \dots, |\text{out}(n)|\}^3$ . The following technical conditions need to be imposed: there does not exist an  $e$  such that  $\mathbf{T} \wedge \text{con}(e)$  is unsatisfiable; there do not exist  $n, e$ , and  $e'$  so that  $e, e' \in \text{out}(n)$  ( $e$  and  $e'$  being distinct),  $\mathcal{A}(e)$  and  $\mathcal{A}(e')$  are defined, and  $\text{pos}(e) = \text{pos}(e')$ . Moreover, the cardinality of the set of reachable states immediately preceding a choice gateway must be bounded from below by the number of outgoing edges from the gateway.

In an execution of process we shall assume the effects of the activities of the process will be dynamically changed with the corresponding knowledge environment. Let us assume the current available information be represented by an a-priori available knowledge base  $\mathcal{K}$ ; and an accumulated effect  $B_0$  that we assume to be true (*i.e.*, to persist) until the next task has been executed. Suppose that a task  $t$  in a process is executed in an instance of a process whose effect can be captured by the belief base  $B = \text{eff}(t)$ . What would be our knowledge after  $t$  is executed? Borrowing concepts from artificial intelligence [5, 7], we revise our knowledge using belief update to capture the changing scenario (we treat belief revision and belief update as same). This can be achieved using an update operators  $\Delta$  details of which can be found in [9].

Let us now formally define semantics of an annotated process graph, a similar semantic annotation for business processes is provided in [4, 15]. A *state*  $s$  of  $\mathbb{G}$  is a pair  $\iota = (\mu, \mathcal{E}_a)$  where  $\mu$  is a token mapping as defined before, and  $\mathcal{E}_a : 2^{\mathcal{T}} \mapsto 2^{(\mathcal{P})}$  is a cumulative effect accumulation function. Assume that the current set of tasks for which the effects are accumulated till date is  $\mathcal{T}_c$  and its accumulated effect is  $\mathcal{E}_a(\mathcal{T}_c)$ . The updated set of tasks will be  $\mathcal{T}'_c$  when a new task is executed. The initial state is  $\iota_0 = (\mu_0, \mathcal{E}_{a_0})$  where  $\mathcal{T}_c = \emptyset$ . By default, we assume  $\mathcal{E}_{a_0}(\emptyset) = \emptyset$ . The effects across the tasks/gateways are going to be

<sup>2</sup> we shall drop the subscript  $\mathbf{P}$  when it is clear from the context.

<sup>3</sup> For the sake of rigor  $\mathcal{P}$  can be partitioned into two sets  $\mathcal{P} = \mathcal{P}_t \uplus \mathcal{P}_x$ , where tasks are annotated with symbols from  $\mathcal{P}_t$ , and conditions on choices come from  $\mathcal{P}_x$ .

accumulated in a recursive manner; the accumulated effect will be denoted as  $\mathcal{E}'_a(\mathcal{T}'_c)$ . Let  $\iota$  and  $\iota'$  be two states. A state  $\iota'$  is reached from state  $\iota$  via node  $n$ , written as  $\iota \xrightarrow{n} \iota'$ , if and only if  $n \in (T \cup g)$  and  $(n_i, n) \in \mathcal{F}$ . We consider different cases of  $n$  being an task, a split gateway, a join gateway etc. in [9]. As before for a given a business process  $\mathbf{P}$  we consider a trace  $\mathbf{P}$  to be a sequence of tasks  $\tau = t_1, \dots, t_l$ , where  $t_i \in \mathcal{T}$ ,  $1 \leq i \leq l$ . Similarly an *effect trace* is defined to be a sequence  $\varepsilon : (t_1, \mathcal{E}_{a_1}), (t_2, \mathcal{E}_{a_2}), \dots, (t_m, \mathcal{E}_{a_m})$ , where  $\tau = t_1, \dots, t_m$  is called the underlying *trace* of  $\varepsilon$ . Given an annotated process model we denote  $\mathcal{E}_{\mathbb{G}}$  to be the set of its effect traces.

Figure 1 depicts an example of a business process for illustrating effect annotation. This process diagram is drawn using BPM notation. We use this process graph as the running exam-

**Table 1.** Effect annotations for process model in Fig. 1

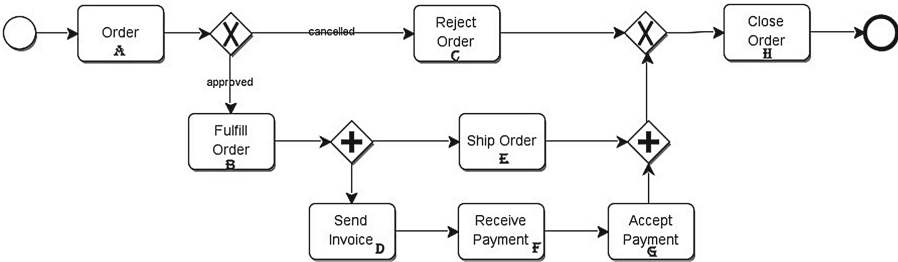
Task	Effects
Order (A)	<i>ordered</i> $\wedge$ <i>received</i>
Reject order (C)	<i>rejected</i>
Fulfill order (B)	<i>fulfilled</i>
Send invoice (D)	<i>invoiceSent</i> $\wedge$ <i>paymentExpected</i>
Ship order (E)	<i>shipped</i>
Receive payment (F)	<i>paymentReceived</i>
Accept payment (G)	<i>paymentAccepted</i> $\wedge$ $\neg$ <i>paymentExpected</i> $\wedge$ <i>paid</i>
Close order	<i>closed</i>

ple throughout this paper. This process contains a start and an end node, and various tasks, such as “order”, “reject order”, “fulfill order” etc. It also has a number of routing constructs such as an XOR-split after the task order and an AND-split after fulfill order. Only one of the branches after the XOR-split is executed depending on the condition (approved or fulfilled) which is true.

In Fig. 1 the semantic annotation of each task is given in Table 1. The knowledge base is given by  $\mathcal{K} = \{\mathcal{P}, \mathbf{T}\}$ , where

$$\begin{aligned} \mathbf{P} &= \{ordered, received, rejected, fulfilled, invoiceSent, paymentExpected, shipped, \\ &\quad paymentReceived, paymentAccepted, closed, approved, cancelled; \text{ and} \\ \mathbf{T} &= \{closed \rightarrow \neg(ordered \wedge received \wedge fulfilled); \\ &\quad (cancelled \wedge rejected) \rightarrow \neg(ordered \wedge received)\}. \end{aligned}$$

Assuming an initial state to be  $\iota_0 = (\mu_0, \mathcal{E}_{a_0})$  where  $\mathcal{T}_c = \emptyset$  and  $\mathcal{E}_{a_0}(\emptyset) = \emptyset$  we can compute the cumulative effect on the execution of each task/gateway. The



**Fig. 1.** An example of a process using BPM notation

**Table 2.** Accumulated effect annotations for process model in Fig. 1

Task	Accumulated effects
Order	{ <i>ordered, received</i> }
Reject order	{ <i>ordered, received, cancelled, rejected</i> }
Fulfill order	{ <i>ordered, received, approved, fulfilled</i> }
Send invoice	{ <i>ordered, received, approved, fulfilled, shipped, invoiceSent, paymentExpected</i> }
	{ <i>ordered, received, approved, fulfilled, invoiceSent, paymentExpected</i> }
Ship order	{ <i>ordered, received, approved, fulfilled, shipped</i> }
	{ <i>ordered, received, approved, fulfilled, invoiceSent, paymentReceived, paymentAccepted, paid, shipped</i> }
Receive payment	{ <i>ordered, received, approved, fulfilled, shipped, invoiceSent, paymentExpected, paymentReceived</i> }
	{ <i>ordered, received, approved, fulfilled, invoiceSent, paymentExpected, paymentReceived</i> }
Accept payment	{ <i>ordered, received, approved, fulfilled, shipped, invoiceSent, paymentReceived, paymentAccepted, paid</i> }
	{ <i>ordered, received, approved, fulfilled, invoiceSent, paymentReceived, paymentAccepted, paid</i> }
Close order	{ <i>cancelled, rejected, closed, ¬fulfilled</i> }
	{ <i>ordered, received, approved, fulfilled, invoiceSent, paymentReceived, paymentAccepted, paid, shipped, closed</i> }

effect accumulation corresponding to the execution of each node is given in the Table 2. The accumulated effect is computed by finding the maximal consistent set computed out of the union of individual effect of the node and the current accumulated effect [9].

### 3 Effect Log Mining

The log entries that we consider contain event of one type (*viz.*, completion of the event), hence we drop event from the log entries in subsequent discussions and consider only tasks (similar convention is followed in ProM-framework). Let  $\mathcal{T}$  to be the set of tasks/activities. We denote a trace as  $\sigma \in T^*$ , where  $\sigma = t_0, t_1, \dots, t_{n-1}$ , such that  $t_i \in \mathcal{T}, 0 \leq i \leq n-1$ . A process log is a defined as a set of traces, denoted as  $W \in \mathcal{P}(T^*)$ . An effect log  $\Theta \subseteq \mathcal{P}(T \times \mathcal{B}(P))$  is defined as a set of effect traces.

The problem of process mining takes process log as input and produces process models as output. That is given a process log  $W \in \mathcal{P}(T^*)$  over a set  $\mathcal{T}$  of activities, find process model  $\mathbf{P} = (\mathcal{N}, \mathcal{F})$  such that  $\mathbb{T}_{\mathbf{P}} = W$ . In this work we propose a variant of process mining problem which we call effect log mining

(also called effect discovery). It says, given an effect log  $\Theta \subseteq \mathcal{P}(\mathcal{T} \times \mathcal{B}(P))$  find annotated process model  $\mathbb{G} = (\mathcal{N}, \mathcal{F}, \mathcal{K}, \mathcal{A})$  such that  $\mathcal{E}_{\mathbb{G}} = \Theta$ .

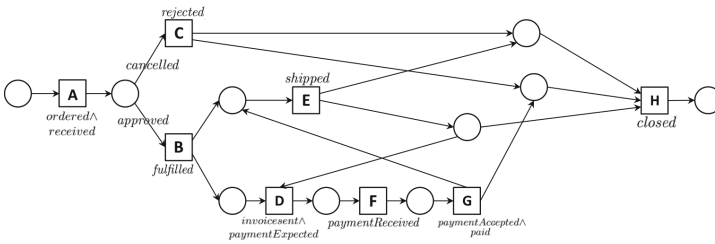
We made some simplifications of our effect mining problem. We assume our knowledge base  $\mathcal{K}$  is a pair  $(\mathcal{P}, \mathbf{T})$ , and  $\mathbf{T}$  is empty, that is, it contains no facts or no rules. Further we make the unique name assumption for tasks, *i.e.*, a task appears only once in the process model, which is just a matter of renaming. Next we propose an algorithm for effect discovery using log abstractions [12] based on log-based ordering relations. Our log-based ordering relation is defined on annotated task nodes (pairs of tasks with accumulated effects). We also define an edge relation between annotated nodes based on the above ordering relation as part of our discovery algorithm. Using this algorithm it will be possible to discover the process and find effect for each individual task in the discovered process. In this work we choose to use a modified  $\alpha$ -algorithm to discover our process model [11] (in spite of its inability to address loops, non-free-choice structures, and limited ability to address variability and closed path). For the details of the algorithm the reader is referred to [9].

We employ our process discovery technique on our example where we assume the knowledge base to be  $\mathcal{K} = \{\mathcal{P}, \emptyset\}$ , where  $\mathcal{P}$  is defined before. We consider the effect log  $\varepsilon$  of the process model in Table 3. The workflow  $\mathcal{W}_{\mathcal{P}}$  discovered thus is shown in Fig. 2. Using our technique we can only recover gross annotation of

**Table 3.** Effect traces and its underlying trace of the process model

Effect trace $\varepsilon$	The underlying trace
$((\text{start}, \emptyset), (\text{A}, \{\text{ordered}, \text{received}\}), (\text{A}, \{\text{ordered}, \text{received}, \text{cancelled}\}), (\text{C}, \{\text{ordered}, \text{received}, \text{cancelled}, \text{rejected}\}), (\text{H}, \{\text{ordered}, \text{received}, \text{cancelled}, \text{rejected}, \text{closed}\}), (\text{end}, \{\text{ordered}, \text{received}, \text{cancelled}, \text{rejected}, \text{closed}\}))$	$(\text{start}, \text{A}, \text{C}, \text{H}, \text{end})$
$((\text{start}, \emptyset), (\text{A}, \{\text{ordered}, \text{received}\}), (\text{A}, \{\text{ordered}, \text{received}, \text{approved}\}), (\text{B}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}\}), (\text{E}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}, \text{shipped}\}), (\text{D}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}, \text{shipped}, \text{invoiceSent}, \text{paymentExpected}\}), (\text{F}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}, \text{shipped}, \text{invoiceSent}, \text{paymentExpected}, \text{paymentReceived}\}), (\text{G}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}, \text{shipped}, \text{invoiceSent}, \text{paymentReceived}, \text{paymentAccepted}, \text{paid}\}), (\text{H}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}, \text{shipped}, \text{invoiceSent}, \text{paymentReceived}, \text{paymentAccepted}, \text{paid}, \text{closed}\}), (\text{end}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}, \text{shipped}, \text{invoiceSent}, \text{paymentReceived}, \text{paymentAccepted}, \text{paid}, \text{closed}\}))$	$(\text{start}, \text{A}, \text{B}, \text{E}, \text{D}, \text{F}, \text{G}, \text{H}, \text{end})$
$((\text{start}, \emptyset), (\text{A}, \{\text{ordered}, \text{received}\}), (\text{A}, \{\text{ordered}, \text{received}, \text{approved}\}), (\text{B}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}\}), (\text{D}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}, \text{invoiceSent}, \text{paymentExpected}\}), (\text{F}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}, \text{invoiceSent}, \text{paymentExpected}, \text{paymentReceived}\}), (\text{G}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}, \text{invoiceSent}, \text{paymentReceived}, \text{paymentAccepted}, \text{paid}\}), (\text{E}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}, \text{invoiceSent}, \text{paymentReceived}, \text{paymentAccepted}, \text{paid}, \text{shipped}\}), (\text{H}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}, \text{invoiceSent}, \text{paymentReceived}, \text{paymentAccepted}, \text{paid}, \text{shipped}, \text{closed}\}), (\text{end}, \{\text{ordered}, \text{received}, \text{approved}, \text{fulfilled}, \text{invoiceSent}, \text{paymentReceived}, \text{paymentAccepted}, \text{paid}, \text{shipped}, \text{closed}\}))$	$(\text{start}, \text{A}, \text{B}, \text{D}, \text{F}, \text{G}, \text{E}, \text{H}, \text{end})$

individual task instead of exact effect, however the satisfiability of the former implies that of the latter.



**Fig. 2.** Discovered workflow  $\mathcal{W}_P$

## 4 Conclusion

Our semantic annotation of process models can help bridge between the high-level process and the actual IT infrastructure as support functions like discovery, composition, mediation can be implemented with concrete services using Semantic Web formalisms. Given an implementation of services and observed snapshots, we should be able to discover the process with its effect annotations using our algorithm. In future we plan to integrate our effect mining technique with data-driven approach to mining effect annotations (and specifically post conditions) from event logs arising in process execution histories [10].

## References

1. Dijkman, R.M., Dumas, M., Ouyang, C.: Semantics and analysis of business process models in BPMN. *Inf. Softw. Technol.* **50**(12), 1281–1294 (2008)
2. Fahland, D., Favre, C., Jobstmann, B., Koehler, J., Lohmann, N., Völzer, H., Wolf, K.: Analysis on demand: instantaneous soundness checking of industrial business process models. *Data Knowl. Eng.* **70**(5), 448–466 (2011)
3. Hinge, K., Ghose, A.K., Koliadis, G., Process, S.: A tool for semantic effect annotation of business process models. In: *Proceedings of the 13th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2009*, pp. 54–63 (2009)
4. Hoffmann, J., Weber, I., Governatori, G.: On compliance checking for clausal constraints in annotated process models. *Inf. Syst. Front.* **14**(2), 155–177 (2012)
5. Katsuno, H., Mendelzon, A.O.: Propositional knowledge base revision and minimal change. *Artif. Intell.* **52**(3), 263–294 (1991)
6. Kiepuszewski, B., ter Hofstede, A., van der Aalst, W.: Fundamentals of control flow in workflows. *Acta Informatica* **39**, 143–209 (2003)
7. Liberatore, P.: The complexity of belief update. *Artif. Intell.* **119**(1–2), 141–190 (2000)

8. Liu, R., Kumar, A.: An analysis and taxonomy of unstructured workflows. In: Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 268–284. Springer, Heidelberg (2005). doi:[10.1007/11538394\\_18](https://doi.org/10.1007/11538394_18)
9. Roy, S., Santriputri, M., Ghose, A.: Annotating and mining for effects of processes. Technical report, University of Wollongong, Australia (2015, Available on request)
10. Santiputri, M., Ghose, A.K., Dam, H.K., Wen, X.: Mining process task post-conditions. In: Johannesson, P., Lee, M.L., Liddle, S.W., Opdahl, A.L., López, Ó.P. (eds.) ER 2015. LNCS, vol. 9381, pp. 514–527. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-25264-3\\_38](https://doi.org/10.1007/978-3-319-25264-3_38)
11. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.* **16**(9), 1128–1142 (2004)
12. Dongen, B.F., Aalst, W.M.P.: Multi-phase process mining: building instance graphs. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) ER 2004. LNCS, vol. 3288, pp. 362–376. Springer, Heidelberg (2004). doi:[10.1007/978-3-540-30464-7\\_29](https://doi.org/10.1007/978-3-540-30464-7_29)
13. Vanhatalo, J., Völzer, H., Koehler, J.: The refined process structure tree. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 100–115. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-85758-7\\_10](https://doi.org/10.1007/978-3-540-85758-7_10)
14. Vanhatalo, J., Völzer, H., Leymann, F.: Faster and more focused control-flow analysis for business process models through SESE decomposition. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 43–55. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-74974-5\\_4](https://doi.org/10.1007/978-3-540-74974-5_4)
15. Weber, I., Hoffmann, J., Mendling, J.: Beyond soundness: on the verification of semantic business process models. *Distrib. Parallel Databases* **27**(3), 271–343 (2010)