UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

TESE DE DOUTORADO

COGNITIVEKiP – A COGNITIVE BPM THEORY FOR KNOWLEDGE-INTENSIVE PROCESSES

João Carlos de Almeida Rodrigues Gonçalves

Orientador
Até 15 de agosto de 2018: Fernanda Araujo Baião
De 15 de agosto até 6 de setembro de 2018: Gleison dos Santos Souza

Co-Orientador
Até 15 de Agosto de 2018: Flávia Maria Santoro

RIO DE JANEIRO, RJ – BRASIL
Setembro de 2018

COGNITIVEKiP – A COGNITIVE BPM THEORY FOR KNOWLEDGE-INTENSIVE PROCESSES

João Carlos de Almeida Rodrigues Gonçalves

TESE APRESENTADA COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE DOUTOR PELO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO).

Prof. Dr. Simone Bacellar Leal Ferreira, D.Sc. – UNIRIO

Prof. Dr. Gleison dos Santos Souza, D.Sc. – UNIRIO

Prof. Dr. Sean Wolfgand Matsui Siqueira, D.Sc. – UNIRIO

Prof. Dr. Giancarlo Guizzardi, Ph.D. - Free University of Bozen-Bolzano

Prof. Dr. Lucinéia Heloisa Thom, D.Sc. - UFRGS

RIO DE JANEIRO, RJ – BRASIL

Setembro de 2018

Acknowledgements

First, I would like to thank all that were at my side during this journey of 4 years of PhD.

A very special “thank you” to my parents, João Carlos and Ignez, for all the support and strength to not let me give up, even in the most difficult times.

To Marcella de Paula Carvalho, my girlfriend, companion and philosophical partner, I thank for all the support, love and patience and also for the innumerable discussions about Linguistics, where many of the ideas that compose this thesis were born.

To my advisors, Fernanda Baião and Flávia Santoro, my sincere gratitude for all those years of work and for that opportunity of learning so much under your advisorship.

To my colleagues of the ALICE research group, I thank you for all those years of work together.

To the colleagues of the RISE-BPM project, thanks for the opportunity of exchanging ideas and learning together.

And, last but not least, a sentiment of gratitude to everyone I forgot to thank in this section!

Almeida Rodrigues Gonçalves, João Carlos de. **CognitiveKiP – A Cognitive BPM Theory for Knowledge-intensive Processes**. UNIRIO, 2018. 200 páginas. Tese de Doutorado. Departamento de Informática Aplicada, UNIRIO.

RESUMO

Gestão de Processos de Negócio (BPM) tem sido aplicada em um grande número de organizações nas últimas décadas, geralmente descrevendo um fluxo de controle de atividades bem estruturadas que uma organização realiza de modo a atingir seus objetivos. Com a evolução das pesquisas em BPM, diversos pesquisadores começaram a perceber um tipo específico de process como crítico para organizações, denominados de Processos-intensivos em Conhecimento (KiP), caracterizados por um fluxo dinâmico e instável de controle que contém atividades complexas e basedas em conhecimento que mudam frequentemente em tempo de execução. Esta tese propõe a caracterização de um KiP em termos de Crenças, Desejos e Intenções dos participantes do processo. Embora estes já sejam os elementos que compõe um KiP, uma descrição detalhada sobre sua precisa conceitualização e como estes elementos influenciam ou determinam o comportamento de atividades durante a execução do processo – uma teoria – permanece como uma lacuna na literatura da área. Esta pesquisa possui um enfoque nas definições teóricas destes elementos, provenientes dos campos da Filosofia e da Pragmática. Esta tese de doutorado visa definir uma conceitualização precisa e bem fundamentada destes conceitos, denominada CognitiveKiP. A Knowledge-intensive Process Ontology (KiPO) é o ponto de partida para caracterizar KiPs de forma independente de domínio, buscando explorar os conceitos presentes em um KiP e descrever precisamente como Crenças, Desejos e Intenções, dentre outros elementos, são interrelacionados em sua caracterização. Em relação à metodologia científica da proposta, estudos de caso foram realizados de modo a explorar as possibilidades da conceitualização proposta em cenários reais, utilizando métodos de análise qualitativa. Os estudos de caso formam um conjunto de evidências para a análise e discussão dos novos conceitos e seu impacto na caracterização e compreensão de um KiP.

Palavras-chave: Cognitive BPM, Sistemas de informação, Processos-intensivos em conhecimento.

ABSTRACT

Business process management (BPM) has been widely applied in several organizations over the past decades, usually depicting the control flow of well-structured activities that an organization performs in order to achieve its goals. As the BPM research field evolved, many researchers started to perceive a specific type of process as being critical to organizations, the so-called Knowledge-intensive Processes (KiP), characterized by a dynamic and unstable control-flow of complex, knowledge-based activities that change frequently at run-time. This research proposes the characterization of a KiP in terms of the Beliefs, Desires and Intentions of the participants. Although those are composing elements of the KiP, a detailed description about their precise conceptualization and how they influence or determine the behavior of activities during the execution of a KiP – a theory – is still missing in the literature of this area. Based on this assumption, the study focuses on the theoretical definitions of these elements, taken from the fields of Philosophy and Pragmatics. This PhD thesis aims to define a precise and well-founded conceptualization of these concepts, named CognitiveKiP. The Knowledge-intensive Process Ontology (KiPO) is taken as a starting point to characterize KiPs in a domain-independent way, towards exploring concepts comprised in a KiP and precisely depicting how Desires, Intentions, Beliefs, among other elements, are inter-related in its characterization. With regard to the scientific methodology for the proposed research, case studies were performed to explore the possibilities of the proposed conceptualization in real-world scenarios, using qualitative analysis methods. The case studies' results formed a set of evidences for analysis and discussion of the new concepts and their impact on the characterization and understanding of a KiP.

Keywords: Cognitive BPM, Information Systems, Knowledge-intensive Process

Catalogação informatizada pelo(a) autor(a)

d635 de Almeida Rodrigues Gonçalves, João Carlos
CognitiveKiP: A Cognitive BPM Theory for Knowledge-intensive Processes / João Carlos de Almeida Rodrigues Gonçalves. -- Rio de Janeiro, 2018.
200

Orientador: Fernanda Araujo Baião e Gleison dos Santos Souza .
Coorientadora: Flávia Maria Santoro.
Tese (Doutorado) - Universidade Federal do Estado do Rio de Janeiro, Programa de Pós-Graduação em Informática, 2018.

1. Cognitive BPM. 2. Sistemas de informação. 3. Processos-intensivos em conhecimento.. I. , Fernanda Araujo Baião e Gleison dos Santos Souza, orient. II. Santoro, Flávia Maria, coorient. III. Título.

Index

Chapter 1 – Introduction 1

1.1) Motivation 1

1.2) Research Contributions 4

1.3) Objectives 4

1.4) Methodology 5

1.5) Intended Contributions 5

1.6) Structure 6

Chapter 2 – Knowledge Intensive Processes 7

2.1) Business Process Management 7

2.2) Knowledge-intensive Processes (KiP) 7

Chapter 3 - The Knowledge Intensive Process Ontology 13

3.1) The main role of Belief, Desire and Intention concepts within KiPO 17

Chapter 4 – CognitiveKiP Theoretical Foundations 19

4.1) Intentionality 19

4.2) On Intentionality and Intentional States 19

4.3) Process Participants and their Intentional States 20

4.4) On Intentionality and the Speech Act Theory 22

4.5) On Social Reality and Collective Intentionality 24

4.6) The BDI paradigm and its limitations 26

4.7) The Belief based Goal Processing theory 29

4.8) The theory of Common Ground Context 32

Chapter 5 - A Cognitive BPM Theory for Knowledge-intensive Processes 34

5.1) Basic Concepts from the Unified Foundational Ontology (UFO) 34

5.2) On Agents, Resources and Business Processes 40

5.3) On Knowledge-intensive Processes 47

5.4) On Communicative Interactions during a KiP 52

5.5) On Common Ground during the Interactions at a KiP 59

5.6) On Collective Intentionality and Social Concepts 62
5.7) Social Objects 63
5.8) Social Commitments and Claims: 64
5.9) Normative Descriptions 66
5.10) On Decisions during the execution of a KiP 69
Chapter 6 – Theory Evaluation 75
6.1) Definition of Theory in Information Systems 75
6.2) A Framework and Criteria for Theory Evaluation 78
6.3) Evaluation of the Parts 78
6.4) Evaluation of the Whole 81
6.5) Evaluating the CognitiveKiP Theory 83
6.6) Scenario 1: ICT Troubleshooting 94
6.7) Scenario 2: Open Source Software Development (GitHub) 108
6.8) Discussion of the scenarios for CognitiveKiP's falseability 131
Chapter 7 - Conclusions and Future Work 133
References 135
Annex A – Audacity repository log excerpt 142
Annex B – Alloy Analyzer code for CognitiveKiP 184

Figure Index

Figure 1 – KiPs in terms of structure, predictability and the difficulty in terms of modeling and automation [Di Ciccio et al., 2014] 8
Figure 2 – Main components of a KiP [Di Ciccio et al., 2014] 11
Figure 3 – KiPCO concepts related to a KiP and its interactions [França et al., 2012] 15
Figure 4 – DO Ontology and its main concepts [Pereira & Santoro, 2010] 16
Figure 5 – CO Ontology and its main concepts [Oliveira, 2009] 17
Figure 6 –Definitions from UFO regarding Agents and Objects [Guizzardi et al., 2008] 36
Figure 7 - Definitions from UFO regarding Actions and Events [Guizzardi et al., 2008] 39
Figure 8 – Basic BPM concepts of the CognitiveKiP Ontology 46
Figure 9 – Basic concepts for the depiction of Knowledge-intensive Process at CognitiveKiP Ontology 51
Figure 10 – CognitiveKiP Ontology elements regarding a Communicative Interaction 53
Figure 11 – Types of Speech Acts involved at a Communicative Interaction 58
Figure 12 – Common Ground in CognitiveKiP 61
Figure 13 – Social concepts of CognitiveKiP 68
Figure 14 – Decision elements of CognitiveKiP 73
Figure 15 – Framework for evaluating a theory by its parts [Weber, 2012] 81
Figure 16 – Framework for evaluating a theory by its whole [Weber, 2012] 83
Figure 17 – Solve ICT Incident as a Knowledge-intensive Process 97
Figure 18 – Solve Incident as a Knowledge-intensive Activity 99
Figure 19 – Communicative Interactions and Decisions (Solve Incident KiA) .. 101
Figure 20 - Contribute code as a Knowledge-Intensive Process 111
Figure 21 - Activities that compose the KiP "Contribute Code" 113

Figure 22 – Knowledge-Intensive Activity "Discuss and review code" 115
Figure 23 – Decision elements involved at Discuss and Review Code KiA 117

Table Index

Table 1 – Evaluation of the Constructs for Intentionality 85
Table 2 – Evaluation of the Constructs for KiP Participants 86
Table 3 – Evaluation of the Constructs for Activities and associated concepts.... 87
Table 4 – Evaluation of the Constructs for Decision 88
Table 5 – Evaluation of the Constructs for Communicative Interaction 89
Table 6 – Evaluation of the Constructs for Social Concepts 89
Table 7 – Evaluation of the States of CognitiveKiP's Constructs 90
Table 8 – Evaluation of the Events of CognitiveKiP's Constructs 91
Table 9 – Dataset field correlation of Scenario 1 with CognitiveKiP concepts.. 102
Table 10 – Ticket #165112 data contents 103
Table 11 – Ticket #240515 data contents 105
Table 12 – Ticket #240986 data contents 106
Table 13 – Dataset field correlation of Scenario 2 with CognitiveKiP concepts 118
Table 14 – Pull Request #130 data contents 119
Table 15 – Pull Request #157 data contents 124
Table 16 – Pull Request #70 data contents 130

Chapter 1 – Introduction

1.1) Motivation

Business Process Management (BPM) as a research field has evolved, and new research questions emerged, mainly involving the dynamics of the knowledge management cycle within an organization [Weske, 2007]. One of the most recent developments is the Cognitive BPM paradigm [Hull & Motahari-Nezhad, 2016], which fosters the application of Cognitive Computing technologies to the BPM ecosystem.

Many researchers already distinguish a specific type of unstructured process as being critical to most business scenarios, a.k.a. Knowledge-intensive Process (KiP). According to Hagen et al. [2005], a business process is knowledge-intensive if its aggregated value can only be reached through the fulfillment of the knowledge requirements of the process participants, while Gronau & Weber [2004] argue that KiPs are also characterized by a dynamic and unstable control-flow, and by the execution of complex activities that frequently change over time and even at runtime. All these characteristics demand for a precise understanding of all the circumstances that lead to each varying process execution; therefore, when it comes to the potential benefits for applying Cognitive Computing techniques, the concept of Knowledge-intensive Process is highly relevant. KiPs have been defined as a type of process that comprises sequences of activities based on intensive acquisition, sharing, storage, and (re)use of knowledge, so that the amount of value added to the organization depends on the actor knowledge. KiPs are complex, less repeatable and require a lot of creativity [Isik et al. 2013].

Based on an extensive literature review, Di Ciccio et al. [2014] affirmed that KiP are processes "whose conduct and execution are heavily dependent on knowledge workers performing various interconnected knowledge-intensive decision-making tasks". According to those authors, the expertise, experience, and decision-making capabilities inherent to knowledge workers essentially determine this kind of processes. Furthermore, they derived

eight key issues that typically characterize a KiP, namely: knowledge-driven; collaboration-oriented; unpredictable; emergent; goal-oriented; event-driven; constraint-and rule-driven; and non-repeatable. Additionally, Little and Deokar [2016] investigated the relevance of knowledge creation in KiP and argued that the expansion and use of knowledge across organizations rely on social processes (both formal and informal) through effective communication.

Marjanovic and Freeze [2011] cite customer support, design of new products/services, marketing, management of data quality, IT governance and strategic planning as examples of KiP. They observe that the way organizations deal with this kind of processes has changed over time, e.g. the customer support processes have evolved from highly structured to knowledge-intensive, and personalized, flexible cases.

Despite the fact that Knowledge-intensive Processes (KiP) are critical processes in organizations, analysts, process participants and stakeholders alike find them difficult to grasp completely, creating several hindrances for their modeling and management. Being complex and human-centric, they generate value through the knowledge exchange between participants, usually involve decision-making tasks with different alternatives for the same activity or different possibilities for the next step on the process flow, and a tacit decision-making rationale.

Due to the dynamic and people-centric traits of a KiP, there are several difficulties related to its analysis and modeling. The main challenge to precisely understand the essence of KiPs is its high variability, since process participants and stakeholders find them difficult to be described completely. We argue that the "human factor" is the main source of complexity, especially due to the difficulty of modeling human behavior in contrast to more structured and less human-centric operational processes.

To precisely conceptualize and analyze a KiP, it is necessary to understand the complexity of the cognitive elements involved during its executions, as well as how process participants interact and exchange knowledge. Therefore, our proposal explores the cognitive aspect of the BPM field, focusing on the advances of related domains of research such as Philosophy, Linguistics and Cognitive Psychology.

This research explores two key issues of a KiP: The first issue is the difficulty to effectively understand the human factor, involving tacit knowledge and overall unpredictability of this kind of process. This thesis tackles this problem by merging the advances of related fields of research such as Philosophy and Psychology into a coherent proposal for analysis of a

KiP and associated elements as this thesis argues that external theories are necessary for a broader understanding of this kind of business process. The second issue is the lack of a comprehensive semantic conceptualization for KiPs, due to a lack of an actionable framework that is feasible for application in the field of Business Process Management and its related disciplines such as modeling and execution.

The main goal is to propose a descriptive Cognitive BPM theory of a KiP, involving mental states (such as Belief, Desire and Intention) as well as their role in determining human actions. Besides, we also propose a comprehensive semantic conceptualization, based on solid foundations from the Unified Foundational Ontology (UFO), providing the theory with a precise semantics, avoiding issues such as conceptual ambiguity, false agreement, and semantic interoperability errors, as well as providing a conceptualization that is feasible for BPM applications.

The proposed theory encompasses concepts from the following works, which provides a theoretically sound basis to describe the cognitive perspective of KiPs:

- Searle's theory of Intentionality [Searle, 1983] as a theory of the mind that explains individual and collective human behavior and is centered on the concept of the Intentional State as a form of describing elements such as beliefs, desires and intentions, enabling a detailed analysis of them;
- The Speech Act Theory [Searle & Vandeveken, 1985] as a method for analyzing the expression of the Intentional States of KiP Agents, making it feasible to approach interactions between process participants as the exchange of Speech Acts expressing their Intentional States.
- The theory of Collective Intentionality [Searle, 1995] to deepen the understanding about interactions between KiP participants and shared externalizations of Intentional States at the social sphere of the process and the organization, such as the instantiation and description of important BPM concepts such as Social Roles, Social Objects, Social Commitments and Claims;
- Stalnaker's theory of Common Ground [Stalnaker, 2002] as a baseline to explain the dynamics of interactions between participants, especially the ones that trigger the execution of actions by process participants at runtime.
- Castelfranchi & Paglieri's Theory of Cognitive Regulation of Action [Castelfranchi & Paglieri, 2007] as a framework for the relationship between Beliefs and Goals, as well

as describing Desires and Intentions in terms of Goals, in contrast to the typical BDI theory applied to BPM.

1.2) Research Contributions

The contributions of the present research are two-fold. First, we propose a theory - aligned with the recent Cognitive BPM paradigm - that precisely and formally defines the cognitive aspect of a KiP. The proposed theory is structured as a well-founded ontology, named CognitiveKiP.

The main motivation of this work is the lack of a comprehensive theory of human action and interaction, based on the elements of Belief, Desires and Intention for Knowledge-intensive Processes, in order to avoid the previously mentioned issues. Although BPM as a discipline has steadily advanced at the past decades, especially broadening its scope towards different types of processes such as KiP [Recker & Mendling, 2016] and the fields of Process Mining [Aalst, 2011], Opinion Mining [Liu, 2012] and Data Science [Cao, 2016] have explored novel algorithms that attempt to extract and analyze human behavior, there is still the fundamental question of understanding how people ponder, interact and act in the scope of a single interaction, as well as on the broader scope of a whole process. In this sense, our proposed ontology is modeled and provides a meta-model for several computational applications - such as KiP learning and mining through NLP techniques - enabling them to analyze the cognitive elements and their relationships with other elements of the process.

1.3) Objectives

The objective of this research is the development of a theory about Beliefs, Desires and Intentions in the context of the research field of BPM, particularly of Knowledge-intensive Processes (KiP). Beliefs, Desires and Intentions are usually described as a set of inherent elements of the KiP participants, but lacking a detailed description about their relationships and their impact on the behavior of stakeholders within the scope of a Knowledge-intensive Process.

Moreover, these concepts have an intrinsic relationship to central elements within a KiP, such as Knowledge-intensive Activities, as well as knowledge exchanges among process participants. This research proposal also argues that, by understanding those elements, a broader understanding about the Knowledge-intensive Process as a whole is attained.

1.4) Methodology

The research proposal depicted in this Thesis aims to prove the following hypothesis in the field of Knowledge-intensive Processes:

H1: “A Knowledge-intensive Process is driven by the Beliefs, Desires and Intentions of its participants”.

The research aims to prove the hypothesis through the following distinct steps:

1) Theoretical Study: A theoretical discussion about the KiP field and its related areas, mainly the Discourse Analysis field, involving Pragmatics, Linguistics as well as Philosophy, Cognitive Psychology and Ontology Engineering.
2) Ontology Design: Specify the proposed theory for Beliefs, Desires and Intentions in the form of a well-founded ontology and analyze the model consistency, using the simulation of different scenarios of instances of the ontology, enabling the refinement of its concepts and axioms and the testing of the model’s boundaries.
3) Empirical Evaluation: A series of case studies to empirically gather evidences to deepen our study, especially within the scope of process participants and their relationships with the different elements of a KiP.
4) Analysis and Discussion: With both the theoretical study and empirical evidence, we can evaluate the initial hypothesis, answer the question proposed and formulate a theory for Beliefs, Desires and Intentions in KiP and the improvements of the initial ontology, reflecting the empirical findings and novel concepts and relationships.
5) Proposal Validation: The case studies and analysis performed in the previous steps will be used to provide input for the validation of the proposal and to define its limitations.

1.5) Contributions

The intended contributions of this research are pointed out as follows:

a) A theory concerning Belief, Desire and Intention and its role for KiP modeling and execution, including traits such as interactions between agents, the knowledge exchange and the flow of knowledge-intensive activities;

b) An extension of the KiPO Ontology, with the additional concepts and relationships concerning the proposed conceptualization on Belief, Desire and Intention;
c) Case studies concerning the proposal´s theory in practice, especially in real-world scenarios;
d) A gathering of evidence from the case studies and experiments performed, and a thorough analysis of their results, in order to validate and fully describe the theory in practice.

1.6) Structure

The structure of this document is the following: Chapters 2 and 3 defines Knowledge-intensive Process and its conceptualization as the well-founded domain ontology, KiPO; Chapter 4 presents theoretical foundations to describe the cognitive perspective of a KiP, from mainly the Discourse Analysis field, involving Pragmatics, Linguistics as well as Philosophy and Cognitive Psychology; Chapter 5 describes the proposed CognitiveKiP theory, and Chapter 6 depicts the theory's evaluation and assessment. Finally, Chapter 7 presents conclusions and future work.

Chapter 2 – Knowledge Intensive Processes

2.1) Business Process Management

Business Process Management (BPM) has been widely applied in several organizations over the past decades to achieve, among other goals, a common understanding of the process itself for a variety of stakeholders and process participants, bridging the gap between the perspective of business and IT alike [Rosemann, 2006]. Traditional approaches for process modeling usually depict a process focusing on the control flow of well-structured activities that an organization performs in order to achieve its goals. However, not all processes present a well-characterized control flow; Eppler et al. [2008] point towards four attributes to evaluate the degree of complexity of a business process: process steps, stakeholders, process dynamics and interdependencies.

Regarding process structure and the flow of activities, Hagen et al. [2005] classifies business processes as structured, semi-structured or unstructured. Structured processes are completely pre-defined, easily modeled using a specific language such as Business Process Model and Notation (BPMN), and repetitive, having a fixed sequence of activities. Examples of structured processes are: attendance orders, deliveries, inventory control, and payroll.

Unstructured (or ad hoc) processes comprise a kind of process that changes frequently, with its instances being very different from each other, both in terms of activities performed and flow. Its nature brings additional difficulty to model with a traditional method or notation. Finally, a semi-structured process shares unstructured and structured parts, sharing traits of both process types on different parts of its flow.

2.2) Knowledge-intensive Processes (KiP)

A traditional definition of a business process usually depicts a process focusing on the control flow, thus defining it as a composition of well-structured activities or other processes (sub-process) that an organization performs in order to achieve its goals [Weske, 2007]. Each activity of the process is characterized by its composing sub-activities, pre-activities, input and output artifacts, required resources and the procedures (methods, techniques) to be followed when

performing the activity. Particularly in the software domain, a software process can interact with other processes in several ways, among them: a process can precede the execution of another, two processes can be executed in parallel, or a process can be executed in a specific moment during the execution of another process [Falbo & Bertollo, 2009].

However, as the BPM field of research evolved, many researchers started to perceive an specific type of unstructured process as being critical to most organizations, a.k.a. Knowledge-intensive Processes (KiP). Among the diverse definitions of Knowledge-intensive Processes, a concise and brief definition is found on [Vaculin et al, 2011], defining KiP as "processes whose conduction and execution are heavily dependent on knowledge workers performing various interconnected knowledge-intensive, decision-making tasks". KiPs are genuinely knowledge-, information- and data-centric and require substantial flexibility at design- and run-time. Therefore, a process is knowledge-intensive if its value can only be created through the fulfillment of the knowledge requirements of the process participants. Moreover, they are characterized by a dynamic and unstable control-flow and complex activities that frequently change over time and even at runtime [Gronau and Weber, 2004].

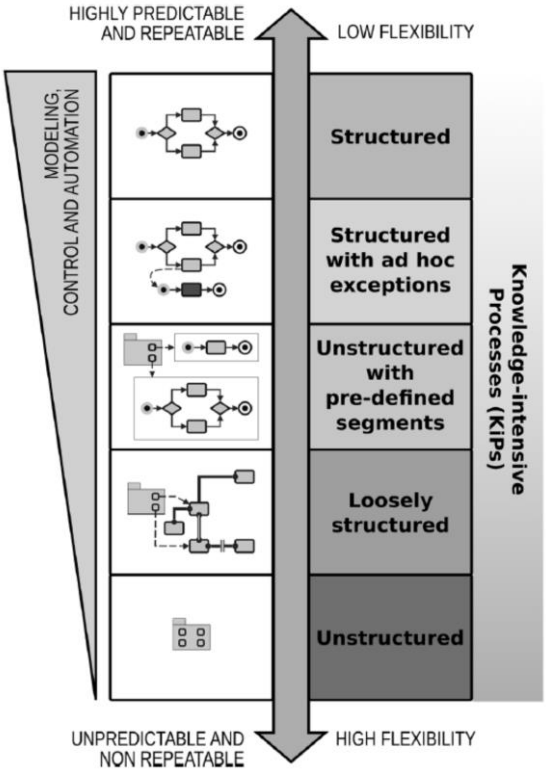

Figure 1 – KiPs in terms of structure, predictability and the difficulty in terms of modeling and automation [Di Ciccio et al., 2014]

Di Ciccio et al. [2014] complements the definition with a scale of classification of a KiP in terms of its flexibility, repeatability, predictability as well as the difficulty towards process-level modeling, control and automation tasks, as depicted in Figure 1. Expanding on this brief definition, recent studies [Unger et al., 2015] point to a set of nine common characteristics for KiPs:

- Knowledge-Prevalence: Knowledge is of utmost importance for the process. Usually knowledge from different sources and/or tacit knowledge is necessary for process execution.
- Collaboration: KiPs include activities often executed by many different process participants and intensive information exchange and coordination between them being a vital part of process execution itself.
- Predictability: Due to its unstructured nature, the flow of activities of a KiP can vary at each instance, due to situation specific needs or constraints.
- Complexity: The coordination of multiple information success, the variety on its execution flow, the variety of both sub-processes and tasks associated with the process itself and large number of participants makes complexity a key characteristic of a KiP.
- Structure: It is only possible to define a workflow that depicts a KiP partially, as unpredictable decisions or tasks guided by creativity are an inherent part of the flow of activities, as well as knowledge flows and knowledge transfers between media and persons being necessary to achieve a successful process completion. [Gronau & Weber, 2004].
- Goal-orientation: Although the unpredictable nature and complexity of KiPs is a hindrance to achieve a consistent structure, a minimum of structure can be achieved by defining milestones or intermediate goals during process execution.
- Event-Driven: Internal and external events may affect the quality of information exchanged during a KiP executing or require a participant to react for the successful achievement of the intended KiP process goal.
- Repeatability: The exact flow and order of activities, during each instance execution of a KiP, depends on several situational and contextual factors as well as possible external events that affect its participants. KiPs tend to be less repeatable than non-KiP processes, so an exact repetition of an instance, in terms of flow and activities of a

previously executed KiP, seems hardly possible, due to the variety of factors affecting each specific execution or instance.

- Frequency and Time-Horizon: KiPs tend to have longer run times than non-KiPs and, due to work changing hands over time and the inherent complexity of the process flow itself, no single individual has a full view of the process instance as a whole. Also, KiPs seem to be executed less frequently and are often of an strategic than operational character.

Focusing on the structure of a process and its activities, we have also found other definitions of a KiP that are complementary to the definitions listed above.

- Rule-based Execution of activities: We can define loosely a knowledge-intensive process (KiP) as a collection of activities of a business process, some activities of which can be knowledge-intensive (called `KiA'). The execution of KIAs depends on information specific for the chosen process instance. This kind of activity can also be modeled during build time, but its execution is triggered or suggested during runtime based on rules [Witschel et al., 2010].
- Contextual Information Dependency: Moreover, the information necessary to decide whether to execute a KIA or not usually come from the context of the process or activity itself, being, for example, application data, process data, functional data, among others [Brander et al., 2011a]. When performing a task, a person often consults resources and its selection based on a number of factors such as her personal skills, experiences or preferences. [Brander et al., 2011b]

Furthermore, Di Ciccio et al. [2014] describe the main components of a Knowledge-intensive Process, as depicted in Figure 2. At its core, we can identify Knowledge Workers collaborating with each other while they perform Knowledge Actions. These actions are integrated with the Data and Knowledge Elements, relying on their availability and content. All of them are tightly integrated and the relations between different Data and Knowledge Elements enables the flow of information to support the Knowledge Actions' performance and Decision Making. Rules and Constraints, often related to guidelines and best practices influence and coordinate the behavior of Knowledge Actions and Elements, as well as dictating their

mandatory or optional nature. All these elements are related to the specific Goals to be achieved, and mainly defined by Knowledge Workers.

The complex inter-dependencies and dynamics between all these elements induce an overall Coordination structure, coupled with the Collaboration structure composed by the Knowledge Workers, changing dynamically in relation to the actual context and Environment.

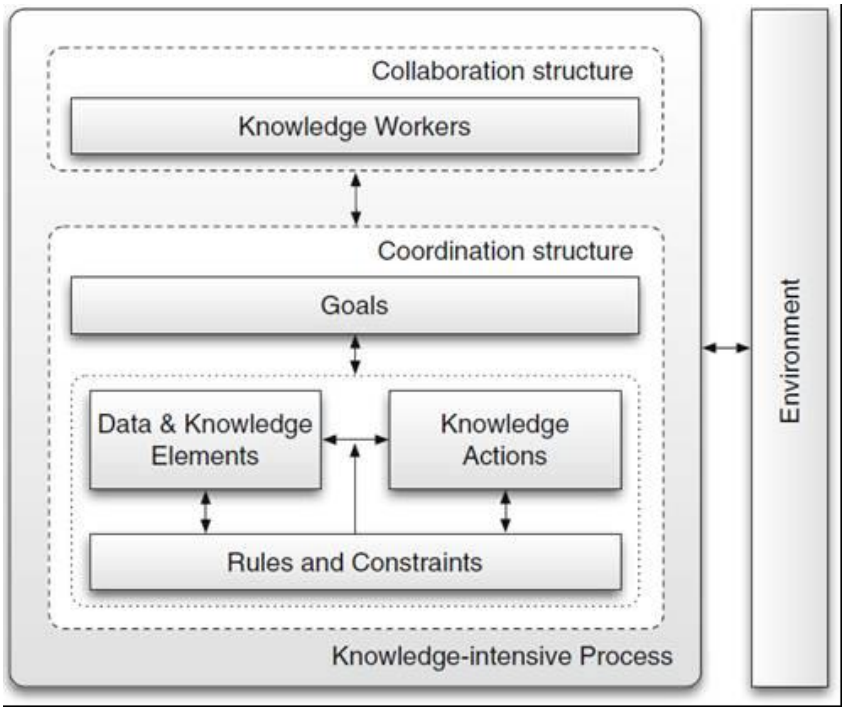

Figure 2 – Main components of a KiP [Di Ciccio et al., 2014]

Therefore, this research argues that an integral part of a KiP is in fact guided by the beliefs, desires and intentions of the process participants. The precise semantics of these elements, how they evolve from one to the other and how they relate to other elements of a KiP - such as Knowledge-intensive Activities and the Interactions among Agents during the execution of a KIA - are extremely important to understand the dynamics of a KiP, and to derive a reasonable explanation of its execution. These three elements are also inherent to the decision-making process embedded in a KiP´s flow of activities, in order to decide which activity is to be executed at each step at the process flow, using contextual information from a variety of sources to make that decision, usually made by stakeholders or process participants.

Due to the amount of tacit knowledge involved in KiPs, methods that use informal data of work practices, such as emails and personal task diaries, have already been used to extract process information [Brander et al., 2011a]. This same trend can be found for information

gathering and process modeling, using different knowledge sources such as collaborative narratives [Gonçalves et al., 2011] or emails [Soares et al., 2013].

Chapter 3 - The Knowledge Intensive Process Ontology

In order to characterize KiPs in a domain-independent and general way, we adopt the conceptualization provided by the Knowledge-intensive Process Ontology (KiPO) [França et al., 2014]. KiPO provides well-founded definitions which enable us to explore the concepts comprised in a KiP and depict how Beliefs, Desires, Intentions and Feelings are inherent to it. As a domain ontology, KiPO is an explicit and formal representation of a shared conceptualization [Guarino, 1995] and an abstraction that depicts BPM and KiP concepts and their relationships of process, independently of process domain.

KiPO precisely defines the semantics of each concept involved within a KiP by referring to the meta properties of the constructs of a top-level ontology, named Unified Foundational Ontology (UFO). UFO is a foundational ontology, in the sense that it provides a system of basic categories and relations whose intended meaning is grounded in very general principles inspired by Formal Ontology, Philosophical Logic, Linguistics, and Cognitive Psychology, and formally characterized by means of logical axioms. UFO consists of three main modules: UFO-A [Guizzardi, 2005], an ontology of Endurants (objects); UFO-B [Guizzardi et al., 2013], an ontology of events (Perdurants); UFO-C [Guizzardi et al., 2008], an ontology of social entities built up on UFO-A and UFO-B; and UFO-S [Nardi et al., 2015], an ontology of services based on commitments. With regard to UFO-A, the set of meta-properties was codified and represented as a UML profile, named OntoUML [Guizzardi, 2005].

The KiPO ontology takes into consideration the meta properties of UFO-based categories. Moreover, KiPO comprises five complementary perspectives, being each of them a sub-ontology itself:

1) **Business Process Ontology (BPO):** containing common process elements such as Activities, Flows and Data Objects.;
2) **Collaborative Ontology (CO)**: depicting concepts common to the knowledge exchange and collaboration between process participants;
3) **Decision Ontology (DO)**: represents the “why” and “how” the decisions are made by process executors, explaining their rationale;

4) **Business Rules Ontology (BRO)**: representing the rules or constraints that must be observed throughout the execution of a KiP, crucial elements to understand and synthesize the high degree of variability typically evidenced in a set of KiP instances since KiPs are typically more declarative than procedural by nature, being constrained by organizational norms and usually described by business rules; and
5) **Knowledge-intensive Process Core Ontology (KiPCO)**: containing concepts and elements that are specific to the Knowledge-intensive Processes and interrelate concepts from the other perspectives.

The Knowledge-intensive Process Core Ontology (KiPCO) is the core ontology of KiPO, dealing mainly with Agents, the Knowledge-intensive Activities they perform and the contextual elements involved on the Knowledge-intensive Activity. Among its elements, Intentions, Desires, Beliefs and Feelings of an Agent are key elements for the execution of the KiP and its activities.

An Agent is defined as "the one who intentionally commits to reach a Goal by executing a Knowledge-intensive Activity. The Agent is motivated by his Desire and acts, according to his Belief." [França et al., 2012].

Moreover, an Agent of a KiP may perform two different kinds of roles at KiPCO: the role of an Innovation Agent or of an Impact Agent. An Innovation Agent has a specialty in some area related to the KiP domain and therefore typically contributes to the execution of a knowledge-intensive activity with innovation and creativity; whereas an Impact Agent is responsible for executing a knowledge-intensive activity and for identifying questions during its execution.

A Fact in the universe of discourse may generate many Evidences in a KiP scenario and an Agent may experience many Feelings, and each one of them is motivated by his Beliefs and/or Evidences presented to him.

Finally, a number of Beliefs of the Agent become one or more Intentions, being specific types of Desire of the Agent. The Intentions impel the Agent to execute the Knowledge-intensive Activity towards the achievement of the Activity Goal. Figure 3 depicts the KiPCO concepts described and its interactions.

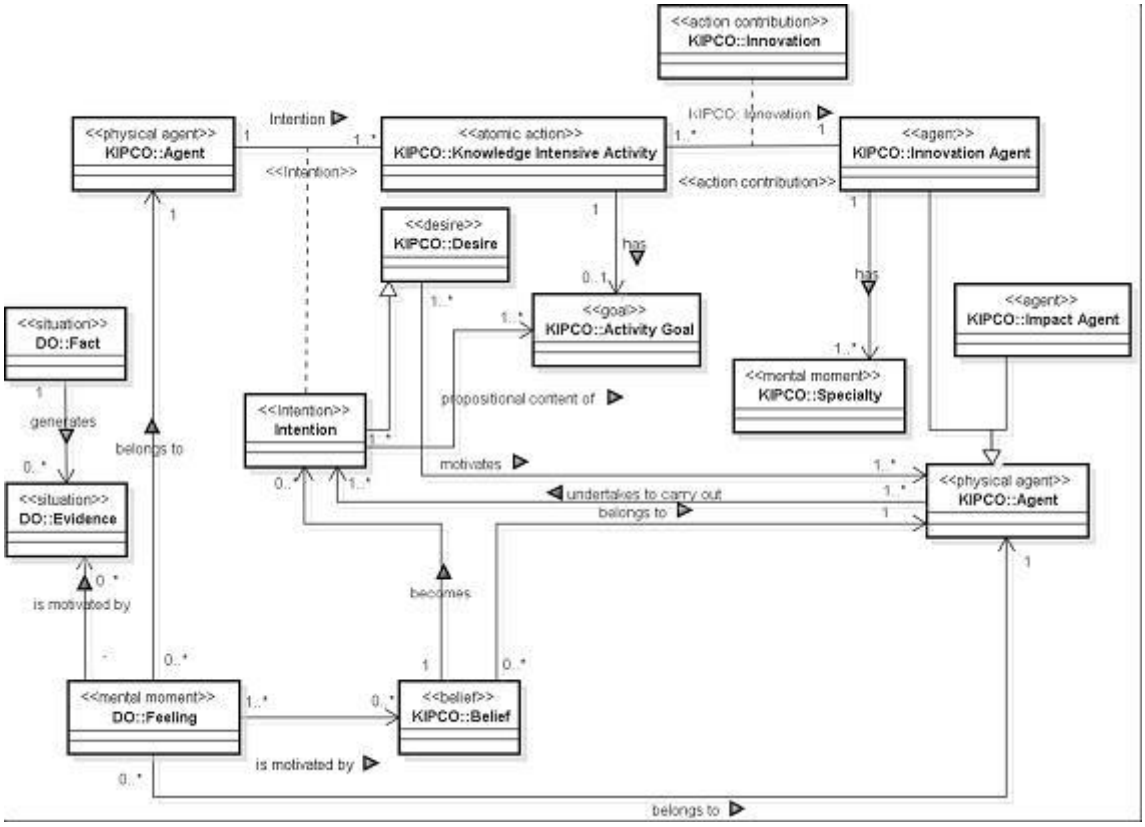

Figure 3 – KiPCO concepts related to a KiP and its interactions [França et al., 2012]

Therefore, we argue that the Intentions and Beliefs of an Agent guide their action when performing a Knowledge-intensive Activity, both being key elements for the modeling and understanding of a KiP.

Intentions are also present in another KiPO perspective, the Decision Ontology (DO), where they play a main role on the decision-making process inherent to a KiP [Pereira and Santoro, 2010], especially on its flow of activities. According to DO, "An Agent is one who intentionally commits to solve a Question by performing a Decision, which is itself a special type of KIA. A Question is then a Contingency (and as such, an event) that triggers the Decision to be made. When making a Decision, the Agent chooses among several Alternatives, which represent potential situations that to be achieved, depending on the decision result. Each Alternative satisfies a set of Advantages and does not satisfy a set of Disadvantages". Figure 4 depicts DO and its main concepts.

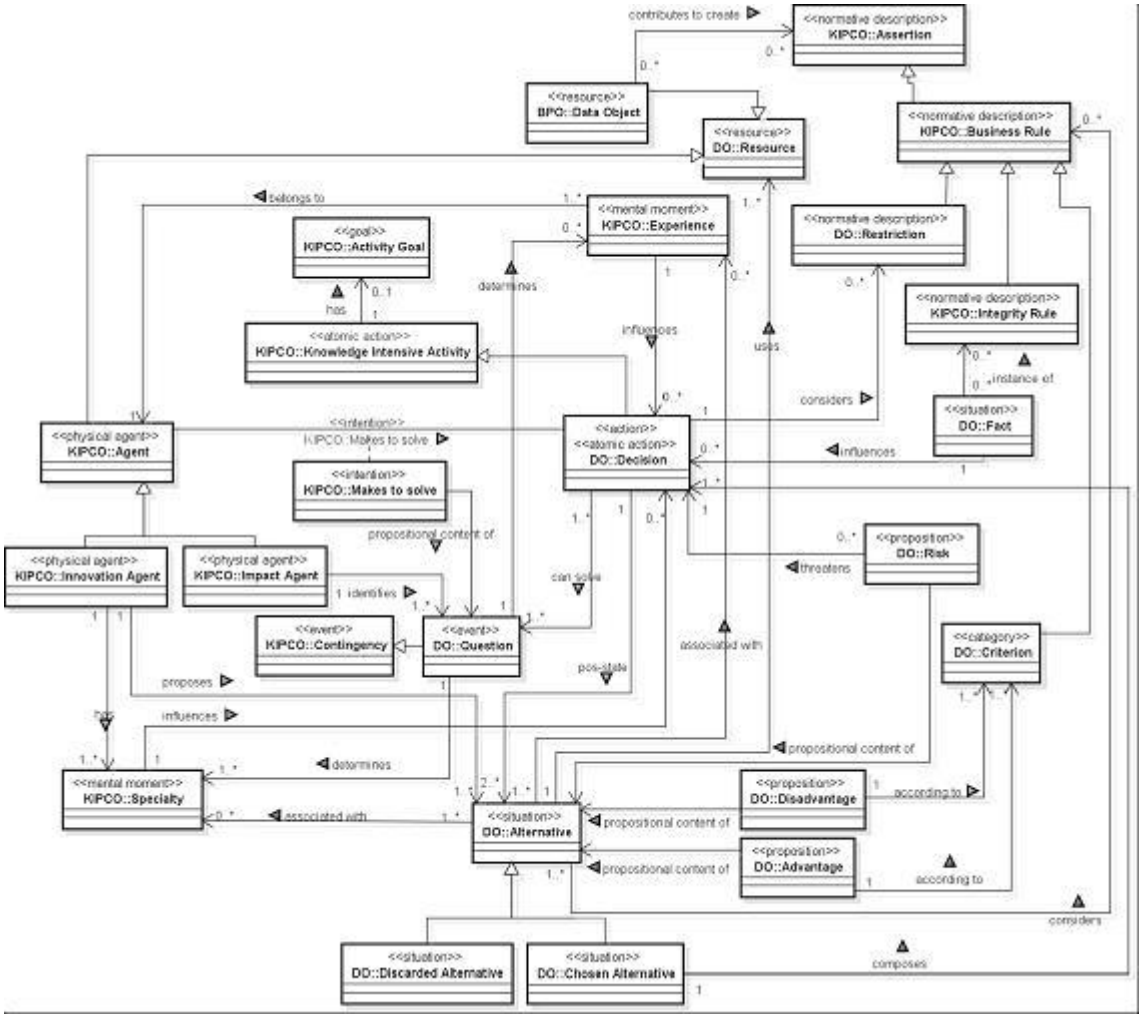

Figure 4 – DO Ontology and its main concepts [Pereira & Santoro, 2010]

As presented above, a Decision is considered a type of Knowledge Activity and thus, the Intentions and Beliefs of the Agents involved are present as well.

Regarding the Collaboration perspective, depicted in CO (Collaboration Ontology), KiPO describes how the Agents interact to exchange knowledge, communicate and socialize, creating mental images or Beliefs that will be applied as they participate at the KiP [Oliveira, 2009]. As the ontology specifies:

"An Agent performs Communicative Interactions, which are composed by Communications and Perceptions. Within a Communication, Messages are exchanged by Agents playing the role of Senders and Receivers. An Agent (as a Sender) sends a Message to start a Communication, which is the propositional content of his Message. The Message reaches the receiving Agent (as a Receiver), who develops a Perception from the Message content. A special type of a Communicative Interaction is a Socialization, which may also involve the

participation of External Agents (by contributing with knowledge when discussing alternatives)" [França et al., 2012].

Socializations are very common in KiPs, which involve discussions or require reaching a consensus among its participants, and may take place through different media, such as a forum discussion or an email track. Figure 5 depicts the CO Ontology.

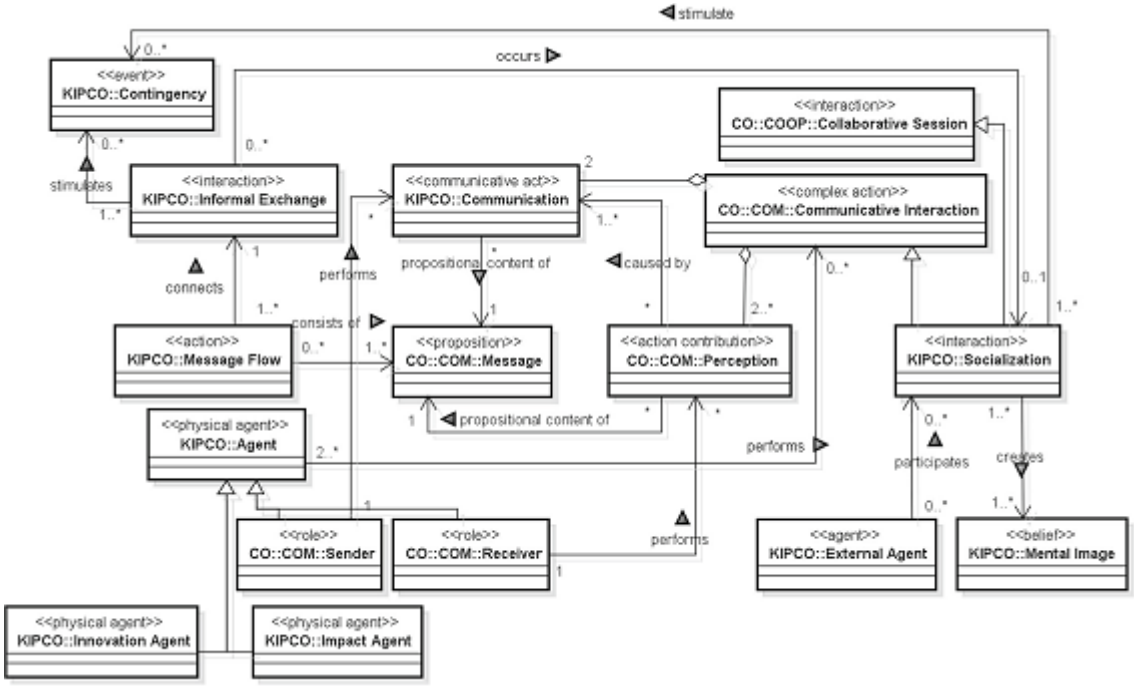

Figure 5 – CO Ontology and its main concepts [Oliveira, 2009]

3.1) The main role of Belief, Desire and Intention concepts within KiPO

Based on the above discussion, belief, desire and intention are elements that are inherent to a KiP and the motivation for its activities. These elements are also involved (especially the beliefs of the Innovation Agents, Impact Agents or External Agents) in the knowledge exchange between process participants. Therefore, the following ideas taken from KiPO, depicted as statements, form the base for the initial exploration of the research. All of them are based on the discussion about the KiPO Ontology and the belief, desire and intention elements:

Statement A: "Communicative Interactions (Socialization being a subtype of them) are performed by Agents, with or without the participation of External Agents, to exchange knowledge and create Beliefs among participants".

Statement B: “A subset of the Beliefs of an Agent becomes Intentions. An Intention is a specific subtype of a Desire of the Agent”.

Statement C: “The Intentions impel the Agent to execute the Knowledge-intensive Activity (or a decision-making inherent to a KiP) towards the achievement of the Activity Goal or the definition of the next activity on the process flow”.

The statements above will guide the thesis towards the exploration of the concepts related to Belief, Desire and Intention at the related fields of knowledge, such as Philosophy and Pragmatics.

Chapter 4 – CognitiveKiP Theoretical Foundations

This Chapter describes the theoretical foundations for the CognitiveKiP theory. Each section will introduce a theory from an external field of research and describe the concepts used for the thesis proposal and their importance and effects at a KiP.

4.1) Intentionality

The discussion about Intentionality is of paramount importance for a precise definition of a Knowledge-intensive process, since it is crucial for understanding the definition of the three intentional states (beliefs, desires, intentions) that we hypothesize as being the ones that determine the activities that take part on each execution of a KiP.

4.2) On Intentionality and Intentional States

Searle [1983] defines Intentionality as "the property of many mental states and events by which they are directed at or about or of objects and states of affairs in the world".

For example, if I have a belief, it must be a belief about a situation or something; if I have a desire, it must be a desire to do something or that something should happen or be the case; if I have an intention, it must be an intention to do something. For the sake of clarity, we consider important to distinguish concepts of Intentionality and Intention: while Intentionality is the directedness (that is, a property or quality of being intended at) of a mental state, an Intention is a (specific subtype of) mental state per se.

We also adopt Searle's definition applicable to the context of mental states. Hence, the mental states that carry Intentionality are defined as Intentional States. Every Intentional State is composed by an intentional content in a psychological mode. The intentional content also determines a direction of fit and conditions of satisfaction. The conditions of satisfaction are the conditions that should be true so that the intentional content (proposition) is satisfied, and the direction-of-fit is the direction of fulfilment of the conditions of satisfaction in relation to the world. The direction-of-fit can be of two kinds: world-to-mind (in the case of the world must correspond to the content proposition) or mind-to-world (the content proposition must

correspond to the world). For example, If I have a belief that it is raining, the intentional content of my belief is: “That it is raining”, and the psychological mode is the belief directed towards the propositional content. The conditions of satisfaction are: “that it is raining” (and not “that the ground is wet” or “that water is falling out of the sky”) and the direction-of-fit is world-to-mind, as the world must satisfy the conditions of satisfaction (i.e. “raining”).

According to Searle [2004], desires and intentions cannot be evaluated to true or false the way beliefs are, because their aim is not to match an independently existing reality, but rather to get reality to match the content of the Intentional state. For that reason, Searle argues they have the “world-to-mind” direction of fit.

An essential property of an Intentional State is that it is intrinsic and inherent to an agent’s mind, in that it belongs to a unique individual, and cannot be shared or transferred among distinct individuals. Therefore, it only takes the existence of one agent for an Intentional State to arise and to exist, and this may take place when the agent is either in solitude (meditating or practicing yoga by himself or being left alone in a desert island) or in social environments (doing his work within an organization or planning the next vacation with his family). For the purpose of this research, however, we are interested in investigating if and how intentional states drive the actions of agents involved in the execution of a KiP in a social environment, more specifically within a business organization.

In such type of environments, it is reasonable to assume that there is the need (of the organization) and commitment (by the agents) of communication among agents who co-participate in the same execution of a KiP, and therefore the agents are (or should be) willing to externalize (at least a subset of) their Intentional States. An Intentional State may be externalized by means of several representations, ranging from graphical representations (such as drawings or diagrams) to textual representations (captured from both written texts or spoken language) and in different media formats. For example, social networks are a popular infrastructure in which these externalizations take place in a very large number of ways and are frequently used by agents involved in process executions as a means for exchanging knowledge among them.

4.3) Process Participants and their Intentional States

Taking into consideration the definition from KiPCO, an Agent is defined as “the one who intentionally commits to reach a Goal by executing a Knowledge-intensive Activity. An Agent

is motivated by his inherent Desires and acts according to his inherent Beliefs." (França et al., 2012). Among other related elements, the mental states of the Agent (such as Intentions, Desires and Beliefs) are key for the execution of the KiP. All these mental states inhere in an Agent and thus are unique to each specific individual. First, we adopt the definition of Belief as a "belief state" from Yalcin [2016] as "a set of metaphysically possible worlds, intuitively the worlds ´left open´ by what is believed.

Thus, propositions are sets of possible worlds, and the propositions an agent believes are those true with respect to all of those worlds the state leaves open". Rather than representing the beliefs of an agent by a single set of possibilities, they are represented by a collection of sets of possibilities. Therefore, agents do not have a single belief state; rather, they have a set of belief states, or "separate systems of belief" (such as beliefs regarding the culture of an specific nation or the organization where the process is executed), the contents of which are each represented by a set of possible worlds. This definition can also be taken as two distinct viewpoints: (i) the map aspect: the state of an agent that represents the world as being a certain way and (ii) the steering aspect: the explanation of the agent's actions. Both aspects are critical to the cognitive aspect of a KiP, as they explain not only the perceptions and viewpoints of an Agent but the rationale of their actions.

Yalcin [2016] also extends the definition to a "Resolution State Model", defining a function between a belief state and a specific question or subject matter: "A belief state is representable as a partial function mapping a resolution of logical space (question or subject matter) to a belief partition (answer or information about the subject matter).

The definition has a direct relationship to a KiP dynamics, specifically to the dynamics of the decision-making aspect inherent at each Knowledge-intensive Activity and at each part of a flow of activities. The agent´s beliefs are directly related to the solving of questions involved on the making of a decision or even on the answering of a request for information about a specific subject.

Second, Malle et al. [2001] define desires and intentions as representational states, and they both express a pro attitude towards the state of affairs that they represent, frequently propelling the agent to act in such a way as to bring about that state. The main difference between them is that the desire is not related to a following action, while an Intention is reached after carefully considering desires, is related to one's own following action and precedes an action.

Moreover, there is a close relationship between an action´s goals and the action itself, in the sense that the goal is the cause for the action to occur and, sometimes, the action will continue until it satisfies its goal [Dretske, 1999]. With both definitions in mind, a relationship between an intention and a goal can also be indirectly stated, since (i) an intention is essentially related to the action it aims to perform, and (ii) an action is related to the goal it aims to fulfill.

Within KiPO, the definition of Intention aligns with this discussion, and is defined as a self-commitment towards the fulfillment of a specific goal (i.e., a commitment of the person that holds the intention with herself directed towards the pursuit of a goal through the execution of an action). In particular, we focus on the textual representations for identifying and analyzing Intentional States, and we argue that one of the most common forms of analysis of their representations is through the theory of Speech Acts, presented as follows.

4.4) On Intentionality and the Speech Act Theory

Speech Act Theory (SAT) can be applied to describe how the experience of Intentional States is shared and communicated between individuals. Although the Intentional State itself is unique to each person, a number of externalizations can be expressed by language in the form of Speech Acts.

A Speech Act can be defined as an illocutionary act, i.e. the act that one performs in producing an utterance, such as an act of asserting a proposition, asking someone a question, or directing someone to do something. A basic definition comes from Austin [1969], depicting the analysis of a speech act at three levels:

- Locutionary act, the performance of an utterance: the actual utterance and its ostensible meaning, comprising phonetic, phatic and rhetic acts corresponding to the verbal, syntactic and semantic aspects of any meaningful utterance.

Ex.: The utterance performance of “Leave the Room”

- Illocutionary act: the pragmatic 'illocutionary force' of the utterance, thus its intended significance as a socially valid verbal action.

Ex.: “Leave the room!” (Order), “Leave the room.” (Request)

- Perlocutionary act: The actual effect of the speech act itself, such as persuading, convincing, scaring, enlightening, inspiring, or otherwise getting someone to do or realize something, whether intended or not. Some speech acts have no perlocutionary act at all.

Ex.: The act of leaving the room by the hearer, after the successful performance of the speech act by the speaker.

Searle and Vanderveken [1985] describe the illocutionary act as the minimal unit of human conversation, examples of it being statements, questions and commands. Whenever a speaker utters a sentence in an appropriate context, with certain intentions, he/she performs one or more illocutionary acts.

Formally, an illocutionary act is defined as having an illocutionary force F and a propositional content P (In the form of "F(P)"), which respectively denotes the speaker´s intention on making the utterance and the meaning of a clause or sentence that is constant of the illocutionary act.

There is a clear distinction between Speech Acts and Intentional States, the former are acts, depending on the production, presentation and/or usage of physical realizations (such as writing on paper or speaking) and the latter are mental states, independent of physical realization. Besides, a Speech Act has a double level of Intentionality: The Intentional State expressed by it and the intention of properly emitting the speech act per se (the communication act). For example, the speech act: "John believes that Steve is bad" has the Intentional State of the belief of John "that Steve is bad" and John's intention of emitting the speech act itself and conveying its associated Belief, both being distinct forms of Intentionality.

Searle and Vanderveken [1986] proposed a taxonomy of Speech Acts, composed of five main classes:

- Assertives: commit a speaker to believing the expressed proposition (e.g.: "I believe Steve is bad")
- Directives: cause the hearer to take a particular action (e.g.: "I am telling you to sit down!")
- Commissives: commit a speaker to doing some future action (e.g.: "I promise to fix the problem.")
- Expressives: express the speaker's attitudes and emotions towards the proposition (e.g.: "I thank you very much!")

- Declaratives: change the social sphere in accordance with the proposition of the declaration (e.g.: “I hereby declare you husband and wife.”)

In order to illustrate the relationship between speech acts and intentional states, we can take the example “I believe that Steve is bad”: There is an assertive illocutionary force and a propositional content “That Steve is bad”, expressing the Intentional State of a Belief, committing the speaker to possess the belief of “that Steve is bad”; whether the belief is true or false, characterizes the speech act as sincere or insincere.

4.5) On Social Reality and Collective Intentionality

Given that a (knowledge-intensive) process is essentially a social concept conducted within an organization by a subset of its agents, it is crucial to extend the discussion on intentionality to address not only its definition from an individual perspective, but also its collective counterpart, in social reality. Thus, intentionality is not only restricted to an individual, there is also the concept of Collective Intentionality.

Collective intentionality is defined by Searle to explain collective work: “The crucial element of Collective Intentionality is a sense of doing something together. Individual Intentionality that each person has is derived from the collective intentionality they share” [Searle, 1995]. A typical example is of a violinist playing a symphony in an orchestra. He has an individual intention of playing the violin as part of the collective intention of the orchestra playing the symphony. One can notice that the violinist is not only an individual violinist anymore, since he/she assumed a function of “Orchestra musician”, thus agreeing to commit to a series of rights and obligations of this function. This phenomenon occurs by the collective acceptance of the violinist as part of the orchestra by the group, in contexts of playing for an audience.

In this sense, a collective intention (for example, “playing a symphony at a concert”) cannot simply be decomposed into several individual intentions such as “playing the violin” or “conducting the symphony” (that is, the whole is not the sum of its parts). Instead, we adopt Searle’s point of view that individual intentionality is derived from the collective intentionality they share while performing an action together. Thus, at the orchestra example, both the violinist and the maestro would have their respective individual intentions to “play the violin together with the orchestra” and “conduct the symphony together with the orchestra”, both

derived from the collective overarching intention of the orchestra “playing the symphony at a concert”.

Another interesting feature of collective intentionality is the phenomenon of imposition of function: According to Searle, “The distinctive feature of human social reality, the way in which it differs from other forms of animal reality known to me, is that humans have the capacity to impose functions on objects and people where the objects and the people cannot perform the functions solely in virtue of their physical structure”. The “imposition of function” operation (or status-function) was also described by Searle [1995] as:

Given a context C, an object X and a function Y, “X counts as Y in context C”.

The new function Y that object X receives is only able to be performed in virtue of the collective acceptance of the new function by the group [Searle, 2003]. As an example, a written sheet of paper, can suffer the imposition of function of “university certificate”, enabling a person to have the status of “PhD”. The potential status will only be enforced by the collective recognition of a relevant group (for example, the counterparts at academia).

By the means of the status-functions, the concepts that compose social reality are created, modified or cease to exist, based on intentional states such as Beliefs. At the sphere of a business process, it can translate as roles such as manager, analyst, technician, team leader and others, being people suffering the imposition of their respective functions, with responsibilities and powers, described by Speech Acts composing work contracts, internal regulations and organizational chart descriptions, for example.

Thus, the three aspects of Intentionality and the effects on business process are: (i) Intentional States, depicting the mental states involved during the execution of the process by the Agents; (ii) Speech Acts, indicating the dynamics of sharing and transmission of representation of each Agent’s Intentional States with other process participants and (iii) Collective Intentionality, describing how the Intentional States and Speech Acts affect the social sphere of the whole process, involving usually organizational features such as roles, resources, responsibilities and claims.

4.6) The BDI paradigm and its limitations

An initial approach on how the participants of a business process' mental states and their consequences on the flow of actions of a process can be further analysis by the Belief-Desire-Intention (BDI) model of human practical reasoning, developed by Michael Bratman. According to the original proposal, Bratman's BDI model proposed a rich analysis where goals are replaced by desires and intentions [Bratman, 1987]. His integrated account is called belief–desire–intention model, BDI model for short.

The BDI paradigm highlighted the fundamental role of an agent's future-directed intentions: they are high-level plans to which the agent is committed and that she refines step by step, finally leading to intentional actions. Intentions therefore play a role that is intermediate between goals, plans, and actions. Intentions are commitments and thus, stable mental attitudes. The strength of the commitment is apparent at the theory, as there are only two possible reasons to abandon an intention: (i) either it turns out to be impossible to satisfy; (ii) or it is only instrumental for another, higher-level intention the agent is about to abandon.

Being high-level plans, Intentions cannot be executed directly: they have to be refined as time goes by, resulting in more and more elaborate plans. At the end of the refinement process there are basic actions, which are the actions the agent can directly execute. While intentions have to be refined in order to obtain executable actions, this should not be done too early. The main cause for this fact is the limitations of computational power and memory of an agent that must be taken into consideration and its inability to store fully elaborate plans for the far ahead future. In a perfect scenario, with unlimited resources, even so the agent would only have imperfect beliefs about the future that may turn out to be wrong: fully worked-out plans would force her to re-plan much more frequently than more abstract, high-level plans would. Therefore, we can state that the process of "refinement of intentions", as described, is a fundamental issue in an agent's management of its intentions.

Forming future-directed intentions enables agents to extend the influence of their deliberations beyond the present moment. This is important given the limited cognitive capacities and time for deliberation of human agents. It also brings into consideration the limitations in terms of content and resources of the deliberating agents.

Collective attitudes such as common goal and joint intention are of utmost importance for the field of BPM, as they are concepts that are inherent to the definition of a Business Process and its inner workings. Notable examples of collaborative activity are painting a house together,

dancing together a tango, or moving a heavy object together. Two or more agents acting together in a collaborative way need to have a common goal and need to form a joint intention aimed at achieving the common goal. In order to make collaboration effective, each agent has to commit to her part in the shared plan and form the corresponding intention to perform her part of the plan. Moreover, she has to monitor the behaviors of the others and, eventually, to reconsider her plan and adapt her behavior to new circumstances.

The interesting aspect of joint intention is the conditional nature of the individual intentions composing it. Specifically, an agent in a group has the intention to do her part in the shared plan conditional on the fact that the other agents in the group also intend to do their part. In this sense and as Bratman emphasizes [Bratman, 1987], the individual intentions composing a joint intention form an interlocking web of individual intentions. From this perspective, joint intention refinement and revision are interdependent as: (i) the refinement of an individual plan by an agent in the group may lead to the refinement of an individual plan by another agent in the group, and (ii) the reconsideration of an individual intention by an agent in the group may trigger the reconsideration of an individual intention by another agent in the group.

An example would be the activity of painting a house together: let's say two agents Mary and Bob have the joint intention to paint a house together. Two options are available: the house can be painted either in blue or in green. Mary refines her individual plan by deciding to paint the house in blue. Consequently, Bob has to refine his individual plan in the same way by deciding to the paint the house in blue. Now, suppose Mary reconsiders her individual intention to paint the house in blue and chooses to paint the house in green. In order to coordinate with Mary effectively, Bob too should change his plan and decide to paint the house in green.

To sum it up, joint intention cannot be considered before individual intention is clearly characterized. It was well received in Artificial Intelligence: numerous approaches adopted the BDI paradigm, either from an implementation perspective—so called BDI agent languages and BDI software agents.

The idea that intentions imply some kind of commitment is explicit in Bratman's theory. It is this peculiarity which qualifies intention for a functional role that mere desires do not play. Once an agent has deliberated in favor of an action and has formed the corresponding intention, he is ''locked into'' the project that he has decided to pursue and, in the absence of relevant new information, the intention to do the action will resist further reconsideration. Consequently, since they are the product of deliberation and having associated a kind of commitment,

intentions are characterized by an intrinsic form of persistence which makes them more resistant to temptations than desires.

The BDI model is a theory that enables the analysis of future-directed intention and had an impact in several areas, ranging from psychology to agent-based computing [Bratman, 1987]. It is also important for the field of BPM, especially for knowledge-intensive processes, since it may be adopted to describe the reasoning process that leads an agent to execute a specific action. The concepts of belief and goal play a central role in the design and implementation of autonomous agents. These concepts are considered to be fundamental mental attitudes of agents: beliefs have a 'mind-to-world' direction of fit (agents try to adapt their beliefs to the truths of the world), while intentions have a 'world-to-mind' direction of fit (agents try to make the world match their goals).

Several works adopted the BDI paradigm, either from an implementation perspective (as BDI-based software agents) or from a formal logic perspective, with Cohen & Levesque's [1990] and Rao & Georgeff's [1991] being the most influential. Recent studies, however, especially by Herzig [2016], point out several limitations of the BDI paradigm. Our study has an specific focus on three of them, which were already pointed out by Herzig [2016]: (i) The lack of analysis in the literature about refinement of intentions (a fundamental concept for the modeling of autonomous agents); (ii) The lack of support for the evolution of an agent's belief (when he/she learns that she was wrong about a proposition); (iii) The lack of support for the evolution of an agent's knowledge and belief (when some external event occurs and is perceived/reacted upon); These limitations impact all fields of research influenced or driven by BDI, including the Knowledge-intensive Processes area of BPM, since its core perspective deals with the relation of BDI concepts with the agents decisions.

Another limitation for its application at the context of the Cognitive BPM paradigm is that Bratman's theory is qualified as a planning theory of intention and traditionally opposed to so-called cognitivist theories of intention [Herzig, 2016], due to the fact that, while according to Bratman's theory, intention has certain distinctive functional properties which cannot be adequately characterized by conceiving it as a combination of a desire to do a certain action plus the belief that one will do the action (or the belief that one will possibly do the action), the cognitivist view defends the idea that intention basically consists in the belief that one will act in a certain way (or, will try to act in a certain way). Thus, according to this view, an agent's intention involves a sort of self-referential aspect: the belief that an intention to perform a

certain action a in the future will be responsible for the future occurrence of action a (or the future attempt to do the action a).

Our proposal aims to address the three limitations mentioned above within the scope of Business Process Management and, more specifically, Cognitive BPM. The first limitation (the lack of analysis about refinement of intentions) will be addressed by adopting the Goal-Processing paradigm from Castelfranchi & Paglieri [2007], while the problems related to the agent's belief evolution (limitations ii and iii) will be addressed by the Common Ground theory, from Stalnaker [2014], each of them described in the following sections.

4.7) The Belief based Goal Processing theory

Castelfranchi & Paglieri [2007] proposed a theory regarding the role of beliefs in goal processing, focusing especially on the cognitive process that leads from a desire to an intention. Their research provided a model of belief-based goal processing, in the sense that there is a relationship between beliefs and goals, in the following way: a belief either supports a specific goal (or a set of goals) or serves as a form of assessment for evaluating a goal as being feasible or not. This theory supports our research by means of a framework that explores the correlation between Beliefs and Goals.

The Goal is a very important concept of BPM, usually defined in both activity-level and process-level (since a process is indeed a complex action). Moreover, the mechanism leading from desires to intentions has a key relevance within a KiP scenario, as decision-making is critical to the process, especially at the instance-level of a KiA selection process, where the "next step" of the activity flow should be chosen. In this sense, the activities executed by an agent are directed towards the achievement of some specific state of the world, and the anticipatory representation of such state represents his goal (or the driving force behind his actions). Even goals that are already achieved in the world can still be pursuable, in the form of "maintenance goals" (i.e., goals about keeping things). Maintenance goals are different from "achievement goals" (those referring to a state of things that is not yet achieved). These definitions were already present in the BDI seminal work of Bratman [1987].

The difference between Bratman's BDI model and Castelfranchi & Paglieri's model is that, in the latter, a goal is not a representation currently and necessarily orienting and guiding an action; instead, it is a representation endowed with this potential function, so that it is somehow "destined" to play this role — but whether or not this role is actually fulfilled depends

on the agent's beliefs. This definition shares many characteristics from notion of "desires" from BDI, but that are two main differences: (i) Desires have a notion of "expected pleasure" ingrained in their definition, so they cannot be induced by external pressure or impositions. Goals, on the contrary, may originate from a duty or an order, possibly disliked by the cognitive agent [Castelfranchi, 1998]. Therefore, not all intentions originate from some endogenous desire of an agent but, arguably, from other sources and (ii) Desires have a strong connotation as non-pursued or even 'non-pursuable' states of affairs. This is one of the reasons in BDI for not using the term in more advanced stages of practical reasoning and deliberation, introducing a new theoretical entity, i.e. intention, as a new primitive. Castelfranchi & Paglieri's proposal enables the technical notion of goal to be applicable both before, without, and after the decision to act, as well as during the persecution of one's aim and the performance of an action. Thus, the notion of goal emphasizes the continuity between (what in BDI are called) desires and intentions, opening the way for an operational model of intention formation. A viewpoint that stands in accordance with the notion of a business process, as there is always goal from the beginning to the end, during the process that leads from a desire to an intention, the same goal is thus transformed in its functional properties by subsequent accretions of relevant belief patterns.

One of the most interesting consequences about the considerations proposed by Castelfranchi & Paglieri [2007] is that as, goals and beliefs are taken as the primitives of a cognitive action, there is a refinement of the process of "intention formation" (the process where the pondering of desires come to actionable intentions) towards a process of goal-processing (i.e., the process that brings the agent from a general interest for a potentially relevant outcome to the subjective commitment to bring it about through adequate planning and action). In this sense, Desires are defined as "Goals still being pondered and not committed to act upon", while Intentions are defined as "Goals that are committed to be acted upon and fulfilled by a Cognitive Agent".

Based on the definitions of Desires and Intention as Goals (admitting Castelfranchi's paradigm), we may know depict the cognitive mechanism behind a rational Agent's action by the usage of the Postulate of Cognitive Regulation of Actions [Castelfranchi, 1996], stated as follows:

Postulate of Cognitive Regulation of Action:
Each goal of a cognitive agent is necessarily supported and justified by this agent's beliefs (i.e., reasons). Cognitive agents cannot activate, maintain, decide about, prefer, plan for or pursue any goal which is not grounded (implicitly or explicitly) on pertinent beliefs.

A goal is sustained (both in its current status and in the continuation of its processing) by a rich structure of beliefs, and these beliefs correspond to (and keep track of) the critical conditions that the goal has already successfully satisfied. This view has two important corollaries, mainly regarding the relationship of specific beliefs towards specific goals and the dependency of the goals on their supporting beliefs:

Corollary 1 (Specificity):
At each stage of its processing, a goal is filtered or supported by specific beliefs, which determine the properties acquired by the goal in the next stage (e.g. from desires to intentions).

Corollary 2 (Dependency):
The destiny of a goal, after its processing has been compromised, strictly depends on the reasons that caused such a failure (i.e. the specific supporting beliefs that were invalidated).

Summing it up, Corollary 1 points out that for a goal to be chosen for pursuit (i.e. become an intention), it must be supported by specific beliefs that enable its selection. In a similar sense, Corollary 2 specifies that if a goal remains unpursued (i.e. still a desire and not an intention), there must have been beliefs, either unsupporting this goal and/or beliefs that previously supported the goal that were invalidated by specific state of affairs.

Due to the important nature of Beliefs during the selection of goals, we must analyze its dynamics in the course of not solely taking into consideration a single cognitive agent but a group of agents collaborating and interacting during the performance of actions, as the next subsection will describe, using a concept called Common Ground Context.

4.8) The theory of Common Ground Context

A knowledge-intensive process is directly impacted by the knowledge exchange that occurs during the interactions among participants, in the form of conversations using several platforms, such as social networks. The conversation among participants during the execution of a KiA (which may be represented in the form of an exchange of speech acts expressed by the speakers that are involved in the conversation) modifies the intermediate and final results of the tasks performed, because the knowledge exchanged brings about situations that are perceived by the participants. Due to the complexity of the topic, a conversation can be defined as an ordered sequence of speech acts, between different speakers and hearers. Each illocutionary act in the sequence creates a limited set of possible replies, in the sense of limited sets of possible speech act to be performed as the next step of the ordered sequence of speech acts at the conversation.

Based on the definition of a conversation as a sequence of speech acts, the dynamics of the perceptions of participants during an interaction was already described by Stalnaker [2002]. Based on the analysis of the interactions, Stalnaker also defined the important concept of Common Ground Context (CG) [Stalnaker, 2014]: "Common Ground Context is composed of common or mutual beliefs plus what a speaker presupposes, i.e., what she believes to be common or mutual belief for all participants during a conversation". From this point on, we will reference this concept as just "Common Ground", as its usually referred at philosophical literature.

The definition of Common Ground includes the notion of pragmatic presupposition, which is the knowledge that is implied during an interaction and it includes at its most basic form the preconditions for linguistic interaction (for example, the mutual public knowledge that we are speaking the same language), the norms of turn-taking in dialogues, and more particularized information about conversational plans and goals [Grice, 1975]. It can also be changed or modified by two kinds of events that can change a conversation's Common Ground: (i) speech acts and (ii) manifest events, the latter being defined as "an event that, when it occurs, that is mutually recognized to have occurred".

From Stalnaker's definition of a Common Ground (described in details at Section 4.2), we adopt a version of the traditional philosophical view of Common ground defined by Harvey [2014] as both "(i) a philosophical construct whose contents are presuppositions adopted by a speaker and actualized by displaying them to an audience. The display is an action intended to communicate something at least in part because it is recognized as so intended, and on the basis

of such a display, each interlocutor is able to align their expectations and assumptions about the topics and content of the conversation and (ii) on the level of representational psychology, common ground is a broader set of entities - not just utterances - that direct joint activities in two ways. This set enables two people to intend to do something, letting interlocutors line up the timing of their respective parts of joint activities such as uttering and perceiving, meaning and understanding, and so on.".

Therefore, the Common Ground of an interaction is not a static construct but a dynamics set of beliefs that change as the speech acts are performed and event perceived by all interaction's participants (manifest events) occur.

The dynamics of the Common Ground changing is called accommodation. According to Harvey [2014], Kai Von Fintel provides the clearest description of this kind of two-step operation:

"An utterance will affect the common ground in two steps: (i) first, the fact that the utterance was made becomes common ground (and the participants may immediately draw inferences based on that fact, and perhaps adjust the common ground accordingly), (ii) then, assuming that the proper (implicit) negotiation has occurred, the asserted proposition is added to the common ground." [von Fintel, 2008 apud Harvey, 2014].

At the context of a knowledge-intensive process, the common ground enables the description of an interaction not solely as knowledge (in the form of propositional content) being exchanged, but also as a representation of the shared viewpoints between the participants, that will influence (as described in the previous subsection) the selection and/or rejection of shared goals during a process execution. This specific consequence of the application of Castelfranchi & Paglieri's model and the notion of Common Ground enables the tackling of a series of issues that form the main hindrances of KiP analysis and modeling. At a KiP scenario, the Common Ground can be used to describe the set of common knowledge presupposed by all participants during the execution of a KiA (as well as during the decision-making task of "which activity will be the next step at the process flow"), forming the rationale of the actions involved.

With the framework of goal-processing as a refinement of the BDI model, together with the theories of Intentionality, Speech Acts and Common Ground, we can now propose a cognitive theory for Knowledge-intensive Processes in the next chapter.

Chapter 5 - A Cognitive BPM Theory for Knowledge-intensive Processes

Our research aims to explore the paradigm called "Cognitive BPM" [Hull & Motahari-Nezhad, 2016], which essentially stands for the application of Cognitive computing technologies to the contexts and aspects of the BPM ecosystem. In particular, our goal is to develop a theory to provide the foundation of a Cognitive Paradigm for Knowledge-intensive Processes. In this direction, we argue that a theory of Intentionality and Speech Acts – allied with the theory of Common Ground – provides a complementary, adequate and precise conceptual basis for understanding, representing and analyzing the dynamic nature of Interactions and their effects within a KiP.

In the present research, we consider these two theories so as to propose a well-founded ontology, named CognitiveKiP, which depicts the involved concepts in a clear and semantically precise way, thus avoiding conceptual ambiguity and logical inconsistences. Thus, our proposal consists of a theory about Knowledge-intensive Processes that is represented and structured as an ontology, providing the necessary conceptualization for the Cognitive BPM effort. The concepts of CognitiveKiP is well-founded in the constructs of the Unified Foundational Ontology (UFO), particularly in its social perspective (UFO-C) and on the constructs related to Services (UFO-S).

5.1) Basic Concepts from the Unified Foundational Ontology (UFO)

Our proposal begins with the discussion of an excerpt of the UFO top-level ontology, whose constructs are central for the creation of a Cognitive BPM paradigm. Two concepts are deemed important for this discussion: the Participants involved in a KiP and what are the different Tasks being performed (and events that can happen during their execution) by the participants. Three notions from UFO-A [Guizzardi, 2005] are required: the notion of Particulars (or Individuals) as "entities that exist in reality possessing a unique identity", in opposition to Universals or Types, that are "pattern of features, which can be realized in a number of different Particulars", the notion of Substance as "existentially independent particulars" (e.g. an specific

person, an specific pencil or a specific dog) and the definition of a Moment as “an individual that can only exist in other individuals” (e.g. a color or a connection).

Moments are existentially dependent on other individuals and the existential dependence is the difference between the Intrinsic Moments (dependent of a single individual) and relators (dependent on a plurality of individuals). An example of the former would be a body temperature of a specific person and of the latter, a marriage. A specific kind of Intrinsic Moment is described in UFO-C [Guizzardi et al., 2008] as an Intentional Moment that represent the “capacity of some properties of certain individuals to refer to a possible situation of reality”. An Intentional Moment has a type, representing the class of situations referred by a specific Intentional Moment (ex: Belief, Desire, Intention, etc.) and a propositional content.

Moreover, an important type of Intentional Moment for our discussion is the Intention, “a specific type of Intentional State that represents a self-commitment towards the fulfillment of a Goal by performing an Action”. A Goal is the propositional content of the Intention. Therefore, to be able to perform an Action, one must be able to have an Intention, i.e., to be agentive.

With the previous definitions in mind, UFO distinguishes between Agents and Objects. While an Agent is defined as an “Agentive substantial particular who inheres Intentional Moments and is able to perform actions or contribute to actions intentionally”, while an Object is “A non-agentive substantial particular that is unable to inhere Intentional Moments and, therefore, is unable to perform actions or contribute to actions intentionally”.

An Agent is further categorized as a Physical Agent (e.g., a person) or a Social Agent (e.g., an organization, a group of people, etc.). In this same way, an Object is either a Physical Object (e.g., a sheet of paper) or a Social Object (e.g., an insurance contract, money, language). Figure 6 illustrates UFO concepts involved in the definitions of Agent and Object.

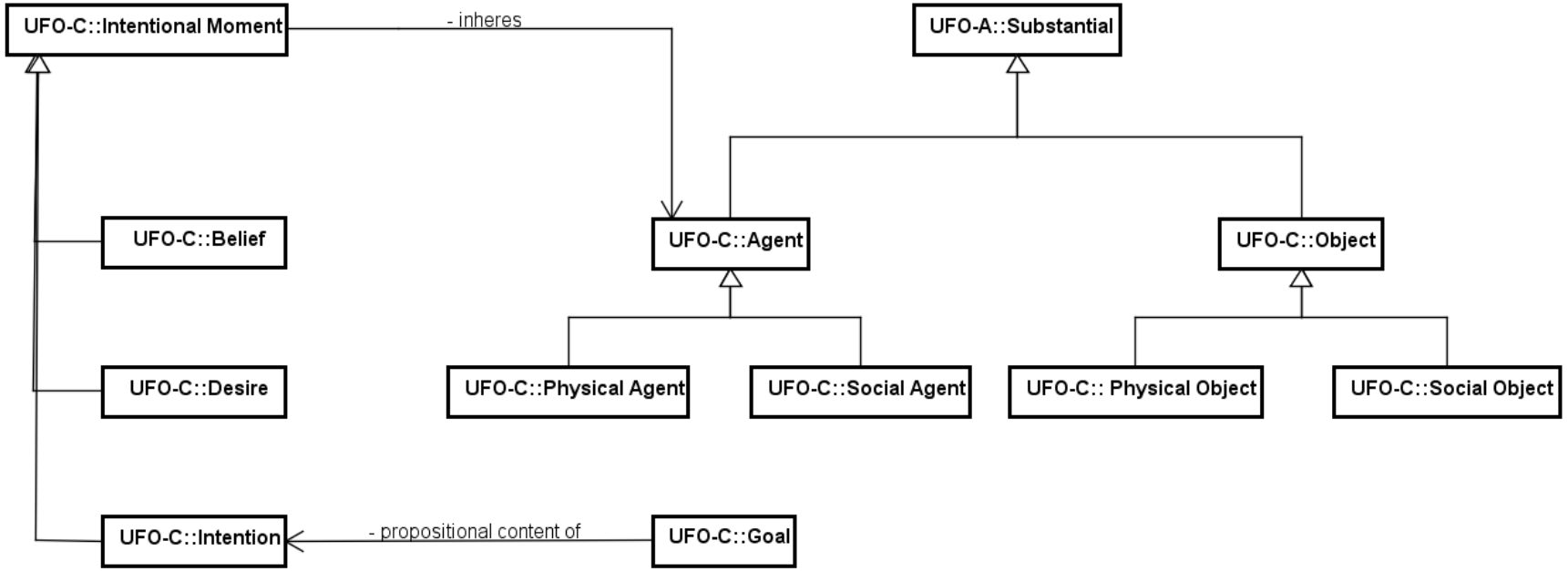

Figure 6 –Definitions from UFO regarding Agents and Objects [Guizzardi et al., 2008]

After presenting well-founded definitions for the concepts of Agents and Objects within KiPs, we move to address the different tasks involving the Agents and Objects within a KiP. UFO-C defines an Action as “an intentional event that instantiate an Action Universal with the purpose of satisfying the propositional content of an Intention”. Expanding our discussion towards the definition of an Event (also called perdurants) in UFO-B [Guizzardi et al., 2013], they are defined as "individuals composed of temporal parts. They happen in time in the sense that they extend in time accumulating temporal parts. Events can be atomic or complex, depending on their mereological structure. Whilst atomic events have no proper parts, complex events are aggregations of at least two disjoint events”. A Complex Event is existentially dependent on all its proper (composing) parts (other Events) and, indirectly, on the objects these proper parts depend on. Thus, an Action, being a specialization of Event, also categorized as Atomic and Complex in the same way as Events. Complex Actions also have the same existential dependency on other Events (including Actions) that compose it, as well as the object they refer to.

An interesting concept from UFO-B related to Events, which can be useful for the characterization of our proposal, is the Participation, defined as “the portion of an event which depends exclusively on a single object”. A Participation is also an Event, and thus can compose Complex Events and Actions as well.

For the specific case of Participations that compose a Complex Action, two types are defined at UFO-B, depending on the Substantial the Event refers to: An Action Contribution (if it refers to an Agent) and a Resource Participation (if it refers to an Object, which is deemed a Resource if it participates in an Action).

The last concept that is crucial for our proposal is the Situation, defined in UFO-B as “a state of the world bound to a specific time point”. It has a close relationship to Events due to the definition of Events as transformations from a portion of reality to another, the “portions of reality” being defined as Situations. Thus, a Situation may trigger an Event *e* (i.e. the state of the world at a specific time point satisfies all the sufficient and necessary conditions for the manifestation of e); also, an Event *e* brings about a Situation *s*, which is the state of the world representative of the results of *e*. Actions, as intentional Events, also may be triggered by (and brings about) specific Situations.

Figure 7 depicts the concepts from UFO-B involved in the definition of the tasks performed during a process.

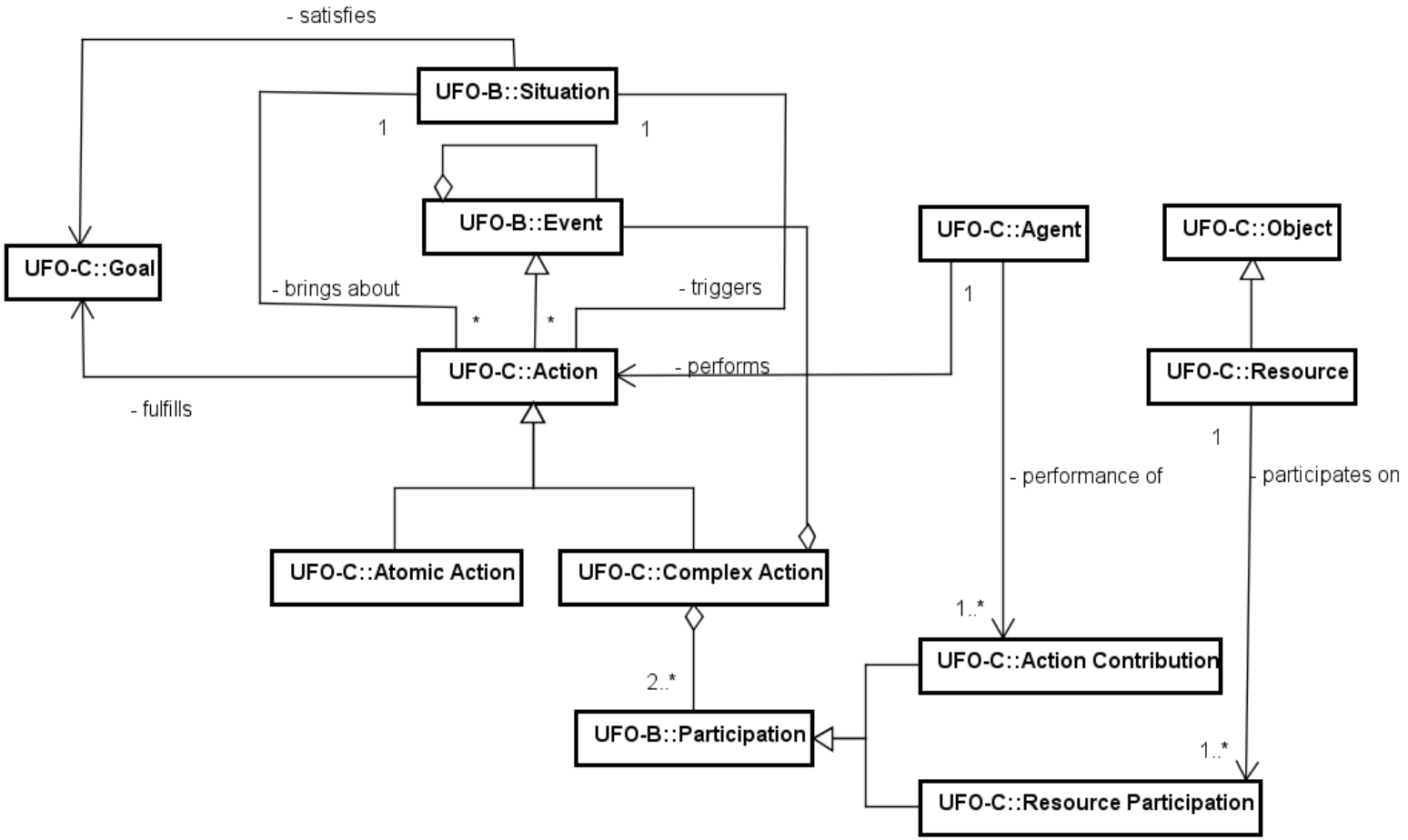

Figure 7 - Definitions from UFO regarding Actions and Events [Guizzardi et al., 2008]

5.2) On Agents, Resources and Business Processes

Based on the discussion at the previous Section, a conceptualization for CognitiveKiP concepts was proposed, being well-founded in UFO with definitions of new concepts and relationships as well as new axioms described at First-Order Logic (FOL). The initial model was proposed and evaluated using Alloy Analyzer [Jackson, 2006]. The initial version of the ontology was translated into Alloy (a logic language based on set theory) and the Alloy Analyzer generated possible instances for the given specification as well as checking the consistency of manually designed assertions. In this process, a visual representation of the possible instances was contemplated and successive rectifications on both the ontology's concepts and axioms were made until a final model was considered satisfactory.

At this section, CognitiveKiP concepts are described using a simple notation as follows: a text box for each concept, with its UFO's stereotype between parenthesis and the full text of the definition below. Also, the axioms involved will be listed at each concept or set of concepts (if applicable). An example of the notation is below:

<Concept> (<UFO stereotype>): <Full definition of the concept>

<Axiom X: Text description>
<AX: Formal definition>

Two central concepts of our proposed theory are the process participant and its inhered intentional states. We define a Participant as an individual (or group of individuals) who inheres intentional states (such as Beliefs, Desires and Intentions) and thus is able to perform actions or contribute to actions intentionally.

We argue that there is a semantic compatibility between the concept of Intentional Moments from UFO-C and Searle's definition of Intentional States. We argue that the concept of an Intentional State in CognitiveKiP is well-founded on the construct of Intentional Moment from UFO. Thus, Intentional States are intrinsic to each individual Agent, that is, two distinct

agents (namely John and Mary, for example) who both want to "study hard for tomorrow's exam" actually inhere distinct desires with equivalent propositional contents, as defined below:

Definition D1: Intentional State (UFO-C::Intentional Moment):
The mental capacity inhered in one Agent to refer to possible situations of reality. Intentional states are further specialized into Belief, Desire, Intention and Feeling. An Intentional State has a propositional content.

Axiom A1: "An Intentional State inheres in one and only one specific Agent"
(A1) $\forall is\ (Intentional\_State(is) \rightarrow \exists! ag\ (Agent(ag) \wedge (inheres\_in(is, ag)))$

Based on the definition of an Intentional State, we can define the specific Intentional States involved at our proposal, namely: Belief, Desire and Intention.

Definition D2: Belief (UFO-C::Intentional Moment):
A specialization of Intentional State, being a mental representation that is used as a plausible substitute for a certain aspect of reality, and that is supposed to be referentially true, i.e. to provide a description that is assumed to correspond, and used as corresponding, to how things actually are.

Definition D3: Desire (UFO-C::Intentional Moment):
A specialization of Intentional State, being a mental representation of a state of affairs that would be desirable to bring to reality, but one is not necessarily committed to.

Definition D4: Intention (UFO-C::Intentional Moment):
A specialization of Intentional State, representative of a self-commitment of a Participant towards performing a course of action, in order to bring a state of affairs to reality.

Definition D5: Feeling (UFO-C::Intentional Moment):
A specialization of Intentional State, representative of a sentiment or affect regarding an object.

We also specialize the definition of Participant into either an Individual Participant (a specialization of UFO-C Physical Agent, representative of a person or a system capable of performing actions) or a Collective Participant (a specialization of UFO-C Social Agent representative of an organization, a group of people). Both subkinds of Participants have intrinsic Intentional States and are capable of performing tasks within a Process.

Definition D6: Participant (UFO-C::Agent):
A Participant of the process, contributing to the process towards the fulfillment of goals.

A2: "A Participant performs an Activity or participates into a Communicative Interaction, performing a SpeechAct"
(A2) ∀pt Participant(pt) → (∃act (Activity(act) ∧ performs(pt, act)) ∨ (∃ci,sp (Communicative_Interaction(ic) ∧ SpeechAct(sp) ∧ performs (pt, sp) ∧ participates_into(pt,ci)))

Definition D7: Individual Participant (UFO-C::Agent):
A specialization of Participant, representing an individual that participates in the process (a person or a system).

Definition D8: Collective Participant (UFO-C::Agent):
A specialization of Participant, representing a collective the participates in the process (groups of people, organizations, etc.).

Besides, a Goal is defined as a proposition that represents the propositional content of an Intentional State (in the logical sense) inhered in a process Participant. Particularly, an Intention is a subtype of Intentional State that represents the commitment of a Participant to act towards the fulfillment of the specific Activity Goal.

The second central concept at the BPM scenario is the Activity, i.e. the task performed by the Agent during the process execution, which is essentially an Action from UFO-B. Considering an Activity as an Action, we argue that the change of state of affairs is precisely the description of the pursuit of the Activity's Goal (the state of the world that the process execution aims to bring about). More importantly, an Activity always involves participants,

performing actions by the way of their own intentions, being a clear distinction from an intentional event (involving agentive function) and an unintentional event (not involving agentive function, such as the consumption of resources during an activity or the effect of natural causes).

An Activity can then be defined as a specialization of UFO-C Action, either Atomic or Complex, the latter being composed of two or more Participations, being at least two of them Activities as well, performed towards the fulfillment of a specific goal.

Definition D9: Unintentional Event (UFO-B::Event):
An Event that occurs without the involvement of the agentive function of a Participant. It can be optionally composed by other Events and other forms of Participations.

Definition D10: Activity (UFO-C::Action):
An intentional Event (i.e. involving agentive function of a Participant) that has the purpose of satisfying the propositional content of a Goal. It can be optionally composed by other Activities, Resource Participations and other forms of Participations and Events.

The definition of Activity brings us to the discussion of a goal within a process. Based on our previous definition of goal, we adopt and specialize the UFO-C definition of Goal as "a proposition representing a state of the world that satisfies the propositional content of an Intention in the logical sense" and bring it closer to a cognitive perspective based on Castelfranchi and Paglieri's work [2007], defining a broader definition: "A goal is defined as an anticipatory internal representation of a state of the world that has the potential for and the function of (eventually) constraining/governing the behavior of an agent towards its realization" and two specific subkinds of Goals, representative of a simplified approach of the authors' approach. First, the Conditioned Goal, is dependent on the fulfillment of the goals of its composing Activities for its own fulfillment. Second, the Executive Goal, is related to a specific Activity and is unconditioned to any other goal. This relation is derived from the existential relation between an event *e1* and its composing event *e2*, as described in the previous subsection.

Definition D11: Goal (UFO-C::Goal):
An anticipatory internal representation of a state of the world that has the potential for and the function of (eventually) constraining/governing the behavior of an agent towards its realization.

Definition D12: Conditioned Goal (UFO-C::Goal):
An specialization of Goal, representative of the state of the world that an Activity aims to bring to reality. It is dependent on the fulfillment of other Goals for its own fulfillment.

Definition D13: Executive Goal (UFO-C::Goal):
A specialization of Goal, representative of the state of the world that an Activity aims to bring to reality. It is independent on the fulfillment of other Goals for its own fulfillment.

Both specializations of Goal, Conditioned Goal and Executive Goal have a relationship with the Intention of a Participant executing an Activity towards the specific Goal's fulfillment. In this sense, the Goal represents the propositional content (in a logical sense) of the Intention inhered by the Participant performing the Activity.

Another important element of a Process is the Resource. The proposal defines a Resource as a UFO-C Object which unintentionally participates into an Activity, either by its use, modification, termination and creation during the Activity's execution. Being an Object, it does not have Mental Moments; thus, a Resource may only participate into an Activity through a Resource Participation.

Definition D14: Resource (UFO-C::Object):
An Object that participates as a resource into an Activity.

Definition D15: Resource Participation (UFO-C::Participation):
An unintentional event triggered by the intended (by the Agent performing the Activity) use, modification, termination or creation of a Resource in that Activity.

A Process (or, more generally, any Activity) performs changes to the states of the world where they occur. These states are composed by different elements, related to the scope of the

specific Process or Activity being performed, or even unintentional Events altering the state of affairs at a given time and, thus, influencing the process or Activity. In order to contemplate this aspect of a Process (or Activity), we define two elements, both as a UFO-C::Situation, depicting the different states of affairs regarding an Activity: (i) The first is the Pre-state, the state of the world, at a given time, that satisfies all the sufficient and necessary conditions for the execution of an Activity or the manifestation of an unintentional Event; (ii) the second is the Post-state, depicting the state of the world, at a given time, that are the results brought about by the execution of the Activity or the manifestation of an unintentional Event into reality.

Definition D16: Pre-state (UFO-C::Situation):
The state of the world, at a given time, that satisfies all the sufficient and necessary conditions for the manifestation of an unintentional Event or of an Activity.

Definition D17: Post-state (UFO-C::Situation):
The state of the world, at a given time, representative of the results brought about by an unintentional Event, by a Task or by an Activity.

Finally, the BPM central concept of a "Process" is denoted in CognitiveKiP as an Activity (keeping in mind that an Activity is composed by Complex Actions, Actions, Resource Participations and Events). Thus, the Conditioned Goal of an Activity is semantically equivalent of a "Process Goal" in BPM. We argue that representing a process as an Activity (stereotyped as an Action from UFO-C), with an arbitrary defined scope, captures the semantic essence of this concept and eliminates the construct overload that exists in most business process representation languages, which provides distinct constructs and terms (task, activity, process, macroprocess) for essentially the same real-world concept of an intentional event. Figure 8 depicts the concepts defined before.

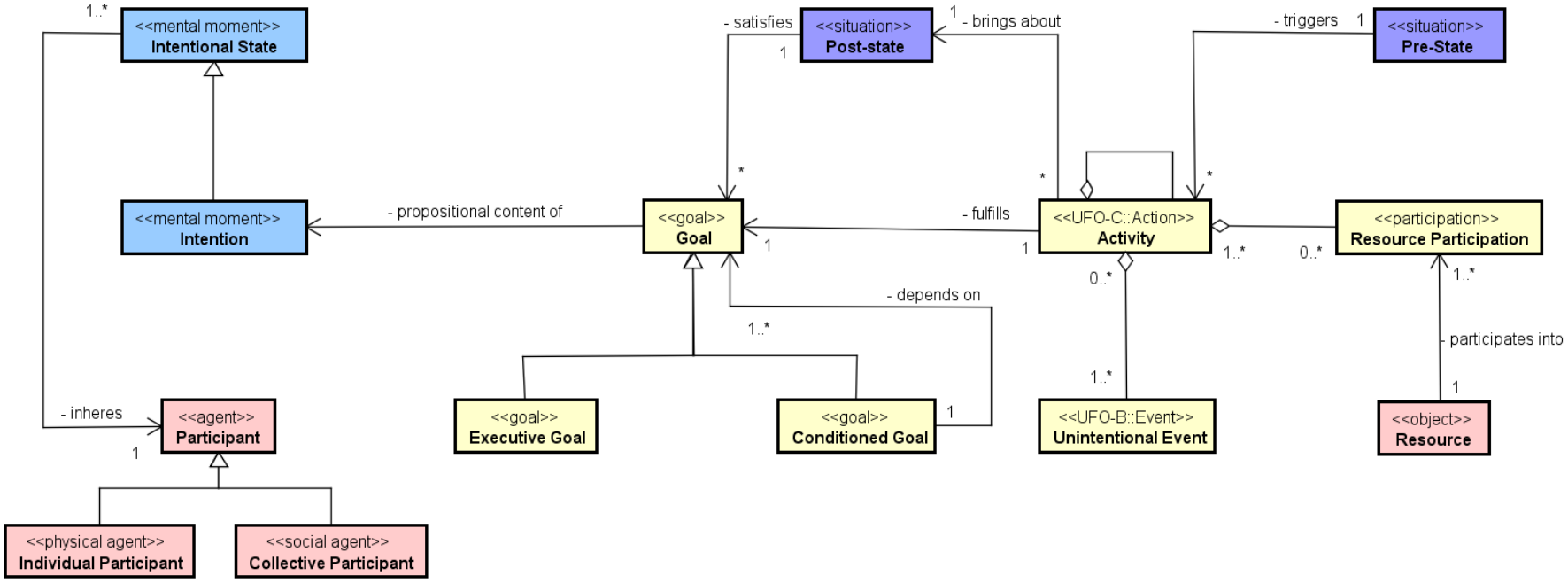

Figure 8 – Basic BPM concepts of the CognitiveKiP Ontology

5.3) On Knowledge-intensive Processes

The framework of BPM concepts of the previous subsection is a representation of the basic elements of a typical scenario of a process instance being executed. However, a Knowledge-intensive Process is a specific subtype of a process and additional concepts must be defined in order to describe its unique characteristics.

Vaculin et al. [2011] describe a KiP as a "process whose conduction and execution are heavily dependent on knowledge workers performing various interconnected knowledge-intensive, decision-making tasks". Two aspects can be perceived from this brief definition. The first is the concept of knowledge-intensity and the other is the decision-making aspect, both inherent to a KiP. Following the idea of a process as a high-level Activity (comprising all the Activities, Tasks, Resource Participations and Events of a process instance), we argue that a Knowledge-intensive Process is a high-level Activity composed by at least one Activity (deemed as a Knowledge-intensive Activity) that (i) has a Communicative Interaction occurring between Participants and (ii) is composed by one or more decision-making activities.

Definition D18: Knowledge-intensive Activity (UFO-C::Complex Action):
An Activity that is necessarily composed by at least one Decision and has a Communicative Interaction occurring during its execution.

In order to precisely define interactions and decisions, we take concepts from the Knowledge-intensive Process Ontology (KiPO) [França et al., 2010] as a baseline, as follows.

A Decision in KiPO is defined as a specific subtype of Knowledge-intensive Activity that is triggered by an unexpected Event, deemed as a Question. For example, during the Knowledge-intensive Process of defining the milestones for next year's strategic planning, a sudden global economic crisis would be the unexpected Event, that represents the Question to be addressed: "Should this year's expected ROI be kept the same as in last year's strategic plan?". Being a subtype of a Knowledge-intensive Activity, a Decision is essentially an Activity (UFO-C::Complex Action) that may be composed by Resource Participations, Activities and other Events (even though it was originally defined in KiPO as an Atomic Action). A Question

is an Event that brings about the Pre-State required by the Decision. The executing Agent of a Decision is the one who intentionally commits to solve a specific Question that occurred during the Knowledge-intensive Activity's execution. Being an Activity itself, solving the Question fulfills the Decision's (Conditioned) Goal and is the propositional content of the Intention of the Impact Agent that performs the Decision.

Definition D19: Question (UFO-B::Situation):
A state-of-affairs (Situation) that raises a question to be answered or problem to be solved, that triggers a Decision.

Definition D20: Decision (UFO-C::Complex Action):
A subtype of a Knowledge-intensive Activity that fulfills the Conditioned Goal of solving a Question, and that is intentionally executed by an Impact Agent.

With regard to interactions, KiPO describes the interaction among process participants as a Communicative Interaction (a UFO-C Complex Action) in which Agents communicate with each other through speech acts [Searle & Vanderveken, 1985], exchanging information and expressing wishes, emotions and ideas. The Speech Act concept at CognitiveKiP will be described at Section 5.6 in detail.

Definition D21: Communicative Interaction (UFO-C::Complex Action)
A complex action that occurs during a Knowledge-intensive Activity, involving the participation of Agents who perform speech acts which compose the communicative interaction during the execution of the activity.

A3: "A Communicative Interaction occurs at one and only one Knowledge-intensive Activity."
(A3) ∀ci (Communicative_Interaction(ci)→ ∃!kia(Knowledge_Intensive_Activity(kia) ∧ occurs_at(ci, kia)))

With regard to Participants, KiPO originally defined three subkinds of process participants: (i) an Impact Agent is a Participant that is responsible for executing a KiA and for identifying Questions during its execution; (ii) an Innovation Agent is a Participant that has a

specialty in some knowledge area related to the KiP domain and that contributes to the execution of a KiA with innovation and creativity (although KiPO did not define neither the nature of this contribution, nor the precise semantics of innovation or creativity); and (iii) an External Agent is a Participant that is outside the scope of the process, participating during Communicative Interactions occurring at a Decision and contributing with knowledge when discussing alter-natives.

In CognitiveKiP, we define both Innovation and Impact agents more precisely. An Innovation Agent is a Participant that contributes to a KiA with innovation and creativity, by performing one or more Speech Acts which compose the Communicative Interaction that occurs during the execution of this KiA. An Impact Agent is a Participant that intentionally executes a KiA and participates into the Communicative Interaction that occurs during the KiA he executes. Finally, there is the External Agent, a special case of Innovation Agent that is external to the Process itself, solely participating only at one or more Communicative Interactions occurring at Decision. Figure 9 depicts the concepts and relationships of the Knowledge-intensive Process elements.

Definition D22: Innovation (UFO-C::Action Contribution):
An specific contribution to a Knowledge-intensive Activity, representative of the novelty or innovation performed by an Innovation Agent to a Knowledge-intensive Activity.

Definition D23: Innovation Agent (UFO-C::Agent):
A specialization of Participant that contributes to a KiA by performing one or more Innovations, being action contributions that compose a KiA.

A4: "For all Innovation Agent, there is one or more Innovation performed by it that compose a Knowledge-intensive Activity"
(A4) $\forall ia$ Innovation_Agent(ia) $\rightarrow \exists in,kia$ (Innovation(in) $\wedge$ performs (ia, in) $\wedge$ composes(in,kia))

Definition D24: Impact Agent (UFO-C::Agent):
A specialization of Participant that intentionally executes a KiA and participates into the Communicative Interaction that occurs during the KiA he executes.

A5: “For all Impact Agents, there is one or more Knowledge-intensive Activities performed by it.”

(A5) ∀ia Impact_Agent(ia) → ∃kia (Knowledge_Intensive_Activity(kia) ∧ performs (ia, kia))

Definition D25: External Agent (UFO-C::Agent):
A specialization of Participant that contributes to a KiA solely by performing one or more Speech Acts which compose the Communicative Interaction that occurs during the execution of a Decision.

A6: “For all External Agents, there is one or more Decisions with Communicative Interactions that they participate into and perform Speech Acts composing them.”

(A6) ∀ia External_Agent(xa) → ∃des,ci,sp (Decision(des) ∧ Communicative_Interaction(ci) ∧ Speech_Act(sa) ∧ performs (xa, sa) ∧ occurs_at (ci, des) ∧ composes (sa, ci))

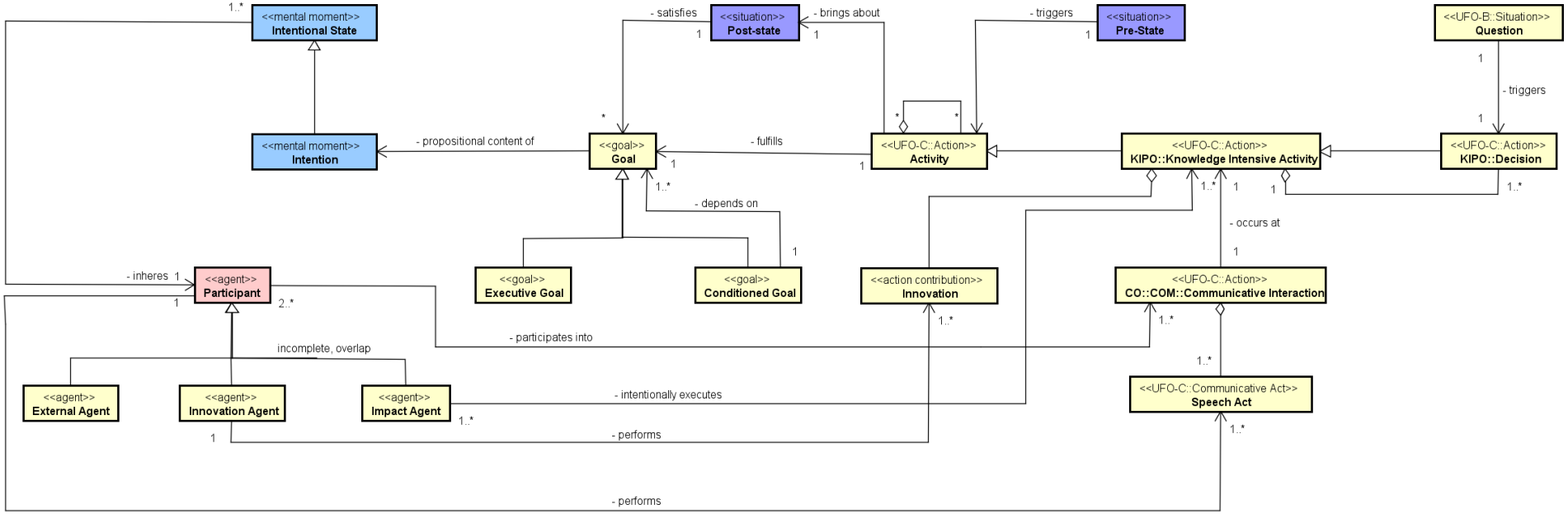

Figure 9 – Basic concepts for the depiction of Knowledge-intensive Process at CognitiveKiP Ontology

5.4) On Communicative Interactions during a KiP

An important aspect of a Knowledge-intensive Process is the Communicative Interaction, being one of most critical forms of knowledge intensity represented by the exchange of ideas, viewpoints and opinions, as well as many other pieces of knowledge among the participants of the process.

The original KiPO ontology encompassed the Collaborative Ontology (CO) [Oliveira, 2009] as one of its components, in order to define the concepts characterizing how collaboration takes place among participants within a KiP. According to KiPO, Communicative Interactions "are composed by Communications and Perceptions. Within a Communication Interaction, Messages are exchanged by Agents playing the role of Senders and Receivers. An Agent (as a Sender) sends a Message to start a Communication, which is the propositional content of his Message. The message reaches the receiving Agent (as a Receiver), who develops a Perception about the Message content."

There are two problems with this definition. The first is related to the concept of a Communication, defined by KiPO as a UFO-C::Communicative Act which is the propositional content of a Message (a UFO-A::Proposition). UFO defines a Communicative Act as "a speech act such as inform, ask or promise" [Guizzardi et al. 2008], bringing it close to Searle's definition of an illocutionary act (as described in Section 3.2). An illocutionary act is composed by an illocutionary force (that describes its type) and a propositional content. In the case of an exchanged message between two participants, the message itself – such as the textual content of an e-mail – is the propositional content of the illocutionary act (and not the other way around, as originally defined in KiPO). The second problem regards the Perception concept in KiPO, defined as the propositional content of a Message, denoting the perception by a receiving Agent of the illocutionary act of a received Message. We argue that it is quite the contrary, that is, a Message constitutes part of the propositional content of a Perception.

Addressing those problems, we propose a new conceptual structure, centered at the Speech Act (as a substitute for the CO::COM::Communication element in KiPO), as defined in Figure 10.

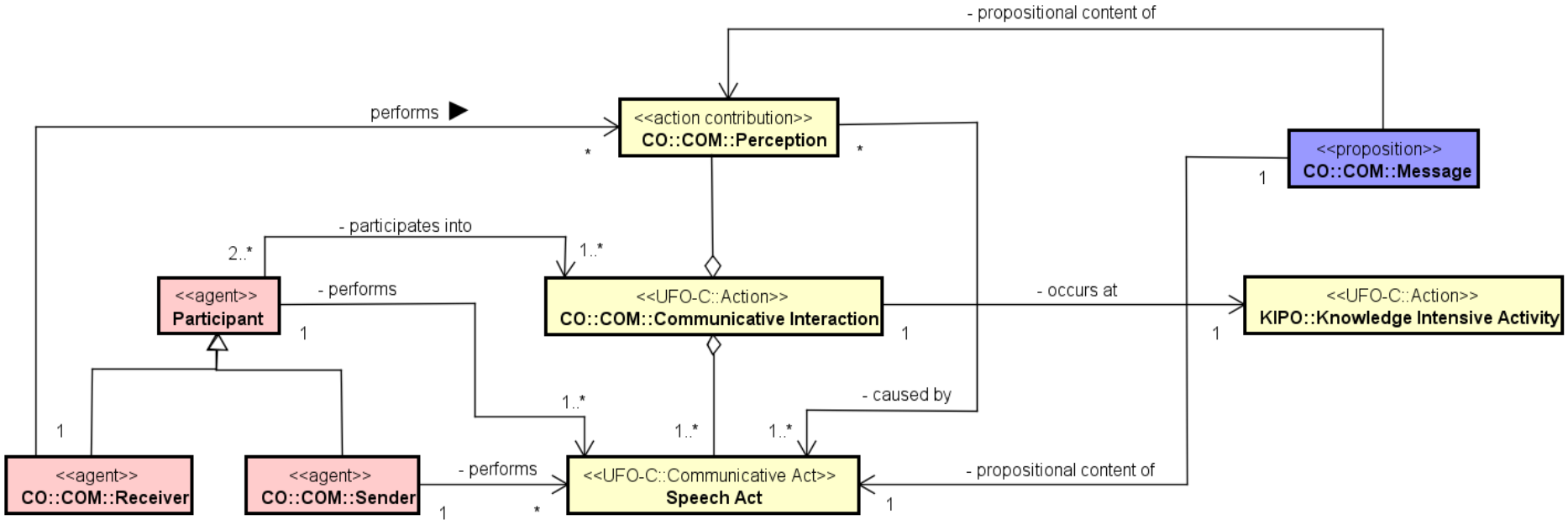

Figure 10 – CognitiveKiP Ontology elements regarding a Communicative Interaction

Another central concept in CognitiveKiP is the Speech Act, and how this novel pragmatic aspect impacts the conceptualization of interactions among participants.

A Speech Act expresses an Intentional State (as defined in Section 6.2) and constitutes the building block for all the mental cognitive concepts we adopt in CognitiveKiP. Intentionality is the aboutness or directedness or reference of mind (or states of mind) to things, objects, states of affairs, events that are involved in a KiP during a Communicative Interaction.

Beliefs, Desires, Intentions and Feelings are UFO-C::Intentional Moments and, as so, they are intrinsic and unique to each individual Agent they inhere in. However, they can be externalized in the form of different representations, representing the intentional content of its corresponding Intentional State. Each representation is essentially a UFO-C::Situation, since it represents the state of affairs that the directedness property of the Intentional States points towards. These are deemed as Externalized Intentional States, with specific types for each relevant Intentional State (Externalized Belief, Externalized Desire, Externalized Intention and Externalized Feeling).

Definition D26: Externalized Intentional State (UFO-C::Situation):
A state of affairs that represents the externalization of an Intentional State inhered in an Agent, depicting a representation of its intentional content.

A7: "An Externalized Intentional State is a representation of one and only one Intentional State"
(A7) ∀xs (Externalized_Intentional_State(is) → ∃!is (Intentional_State(is) ∧ is_representation_of(xs, is)))

Definition D27: Externalized Belief (UFO-C::Situation):
A specialization of an Externalized Intentional State, representing an externalized representation of an Agent's Belief, depicting a representation of its intentional content.

A8: "An Externalized Belief is a representation of one and only one Belief"
(A8) ∀xb (Externalized_Belief(xb) → ∃!bl (Belief(bl) ∧ is_representation_of(xb, bl)))

Definition D28: Externalized Desire (UFO-C::Situation):
A specialization of an Externalized Intentional State, representing an externalized representation of an Agent's Desire, depicting a representation of its intentional content.

A9: "An Externalized Desire is a representation of one and only one Desire"
(A9) ∀xd (Externalized_Desire(xd) → ∃!ds (Desire(ds) ∧ is_representation_of(xd, ds))

Definition D29: Externalized Intention (UFO-C::Situation):
A specialization of an Externalized Intentional State, representing an externalized representation of an Agent's Intention, depicting a representation of its intentional content.

A10: "An Externalized Intention is a representation of one and only one Intention"
(A10) ∀xi (Externalized_Intention(xi) → ∃!it (Intention(it) ∧ is_representation_of(xi, it))

Definition D30: Externalized Feeling (UFO-C::Situation):
A specialization of an Externalized Intentional State, representing an externalized representation of an Agent's Feeling, depicting a representation of its intentional content.

A11: "An Externalized Feeling is a representation of one and only one Feeling"
(A11) ∀fl (Externalized_Feeling(xf) → ∃!fl (Feeling(fl) ∧ is_representation_of(xf, fl))

We can finally define a Speech Act as an specialization of a Communicative Act that expresses an Externalized Intentional State:

Definition D31: Speech Act (UFO-C::Communicative Act):
An illocutionary act containing a propositional content and an illocutionary force, expressing an Externalized Intentional State.

A12: "A Speech Act is an expression of one and only one Externalized Intentional State"
(A12) ∀sa (Speech_Act(sa) → ∃!xs (Externalized_Intentional_State(xs) ∧ is_expression_of(sa, xs)))

A13: For each pair Participant, Speech Act: There is an Intentional State inherent to the Participant with a representation, that is the Externalized Intentional State, that the Speech Act performed by the Participant expresses.

(A13) ∀pt,sp Participant(pt), Speech_Act(sp) → ∃is,xs (Intentional_State(is) ∧ Externalized_Intentional_State(xs) ∧ expression_of(sp,xs) ∧ representation_of(xs,is) ∧ inheres(is,pt))

A14: For all Speech Act, there is a Participant and Communicative Interaction performed by a Participant that participates into a CommunicativeInteraction composed by the Speech Act

(A14) ∀sp(Speech_Act) → ∃pt,ci (Participant(pt) ∧ Communicative_Interaction(ci) ∧ participates_into(pt,ci) ∧ performs (pt, sp) ∧ composes(sp,ci))

Following Searle's taxonomy [Searle & Vanderveken, 1986], we have four types of speech acts to be respectively represented as expressions of each Externalized Belief, Desire, Intention and Feeling of the Participant that performs it: Assertive, Directive, Commissive and Expressive Speech Acts, as described in Figure 11.

Definition D32: Assertive Speech Act (UFO-C::Communicative Act):
A Speech Act that express an Externalized Belief.

A15: "For all Assertive Speech Act performed by a Participant, there is a Belief that is inherent to the Participant and an Externalized Belief that is a representation of the Belief and is expressed by the Speech Act"

(A15) ∀asa (Assertive_Speech_Act(asa) → ∃!bl,xb,pt (Participant(pt) ∧ Belief(bl) ∧ Externalized_Belief(xb) ∧ performs(pt,asa) ∧ inheres(bl,pt) ∧ is_representation_of (xb,bl) ∧ is_expression_of(sa, xb)))

Definition D33: Directive Speech Act (UFO-C::Communicative Act):
A Speech Act that express an Externalized Desire.

A16: “For all Directive Speech Act performed by a Participant, there is a Belief that is inherent to the Participant and an Externalized Desire that is a representation of the Desire and is expressed by the Speech Act”

(A16) ∀dsa (Directive_Speech_Act(asa) → ∃!bl,xd,pt (Participant(pt) ∧ Desire(ds) ∧ Externalized_Desire(xd) ∧ performs(pt,dsa) ∧ inheres(ds,pt) ∧ is_representation_of (xd,ds) ∧ is_expression_of(sa, xd)))

Definition D34: Commissive Speech Act (UFO-C::Communicative Act):
A Speech Act that express an Externalized Intention.

A17: “For all Commissive Speech Act performed by a Participant, there is an Intention that is inherent to the Participant and an Externalized Intention that is a representation of the Intention and is expressed by the Speech Act”

(A17) ∀csa (Commissive_Speech_Act(csa) → ∃!it,xi,pt (Participant(pt) ∧ Intention(it) ∧ Externalized_Intention(xi) ∧ performs(pt,csa) ∧ inheres(it,pt) ∧ is_representation_of (xi,it) ∧ is_expression_of(sa, xi)))

Definition D35: Expressive Speech Act (UFO-C::Communicative Act):
A Speech Act that express an Externalized Feeling.

A18: “For all Expressive Speech Act performed by a Participant, there is a Feeling that is inherent to the Participant and an Externalized Feeling that is a representation of the Feeling and is expressed by the Speech Act”

(A18) ∀esa (Expressive_Speech_Act(esa) → ∃!fl,xe,pt (Participant(pt) ∧ Feeling(fl) ∧ Externalized_Feeling(xf) ∧ performs(pt,esa) ∧ inheres(fl,pt) ∧ is_representation_of (xf,fl) ∧ is_expression_of(sa, xf)))

We leave the fifth Speech Act type, the Declarative Speech Act, to be described at Section 5.6, due to its complexity in changing the social sphere.

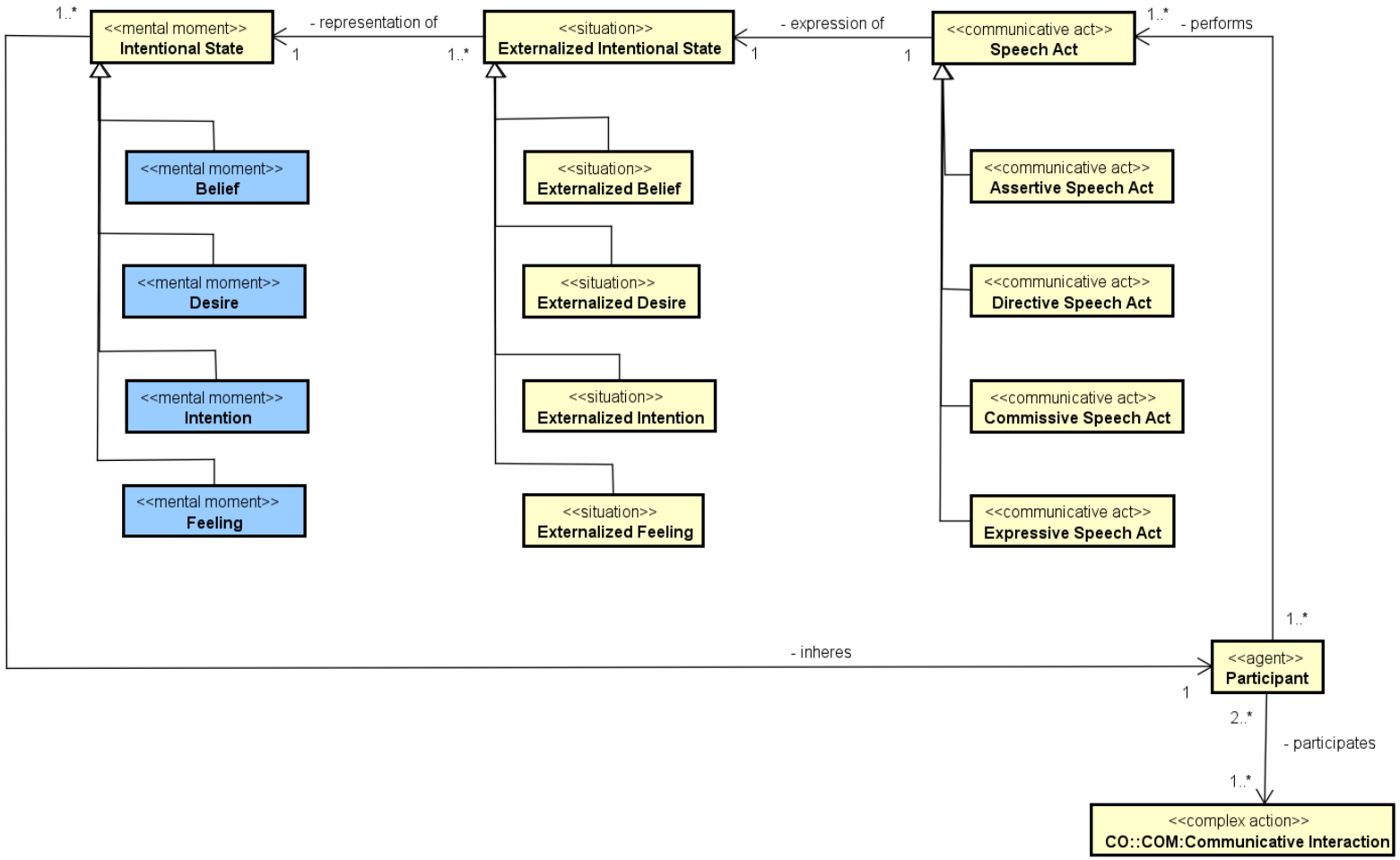

Figure 11 – Types of Speech Acts involved at a Communicative Interaction

5.5) On Common Ground during the Interactions at a KiP

An important part of a KiP is the exchange of knowledge during the Knowledge-intensive Activities, as not only a KiP is driven by knowledge, but interactions and collaboration play an important role on its flow. After we defined the Intentional States and how they are represented and expressed, in the form of Speech Acts and its different types, we must now define the dynamics of Common Ground reflecting how the interchange of representations of Intentional States between Agents and their reactions towards events affect the process.

Since each Belief is inherent to a specific Agent and Common Ground is the set of Beliefs presuposed to be shared by all Agents involved in a Communicative Interactions, we define a new concept, Shared Presupposition, as a specialization of the externalized representation of a Belief (Externalized Belief) as the basic building block for the Common Ground.

Definition D36: Shared Presupposition (UFO-C::Situation):
A specialization of Externalized Belief that is representative of the belief presupposed by all participants at a Communicative Interaction and created by the successive modifications of the Common Ground.

Each Shared Presupposition can occur either at the beginning of the Communicative Interaction, being the initial assumptions the Participants have already made previously and they can be modified during the interaction, at two specific conditions: (i) the performance of a speech act during the interaction or (ii) the occurrence of an event that is presupposed to be perceived by all participants, called a Manifest Event. We also define this important concept at CognitiveKiP.

Definition D37: Manifest Event (UFO-C::Event):
A specialization of Event that occurs during the Communicative Interaction and its occurrence is presupposed to be known by all Participants involved at the Interaction.

Based on the two new concepts, the Common Ground as a concept can also be defined, being described as a set of Shared Presuppositions that is modified during the Communicative Interaction.

Definition D38: Common Ground (UFO-C::Situation):
During an interaction, the Common Ground (CG) is defined as the set of Shared Presuppositions either (i) shared by all participants or (ii) presupposed to be shared by the interaction's participants. The CG can be modified by either speech acts or manifest events, i.e. events that are presupposed to be known by all participants.

A19: "The Common Ground is composed by the Shared Presuppositions brought about either by (shared by all Participants participating at a Communicative Interaction"
(A19) ∀cr (Common_Ground(cr) → ∃sp, ci, (Shared_Presupposition(sp) ∧ Communicative_Interaction(ci) ∧ composes(sp, cr) ∧ brings_about(ci,cr) ∧ ((∃sa (Speech Act(sa) ∧ composes(sa,ci) ∧ modifies(sa,cr) ∧ brings_about(sa,sp)) ∨ (∃me (Manifest_Event(me) ∧ modifies(me,cr) ∧ brings_about(me,sp))))

A20: "For all Shared Presupposition, composing a CommonGround brought About by a CommunicativeInteraction, is presupposed by the Participants that participate into the Communicative Interaction"
(A20) ∀sp(Shared_Presupposition) → ∃cg,ci,pt (Common_Ground(cg) ∧ Communicative_Interaction(ci) ∧ Participant(pt) ∧ participates_into(pt,ci) ∧ presupposed_by(sp,pt))

Figure 12 depicts the concepts involved in this part of CognitiveKiP.

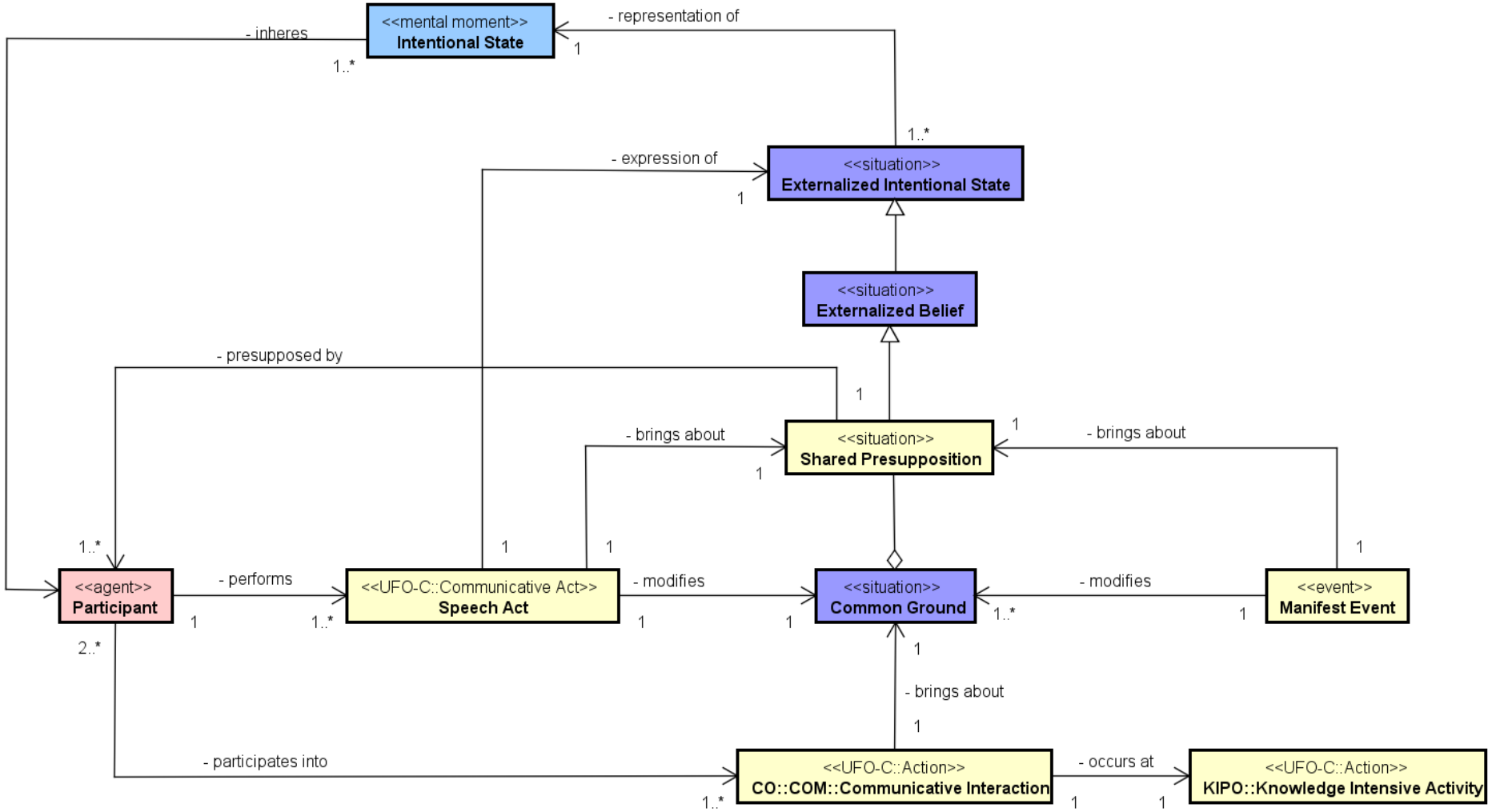

Figure 12 – Common Ground in CognitiveKiP

5.6) On Collective Intentionality and Social Concepts

Largescale groups and institutions have often been cited as a type of group that genuinely have a mind of its own [Ludwig, 2017] due to their complexity, hierarchical structure and differentiation of roles that are successively filled in by different individuals along the time. These roles are attributed to either objects (physical objects, data objects) or persons (participants of the process) and form the core definition of what is deemed as a "status-function" [Searle, 1995]: "Given a context C, an object/person X and a function Y; X counts as Y in context C. It is only able to be performed in virtue to the collective acceptance of the function by the group."

The status-function Y is a function that the object or person X was not able to perform before the imposition of function Y by the collective intentionality and, thus, it is only able to be performed in virtue of its collective acceptance by the group, at a specific context [Searle, 2003].

Within a KiP, the status-functions affect either Participants or Resources in different ways, describing both an organizational reality of participants' roles (such as manager, analysis, or an operator) and functions of a resource within a specific activity (such as a timetable, or a system log).

An important concept for understanding the enactment of status-functions is a special type of Speech Act called Declarative Speech Act [Searle & Vanderveken, 1989], that modify the social sphere by its performance, creating new social concepts.

Definition D39: Declarative Speech Act (UFO-C::Communicative Act):
A Speech Act that modify the social sphere of the Process.

A21: "All Declarative Speech Acts can create either (i) a Process Role, (ii) Participant Social Commitment or (iii) a Social Object"

(A21) ∀dsa (Declarative_Speech_Act→ (∃pr(Process_Role(pr) ∧ creates(dsa, sr)) ∨ (∃pc(Participant_Social_Commitment(pc) ∧ creates(dsa, so)) ∨ (∃so(Social_Object(so) ∧ creates(dsa, so)))

We argue that the imposition of a function within a KiP can happen in three specific cases: as (i) Social Commitments and their respective counterparts, the Social Claims, describing relations between Participants; as (ii) Process Roles, composed by sets of Social Commitments and Claims, describing specific roles with their respective powers and responsibilities within the social structure; and as (iii) Social Objects imposed upon physical or data objects, describing specific function that the object can perform, for example during one or more Resource Participations. All of them can be created by the performance of a Declarative Speech Act. Social Commitments and Claims can be created by Commissive Speech Acts. The details of each of these three effects of Collective Intentionality will be described in the next section.

5.7) Social Objects

In a Knowledge-intensive Process scenario, the dynamics of the social objects is an important topic, depicting the relationship between the cognitive aspects of BPM (such as Collective Intentionality) and classic process concepts such as the different roles and types of objects. One kind of function imposition following Collective Intentionality is imposition of Social Objects.

An example of the imposition of a social object is a piece of paper becoming a contract, by the collective intentionality of the parties involved writing and signing it; the contract (originally a piece of paper) has the function of creating a Social Commitment and Claim between the signing parties.

Definition D40: Social Object (UFO-C::Social Object):
A Social Object is an abstract object representing the attribution of capacities and functions that were not previously existent to the physical or data object it is imposed upon.

It is also helpful in representing the reality of a KiP instance as there are typical occurrences of, for example, commitments being generated by messages exchanged by process agents and a whole range of resources (documents, spreadsheets, software) assuming functions that are, by definition, beyond their original meaning (for example, a spreadsheet that is defined as the recognized schedule of a project).

5.8) Social Commitments and Claims:

UFO describes Social Commitments and Claims as Social Moments from UFO-C, being "Types of intentional moments that are created by social actions" [Guizzardi et al., 2006]. Social commitments and Social claims are types of social moments. A social commitment is a commitment of an agent A towards another agent B. As an externally dependent moment, a social commitment inheres in A and is externally dependent on B. The social commitments necessarily cause the creation of an internal commitment in A. Also, associated to this internal commitment, a social claim of B towards A is created. Commitments and claims always form a pair that refers to a unique propositional content, represented by a Social Relator. Social Relators are composed of one or more pairs of social commitments and social claims. Like all UFO Relators, a Social Relator is founded on a particular Event.

In CognitiveKiP, the founding event of a Social Relator depends on a social context, either as (i) a description of the Social Relator in a Normative Description that is valid in that context (e.g. an specific KiA) or (ii) a Declarative Speech Act performed by an Agent. Both cases are only valid if they are recognized by the KiA Participants (i.e. the collection of Participants involved in the KiA execution).

There are two main types of commitments: explicit and implicit. Explicit commitments are represented or communicated in an explicit way between agents, usually through the expression of speech acts in a variety of forms (such as messages) or specified in normative descriptions (such as written contracts). Implicit commitments are those that the agents do not need to communicate, since they are implicitly defined as common knowledge among them. Usually, explicit commitments are represented in a variety of forms, such as messages, contracts, business rules specifications and laws, while Implicit Commitments may be composed by habits and accepted practices within a community.

An important class of Commitments that can be either explicit or implicit is called Meta-commitment, defined as a "commitment to accept commitments" [Castelfranchi, 1995]. They are usually associated with a Social Role, as potential commitments to become operational due to the need or circumstance. An example would be the meta-commitment of a software developer to instruct his fellow team members on a specific technology or programming language that can be useful for a project.

With the following concepts in mind, we can define the concepts of Participant Commitment, Participant Claim (its counterpart), Social Commitment and its counterpart, Social Claim, such as follows:

Definition D41: Participant Commitment (UFO-C::Social Commitment):
A commitment of an agent A towards another agent B. As an externally dependent moment, a social commitment inheres in A and is externally dependent on B.

Definition D42: Participant Claim (UFO-C::Social Claim):
A Claim of an agent A that receives a commitment from an agent B. It is the counterpart of a Social Commitment.

Definition D43: Process Role (UFO-C::Social Relator):
An UFO-C::Social Relator composed of two or more pairs of associated Participant commitments/claims.

A22: "A Participant Social Commitment is a Social Commitment between two Participants of the KiP, where one commits to the Social Commitment, and the other receives the corresponding Participant Social Claim that is counterpart of the Participant Social Commitment. It is valid until the occurrence of one of the possible Discharge Conditions of the Participant Social Commitment."

(A22) ∀psc,psm (Participant_Social_Commitment(psc) ∧ Participant_Social_Claim(psm) → ∃pt1, pt2, dc(Participant(pt1) ∧ Participant(pt2) ∧ Discharge_Condition(dc) ∧ commits(pt2, psc) ∧ receives(psm, pt1) ∧ discharges(psc, dc)) ∧ is_counterpart_of(psc,psm) ∧ (pt1 ≠ pt2)))

A23: "A Commissive Speech Act can create a Commitment between the Agent that performs it and one or more Agents."

(A23) ∀csa (Commissive Speech Act → ∃psc,psm,pt1,pt2(Participant_Social_Commitment(psc) ∧ Participant_Social_Claim(psm) ∧ Participant(pt1) ∧ Participant(pt2) ∧ performs(pt1,csa) ∧ creates(csa, psc) ∧ commits(pt1, psc) ∧ receives(psm, pt2)))

Note that commitments do not specify activities, but rather they constitute a high-level social abstraction that the participants in a business process enact [Singh, 1999], together with the discharge conditions that must hold when a commitment is fulfilled.

There are interesting traits of the dynamics of commitments between agents [Dalpiaz et al., 2015]: "The social commitments of an Agent typically constrain him to act in accordance with them. A social commitment is discharged when a desired discharge condition is satisfied. The condition can be an event, or a condition relative to one of the agents, typically involving contextual elements."

Definition D44: Discharge Condition (UFO-B::Event):
An Event that discharges the Social Commitment (and its associated Claim).

5.9) Normative Descriptions

A special type of Social Object is a Normative Description defines one or more rules/norms recognized by at least one Social Agent and that can define nominal universals such as Social Moment universals (that is, social commitment types such as "PhD Candidates must attend the weekly meeting on friday"), Social Objects (the Brazilian Constitution, the crown of the king of Spain) and Process Roles (such as president, prime minister, PhD candidate or pedestrian).

Definition D45: Normative Description (UFO-C::Social Object):
A type of Social Object composed of Declarative Speech Acts that enable the enactment of Social Objects, Participant Commitments, Participant Claims and Social Relations.

A24: For each Activity, All Participants either (i) participating at its Communicative Interactions or (ii) performing the Activity, recognize the social concept involved.
(A24) ∀act(Activity) → ∃pt,ci,cm,cl,sr,so (Participant(pt) ∧ Communicative_Interaction(ci) ∧ Commitment(cm) ∧ Claim(cl) ∧ Process_Role(pr) ∧ Social_Object(so) ∧ occurs_at(ci,act) ∧ (performs(pt,act) ∨ participates_into(pt,ci)) ∧ recognized_by(cm,pt) ∧ recognized_by(cl,pt) ∧ recognized_by(pr,pt) ∧ recognized_by(so,pt))

In our proposal, Social Concepts are either (i) pre-existent, being created before the execution of the KiP instance or (ii) created in ad-hoc manner during the execution of the KiP. Pre-existent social objects, participant commitments, claims and process roles should be described in a Normative Description recognized by all Participants. Ad-hoc commitment and claims are created by the exchange of Speech Acts during the Communicative Interaction that occurs during a Knowledge-intensive Activity. Figure 13 depicts the Social Concepts in CognitiveKiP.

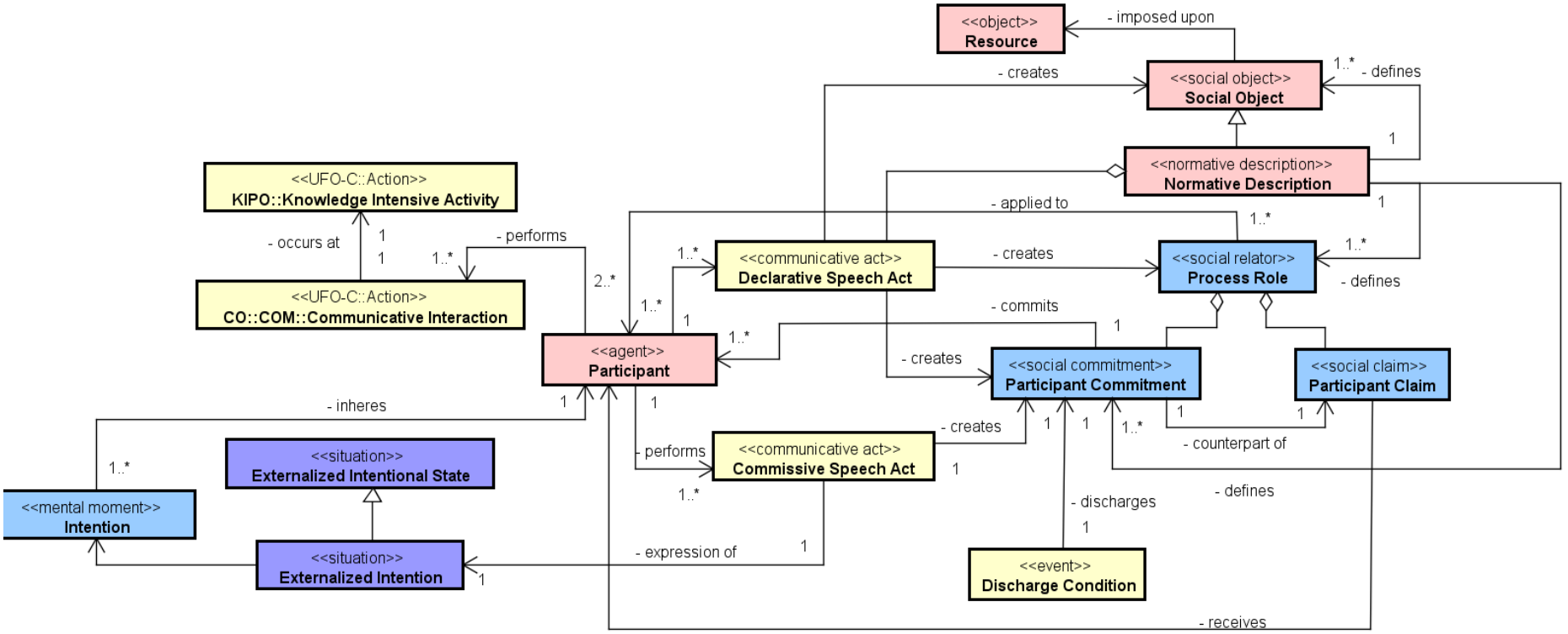

Figure 13 – Social concepts of CognitiveKiP

5.10) On Decisions during the execution of a KiP

While the communicative interactions and the exchange of speech acts between process participants form a very important part of a KiP execution, another critical element is the nature of the decision-making tasks. KiPO depicts the elements involved during decision-making rationale by means of its Decision (sub)Ontology (DO), with its central element called Decision.

At CognitiveKiP, we have adapted the KiPO definitions as follows: "a Decision is a type of Knowledge-intensive Activity triggered by a Question (an UFO-B Event) and executed by a Participant (an UFO-C Agent), who intentionally commits to solve the Question by making (performing) the Decision. When making a Decision, the Agent chooses among several Alternatives, which represent potential situations (UFO-B Situations) that will be achieved (Chosen Alternative) or not (Discarded Alternative) depending on the Decision result" [França et al., 2014].

We argue that there are two distinct activities involved in a decision-making task during the execution of a KiP: (i) the decision-making task itself (depicted in KiPO as a Decision) and (ii) the chosen course of action (a distinct Activity), triggered by the Chosen Alternative and determined by the previously Decision made. Also, the Situation brought about by the Decision is not only the Chosen Alternative, but a new concept, Decision Results, composed by all the Alternatives involved at a Decision, either Chosen Alternatives or Discarded Alternatives.

Definition D46: Decision Results (UFO-B::Situation):
A state-of-affairs that is composed by all Alternatives involved at a Decision, including Discarded Alternatives and Chosen Alternative.

Therefore, taking into consideration the Postulate of Cognitive Regulation of Action [Castelfranchi & Paglieri, 2002] described in Section 4.6, we have adapted and simplified the Goal processing model so that the Alternatives depict Situations that fulfill Active Goals, which have the propositional content of Desires that are still not acted upon or pursued by an Action. The Chosen Alternative fulfills a Chosen Goal, which has the propositional content of an

Intention and therefore is acted upon by an Action; the Chosen Goal is therefore the Goal of the chosen course of action. Finally, the execution of a Decision brings about the situation of its results (i.e. after the Decision is made, a course of action is chosen), triggering the Chosen Course of Action towards fulfilling a Chosen Goal.

Definition D47: Active Goal (UFO-C::Goal):
A type of Goal that has the propositional content of a Desire and, thus, is still not acted upon or pursued by an Activity.

A25: "A Goal is deemed an Active Goal, being the propositional content of an Agent's Desire, if either (i) it is not supported by any Externalized Beliefs composing the Common Ground OR (ii) it is unsupported by at least one Externalized Belief composing the Common Ground."
(A25) ∀acg (Active_Goal(acg) → (∃xb, ds (Externalized_Belief(xb) ∧ Desire(ds) ∧ propositional_content_of (acg, ds) ∧ (unsupports(xb, acg)) ∨ ¬(∃xb (supports(xb, acg))))

Definition D48: Chosen Goal (UFO-C::Goal):
A type of Goal that has the propositional content of an Intention and, thus, is acted upon or pursued by an Activity.

A26: "An Goal is deemed a Chosen Goal, being the propositional content of a Participant's Intention, if it is both (i) supported by at least one of the Externalized Beliefs AND (ii) NOT unsupported by any of the Externalized Beliefs composing the Common Ground for the Knowledge-intensive Activity to be executed."
(A26) ∀csg (Chosen_Goal(csg) → (∃xb, it (Externalized_Belief(xb) ∧ Intention(it) ∧ propositional_content_of(csg , it) ∧ supports(xb, csg))) ∧ ¬(∃xb (unsupports(xb, csg)))

The Goals involved in a KiP (and its composing KiAs) must be defined and shared among the Participants, usually being defined either by a third-party (at the strategic planning meeting of the organization) or during the execution of the process. Nevertheless, prior to being chosen and acted upon, they are initially considered as Active Goals and, as such, as expressions of Desires of a Participant.

The decision-making, i.e. the selection of Goals for pursuit, is done based on the Beliefs of the Participant performing the Decision. We have two types of Externalized Beliefs involved: (i) Support Belief, being an externalized Belief that supports a specific Goal and (ii) Unsupport Belief, being an externalized Belief that unsupports an specific Goal. All of them being representations of the Beliefs of the Participant that performs the Decision.

Definition D49: Support Belief (UFO-B::Situation):
A type of Externalized Belief that provides support for a Goal.

Definition D50: Unsupport Belief (UFO-B::Situation):
A type of Externalized Belief that takes away support (unsupports) for a Goal.

An interesting feature of this Decision model is that, Support Beliefs and Unsupport Beliefs, being subtypes of Externalized Belief, can be either representation of a Belief that only the Participant performing the Decision has (for example, a team member that disagree on a choice while the rest of team agrees) or a Shared Presupposition coming from a Communicative Interaction, representative of the shared consensus of a group. As the Participant is about to perform the decision, it ponders all the choices available (Alternatives), based on his Beliefs (either from individual opinion or presuppositions from interactions) and chooses the Alternative to be acted upon, as depicted in the axioms below.

A27: "For each Alternative, there is a Decision and a Decision Results brought about by the Decision, so that the Alternatives compose a Decision Results and the Decision brings about a Decision Results."
(A27) ∀at(Alternative) → ∃ds,dr (Decision(ds) ∧ Decision_Results(dr) ∧ brings_about(ds,dr) ∧ composes(at,dr))

A28: "For each Chosen Alternative, there is an Activity triggered by a Chosen Alternative and a Chosen Goal, fulfilled by the Activity."
(A28) ∀ca(Chosen_Alternative) → ∃act(Activity) ∧ Chosen_Goal(csg) ∧ Intention(it) ∧ triggers(ca,act) ∧ fulfills(act,csg) ∧ propositional_content_of(ca, csg) ∧ propositional_content_of(csg, it)

A29: “For each Discarded Alternative, there is an Active Goal that is the propositional content of a Desire.”

(A29) ∀ca(Discarded_Alternative) → ∃ag,de (Active_Goal(ag) ∧ Desire(de) ∧ propositional_content_of(ag, de)))

Figure 14 depicts the Decision elements of CognitiveKiP.

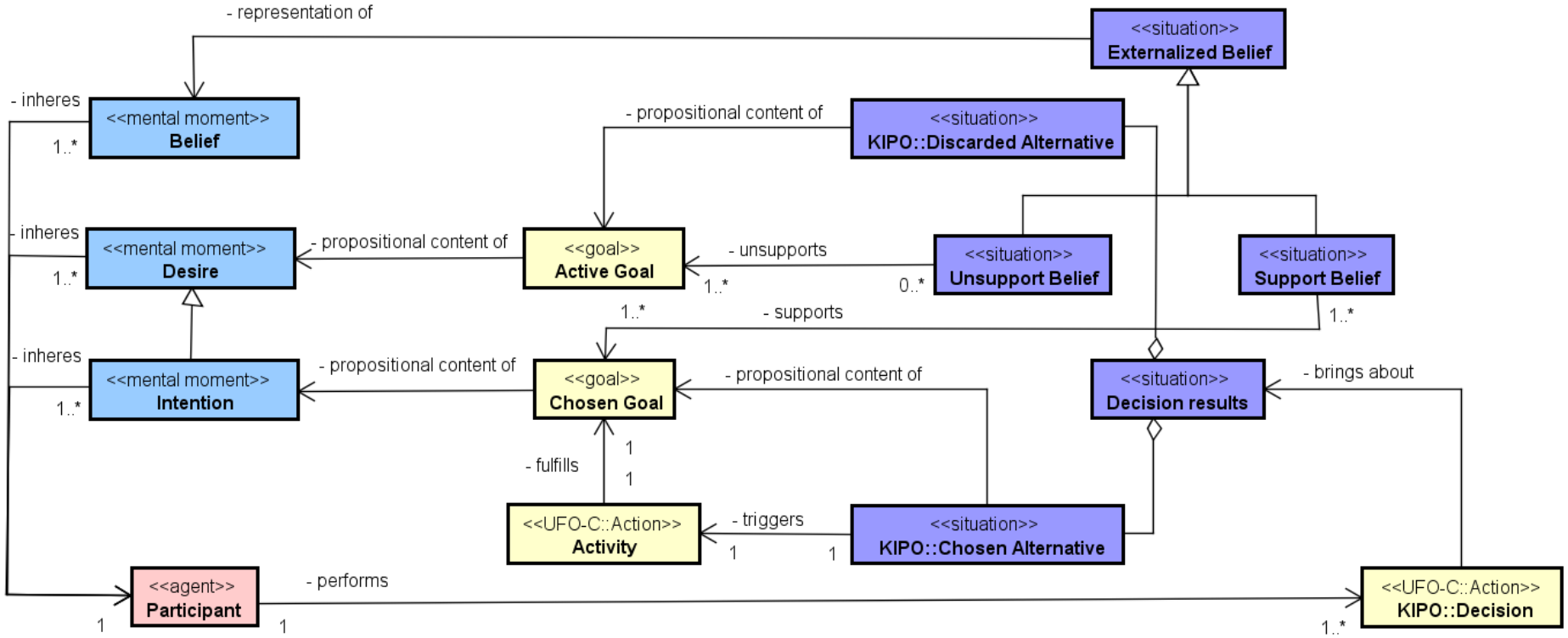

Figure 14 – Decision elements of CognitiveKiP

In the previous subsections, we have defined the main BPM concepts (Participant, Resource, Activity, Goal) within a Cognitive BPM paradigm, relating each of them to a cognitive dimension by defining their relationships with concepts such as Intentional States, Speech Acts and Common Ground. As the CognitiveKiP theory is defined and the conceptual modeling of the ontology is described, we can now proceed to a proof of concept study at a real scenario and the theory evaluation in the following section.

Chapter 6 – Theory Evaluation

The research described in this thesis proposes a new theory for KiP. This chapter addresses the definition of a theory - especially in the field of Information Systems - as well as the evaluation of the proposal.

6.1) Definition of Theory in Information Systems

For many disciplines, the proposal and evaluation of a theory is a central goal of their research endeavors. By proposing high-quality theory, we are more likely to enhance our domain of knowledge, expand the research field and contribute to the work of other researchers, who will further improve, criticize and enhance the theory with their own contributions and work. Information Systems is an applied science research field; however, in spite of the importance that theory development has in IS research, the development of new theories and the refinement of existing theories have been relatively neglected features of research within the information systems discipline. [Weber, 2006].

Research in the information systems field examines more than just the technological system, or just the social system, or even the two side-by-side; in addition, it investigates the phenomena that emerges when the two interact" [Gregor, 2006]. Another concern is the difference between native (Information Systems theories) and imported (theories from other fields of research) theories. Moody et al. [2009] provide a useful distinction between native and imported theories: “A native (indigenous) theory is a theory specifically developed to describe, explain, predict, or design IS phenomena. An imported (exotic, introduced) theory is a theory borrowed from an external (reference) discipline to describe, explain, predict, or design IS phenomena”.

This research adopts the following definition of a theory: “A theory is an abstract entity that aims to describe, explain and enhance understanding of the world and in some cases to provide predictions of what will happen in the future and to give a basis for

intervention and action" [Gregor, 2009]. Central to many understandings of theory are the twin goals of explanation and prediction. Explanation is closely linked to human understanding, as an explanation can be provided with the intent of inducing a subjective state of understanding in an individual. Apart from explanations, theories can also aim at making predictions, which allows the theory both to be tested and to be used to guide action.

The two main goals of a theory - the explanatory power and the predictive power - form what Dubin [1979] calls the precision paradox and the power paradox, respectively. Some theories can focus on one goal, either explanation or prediction, at the expense of the other. That is, it is possible to achieve precise predictions without necessarily having understanding of the reasons why outcomes occur. For example, it is possible to predict the weather by the observation of the clouds or to predict that a computer network will collapse by the analysis of its lost packets and transmission glitches (predictive power), without having the understanding of how precipitation of rain or a network failure occur (explanatory power). In contrast, it is possible to have models that are powerful in contributing to understanding of processes without providing, at the same time, precision in prediction.

Case studies of information systems implementation might give us a good understanding of how lack of involvement of users can lead to user dissatisfaction with a completed system. It would still be difficult to predict with any degree of accuracy the degree of user dissatisfaction arising from lack of involvement over a wide range of systems and settings.

Gregor [2006] proposed a method for classifying Information Systems theories, with a focus on the four primary goals of an IS theory:

1. **Analysis and description**. The theory provides a description of the phenomena of interest, analysis of relationships among those constructs, the degree of generalizability in constructs and relationships and the boundaries within which relationships, and observations hold;
2. **Explanation**. The theory provides an explanation of how, why, and when things happened, relying on varying views of causality and methods for argumentation. This explanation will usually be intended to promote greater understanding or insights by others into the phenomena of interest;

3. **Prediction**. The theory states what will happen in the future if certain preconditions hold. The degree of certainty in the prediction is expected to be only approximate or probabilistic in IS;
4. **Prescription**. A special case of prediction exists where the theory provides a description of the method or structure or both for the construction of an artifact (akin to a recipe). The provision of the recipe implies that the recipe, if acted upon, will cause an artifact of a certain type to come into being.

The combinations of the four goals lead to five classifications of IS theory [Gregor, 2006], as follows:

Type I. **Analysis**. The theory does not extend beyond analysis and description. No causal relationships among phenomena are specified and no predictions are made.

Type II. **Explanation**. The theory provides explanations but does not aim to predict with any precision. There are no testable propositions.

Type III. **Prediction.** The theory provides predictions and has testable propositions, but does not have well-developed justificatory causal explanations.

Type IV. **Explanation and Prediction**. The theory provides predictions and has both testable propositions and causal explanations.

Type V. **Design and action**. The theory gives explicit prescriptions (e.g., methods, techniques, principles of form and function) for constructing an artifact.

From this initial taxonomy, Weber [2012] performed an evaluation of the five types proposed by Gregor [2006] and stated that Type I theories are typologies and Type V theories are solely models and, thus, not real theories. Type II and III, respectively theories for explaining and predicting, may constitute theories depending on how rigorously their elements and its whole are defined. Finally, the author concludes that a real theory in IS is best aligned with the Type IV theory classification, a theory for both explanation and prediction.

Considering these two aspects of an IS theory, we adopt the definition of Weber [2012] and we apply the framework for the evaluation of the proposal as an IS Theory.

6.2) A Framework and Criteria for Theory Evaluation

Weber [2012] proposes a framework for Information Systems theory's evaluation that begins by establishing a distinction between models and theories:

"All theories are models, but not all models are theories. A model must satisfy certain conditions before I deem it to be a theory – conditions that relate to rigorous specification of its "parts" and particular qualities of its "whole". Thus, the existence of a model is a necessary condition for the existence of a theory, but it is not a sufficient condition. The existence of a theory, however, is a sufficient condition for the existence of a model."

From this initial premise, the author describes his proposal, defining that a theory should be evaluated from two perspectives: The first is the "parts" perspective (the quality of the theory's individual components; The second is the "whole" (the quality of the theory considered *in toto*).

6.3) Evaluation of the Parts

An IS theory usually has three parts: (i) its constructs; (ii) its associations and (iii) the states they cover. In addition, theories that cover dynamic phenomena have a fourth part, (iv) the events they cover.

The description of the parts of a theory is of paramount importance as it defines and circumscribes the boundary or domain of the theory, being the precise description of the phenomena it is intended to cover (i.e. its focal phenomena). How precise a theory's parts is described has a direct relationship to how better able the researchers are to design tests that fall within the theory's domain and establish a specific context where the evaluation of the theory can be performed. The definition of a boundary is also beneficial, in the case of empirical evaluation and testing, to the filtering of data collected to perform tests and experiments in the theory's domain.

1. Constructs

A construct in a theory represents "an attribute in general of some class of things in its domain" (as opposed to a particular attribute of a specific thing). Being general attributes,

they need to be defined precisely to ensure that the meanings of each class and the things in each class are clear.

Once the meanings of the classes of things that a theory covers are clear, the nature of each attribute in general that pertains to a particular class ought to be defined precisely, as developing valid and reliable empirical indicators of the attributes in general will be difficult if there is ambiguity or lack of clarity in the definition of the constructs, affecting also the interpretation of data collected for the theory's evaluation.

2. Associations

An association shows that the values of one construct are somehow related to the values (static phenomena) or history of values (dynamic phenomena) of another construct. Associations in a theory can have multiple meanings. When evaluating the meaning to ascribe to an association at the outset it is important to reflect upon whether a theory covers only static phenomena, dynamic phenomena, or a combination of both static and dynamic phenomena.

If the theory covers static phenomena, an association shows that the values of one construct are somehow related to the values of another construct. For instance, when "snapshots" of the phenomena that pertain to things in a class are taken at some point in time and the values of attributes of things in the class are examined, the theory might predict that high values for instances of one construct will tend to be associated with low values for instances of another construct.

Otherwise, in the case of a theory that covers dynamic phenomena (events), an association shows that a history of values for instances of one of the constructs is conditional on a history of values for instances of the other construct.

3. States

A state in a theory can be defined as the complex attribute that represents a state of a theory's elements (its attributes in general along with their associated values). A theory should specify clearly and as precisely as possible the things in the class or classes of things that it is intended to cover (the "state space" of the theory). In other words, it should stipulate those states that might arise for things in the class or classes of things that fall within its domain and for which it is intended to have explanatory and predictive power.

The theory boundary must be also considered to discard some the combinations composing the state space that either cannot occur naturally or fall outside the theory boundary. The three parts above are sufficient for theories that cover static phenomena. A fourth part is necessary for a theory that cover dynamic phenomena.

4. Events

The specific case of a theory that covers dynamic phenomena brings into question a fourth element, the event. An event is an element of a theory that is representative of a theory's element changing from one of its states to another. Like the other elements, an evaluation of the theory's boundary and which events fall within or outside the theory must be done.

If a theory is intended to cover events, the event space that falls within the theory's boundary must also be articulated. At the outset, all conceptually possible pairs of inside-boundary states must be considered (recall, each event can be conceived as a before-state, after-state pair). These constitute the conceivable event space covered by the theory. Some combinations can be eliminated because they cannot occur naturally (they are unlawful). Those that remain must be evaluated to determine whether they are covered by the theory. In other words, they must be partitioned into inside-boundary events and outside-boundary events.

Figure 15 depicts the framework from Weber [2012] and its analysis of the parts of an IS Theory.

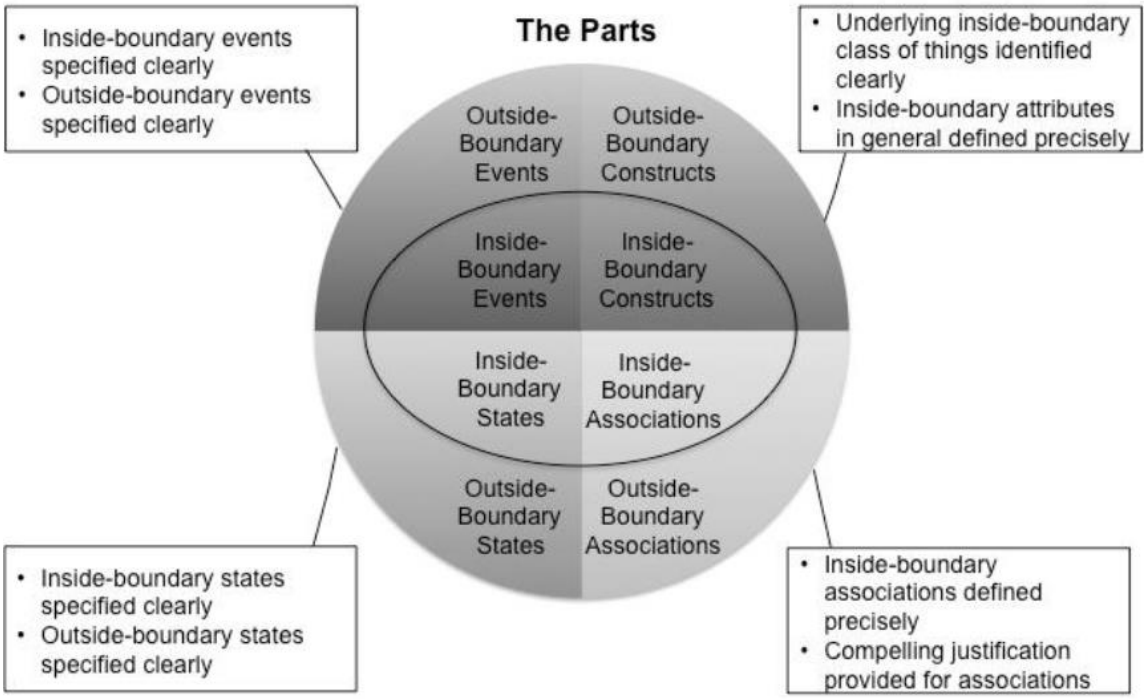

Figure 15 – Framework for evaluating a theory by its parts [Weber, 2012]

When evaluating a theory, the focus initially should be on the quality of its parts. The parts of a theory need to be described precisely because they circumscribe the boundary or domain of the theory – that is, the phenomena it is intended to cover. If researchers have a clear understanding of the theory's parts, they are better able to design tests that fall within the theory's domain rather than unwittingly testing the theory in an inappropriate context. Moreover, they should be able to filter data they have collected so they undertake tests on only the subset representing phenomena in the domain the theory covers.

6.4) Evaluation of the Whole

A second step for the evaluation of an IS theory is also necessary, being the assessment of the IS theory as a Whole with respect to the following criteria.

1) Importance

The importance (or utility) of a theory is assessed via judgments made about the importance of its focal phenomena, as it would be pointless to have a theory with well-defined parts but addressing uninteresting phenomena. The assessment of how interesting is the focal phenomena is analyzed from two distinct viewpoints: the theoretical

viewpoint of research (e.g.: improving science with new perspectives about the phenomena) and the empirical viewpoint of practice (e.g.: improving the effectiveness and efficiency of some entity's activities). Potentially, enhanced understanding of the focal phenomena will provide key insights that enable both theoretical or empirical progress to be made on some problem within a discipline.

2) Novelty

A theory will be assessed on its novelty to the extent it changes the paradigms used by researchers to investigate phenomena within their discipline. Usually, it is considered novel by most researchers if it provides a way of resolving "anomalies" within their discipline that existing theories are unable to explain or predict. Other possibilities are that a theory enables to conceive new and interesting phenomena or to re-conceptualize existing phenomena in new and interesting ways, as well as theories that break the cycle of "normal science" within a discipline and set new paths for the discipline to follow.

3) Parsimony

Parsimony is the measure on how a theory achieves good levels of predictive and explanatory power in relation to their focal phenomena using a small number of constructs and associations. It is an important measure due to the fact that, by using a small number of constructs, they also limit the size of both its conceivable state space and event space. As a result, it is often easier to analyze the elements that fall within the boundary of the theory.

4) Level

The level of a theory means how broad or specific it is regarding to its focal phenomena. Usually, Generality is attained as a trade-off between a theory's accuracy and simplicity. The measure of a theory's "appropriate level" is usually based on subjective criteria, as it varies between different disciplines and research fields. In the context of their discipline, researchers make judgments about whether a theory is formulated at an appropriate level to be interesting or useful.

5) Falsifiability

Falsifiability is an important criterion due to the fact that most, if not all, theories cannot be proven via empirical tests, as it is impossible to test the theory for all possible scenarios or applications within its boundary. However, if a theory has been clearly articulated, tests can be designed and applied to examine the conditions that researchers believe are most likely to lead to the theory being falsified (i.e. failing empirical testing) rather than supported. In this sense, support for a theory grows when its powers of prediction and/or explain remain robust across different empirical tests.

Researchers must be able to generate precise predictions about the focal phenomena in order to be capable of falsifying a theory, as there is the risk of compromising the results of the empirical testing of the theory if the predictions used in falsifiability tests are vague.

Figure 16 depicts the perspective of the "whole" for the evaluation of an Information Systems Theory [Weber, 2012].

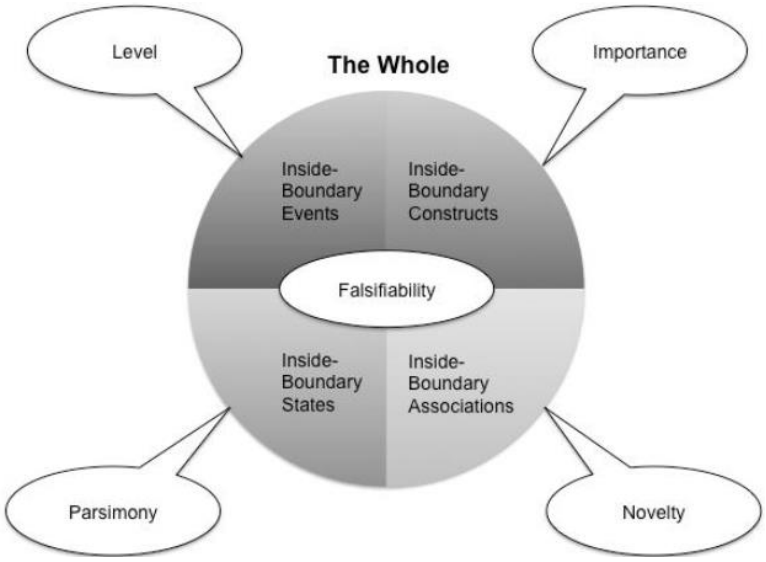

Figure 16 **– Framework for evaluating a theory by its whole [Weber, 2012]**

6.5) Evaluating the CognitiveKiP Theory

After the description of Weber's framework for IS theory evaluation, we can analyze the thesis' proposal, both the theory and its materialization and model (the CognitiveKiP Ontology). Due to the impossibility of testing the theory at an infinite or exhaustive number of possible scenarios, its evaluation will be performed in two steps, following Weber's proposal: A first step will be the evaluation of the theory's parts, discussing how

precise its definitions are as well as its scope, in comparison to the theory boundaries. The second step will be the evaluation of the theory as a whole, discussing how it depicts the focal phenomena within its boundaries. Finally, a third and important step is the falsifiability analysis, where data from two different scenarios will be used to test the theory's possible limitations and the thresholds of its boundary.

A. Evaluation of Theory as Parts

Beginning with the "parts" section of the framework, it is necessary to assess each concept and axiom proposed at CognitiveKiP and evaluate its boundary and, if it is the case, its limitations.

We start by the constructs involving Intentionality. We have proposed a novel concept "Intentional State", being an specialization of UFO-A::Intentional Moment, in order to adapt Searle's concepts of Intentionality for our extension of KiPO (that uses UFO as its foundational ontology). It was chosen since it is a concept from a theory that is external to both UFO and KiPO, thus a new concept would be a way of "bridging" the theory without implications of semantic equivalence.

Going further to the specializations of an Intentional State, Belief, Desire, Intention and Feeling were already present in KiPO and well-founded in UFO-C::Mental Moment (a Mental Moment being an specialization of Intentional Moment, the concept brought from "adapted" from UFO for the proposal of Intentional State). The reason for this new specialization is their new semantics, as they are depicted as possessing possible externalized representations (Externalized Intentional States) and expressions (Speech Acts).

Lastly, an important point particularly concerns Feeling, which is expressed and represented also as Intentional State, but it has no direct effect to the dynamics of goals and decisions about activities performed during a KiP execution (as described at CognitiveKiP) and therefore, its effect on the process, if any, are considered to be external to the boundary of this thesis. This is an important part of the boundary of CognitiveKiP, although the dynamics of representation and expression of any Intentional State is thoroughly described. Table 1 presents the assessment of Constructs and Associations for the concepts of Intentionality. The assessment focus on how clearly defined is each CognitiveKiP concept and its associations. In the case of a more complex or overlapping

definition with other concepts, a complementary explanation will be provided at the evaluation. Table 1 shows the assessment for the Intentionality's Constructs.

Table 1 – Evaluation of the Constructs for Intentionality

Construct	Assessment of Construct and Associations
Intentional State	Defined clearly at D1 in order to represent Searle's concept of Intentional State.
Belief	Defined clearly at D2.
Desire	Defined clearly at D3.
Intention	Defined clearly at D4.
Feeling	Defined clearly at D5. CognitiveKiP defines precisely the dynamics of its representation and expression through speech acts, but the concept's possible effects on a KiP are considered external to the thesis' boundary.

The process participants are defined as the Participant concept, which is specialized into two generalization sets. The first regarding the number of Physical Agents participating into the business process (Individual Participant and Collective Participant) and the other regarding the nature of the participation into a Knowledge-intensive Process (Innovation Agent, Impact Agent and External Agent).

The two generalization sets are orthogonal; for example, an instance of Impact Agent may also instantiate either an Individual or a Collective Participant, while an instance of an External Agent may also instantiate an Individual or Collective Participant. Another interesting feature is that the KiP participant concepts are activity-specific, which means that the scope of a single participation refers to each activity composing the KiP. For example, given two Activities x and y, the same Participant can play the role of an Impact Agent in Activity x and play the role of an Innovation Agent in Activity y.

The concept of Innovation is taken from the original KiPO definition, being an important element to the definition of a Knowledge-intensive Process. Although "Novelty" and "Innovation" are subjective terms, they are clearly defined as a contribution to a specific KiA, usually in the form of knowledge exchange between the

Innovation Agent and other participants that participate in the KiA (possibly, but not restricted to, a Communicative Interaction). Finally, the External Agent is precisely defined as a participant of a Communicative Interaction occurring at a Decision. Table 2 depicts the assessment of Constructs and Associations for the concepts related to KiP Participants.

Table 2 – Evaluation of the Constructs for KiP Participants

Construct	Assessment of Construct and Associations
Participant	Defined clearly at D6.
Individual Participant	Defined clearly at D7. Overlaps with Innovation Agent, Impact Agent and External Agent.
Collective Participant	Defined clearly at D8. overlaps with Innovation Agent, Impact Agent and External Agent.
Innovation	Defined at D9. Concept is subjective by nature, but essential to the definition of a Knowledge-intensive Process and its activities.
Innovation Agent	Defined clearly at D10.
Impact Agent	Defined clearly at D11.
External Agent	Defined clearly at D12.

For the Event- and Activity-related constructs of CognitiveKiP, there is a clear distinction between agentive and non-agentive events in the theory. The concepts of Unintentional Event and Activity are clearly distinct. Also, Goal and its specializations (Conditioned Goal and Executive Goal) are also only associated with Activity, thus being clearly defined in the depiction of intentional acts during a KiP execution.

The Resources and its correlated Resource Participants are also precisely defined as non-agentive Participations in an Activity, as well as Pre-state and Post-state, formally characterized as UFO-C Situations related to both Unintentional Events or Activities, following the definitions of UFO-C [Guizzardi, 2006].

The Knowledge-intensive Activity construct specializes Activity and is necessarily composed by at least one Decision and at least one Communicative Interaction, making it distinct from a non-KiA (generic) process Activity. Table 3 lists the assessment of Constructs and Associations for Activities and Events.

Table 3 – Evaluation of the Constructs for Activities and associated concepts

Construct	Assessment of Construct and Associations
Unintentional Event	Defined clearly at D13 and distinct from Activity
Activity	Defined clearly at D14 and distinct from Unintentional Event.
Goal	Defined clearly at D15.
Conditioned Goal	Defined clearly at D16. Overlap with Chosen Goal and Active Goal.
Executive Goal	Defined clearly at D17. Overlap with Chosen Goal and Active Goal.
Resource	Defined clearly at D18.
Resource Participation	Defined clearly at D19.
Pre-state	Defined clearly at D20.
Post-state	Defined clearly at D21.
Knowledge-intensive Activity	Defined clearly at D22.

The evaluation of Decision-related constructs starts with Decision, the decision-making Activity *per se* and Question, the Situation that triggers the Decision to be performed. In the original KIPO, Question was defined as an Event [França et al., 2012] but, in order to improve the precision of this definition, we follow the UFO-B definition of a Situation as a trigger of an Event or Action [Guizzardi et al., 2013].

The Active Goal and Chosen Goal, with their respective associations to Desire and Intention, are precisely defined; moreover, given that both Conditioned Goal and Executive Goal are necessarily associated with an Intention and that an Active Goal is never acted upon, it may be inferred that only Chosen Goal can become a Conditioned or Executive Goal. At last, both Support Belief and Unsupport Belief are precisely defined, being Externalized Beliefs that are associated with Goals. Table 4 depicts the assessment of Constructs and Associations for the concepts for Decision and its related elements.

Table 4 – Evaluation of the Constructs for Decision

Construct	Assessment of Constructs and Associations
Question	Defined clearly at D23. Overlap with Pre-state and Post-state.
Decision	Defined clearly at D24.
Decision Results	Defined clearly at D25.
Chosen Alternative	Defined clearly at D26.
Discarded Alternative	Defined clearly at D27.
Active Goal	Defined clearly at D28. Overlaps with Conditioned Goal and Executive Goal.
Chosen Goal	Defined clearly at D29. Overlaps with Conditioned Goal and Executive Goal.
Support Belief	Defined clearly at D30. Overlaps with Shared Presupposition.
Unsupport Belief	Defined clearly at D31. Overlaps with Shared Presupposition.

The representations of Intentional States, depicted by the construct Externalized Intentional State, are precisely defined, as well as their specializations for each Intentional State. CognitiveKiP assumes that the Speech Act is their form of expression and the concepts of Assertive Speech Act, Directive Speech Act, Commissive Speech Act and Expressive Speech Act are precisely-defined, along with their specializations. The fifth Speech Act type, the Declarative Speech Act, is a special case, since any other types of speech acts can be also classified as a Declarative Speech Act (e.g.: A request for a technician to be responsible for monitoring internet traffic is both a Directive Speech Act and a Declarative Speech Act).

The Communicative Interaction is associated to Common Ground. Common Ground is a complex concept that is well defined, being composed of Shared Presuppositions; Common Ground accommodation (i.e., how its instantiation changes in a particular scenario) as a result of the occurrence of Manifest Events and/or Speech Acts during the Communicative Interaction and their perception by its participants. Table 5 depicts the assessment of Constructs and Associations for the concepts of Communicative Interactions and Speech Acts.

Table 5 – Evaluation of the Constructs for Communicative Interaction

Construct	Assessment of Construct and Associations
Externalized Intentional State	Defined clearly at D32.
Externalized Belief	Defined clearly at D33.
Externalized Desire	Defined clearly at D34.
Externalized Intention	Defined clearly at D35.
Externalized Feeling	Defined clearly at D36.
Communicative Interaction	Defined clearly at D37.
Speech Act	Defined clearly at D38.
Assertive Speech Act	Defined clearly at D39.
Directive Speech Act	Defined clearly at D40.
Commissive Speech Act	Defined clearly at D41.
Expressive Speech Act	Defined clearly at D42.
Declarative Speech Act	Defined at D43. Due to its broad definition. It can overlap with all other speech act types.
Shared Presupposition	Defined clearly at D44.
Manifest Event	Defined clearly at D45.
Common Ground	Defined clearly at D46.

The constructs describing Collective Intentionality are well defined, with no inconsistences or ambiguities in their definitions and associations with other CognitiveKiP constructs. Table 6 depicts the Constructs assessment for Collective Intentionality and Social Concepts.

Table 6 – Evaluation of the Constructs for Social Concepts

Construct	Brief Assessment of Construct Definition
Social Object	Defined clearly at D47.
Participant Commitment	Defined clearly at D48.
Participant Claim	Defined clearly at D49.
Process Role	Defined clearly at D50.
Discharge Condition	Defined clearly at D51. Overlaps with Activity and Unintentional Event.
Normative Description	Defined clearly at D52.

After the evaluation of Constructs and Associations, we proceed to analyze how precisely defined are the States and Events of CognitiveKiP.

Participant Commitments and Claims, due to their relationships with specific Discharge Conditions, have two states: Active (enacted) or Discharge (inactive, due to the occurrence of the Discharge Condition). Similarly, a Goal can be an Active Goal (the propositional content of an Agent's Desire) or a Chosen Goal (the propositional content of an Agent's Intention); the former being not acted upon and the latter being the selection of a goal prior to the execution of an Activity aiming towards its fulfillment. Alternatives follow a similar behavior, being either a Chosen Alternative or a Discarded Alternative as a consequence of the Beliefs involved at a Decision, either supporting or taking away support for a specific Alternative.

Common Ground is a bit more complex in the sense that, during a Communicative Interaction, it can have three distinct states: (i) Initial, occurring before the first speech act of the Communicative Interaction be performed; (ii) Modified, occurring after the performance of each Speech Act or the occurrence of each Manifest Event; and (iii) Final, composed of the final presuppositions after the Communicative Interaction ends. Therefore, all the possible states are precisely defined and depicted at Table 7.

Table 7 – Evaluation of the States of CognitiveKiP's Constructs

Construct	Assessment of States
Participant Commitment	Two states: Active or Discharged.
Participant Claim	Two states: Active or Discharged.
Common Ground	Three states: Initial (pre-conditions for interaction), Modified (at each manifest event and/or speech act), Final (at the end of the Communicative Interaction).
Goal	Two states: Active (Not acted upon its fulfillment) and Chosen (To be acted upon its fulfillment).
Alternative	Two states: Chosen and Discarded.

Finally, the Events of the theory can be analyzed in a similar sense. One of the most important dynamic events of CognitiveKiP is the Common Ground accommodation, resulting from the performance of each Speech Act or occurrence of each Manifest Event. The other dynamic events (Belief Support/Unsupport for Goals, Selection of Alternatives at a Decision and the Enactment/Discharge of Commitments) are precisely defined as well. Table 8 describes the events deemed relevant to the cognitive perspective of a KiP and how they are depicted in the CognitiveKiP Ontology model.

Table 8 – Evaluation of the Events of CognitiveKiP's Constructs

Construct	Assessment of Events
Common Ground Accommodation (Communicative Interaction)	Defined clearly by D38 and Axioms A14 and A15.
Belief Support/Unsupport for Goal (Decision)	Defined clearly by Axioms A23 and A24.
Alternative Selection (Decision)	Defined clearly by Axioms A20, A21, A22, A23 and A24.
Enactment/Discharge of Commitments (Commitment/Claim)	Defined clearly by Axioms A17, A18 and A19.

B. Evaluation of Theory as Whole

We present as follows the analysis of our proposed CognitiveKiP Theory with respect to each criterion proposed by the framework of Weber [2012].

1) Importance

CognitiveKiP brings an enhanced understanding of Knowledge-intensive Processes dynamics, due to its key insights about the role of interactions within the tasks of modeling and executing a Knowledge-intensive Process. Also, it constitutes a precise conceptualization about the nature of "Knowledge Intensity" of a process and brings a novel and semantically precise viewpoint on the cognitive aspect of KiPs and how it drives the process execution.

There is also an important contribution in terms of the CognitiveKiP ontology, since it serves as a meta-model for the extraction of speech acts and the exploration of Process Mining techniques in real-world scenarios, using the records of interactions digitally available in infrastructures such as social networks and organizational repositories for emails or BPMS-aware logs [Ricchetti et al., 2017].

2) Novelty

CognitiveKiP is a novel theory as it expands the Cognitive BPM paradigm towards a more "actionable" direction, with a new set of concepts to be employed towards managing KiPs. Being a well-founded ontology, the formal ontology that materializes the CognitiveKiP theory can be applied as a meta-model for extracting and modeling KiPs, as well as in other applications.

Also, the introduction of concepts from Pragmatics and Cognitive Psychology - such as Speech Acts and Intentionality - enables the analyst to explain the effects of these elements on the process and to predict outcomes. For example, based on the speech acts and expressed intentional states that are being performed at a Communicative Interaction during a KiP's execution, the analyst can predict which goals or alternatives are not prone to be chosen and the rationale behind the choice of a specific course of action.

3) Parsimony

Using UFO Constructs and KiPO as a base ontology, we obtained a non-ambiguous set of constructs. The number of constructs in the theory is minimal, since each and every construct is used in at least one definition of the focal phenomena of the theory, and therefore is strictly necessary for its description. By materializing the theory as a formal ontology, we also define the elements (interactions, speech acts, etc.) in a sufficient way for explanatory/prediction power with a reduced number of constructs.

4) Level

We define the theory elements in a general fashion, in order to characterize KiPs in general and their cognitive/intentionality aspects. Each concept could be detailed much further, especially if the theories of Intentionality and Pragmatics were explored in all its features. However, CognitiveKiP aims to describe a KiP dynamics in a general way, on a trade-off between detail and predictive/explanatory power.

5) Falseability

Most theories, as previously stated, cannot be proven empirically, due to the impossibility of testing every possible scenario of its application within the theory's boundary. However, it is possible to evaluate a theory's support, in the sense of how its powers of prediction and explanation remain robust across its different tests and at which degree empirical tests are evaluating the theory's elements within or outside its boundaries [Godfrey-Smith, 2003, pp. 202-218; Hempel, 1966, pp. 33-46].

For the falseability analysis of CognitiveKiP, we begin by defining its boundaries. As shown in the Evaluation of Parts Section, most concepts are precisely defined, while a few constructs have arguably broader definitions. In order to clarify these concerns, some points about the boundary of the thesis must be specified (that is, the theory limitations):

- The concept of Feeling, as an Intentional State, is considered within the boundaries of the theory in its dynamics of representation (Externalized Intentional States) and expression (Expressive Speech Acts). However, its effects on the dynamics of a KiP is outside the boundaries of the theory.
- The different forms of Beliefs involved in the selection of Goals to be pursued, as described in Castelfranchi & Paglieri's [Castelfranchi & Paglieri, 2007] work, were simplified in two Cognitive concepts: Support Belief and Unsupport Belief. Also, the concept of "Means-end Belief", present in the author's taxonomy and involved in the distinction between a Conditioned Goal (dependent on other goals for its fulfillment) and an Executive Goal (independent of other goals for its fulfillment) is also outside the boundaries of the theory.
- The Intentional States that are not expressed as Speech Acts are outside the boundary of the theory. The broad definition of a Speech Act in CognitiveKiP encompasses both spoken and written interactions between participants, as well as synchronous and asynchronous interactions. This includes most forms of communication supported by digital social platforms, such as social network posts, chats and emails, as well as indirect communication that occur through

books, organizational reports and specifications, and other forms of written exchanges, including graphic representations.

- More advanced elements of Speech Act Theory, such as indirect speech acts (i.e. speech acts with multiple meanings or speech acts that are presupposed to be more than a single speech acts) are outside of the theory's boundary, but the simplified depiction of the general concept of Speech Act theory serves as a base conceptualization for the analysis of a KiP.

With the clear definition of the theory's boundary above, the falseability assessment begins with the proposal of null hypothesis.

Thus, based on the original hypothesis of the research (H1), the Null Hypotheses (H') can be formulated, as follows:

H1: "*A Knowledge-intensive Process is driven by the Beliefs, Desires and Intentions of its participants"*

H1': *"A Knowledge-intensive Process is* ***not*** *driven by the Beliefs, Desires and Intentions of its participants"*

In the following subsections, two distinct case studies will be described as both proof-of-concepts for CognitiveKiP as well as for the evaluation of its falsifiability. To improve clarity at the scenario diagrams and due to space restrictions, the following subsections adapted the modeling notation, in order to use the CognitiveKiP concepts are used instead of UFO stereotypes.

6.6) Scenario 1: ICT Troubleshooting

The first scenario for the theory evaluation is a process for solving ICT (Information and Communications Technology) incidents, which follows the "Incident Management" specification from ITIL [Taylor et al., 2007]. This scenario definition and all its data considered for the evaluation was taken from a real Brazilian ICT outsourcing company, which has about a hundred contracts with diverse clients. One of the main services provided by the company is customer support, which intends to fulfill technical requests

(e.g. e-mail configurations, backup and restore) or solve technical problems (e.g. system failures) that are reported by clients to the company call-center.

The process is called here as “Solve ICT Incident”. In this process, contractors (client) contact the company’s call center with an incident solving request; then, the company’s employee registers the incident (which is called “opening a ticket” in the system) and forwards the request to the specific area and, after the service is finished, the ticket is closed. During each process instance, messages are logged into the company’s systems (both internal - between employees - and external - between employees and customers - messages), enabling a detailed analysis of the communicative interaction that occurs during each instance.

When a client reports a new problem, this triggers the creation of an incident ticket in a process-aware system called OTRS that supports company’s operations. Within OTRS, incident tickets are registered, alternative solutions are considered, a solution approach is defined, executed, validated and then deployed. During the resolution of a ticket, messages are exchanged among process participants (both technical teams and client) and associated to a ticket in OTRS. These messages contain natural language texts in chronological order; from them, it is possible to retrieve speech acts uttered from process participants.

Processes of this nature essentially involve the application of technical skills, troubleshooting abilities, collaboration and information exchange between stakeholders, and ad-hoc decisions are frequently discussed and made, since most of the problems are situational. For all these reasons, this process is characterized as a KiP and should be managed as so. The process activities are summarized as follows:

1. Contact Call Center: The customer sends an e-mail to the Call Center of the company requesting maintenance in some equipment or software service in its environment.
2. Open Ticket: The Call Center opens a ticket and forwards it to the responsible area.
3. Solve Incident: This ticket can be further handled through message exchanges between different areas of the company, and the customer, every time more information is necessary. At each routing, a code is generated for the message. Thus, it is possible to record the history (log) of this ticket until its conclusion.

4. Assess Ticket Resolution: The ticket is only closed when the customer receives confirmation that his request was executed successfully.

A model based on the process description above and using CognitiveKiP concepts is depicted in Fig. 17.

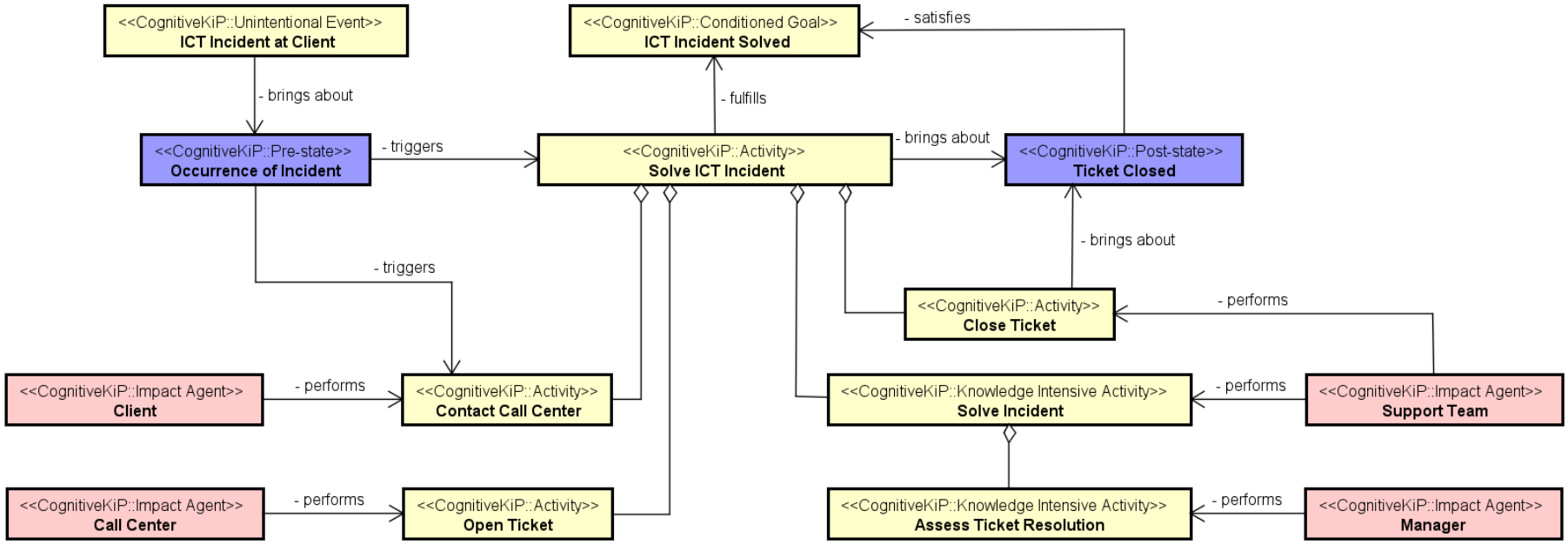

Figure 17 – Solve ICT Incident as a Knowledge-intensive Process

Four Participants are described at the model: (i) Client, the company's client that reports the incident; (ii) Call Center, the team responsible for opening the ticket, make the initial contact with the client and forward the support request for the team with the most adequate skillset to solve the problem; (iii) Support Team, the team responsible for solving the incident and closing the ticket upon the resolution of the incident; and (iv) Manager, the person responsible for evaluating the service and assess whether the incident was solved or not.

The process starts with the Occurrence of an Incident (Depicted as its pre-state) and ends with the Ticket being closed. All lower-level activities compose the higher-level activity (representative of the KiP) and two of the activities are Knowledge-intensive Activities: "Solve Incident" and "Assess Ticket Resolution" (being a Decision i.e. a type of KiA), the latter composing the former, being a sub-activity of Solve Incident.

Furthermore, "Solve Incident" must be characterized as a KiA, with its component Decision "Assess Ticket Resolution". Each has a Communicative Interactions occurring during its execution, respectively: (i) A discussion of the resolution of the incident itself, focusing on a feasible solution for the problem and (ii) A discussion on whether the incident was solved, the client is satisfied and the ticket must be closed.

Fig. 18 depicts the Knowledge-intensive Activity and its main elements.

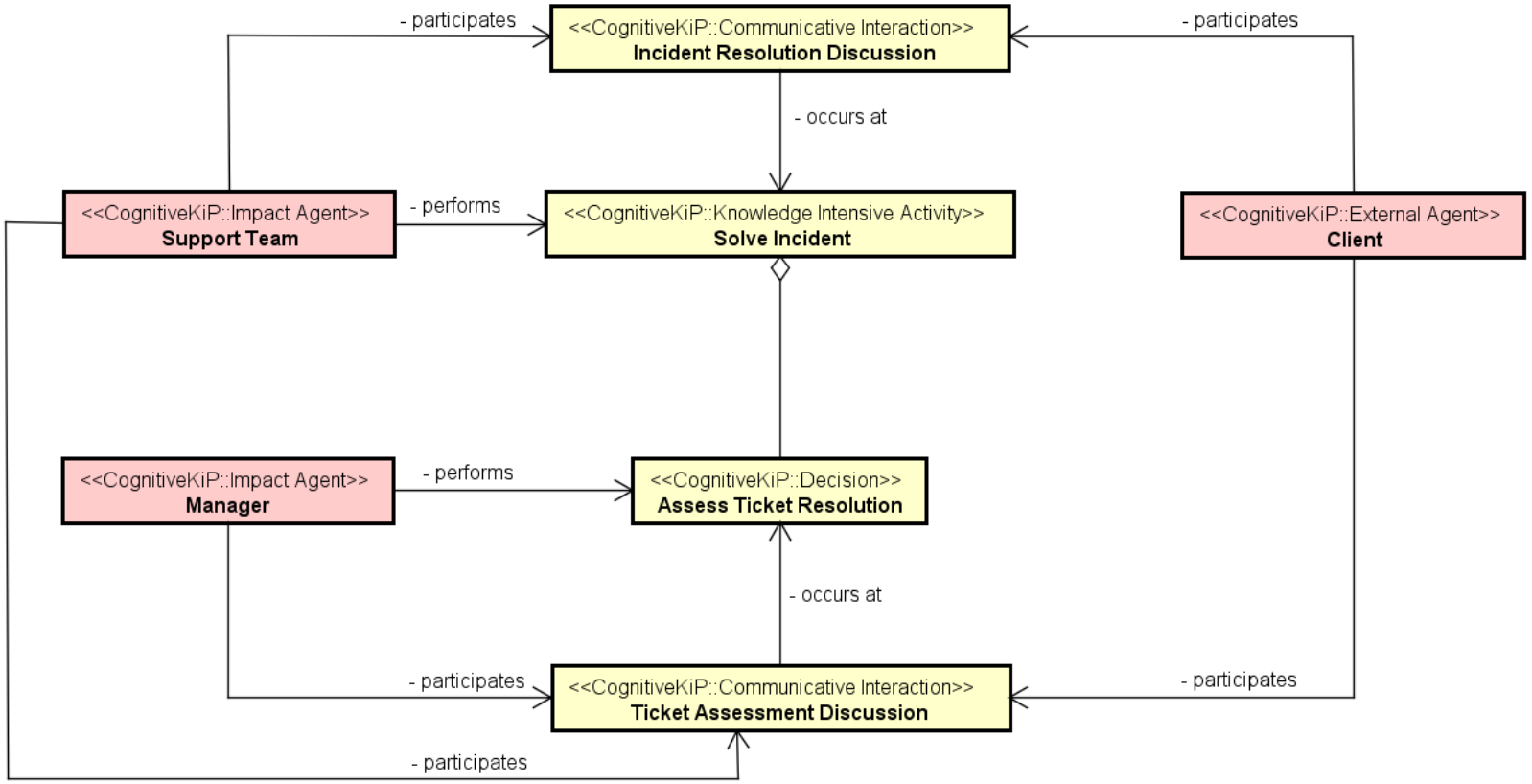

Figure 18 – Solve Incident as a Knowledge-intensive Activity

Focusing on the Decision and Communicative Interaction of "Solve Incident", there are both Common Ground elements, one for each Communicative Interaction. During the Incident Resolution Discussion, there is a Shared Presupposition by all Participants on a consensus for a solution for the incident; during the Ticket Assessment Discussion, one of the Externalized Beliefs composing its Common Ground plays the role of a Support Belief for the "Close Ticket and notify Client" Alternative, and also plays the role of an Unsupport Belief for the "Keep Ticket" Alternative, thus characterizing a Chosen Alternative and a Discarded Alternative, respectively.

Both Alternatives compose the Decision Results of the "Assess Ticket Resolution" activity, depicting all the elements involved in the discussion and interaction between participants for the solution of the incident as well as the Decision involved in the KiA. Fig.19 depicts the model for this scenario.

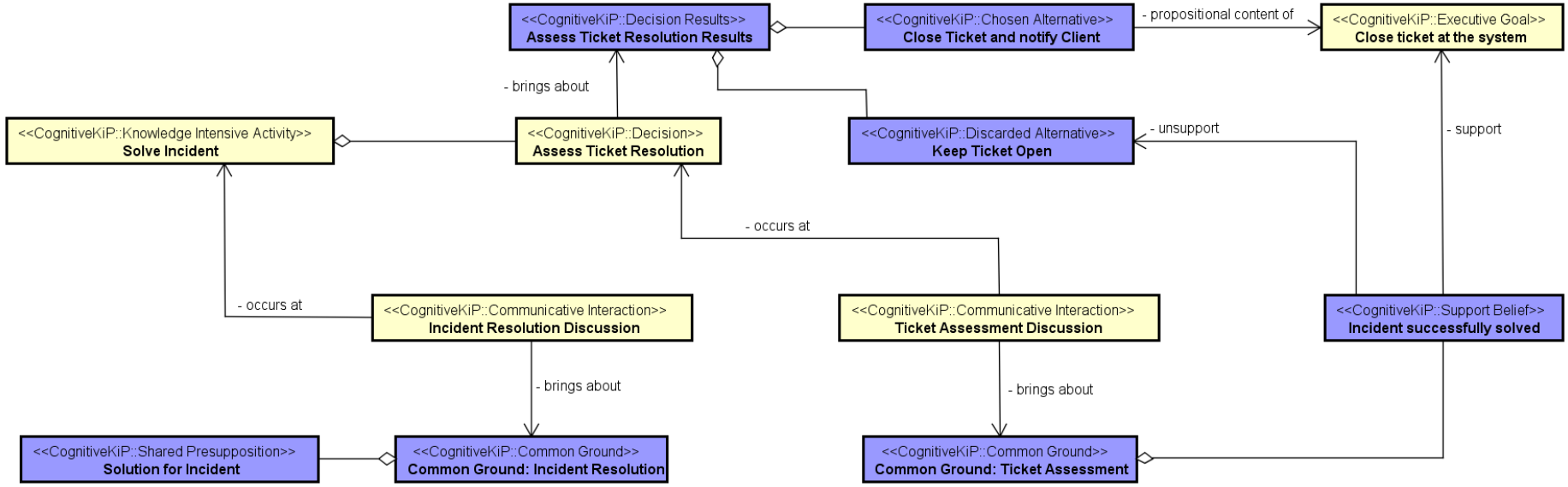

Figure 19 – Communicative Interactions and Decisions (Solve Incident KiA)

After the description of the scenario and the illustration of how it could be modeled using CognitiveKiP, we can now perform a series of analysis on real world data about the process.

A dataset was extracted from the OTRS repository with all tickets labeled as as “Incident" reported in the second semester of 2015. A total of 5,714 tickets were gathered, comprising 25,380 messages exchanged in the system during the troubleshooting process. Each exchanged message was composed by the following fields:

- Ticket identifier,
- Message identifier,
- Date and time of the sent message,
- Name and origin email sender,
- Name(s) and e-mail(s) of email destination, and
- The message itself (translated into English).

A correlation between the dataset fields and CognitiveKiP elements was made, as described below in Table 9:

Table 9 – Dataset field correlation of Scenario 1 with CognitiveKiP concepts

Dataset 1	
Dataset Field	**Corresponding construct in CognitiveKiP**
Ticket_id	Instance of Cognitive::Activity "Solve ICT Incident" (KiP)
Message_id	CognitiveKiP::Message at CognitiveKiP::Communicative Interaction occurring at "Solve ICT Incident
Sender	CognitiveKiP::Sender / Cognitive::Participant
Receiver	CognitiveKiP::Receiver / Cognitive::Participant
Body	Propositional content of CognitiveKiP::Message

The analysis has two specific aims: first, to discover process instances that validate the theory, being typical examples of its application and, second, to discover cases that are questionable and bring evidence to discuss the theory’s boundary and the null hypothesis.

First, a typical ticket was selected, representing a scenario of the theory’s application will be analyzed, where the thesis hypothesis is true. Due to privacy concerns

of the ICT company, we have anonymized the dataset personnel and client information. The Participants involved will be named as: Cecilia (Call Center), Stan (Support Team), Clive (Client) and System (the OTRS system). The log excerpt of this specific incident ticket is presented below in Table 10.

Table 10 – Ticket #165112 data contents

Ticket ID	Msg ID	Sender	Receiver	Body
165112	1	Cecilia	Stan	Dears, good morning! I urgently request to check the problem below: Looking forward. Regards, Cecilia
165112	2	Stan	System	Rated.
165112	3	Stan	System	Rated.
165112	4	Stan	Clive	Mr. Clive, good morning. You can check this problem on the spot, please?
165112	5	Clive	Stan	Good afternoon! already checked and this with the wrong password and is requesting a new password, which I do not have
165112	6	Stan	Clive	What is the e-mail account that the user uses? How many seasons have this account set up?
165112	7	Stan	System	Rated.
165112	8	Stan	System	Moved queue for verification.
165112	9	Stan	Clive	Good morning, Mr. Clive. Could you please check, which account and how many computers receive these emails and tell me the reason why I am addressing

				the issue of migration of Office 365 accounts.
165112	10	Stan	System	Moved queue for treatment.
165112	11	Clive	Stan	The account is clive@client.com.br and there are two computers
165112	12	Stan	Clive	Paul, please configure the account at stations as email that shot to the SPO IT account, please.
165112	13	Stan	System	Owner update.

In the example of Ticket #165112, Message #1 depicts a Directive Speech Act from the Client to the Support Team, expressing a Desire to check the problem/incident. Message #2 and Message #3 are Declarative Speech Acts, sent from the Support Team to the System, in order to modify a Social Object (the incident's ticket itself), changing its priority rate in the System.

Message #4 is another Directive Speech Act, this time from the Support Team to the Client, requesting the client to verify the problem locally and Message #5, the response to the request, is an Assertive Speech Act, representing the perceptions of the client regarding the problem (representative of a Belief, even if it is found to be true in reality). The messages exchange continues until Message #13, with a final Declarative Speech Act, closing the ticket due to the client updating the status of the incident as "solved".

Another interaction was selected from the dataset is depicted below (Ticket #240515), and illustrates a scenario in the boundary of CognitiveKiP. In the first message, we noticed that the Client ("Ursula") made a contact with the support technician ("Stan") prior to the exchange of emails and interaction registered in the System, as the message states the problem found by Ursula through an email sent by Stan to both Ursula and his Manager ("Matt"). Message #2 is Ursula's reply to the message, with a screenshot of the problem, followed by Message #3, where Stan informs the system about the update on

the ticket (the screenshot sent by Ursula) and in Message #4 Stan states that contact was made with Ursula and the problem persists. The ticket then closes, inferring that the incident was solved. This ticket represents a scenario where part of the Communicative Interaction occurred outside the System, by other channels such as conversations at the client's site, telephone or other means of communication. CognitiveKiP is still valid for this case, as the same dynamics of speech acts are occurring during the interactions outside the system's registration. Table 11 shows Ticket #240515 data contents.

Table 11 – Ticket #240515 data contents

Ticket ID	Msg ID	Sender	Receiver	Body
240515	1	Stan	Matt, Ursula	Client Ursula, trying to access the Remote Desktop, clicks on the shortcut of the management system and the program gives an error message.Client will send us a screenshot of the error, so that we can proceed with the work under a new ticket.
240515	2	Ursula	Stan	<Attached image file of the problem>
240515	3	Stan	System	Client Update!
240515	4	Stan	Matthew	Contact was made with client and she informed that the problem persists.

Another illustrative example on how Beliefs, and more importantly, the Shared Presuppositions composing the Common Ground, interfere on the "Solve Incident" KiA is depicted at Table 12.

Table 12 – Ticket #240986 data contents

Ticket ID	Msg ID	Sender	Receiver	Body
240986	1	Darrell	Stan	<Attached Error message screenshot> Regards, Darrell
240986	2	Cecilia	System	Ranked according to client's request.
240986	3	System	Darrell	Dear Darrell, your support request was registered with the subject: "User Darrell – Network problem" Register number at the system is # 2015122210221790. Feel free to reply directly to this message if you need any additional information regarding the request. Thank you very much for the attention. "
240986	4	Cecilia	Stan	Dear Stan, Could you please attend to this request?
240986	5	Stan	Darrell	Dear Darrell, We have tried to contact you through the telephone at your email signature, however without success. As soon as possible please contact us

				to proceed with your request. Regards, Stan.
240986	6	Darrell	Stan	I am sorry. I have a hearing impairment. I only use Whatsapp. I have recalled my network password and everything is fine now. Thank you very much!
240986	7	Stan	Darrell	Dear Darrell, I am sorry for the inconvenience and thank you very much for your reply. If the problem happens again, please contact us. Regards, Stan.

The first message is solely an attachment of a screenshot of the error by the Client ("Darrell"), followed by the Call Center operator ("Cecilia")'s interpretation of the error type from the image and its registration at the System in Message #2 and the automated message from the System to the Client in Message #3.

In Message #4, Cecilia contacts the support technician ("Stan"), authorizing the analysis of the problem. At this point, we have a Manifest Event during this interaction, in which Darrell does not reply to any of Stan's calls. According to usual IT incident protocol, the lack of communication from the client side can lead to the delaying of the service, a reduction of its priority level or even the closing of the ticket as "unable to solve".

Also, in Message #6, the client Darrell finally answers with a compelling reason for its out-of-reach status, a hearing impairment and the fact that he only uses another form of contact (Whatsapp) due to that limitation. Also, the absence of contact was due to the problem already being solved by the client himself. This illustrates clearly how the dynamics of Common Ground presupposition can provide a form of analysis of this kind of atypical scenario and how it would explain the rationale (and the mistake by the support team) of erroneously closing a ticket as unsolved in this situation.

6.7) Scenario 2: Open Source Software Development (GitHub)

For the second scenario of evaluation with real data, the open source software development domain was chosen for our study, as it is a field that necessarily involves different participants exchanging knowledge and a great variability during the execution of each process instance.

Open source Software (OSS) covers a variety of software artefacts including source code, licenses, innovation, ethics, philosophy, social movement, community, culture, governance and organizational engagement. [Franco-Bedoya et al., 2017]. Typically, the developers are primary volunteers. In addition, the software emerges from a loosely coordinated, unsupervised community of developers and other contributors [van Angeren et al., 2011].

OSS projects are typically initiated by an individual or a small group with a specific necessity. This necessity is the motivation for the creation of the OSS project [Uden et al., 2007]. Rather than a single corporate entity owning the software, a sometimes broad community of volunteers determines which contributions are accepted into the source code base and where the OSS project is heading [Riehle, 2007]. The key actors of Open Source Software are the OSS communities, usually gathered around specific software projects. They guarantee the development, support, and maintenance of OSS [Foulonneau et al., 2013]. An OSS community involves organizations and individuals producing and consuming OSS components. There are many roles in an OSS community with different levels of participation e.g., users, reviewers, contributors, administrators, partners, and developers [Squire and Williams, 2012].

One of the most popular OSS platform is GitHub, a web-based, social software development environment that provides source code management, version control, issue tracking and other features. GitHub allows users to set up a public repository that anyone can fork and use for their own code and/or to contribute changes to the code. Three important concepts are critical for its usage: (i) Creating a “Fork”, an operation where the user makes clone or copy of a repository, usually for two reasons: either to reuse the code or as a starting point to contributing back to the original project through a pull request, which can then be merged with the main branch of code; (ii) a Commit, a change of an individual or set of files/folders and (iii) a Pull request, an operation on which code from

one developer (that previously performed a fork, modified the code with different Commits and now have a different version of the software) is contributed back to a GitHub repository.

Among the typical OSS business processes involved in a Github repository, one of the most important is the "Contribute Code to a Software Project" process, involving the software development tasks, as well as sharing the code, comments, viewpoints, bug reports and other forms of knowledge between developers, user and members of the OSS community. The goal to be reached in this process is to contribute successfully with a code for a specific version of an open-source software.

The process can be described by CognitiveKiP as the higher-level Activity of the model. A Conditioned Goal "Successful Code Contribution" is associated with the Activity, being dependent of all the Goals (either Conditioned or Executive Goals) of the other Activities composing the higher-level Activity. There is a clear necessity of communicating your code contributions and receiving feedback (in the form of cognitive aspects such as Beliefs, Desires and Intention) from contributors and users of the software (represented as a Communicative Interaction) and critical decision-making tasks (characterized as Decisions), such as accepting or rejecting an specific functionality to a future version of the software, or deciding whether or not a contribution is sufficiently stable to be merged with the software's repository. All these characteristics enable the description of "Contribute Code to a Software Project" as a Knowledge-intensive Process. Also, there are specific business rules, either pre-defined (e.g. the division of roles and responsibilities between developers, especially at mature projects such as the Linux Kernel) or ad-hoc or implicit (rules of conduct, code conventions, etc.) Moreover, for open source software (OSS) communities, such practices afforded by IT platforms include modularization of software code and collaboration through incremental layering of contributions [Howison and Crowston 2014, Mac-Cormack et al. 2012, Puranam et al. 2014].

For our case study, we adopt the basic GitHub workflow of code contribution to a repository as a base model for the analysis of the process "Contribute Code to a Software Project" and its activities. This model is based on the description of the basic GitHub flow at the official website.

There is a pre-state situation "Remote repository available" that triggers the "Contribute code to project" process and its initial composing activity "Create a new

branch", and a final resulting (post-state) situation "Code merged" that represents the end result of the process as a whole. The code contribution to a repository usually involves a "Pull Request", a set of proposed changes to a repository's code submitted by a user that can be accepted or rejected by a repository's team. An important observation about the nature of a pull request is that it contains an interaction between process participants (a discussion regarding whether the code is accepted or reject and the necessary modifications to be performed as well as a decision regarding the acceptance of the contribution to be merged with the repository's code contents.

We begin by depicting the process as a whole as a higher-level Knowledge-intensive Activity (i.e., the KiP in question) that is going to be analyzed. We have depicted our KiP as an Activity called "Contribute code to Project", being composed by the KiP's sub-processes and activities modeled as an instance of CognitiveKiP::Activity, Three distinct Participants are involved at the process: (i) Contributor, an independent developer that is willing to improve the code and contribute to the repository; (ii) Maintainer, a team member of the repository's project responsible with its version management and code consistency and (iii) User, a person with access to the repository that can give opinions, usage experience reports and viewpoints about the software being developed.

The Contributor is responsible for creating a new branch (i.e. a parallel version of a repository) and modifying the code, either adding new lines of codes and/or editing or removing other parts (defined as a commit or revision at GitHub, being an individual change to a file or set of files) at his local copy of the repository's code and then submitting his modifications to the repository as a pull request. Then, the Maintainer will review and discuss the code with the Contributor until a decision is made whether the Contributor's code is ready to be merged with the repository's code or not. A series of new verifications and tests will be executed until the code is merged and deployed as a new version of the code at the remote repository. Figure 20 models this process using CognitiveKiP concepts, based on the description of the basic GitHub flow at the official website[1].

[1] https://guides.github.com/introduction/flow

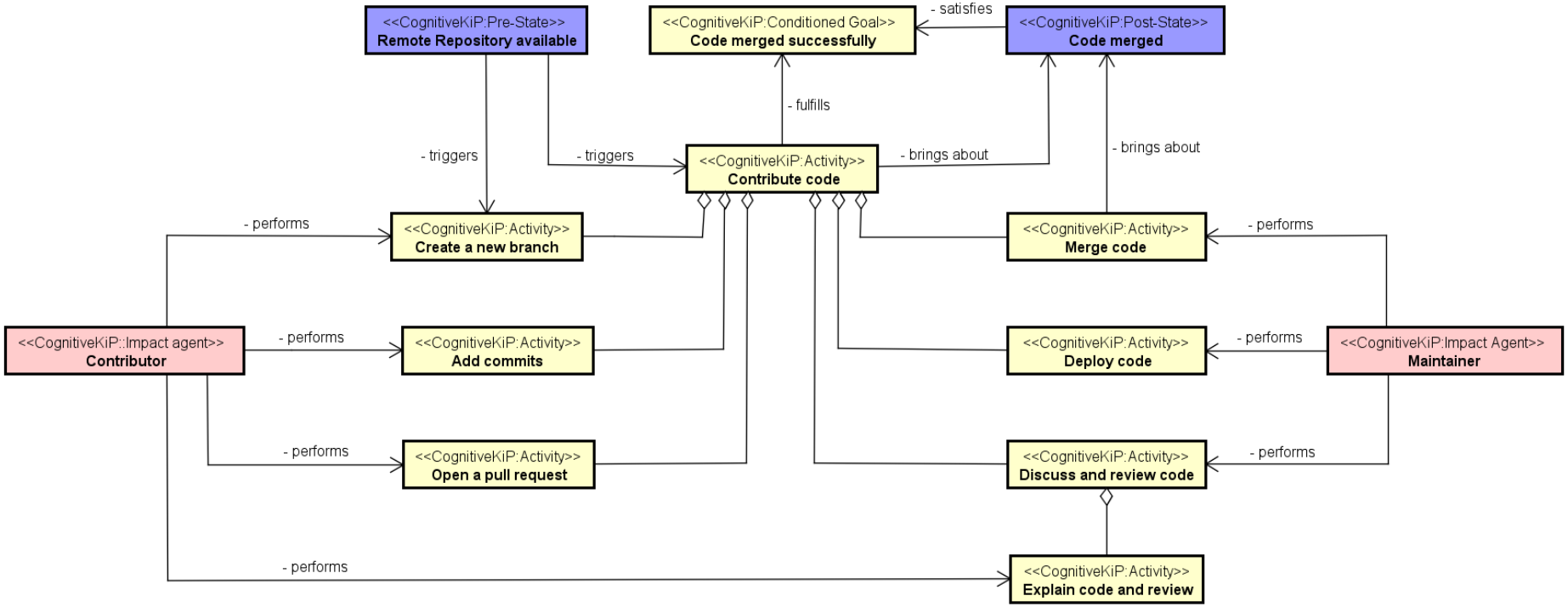

Figure 20 - Contribute code as a Knowledge-Intensive Process

Among the Activities that compose our KiP, we have: (a) “Create a new branch”, (b) “Add commits”, (c) “Open a pull request” and “Explain code and review” being performed by the Contributor. The Maintainer performs the activities “Discuss and review code”, “Deploy code” and “Merge code”, all of them composing directly or indirectly (as in the case of “Explain code and review” as a Task composing “Discuss and review code”) the KiP, represented by the highest level Activity “Contribute with Code”.

Taking the triggering CognitiveKiP pre-states in consideration, we have a flow of activities. We analyze its flow in terms of Situations “triggering” and “being brought about by” different actions and fulfilling distinct goals. Figure 21 depicts the different Activities that compose the KiP “Contribute with Code”.

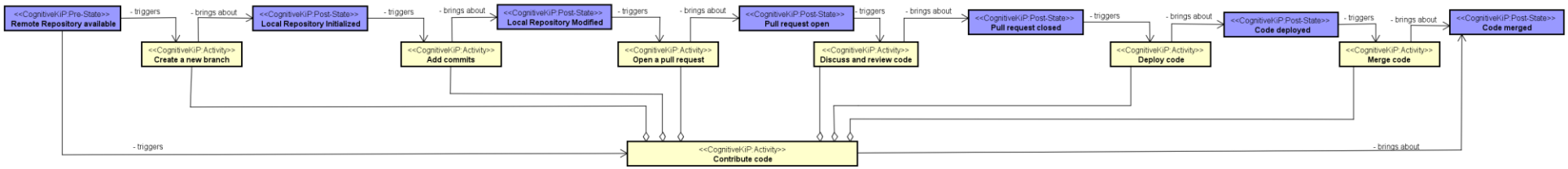

Figure 21 - Activities that compose the KiP "Contribute Code"

By definition, a KiP is composed by at least one KiA, which is necessarily composed by exactly one Communicative Interaction and at least one Decision. We take the Activity “Discuss and review code” as a KiA for our analysis. We argue that it is clearly a Knowledge-intensive Activity due to the occurrence of a Communicative Interaction (the discussion between the Contributor and Maintainer about the code review) and the occurrence of a Decision (the decision-making activity of whether or not to close the pull request at a specific time during the review process). Figure 22 depicts the concepts involved at the “Discuss and review code” KiA.

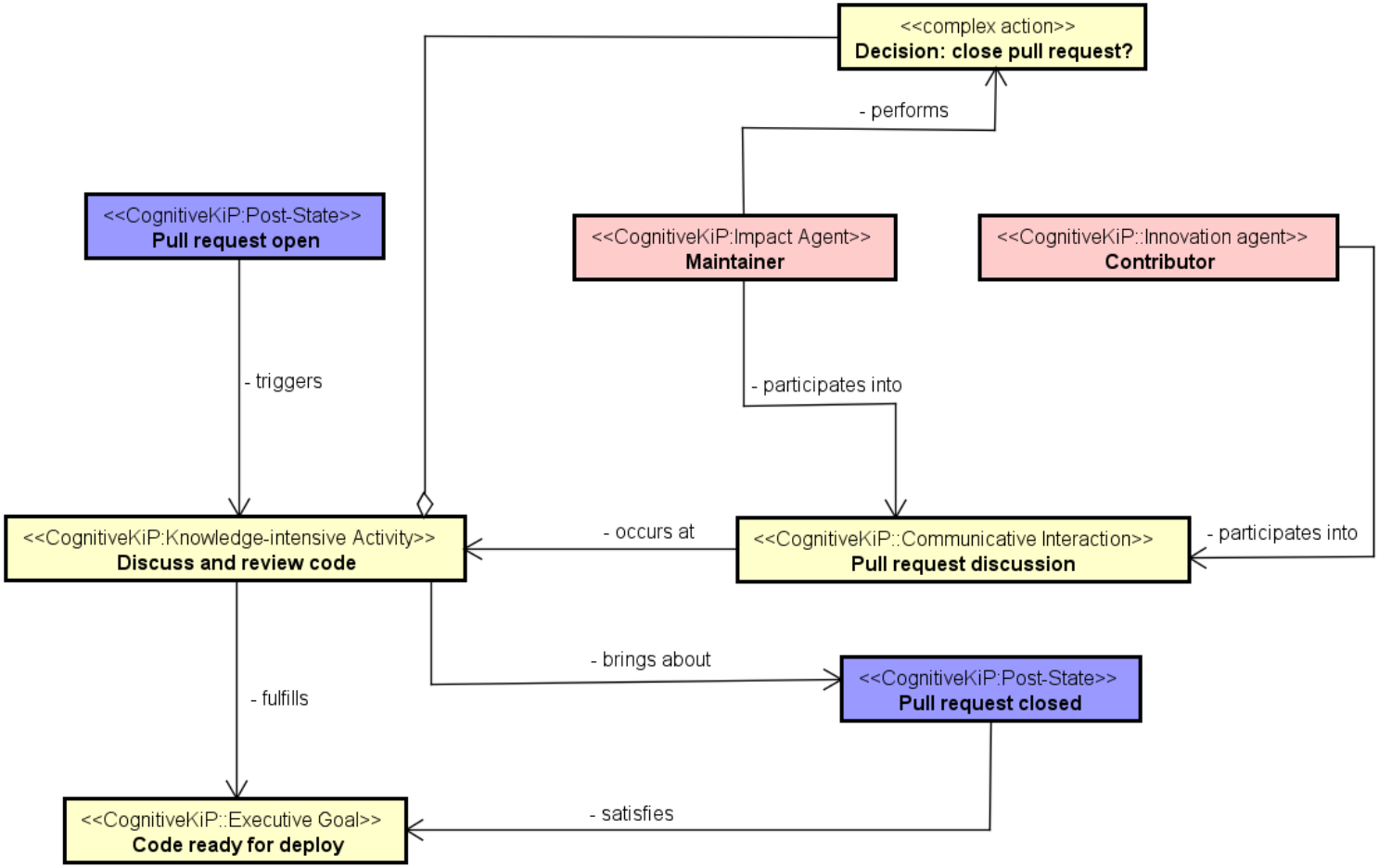

Figure 22 – Knowledge-Intensive Activity "Discuss and review code"

The communicative interaction “Pull request discussion” occurs during the execution of the KiA “Discuss and review code”, involved the exchange of different Speech Acts between Contributor and Maintainer. The exchange of Speech Acts (and therefore of knowledge in the shape of different points of view, beliefs, desires, intentions, etc) modifies the Common Ground of “Pull request discussion”. The set of presupposed beliefs composing the Common Ground (depicted as its composing Externalized Beliefs) are presupposed by both Maintainer and Contributor and forms the base assumptions for making the decision. The chosen course of action (either keeping the pull request open or closing the pull request) will be supported by the Externalized Beliefs that compose the Common Ground as a Goal to be pursued after the Decision, being fulfilled by the Action “Perform closing of pull request”. Figure 23 depicts the dynamics of the pull request decision during a “Contribute with Code” instance.

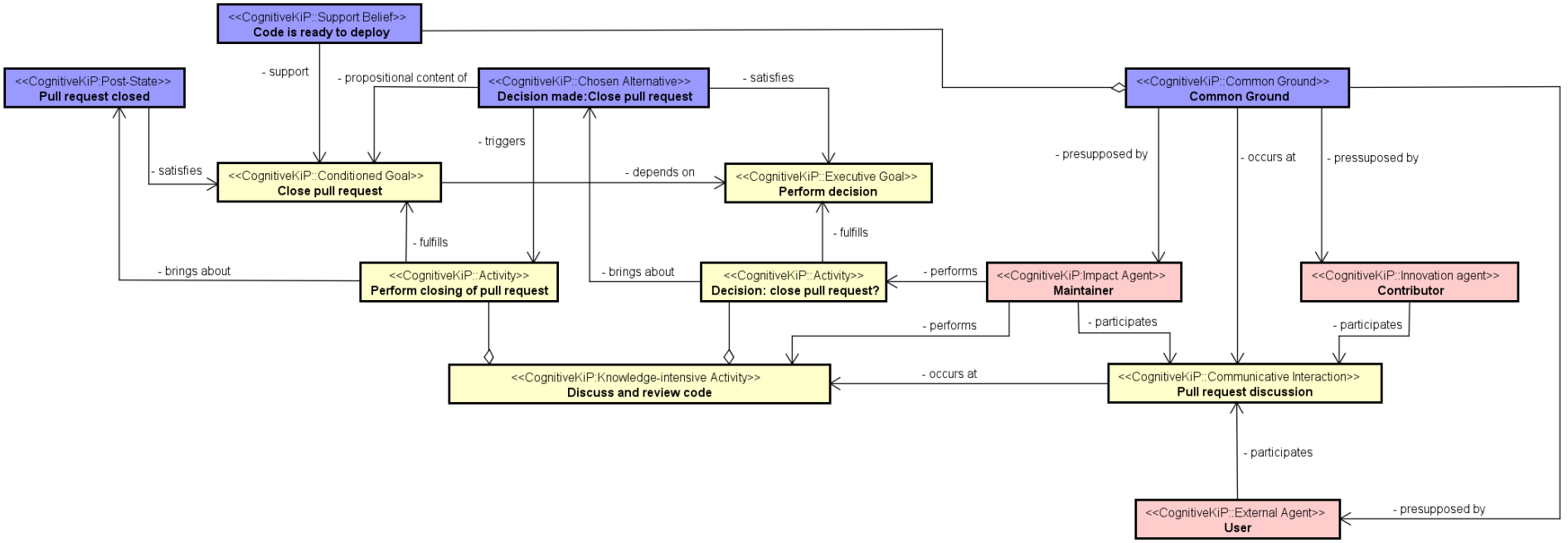

Figure 23 – Decision elements involved at Discuss and Review Code KiA

For the analysis based on real data, the Audacity project[2] Github repository was chosen for our study. The Audacity project is a Free, open source, cross-platform audio software and multi-track audio editor and recorder for Windows, Mac OS X, GNU/Linux and other operating systems and developed by a group of volunteers. We extracted a dataset composed of 231 Closed Pull requests (KiP instances that were finished by the time of extraction) from March, 2015 until March, 2018, with a total of 503 Messages.

A correlation between dataset features and CognitiveKiP elements was applied, as described in Table 13.

Table 13 – Dataset field correlation of Scenario 2 with CognitiveKiP concepts

Scenario 2 Dataset	
Field in the process log	**Corresponding construct in CognitiveKiP**
PullRequestID	Instance of Cognitive::Activity "Solve ICT Incident" (KiP)
MsgID	CognitiveKiP::Message at CognitiveKiP::Communicative Interaction occurring at "Solve ICT Incident"
User_login	CognitiveKiP::Sender / CognitiveKiP::Participant
Body	Propositional content of CognitiveKiP::Message

Beginning the analysis, a typical pull request was selected from the repository. Pull Request #130 (named "Several build failure fixes") was opened on 04/28/2016 by the user "MaxKellermann" with a commit with 4 file modifications to the repository code. Due to the lack of clarity on which bugs the code intended to fix, a Communicative Interaction was started, involving the Contributor ("MaxKellermann") and three Maintainers ("Paul-Licameli", "waliser" and "windinthew"). This pull request interaction involves a higher number of messages than usual and thus illustrates the theory in all its elements. Table 14 show the dataset content for Pull Request #130.

[2] https://github.com/audacity

Table 14 – Pull Request #130 data contents

PullRequestID	MsgID	User_login	Body
130	1	Paul-Licameli	Please explain which build failures you are fixing with each of these commits. Please explain more about how the wxFileNameWrapper kludge causes a crash. I would like to keep that kludge for the compilers that let us get away with it.
130	2	MaxKellermann	Ok, Ill add error messages to the commit messages.
130	3	MaxKellermann	In any case, the wxFileNameWrapper kludge causes severe double free bugs. Valgrind is screaming loudly, and sometimes, even glibc notices heap corruption. Ill post details.
130	4	Paul-Licameli	According to the travis output linked herein, commit X is causing build failures.
130	5	MaxKellermann	Apparently, Travis builds with a very old FFmpeg version which doesnt have the `const` yet. Without that commit, build fails here (FFmpeg 3.0 and 2.8.6).
130	6	Paul-Licameli	I cherry-picked the fix for Track.cpp
130	7	MaxKellermann	So Im using my system FFmpeg, because I hate projects which ship (outdated) copies of other libraries. The FFmpeg version in `lib-src/ffmpeg/` is 2.2.2 Would you agree to update those headers again? (I'd remove them completely, but thats just my opinion.)
130	8	MaxKellermann	I have added a valgrind log to the wxFileNameWrapper commit. Do you need to know anything else?
130	9	Paul-Licameli	I am not qualified to make the decision about FFmpeg. I suggest you ask the question also at the audacity list. Can you figure out conditional compilation that could make the build work with either version?

130	10	MaxKellermann	Ok, wrapped in preprocessor version checks.
130	11	Paul-Licameli	Please see my commit, it implements a different cheat for wxFileNameWrapper which might not crash you.
130	12	MaxKellermann	This will fall apart as well any day. I wouldnt do that. I understand your desire for move operations, but in this case Id wait for WX to support it. Everything else is a kludge thats just waiting to crash (or corrupt data randomly).
130	13	Paul-Licameli	Its ugly either way, but humor me and see if it compiles and runs and lets you load and save a project without apparent trouble.
130	14	MaxKellermann	Compiles and doesnt crash, no valgrind warning.
130	15	Paul-Licameli	Good, I enabled the less evil swap function. It should be safe so long as we do not change version of wxWidgets.This discussion page is still telling me you did not satisfy the Travis build for FFmpeg functions.
130	16	MaxKellermann	Thats because your FFmpeg version numbers are inconsistent!In upstream FFmpeg, the `const` was added in commit https://github.com/FFmpeg/FFmpeg/commit/ec4f04da1 and `version.h` said:```#define LIBAVFORMAT_VERSION_MAJOR 55#define LIBAVFORMAT_VERSION_MINOR 20#define LIBAVFORMAT_VERSION_MICRO 0```Now your `version.h` without the `const` says:```#define LIBAVFORMAT_VERSION_MAJOR 55#define LIBAVFORMAT_VERSION_MINOR 33#define LIBAVFORMAT_VERSION_MICRO 100```.. which is the more recent version number set by commit https://github.com/FFmpeg/FFmpeg/commit/db3c9701f46d20fd7e94c3222cf4fd4524a16414 . The real problem is that FFmpeg applied those two changes in

			different branches, and your copy is a newer branch but without the `const` change.So now Ive changed the minimum version expected for `const` to 55.33.101, one more than Audacitys FFmpeg copy. I hope this covers all relevant versions.
130	17	walisser	What needs to be done to get the FFmpeg patch through?
130	18	windinthew	For a start, the commit has conflicts. And when that is solved, we have to be sure that the currently recommended FFmpeg 2.2.3 maximum is still supported (so e.g. the recommended Windows/Mac FFmpeg downloads at http://manual.audacityteam.org/man/faq_installation_and_plug_ins.html#ffdown still work).Or upgrade Audacity to support later FFmpeg/libav (what range of versions?) Or migrate to gstreamer and use their FFmpeg support rather than hardcoding our own. The last two are major undertakings but I am not an expert in this. I doubt we will want to make any changes now before 2.1.3 release but I suggest taking Pauls recommendation to ask on the -devel list.
130	19	walisser	I was hoping to give Max a chance to do this, if he doesnt I will submit a new pull request and this one can be closed.I cherry picked the FFmpeg commit only, and it doesnt break the current master branch. I dont know about Travis yet. I dont expect any change in behavior (or supported versions) as a result.FWICT this patch is about fixing a build failure since the function prototypes changed in the new headers. But since theyre only adding a const where there wasnt one, that wont break anything (const isnt part of the C calling conventions AFAIK). So functionally, nothing

			has changed, the same symbols are still loaded from the libraries. This can all be verified of course without much effort.
130	20	MaxKellermann	As of yet, I have no feedback whether my changes will be merged. I dont want to waste time on something that will go to the trash can. When I submitted this PR, there was no conflict - the Audacity project ignored my PR for so long, and **after** that merged changed which conflicted with this PR. Yes, I will do you the favor and resolve the conflicts - if you want me to. About the `const`: the function pointers are not compatible if the constness of pointer targets is not the same. This has nothing to do with calling conventions - calling conventions describe how calls are made on the machine language level. Thus, calling conventions have little to do with the C language, they’re a lower level, and semantic API declarations like constness do not matter on that level.
130	21	Paul-Licameli	As I said to Max before, I do not consider myself competent to decide which version of the library should be used in our released binaries for Mac andWindows.However I reexamined commit 41de5b385b2ca721a71871fcdb00ec3fa4441b5band I saw that all changes are conditionally compiled, so that it leavesour source code compatible with use of either version of the library.Therefore I have cherry-picked and commited that. Alone, it was unconflicting.The other commits are not relevant to upgrading the library version. Updates of Makefile.in are periodically done by other people, and simply commit the results of an automated tool, which doesnt demand realprogramming effort. And from discussions earlier

			with Max, I found other changes towxFileNameWrapper that do not give up the performance improvements Iintended, while also fixing the complaints from valgrind. Therefore I do not intend to take those commits. The changes to .gitignore give developers some convenience while not affecting the build at all. No development really depends on them. Therefore I intend to close the commit request, having selected the onemost important part. PRL
130	22	Paul-Licameli	... And the Travis build has succeeded. PRL

In Message #1, the repository maintainer ("Paul Licameli") performs two Directive Speech Acts ("Please explain which build failures you are fixing with each of these commits" and "Please explain more about how the wxFileNameWrapper kludge causes a crash"). In Message #2, the contributor ("MaxKellermann"), responsible for the opening of the pull request, replies to the first message with a Commissive Speech Act ("Ok, I'll add error messages to the commit messages") describing his Intention of adding error messages to the commit messages of his code.

The interaction proceeds with a number of Speech Acts being exchanged, but with a higher frequency of Assertive Speech Acts, depicting different viewpoints about the code (e.g.: "For a start, the commit has conflicts" in Message 18). Even Beliefs about the specific skills necessary for the code's testing are present ("As I said to Max before, I do not consider myself competent to decide which version of the library should be used in our released binaries for Mac and Windows" in Message #9).

The last Message has a Commissive Speech Act involving the closing of the commit request ("Therefore I intend to close the commit request, having selected the one most important part"). In fact, the dataset shows that the commit code was merged to the repository about 10 minutes after the last message and the pull request was closed.

There is an absence of Declarative Speech Acts in this example pull request, due to the fact that the Github platform already performs most of the status changing of data objects involved (such as code, files, bug issues, user status at a repository), discouraging the use of Speech Acts in a pull request interaction for the modification of the Github social reality.

The following cases were considered useful to test the boundaries of the theory. The first, Pull Request #157, was opened by one of the Maintainers ("waliser") with the title "OpenMP-ized SpecCache::Populate" and was a special case, as another Maintainer merged the code with the repository prior to discussing it with his fellow Maintainers. Table 15 shows the dataset contents for Pull Request #157.

Table 15 – Pull Request #157 data contents

PullRequestID	MsgID	User_login	Body
157	1	JamesCrook	Thank You! I have checked this in to master. Very welcome change. Please start a discussion on https://lists.sourceforge.net/lists/listinfo/audacity-devel about the next steps, e.g. compiling for windows.
157	2	Paul-Licameli	James, I have a serious reservation about this. I took a look at this, and I believe it may give incorrect results when the reassignment algorithm is used. It is the outer loops over times that are parallelized, not the inner loopsover the frequency bins. Usually for each time, there is one column ofpixels in the display, and values for that column only in thetwo-dimensional array are updated.But in the case of reassignment, there are time and frequency corrections,which move the contribution of a time/frequency bin of the FFT, adding tisometimes into the cells corresponding to another column. Therefore, there may be a race condition to update those cells, and nothingin this code guards against that. Indeed I am not sure how to prevent theraces in a simple way.
157	3	Paul-Licameli	James has merged the request already, but I have a serious reservation about this. I took a look at this, and I believe it may give incorrect results when the reassignment algorithm is used. It is the outer loops over times that are parallelized, not the inner loops over the frequency bins. Usually for each time, there is one column of pixels in the display, and values for that column only in the two-

			dimensional array are updated. But in the case of reassignment, there are time and frequency corrections, which move the contribution of a time/frequency bin of the FFT, adding ti sometimes into the cells corresponding to another column. Therefore, there may be a race condition to update those cells, and nothing in this code guards against that. Indeed I am not sure how to prevent the races in a simple way.PRL
157	4	Paul-Licameli	Put this another way. Sometimes in parallelizing a loop, you must identify the variables that are accumulators, and give each thread its copy of the accumulation variable, and then have extra work (reduction) after the loop to combine the answers.In the case of the reassignment algorithm, the entire vector freq is the accumulator that should be replicated among the threads, and the reduction step is itself another loop that sumes corresponding array elements into one array.
157	5	JamesCrook	Thanks Paul. Do you have a concern that it might give incorrect (or slower than before) results when not using OpenMP? Im willing to have it in as experimental support for OpenMP. Walisser believes locking isnt needed - and if it is needed is the downside of not having it that we get black pixels or other artifacts in the reassigned spectrogram? People may accept that for speed, if the output is mostly OK? I think we need a concrete example of a bad spectrogram before and after, and then Walisser can work to fix that. I am pretty sure OpenMP will be for the adventurous, not something we offer precompiled or by default, at least for a while.
157	6	Paul-Licameli	I also question the need for thread-local storage. Could not stack-allocated arrays of caches and scratches do, with each thread simply fetching from its own slot in the array?

			Could not std::vector<float> scratch simply be made larger, each thread using a different part of it?I also dont like the introduction of a few naked new and delete when I am working to eliminate those.I also wonder if there might be a smarter way to avoid the replication of WaveTrackCache objects. This could avoid some redundant reads from the disk and make the algorithm even faster. It would require that the loopfor (auto xx = lowerBoundX; xx < upperBoundX; ++xx)would be factored into an outer loop over appropriate column ranges that would each require WaveTrackCache to be populated just once, and then the inner loop over columns, which would be parallelized as now, and could freely use the common cache without contention because none would cause a miss in it that changes its contents.
157	7	Paul-Licameli	James, I saw nothing that makes me doubt the serial correctness.The consequence of the race condition to accumulate sums may be that the sums come out too small. These numbers (after conversion to dB) provide the input to TrackArtist::DrawClipSpectrum, which transforms them into a colored bitmap. So some pixels in the reassigned spectrogram may be colored as less intense than they should be (red when they should be white, magenta when red, etc.) The simplest fix for now might be just to cause only one thread to run, on condition that there is reassignment, but let us have the advantage of OpenMP otherwise. Is there some library call to OpenMP to instruct it to do that?Perhaps TrackArtist::DrawClipSpectrum affords other opportunities to use OpenMP, without any races to worry about (whether with reassignment or not). Perhaps these are less valuable. Scroll with the mouse wheel in the vertical frequency ruler -- I believe that display update involves only these loops, not the

			WaveClip.cpp computations. Maybe performance of this scrolling doesnt need the help.
157	8	Paul-Licameli	I can suggest some test cases.To get the most extreme effect of time reassignment, try generating a click track, or even just making an impulse by generating silence, then pulling one sample up with the draw tool.View that in spectrogram with the longest window. View it with and without reassignment to see the big difference that makes in the display.Then contrast the reassigned picture with the serial and parallel algorithm.I have not tried it. This is mostly likely to expose problems with race conditions. But if you dont see a difference, that is not a proof that there is not a problem.What pattern the problem takes may depend on how OpenMP partitions the loop passes among threads, and I dont have the experience to understand that.
157	9	walisser	I am looking at potential data races now. There are some in BlockFile for example. I missed these because I did not properly instrument with helgrind the first time. There are some data members declared mutable which throws the whole const-correctness assumption out the window.As for OMP partitioning, in the static schedule mode (default, used here), it will evenly divide the work between threads, with the last thread doing a little bit less; e.g. for an updates size of 800 samples wide, each thread does 100 samples in one go.
157	10	Paul-Licameli	Can you explain more about BlockFile and which mutable fields you mean?
157	11	Paul-Licameli	I think there are enough doubts about this, but also enough willingness from walisser to work on them, that this merge should be reverted for now and a better pull request made again later.

157	12	walisser	I agree this should be put on hold until more testing can be done to find and correct races.BlockFile.h contains a few members declared mutable to get around const. For example in PCMAliasBlockFile::ReadData (line PCMAliasBlockFile.cpp:117) it gets written from a const method.
157	13	Paul-Licameli	I dont know helgrind at all, but I think this is irrelevant. I doubt you construct any PCMAliasBlockFile objects in the usual use of Audacity. They happen only in certain circumstances involving imported sound files.
157	14	walisser	I got that from the helgrind trace, so they are constructed in my test case which loads a .aiff file. Basically helgrind is telling us there was a read and write to some memory location from different threads while there was no mutex lock held between them. This does not mean it is an actual race of course, only potential. There could be some other form of race prevention that helgrind cant detect.Here is what Ive distilled from the trace so far (there is a lot more in the trace, mostly from system libraries).https://gist.github.com/walisser/951929d36a998aa8b7209ebd88efdacf
157	15	walisser	You are right about reassignment, definitely race condition there and confirmed with helgrind. However, the effects to my eyes are imperceptible.It seems the probability of the race occurring is low since there are relatively few threads compared to samples (e.g. with 4 threads, a 400 sample width, each thread does 100 sequential sample-windows). You may only see any artifacts near the border of those work groups, the lookahead/lookbehind is at most 0.5s if Im reading it right.The worse case could be scrolling, as there are at most 30 sample-windows in a scroll event (at least on my system), all of which could fall in the 0.5s

			window. I will look at that.If youre interested, here are a few test images from a click track as suggested, comparing single thread vs multi-thread.https://goo.gl/9kIMry

Message #1 (“Thank You! I have checked this in to master. Very welcome change. Please start a discussion on https://lists.sourceforge.net/lists/listinfo/audacity-devel about the next steps, e.g. compiling for windows”) depicts the anomalous situation of a Decision in which an Impact Agent chooses an Alternative, supporting it with a Support Belief of his own (as described in axiom A21) and the Communicative Interaction proceeds with other participants questioning the Decision being made. This dynamic is common in KiP instances and this specific pull request scenario is illustrative of how the theory has explanatory power (the description of what occurred) and predictive power (the detection of the “anomaly” of a decision without a discussion and its possible consequences during the process).

Message #3 (“James has merged the request already, but I have a serious reservation about this. I took a look at this, and I believe it may give incorrect results when the reassignment algorithm is used”) depicts the perception of a Manifest Event (the merging of the pull request by James) and an Unsupport Belief (“I believe it may give incorrect results”) that turns the Chosen Alternative of merging the pull request into a Discarded Alternative (as described in axiom A20) at this point of the process.

It can be perceived that CognitiveKiP still preserves its explanatory power, describing how the Decision’s dynamics will proceed with the interplay of Support and Unsupport Beliefs being expressed by Assertive Speech Acts, and its predictive power, as it predicts that new speech acts will be exchanged until a new Alternative (the propositional content of a Chosen Goal) is achieved, as the previously Chosen Goal was invalidated by the Unsupport Belief expressed at Message #3. At the end of this pull request, a new pull request was opened (#158), a new Communicative Interaction was instantiated for the discussion to proceed and a new Decision was performed to merge the code again, after the required bug fixes were performed.

A pull request with a smaller interaction but a very interesting scenario is Pull Request #70, as depicted below in Table 16:

Table 16 – Pull Request #70 data contents

PullRequestID	MsgID	User_login	Body
70	1	Paul-Licameli	Steve was not satisfied with the completeness of my recent P2 bug fix. I will leave this here.
70	2	SteveDaulton	This still does not work right for me. I've made a quick video to demonstrate the problem. In short, the track spectrogram settings are now causing the Spectrogram Preferences to change:https://drive.google.com/file/d/0Bwz8k7IbIHRyR3NaRHhNc0hhcUU/view?usp=sharing
70	3	JamesCrook	It was 50:50 whether to merge this or close the pull request. In the end I closed it because it was done to meet Steves reservations, and Steve was not OK with it.

Taking into consideration the Decision involving the pull request and the Assertive Speech Act expresses by the three Participants (Paul-Licameli, SteveDaulton and JamesCrook), a conflict of viewpoints is present, as depicted in Messages #1 and #2 (e.g.: "Steve was not satisfied with the completeness of my recent P2 bug fix." and "This still does not work right for me"). The final outcome of the interaction and decision is described in Message #3, as the pull request is closed without merging the code, due to an Unsupport Belief inhered in one of the Maintainers towards the Alternative of "Merging the code of Pull Request #70 to the repository".

This is a scenario that tests the boundary of CognitiveKiP even further, as there is insufficient data into the dataset to take any conclusions. However, a series of possible causes can be raised, and each could be partly described by the Theory: First, a Business Rule of the repository (as some OSS repositories have agreements and guidelines, such

as code conventions that all users must follow) that enforces that every Pull Request merge must be decided as a consensus between all Participants involved or, in a more anomalous case, if there is a form of hierarchy between the Maintainers, in which the Beliefs of user JamesCrook, expressed by Message #3, would have a bigger weight than the Beliefs of the others, in practice overriding the other Maintainers objections and deciding which alternative to pick by himself.

Although, the specific dynamics of different weights between beliefs of different participants (due to hierarchical levels, etc.) at a Decision is not depicted at CognitiveKiP, being a "threshold case" for the theory. However, the case is partly described by the theory and does not invalidate the hypothesis H1 that forms the core of the thesis' proposal.

6.8) Discussion of the scenarios for CognitiveKiP's falseability

The set of scenarios described in Section 6.7 include both typical and threshold cases. A typical case is the one in which CognitiveKiP clearly has explanatory and predictive powers, while a "threshold case" is the one in which the boundary of the theory can be assessed and its falseability can be evaluated.

Taking into account the Null Hypothesis H1', we can state that H1' is False due to the following statements, based on the ICT incidents (Scenario 1) and Pull Requests (Scenario 2) real data.

- There is no performing of a Decision or choosing of a Goal to pursue during a KiP without Beliefs, as they are inhered in a Participant and form the factor responsibility for its selection in the form of Support or Unsupport Beliefs.
- There is no depiction of possible courses of action during a KiP without Desires, as they are the propositional content of Active Goals (Goals not yet chosen to be pursued but already present in the process).
- There is no chosen and actionable course of action without Intentions, as they form the propositional content of a Chosen Alternative, a state-of-affairs that triggers the chosen course of action. This specific course of action is an Activity, and therefore can only be intentionally performed by a Participant towards the fulfillment of his Goals.

Based on the three points above, the real data and the falsity of H1', we state the thesis hypothesis H1 is True and thus the falseability evaluation is finished. The falseability assessment has evidenced that the boundaries of the theory are well-defined and that the cases that are questionable (threshold cases) have been described, in all its possibilities with concepts that fall within the boundaries of the theory.

Chapter 7 - Conclusions and Future Work

This thesis proposes a new theory for Knowledge-intensive Processes, focusing on the Belief, Desire and Intention of the process participants as the core concepts to characterize a KiP, thus providing a precise understanding of its inner workings. This new paradigm for Knowledge-intensive Processes within a Cognitive BPM framework applied theories from Pragmatics and Cognitive Psychology and harness the formal precision of the KIPO Ontology and its foundational ontology, the Unified Foundational Ontology (UFO).

The relationships described between the speech acts being exchanged among Participants during interactions (which represents ways of exchanging representations of mental states) brings up a novel cognitive dimension compared to the traditional methods and notations of business process modeling and analysis. Moreover, the relationship of the mental states with other elements of the process (such as a decision being related to the selection of goals based on shared beliefs between agents) enabled a form of analysis that brings new insights about the nature of KiPs and the best ways of modeling and understanding this particular kind of process.

The proposal of CognitiveKiP - both as an IS theory based on imported theories from different fields of research such as Linguistics, Pragmatics and Cognitive Science - and the theory's materialization as an Ontology enabled the evaluation of its concepts and the theory's boundaries, as discussed during the analysis of the "threshold" scenarios. The usage of a solver (Alloy Analyzer) for the simulation of instances, from the model that was materialized from the theory, was much useful for the evaluation of the consistency of the model constructs as well as visualize possible scenarios of application.

A number of works based on the theory were published, involving fellow researchers that worked together with us. The works serve as the experimentation with different perspectives based on the broader outlook of the CognitiveKiP theory and its ontology. The application of the theory for Process Mining [Ricchetti et al., 2016]; the impact of the speech acts at the decision-making tasks during a KiP [Barboza et al., 2018], as well as other undergraduate students' publications and final projects.

The limitations of the theory are several: (i) the empirical analysis is limited to the gathered data. Further case studies can expand the empirical analysis with new evidence

about how the theory works for real-world scenarios; (ii) Castelfranchi & Paglieri's theory for goal processing has a much detailed taxonomy of Beliefs that was simplified in Support and Unsupport Beliefs for the scope of CognitiveKiP; (iii) The Intentional State of Feeling has its representation and expression described at the CognitiveKiP's dynamics but its impact at the KiP and composing KiAs is not depicted and (iv) The Intentional States that are not expressed as Speech Acts are outside the boundary of the theory as the broad definition of a Speech Act in CognitiveKiP encompasses both spoken and written interactions between participants, as well as synchronous and asynchronous interactions.

Future work includes extending the application of CognitiveKiP as a metamodel for the extraction of speech act patterns during the interactions occurring along the execution of real-world KiP instances, as well as exploring the influence of Feeling in KiPs and its relationships with the other concepts of CognitiveKiP. Finally, the improvement of the CognitiveKiP Ontology using a detailed analysis of its structure and axioms based on theories of ontology patterns and anti-patterns will also be performed, in order to improve the ontology and expand its applications.

References

- Aalst, W. M. P. "Process Mining: Discovery, Conformance and Enhancement of Business Processes", Springer Verlag, 2011.
- Austin, I. L. “How to Do Things with Words”, Oxford: Oxford University Press, 1962.
- Barboza, T., Richetti, P., Baião, F., Santoro F. M., Gonçalves, J. C. A. R., Revoredo, K. “26th International Conference on Cooperative Information Systems”. Valetta, Malta, 2018 (to appear).
- Brander S., Hinkelmann K., Martin A., Thönssen B. “Mining of agile business Processes” In: Proceedings of the AAAI Spring Symposium on AI for Business Agility”, 2011a.
- Brander S., Hinkelmann K., Hu B., Martinz A., Riss, U. V., Thönssen B., Witschel H. F. “Refining Process Models through the Analysis of Informal Work Practice,” In: 9th International Conference on Business Process Management, vol. 6896, pp. 116-131, 2011b.
- Bratman, M. “Intention, plans, and practical reason” Cambridge: Harvard University Press, 1987.
- Cao, L. "Data science and analytics: a new era" International Journal Data Science Analytics, 1 (1), pp. 1–2, 2016.
- Castelfranchi, C. “Commitments: From Individual Intentions to Groups and Organizations,” in Proceedings of the First International Conference on Multiagent Systems, pp. 41–48, 1995.
- Castelfranchi, C. “Reasons: Belief support and goal dynamics”. Mathware & Soft Computing, v.3, pp. 233-247, 1996.
- Castelfranchi, C. “Modelling social action for AI agents”. Artificial Intelligence 103, pp. 157-182, 1998

- Castelfranchi, C., Paglieri F. "The role of beliefs in goal dynamics: Prolegomena to a constructive theory of intentions" Synthese, v.155(2), pp. 237–263, 2007.
- Castelfranchi, C. "Goals, the true center of cognition" In: The goals of cognition. Essays in Honor of Cristiano Castelfranchi, pp. 825-870, 2012.
- Cohen, P. R., Levesque, H. J. "Intention is choice with commitment". Artificial Intel-ligence, 42(2), pp. 213–261, 1990.
- Di Ciccio, C., Marrella, A., Russo, A. "Knowledge-intensive processes: Characteristics, requirements and analysis of contemporary approaches" Journal of Data Semantics, 2014.
- Dretske, F. "Explaining behavior: Reasons in a world of causes". MIT press, 1988.
- Dubin, R. "Theory building (Rev. ed)". London: Free Press, 1979.
- Eppler, M. J., Seifried, P., Roepnack A. "Improving konwledge-intensive processes through an enterprise knowledge medium" Kommunikationsmanagement im Wandel, pp. 371-390, 2008.
- Falbo, R. A., Bertollo, G. "A software process ontology as a common vocabulary about software processes" International Journal of Business Process Integration and Management vol. 4, pp. 239–250, 2009.
- Franco-Bedoya, O., Ameller, D., Costal, D., Franch, X. "Open Source Software Ecosystems: A Systematic Mapping" Information and Software Technology, 91, pp. 160-185, 2017.
- França, J. B. S., Santoro, F. M., Baião, F. A. "Towards characterizing konwledge-intensive processes" In: Proceedings of the IEEE 16th International Conference on Computer Supported Cooperative Work in Design, pp. 23-25, 2012.
- França, J., Netto, J., Carvalho, J., Santoro, F., Baião, F., Pimentel, M. "KiPO: The Knowledge-intensive Process Ontology", Software & Systems Modeling April 2014, Springer, 2014.
- Foulonneau, M., Pawelzik, R., Gregoire, B., Donak, O. "Analyzing the Open Source Communities' Lifecycle with Communication Data". In: Proceedings of the 5th MEDES. ACM, pp. 340–344, 2013.

- GODFREY-SMITH, P. "THEORY AND REALITY: AN INTRODUCTION TO THE PHILOSOPHY OF SCIENCE". UNIVERSITY OF CHICAGO PRESS, 2003.
- GONÇALVES, J. C. A. R., SANTORO, F. M., BAIÃO, F. A. "LET ME TELL YOU A STORY - ON HOW TO BUILD PROCESS MODELS". JOURNAL OF UNIVERSAL COMPUTER SCIENCE, VOL. 17, NO. 2, PP. 276-295, 2011.
- GREGOR, S. "THE NATURE OF THEORY IN INFORMATION SYSTEMS". MIS QUARTERLY, 30(3), PP. 611-642, 2006.
- GRICE, H. P. "LOGIC AND CONVERSATION", IN: SYNTAX AND SEMANTICS, 3: SPEECH ACTS, NEW YORK: ACADEMIC PRESS, PP. 41-58, 1975.
- GRONAU, N, WEBER E. "MANAGEMENT OF KNOWLEDGE INTENSIVE BUSINESS PROCESSES" BUSINESS PROCESS MANAGEMENT, SPRINGER, HEIDELBERG, 2004.
- GUARINO, N. "FORMAL ONTOLOGY, CONCEPTUAL ANALYSIS AND KNOWLEDGE REPRESENTATION", INTERNATIONAL JOURNAL OF HUMAN AND COMPUTER STUDIES, 43 (5/6), PP. 625-640, 1995.
- GUIZZARDI, G. "ONTOLOGICAL FOUNDATIONS FOR STRUCTURAL CONCEPTUAL MODELS," UNIVERSITY OF TWENTE, 2005.
- GUIZZARDI, G., FALBO, R. A., GUIZZARDI, R. S. S. "GROUNDING SOFTWARE DOMAIN ONTOLOGIES IN THE UNIFIED FOUNDATIONAL ONTOLOGY (UFO): THE CASE OF THE ODE SOFTWARE PROCESS ONTOLOGY. IN: PROCEEDINGS OF THE XI IBEROAMERICAN WORKSHOP ON REQUIREMENTS ENGINEERING AND SOFTWARE ENVIRONMENTS, (IDEAS'2008), PP. 244-251, 2008
- GUIZZARDI, G., WAGNER, G., FALBO, A., GUIZZARDI, R. S. S., ALMEIDA, J. P. A. "TOWARDS ONTOLOGICAL FOUNDATIONS FOR THE CONCEPTUAL MODELING OF EVENTS" IN: 32TH INTERNATIONAL CONFERENCE ON ENTERPRISE MODELING (ER 2013), PP. 327– 341, 2013.
- HAGEN, C. R., RATZ, D., POVALEJ, R. "TOWARDS SELF-ORGANIZING KONWLEDGE-INTENSIVE PROCESSES" JOURNAL OF UNIVERSAL KNOWLEDGE MANAGEMENT, V.0, N.2, PP.148-169, 2005.
- HARVEY, M. "MAKING SENSE OF COMMON GROUND: PRESUPPOSITION, JOINT ACTION, AND COOR-DINATION" MASTER OF PHILOSOPHY THESIS. UNIVERSITY OF CAMBRIDGE, 2014.

- HERZIG, A., LORINI, E., PERRUSSEL, L., XIAO, Z.: BDI LOGICS FOR BDI ARCHITECTURES: OLD PROBLEMS, NEW PERSPECTIVES. KUENSTLICHE INTELLIGENZ (TO APPEAR)
- HOWISON J., CROWSTON, K. "COLLABORATION THROUGH OPEN SUPERPOSITION: A THEORY OF THE OPEN SOURCE WAY". MIS QUARTERLY, 38(1), PP. 29–50, 2014.
- HULL, R., MOTAHARI-NEZHAD, H.R.: RETHINKING BPM IN A COGNITIVE WORLD: TRANS-FORMING HOW WE LEARN AND PERFORM BUSINESS PROCESSES. IN: LA ROSA, M., LOOS, P., PASTOR, O. (EDS.) BUSINESS PROCESS MANAGEMENT. BPM 2016. LNCS, VOL. 9850, PP. 3-19. SPRINGER, 2016.
- ISIK, Ö. MERTENS, W., VAN DEN BERGH, J. "PRACTICES OF KONWLEDGE-INTENSIVE PROCESS MANAGEMENT: QUANTITATIVE INSIGHTS" BUSINESS PROCESS MANAGEMENT JOURNAL, 19(3), PP. 515-534, 2013.
- JACKSON, D. "ALLOY: A LIGHTWEIGHT OBJECT MODELLING NOTATION" IN: ACM TRANSACTIONS ON SOFTWARE ENGINEERING AND METHODOLOGY (TOSEM) 11, 2, PP. 256 – 29, 2002.
- LITTLE, T. A., DEOKAR, A. V. "UNDERSTANDING KNOWLEDGE CREATION IN THE CONTEXT OF KNOWLEDGE-INTENSIVE BUSINESS PROCESSES", JOURNAL OF KNOWLEDGE MANAGEMENT, VOL. 20, ISSUE 5, PP. 858 – 879, 2016.
- LIU B. "SENTIMENT ANALYSIS AND OPINION MINING". MORGAN AND CLAYPOOL, 2012.
- MACCORMACK, A., BALDWIN, C., RUSNAK, J. "EXPLORING THE DUALITY BETWEEN PRODUCT AND ORGANIZATIONAL ARCHITECTURES: A TEST OF THE "MIRRORING" HYPOTHESIS". RES. POLICY, 41(8), PP. 1309–1324, 2012.
- MALLE, B. F., MOSES, L. J., BALDWIN, D. A., "INTENTIONS AND INTENTIONALITY". MIT PRESS, 2001.
- MARJANOVIC, O., FREEZE, R. D. "KNOWLEDGE INTENSIVE BUSINESS PROCESSES: THEORETICAL FOUNDATIONS AND RESEARCH CHALLENGES" IN: PROCEEDINGS OF 44TH HAWAII INTERNATIONAL CONFERENCE ON SYSTEMS SCIENCE (HICSS-44 2011), KOLOA, KAUAI, HI, USA, IEEE COMPUTER SOCIETY, PP. 1-10, 2011.
- MOODY, D., IACOB, M.-E., AMRIT, C. "IN SEARCH OF PARADIGMS: IDENTIFYING THE THEORETICAL FOUNDATIONS OF THE IS FIELD," IN: PROCEEDINGS OF THE EUROPEAN CONFERENCE ON INFORMATION SYSTEMS, PRETORIA, SOUTH AFRICA, PP. 1-13, 2010.

- NARDI, J. C., DE ALMEIDA FALBO, R., ALMEIDA, J. P. A., GUIZZARDI, G., PIRES, L. F., VAN SINDEREN, M. J., GUARINO, N., FONSECA, C. M. "A COMMITMENT-BASED REFERENCE ONTOLOGY FOR SERVICES", INFORMATION SYSTEMS, ELSEVIER, 2015.
- OLIVEIRA, F. F. "ONTOLOGY COLLABORATION AND ITS APPLICATIONS. M.SC. DISSERTATION" PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO, VITÓRIA, BRAZIL (IN PORTUGUESE), 2009.
- PEREIRA, A., SANTORO, F. "COGNITIVE DECISION MAKING PROCESS AS CONTEXT INFORMATION" IN: THE 15TH IFIP WG8.3 INTERNATIONAL CONFERENCE ON DECISION SUPPORT SYSTEMS (DSS 2010), LISBOA, PORTUGAL, 2010.
- PURANAM, P., ALEXY, O., REITZIG, M. "WHAT'S 'NEW' ABOUT NEW FORMS OF ORGANIZING?" ACAD. MANAGEMENT REV, 39(2), PP. 162–180, 2014.
- RAO, A. S., GEORGEFF, M. P. "MODELING RATIONAL AGENTS WITHIN A BDI-ARCHITECTURE". IN PROC. OF THE 2ND INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING (KR), PP 473–484, 1991.
- RECKER, J., MENDLING, J. "THE STATE OF THE ART OF BUSINESS PROCESS MANAGEMENT RESEARCH" BUS. INF. SYST. ENG. , 58, PP. 55-72, 2016.
- RICHETTI, P.H.P., GONÇALVES, J.C.A.R., BAIÃO, F.A., SANTORO, F.M. "ANALYSIS OF KNOWLEDGE-INTENSIVE PROCESSES FOCUSED ON THE COMMUNICATION PERSPECTIVE". IN: CARMONA, J., ENGELS, G., KUMAR, A. (EDS.) BPM 2017, BARCELONA, SPAIN, SEPTEMBER 10-15, 2017. LNCS, VOL. 10445, PP. 269–285, 2017.
- RIEHLE, D., "THE ECONOMIC MOTIVATION OF OPEN SOURCE SOFTWARE: STAKEHOLDER PERSPECTIVES". IEEE COMPUTER, 40 (4), PP. 25–32, 2007.
- ROSEMANN M. "POTENTIAL PITFALLS OF PROCESS MODELING: PART A," BUSINESS PROCESS MANAGEMENT JOURNAL, VOL. 12, NO. 2,PP. 249–254, 2006.
- STALNAKER, R. "CONTEXT". OXFORD UNIVERSITY PRESS, 2014.
- STALNAKER, R. "COMMON GROUND". LINGUISTICS AND PHILOSOPHY 25 (5-6), PP. 701-21, 2002.
- SEARLE, J. R., VANDERVEKEN, D. "FOUNDATIONS OF ILLOCUTIONARY LOGIC" CAMBRIDGE UNIVERSITY PRESS, 1985.

- SEARLE, J. R. INTENTIONALITY: AN ESSAY IN THE PHILOSOPHY OF MIND. CAMBRIDGE: CAMBRIDGE UNIVERSITY PRESS, 1983.
- SEARLE, J. R., "THE CONSTRUCTION OF SOCIAL REALITY", THE PENGUIN PRESS, LONDON, 1995.
- SEARLE, J. P. "MIND: A BRIEF INTRODUCTION", CAMBRIDGE UNIVERSITY PRESS, 2004.
- SOARES D., SANTORO F. M., BAIÃO F. A., "DISCOVERING COLLABORATIVE KNOWLEDGE-INTENSIVE PROCESSES THROUGH E-MAIL MINING", JOURNAL OF NETWORK AND COMPUTER APPLICATIONS, VOL. 36, NO. 6, PP.1451-1465, 2013.
- SQUIRE, M., WILLIAMS, D. "DESCRIBING THE SOFTWARE FORGE ECOSYSTEM". IN: PROCEEDINGS OF THE 45TH HICSS. IEEE, PP. 3416–3425, 2012.
- TAYLOR, S., LLOYD, V., RUDD, C. "ITIL: SERVICE DESIGN", NORWITH, U.K.:THE STATIONERY OFFICE, 2007.
- UDEN, L., DAMIANI, E., GIANINI, G., CERAVOLO, P. "ACTIVITY THEORY FOR OSS ECOSYSTEMS". IN: PROCEEDINGS OF THE DEST. INAUGURAL IEEE-IES, PP. 223–228, 2007.
- UNGER, M., LEOPOLD, H., MENDLING J. "HOW MUCH FLEXIBILITY IS GOOD FOR KNOWLEDGE INTENSIVE BUSINESS PROCESSES: A STUDY OF THE EFFECTS OF INFORMAL WORK PRACTICES", HICSS 2015, PP. 4990-4999, 2015.
- VACULIN R., HULL R., HEATH T., COCHRAN C., NIGAM A., SUKAVIRIYA P. "DECLARATIVE BUSINESS ARTIFACT CENTRIC MODELING OF DECISION AND KNOWLEDGE-INTENSIVE BUSINESS PROCESSES". IN: THE FIFTEENTH IEEE INTERNATIONAL ENTERPRISE COMPUTING CONFERENCE (EDOC 2011), PP. 151–160, 2011.
- VAN ANGEREN, J., KABBEDIJK, J., JANSEN, S., POPP, K. "A SURVEY OF ASSOCIATE MODELS USED WITHIN LARGE SOFTWARE ECOSYSTEMS. IN: PROCEEDINGS OF THE 3RD CEUR-WS. PP. 27–39, 2011.
- WEBER, R. "EVALUATING AND DEVELOPING THEORIES IN THE INFORMATION SYSTEMS DISCIPLINE," JOURNAL OF THE ASSOCIATION FOR INFORMATION
- SYSTEMS (13:1), PP. 1-30, 2012.
- WESKE, M. "BUSINESS PROCESS MANAGEMENT: CONCEPTS, LANGUAGES, ARCHITECTURES", 1ST ED., SPRINGER, 2007.

- WITSCHEL H. F., HU B., RISS U.V., THÖNSSEN B., BRUN R., MARTIN A., HINKELMANN K. “A COLLABORATIVE APPROACH TO MATURING PROCESS-RELATED KNOWLEDGE”, BUSINESS PROCESS MANAGEMENT, PAGES 343–358, BERLIN, HEIDELBERG, 2010.
- VON FINTEL, K. “WHAT IS PRESUPPOSITION ACCOMMODATION AGAIN?” PHILOSOPHICAL PERSPEC-TIVES, 22(1), PP. 137-170, 2008.
- YALCIN, S. "BELIEF AS QUESTION-SENSITIVE." PHILOSOPHY AND PHENOMENOLOGICAL RESEARCH, 2016.

ANNEX A – AUDACITY REPOSITORY LOG EXCERPT

PullRequest ID	Type	User_login	Role	Title	Body
115	Title	Paul-Licameli	MEMBER	Fix two effects that declared parameters but did not use them...	... Travis build warnings discovered this problem for me.
115	Comment	lllucius	CONTRIBUTOR		There was actually a reason those were not verified...they arent part of the saved effects settings.
115	Comment	Paul-Licameli	MEMBER		You refer to Amplify or Equalization or both?PRLOn Wed, Mar 2, 2016 at 1:10 AM, Leland Lucius notifications@github.comwrote:> There was actually a reason those were not verified...they arent part of> the saved effects settings.> > —>

					Reply to this email directly or view it on GitHub> https://github.com/audacity/audacity/pull/115#issuecomment-191083274.
115	Comment	lllucius	CONTRIBUTOR		Both. I actually did include the amp value when I first converted the effects, but it was determined that it didnt make any sense to save it. The EQ fellas are basically in the same boat.
115	Comment	Paul-Licameli	MEMBER		Why does it not make sense for Amplify?If I use Amplify, then use it again in-session, the last-used amplificationfactor is forgotten.On Wed, Mar 2, 2016 at 3:11 AM, Leland Lucius notifications@github.comwrote:> Both. I actually did include the amp

					value when I first converted the> effects, but it was determined that it didnt make any sense to save it.> The EQ fellas are basically in the same boat.> > —> Reply to this email directly or view it on GitHub> https://github.com/audacity/audacity/pull/115#issuecomment-191121255.
115	Comment	lllucius	CONTRIBUTOR		Unfortunately, I dont recall all of the details, but it was all discussed back in April/May of last year.However, there may have been further changes since then that have broken this or that we didnt get it right back then.
115	Comment	lllucius	CONTRIBUTOR		I believe it was because mAmp isnt really used

					when the effect is applied. Only the ratio is used.
115	Comment	JamesCrook	OWNER		Amplification is supposed to be recalculated based on the actual loudness of the audio selected. This pull request has been around since March. I figure if Paul would have committed it, and still can even with me closing the pull request, as he has commit rights, if he thought it was OK.
116	Title	ThomasFeher	CONTRIBUTOR	fix -Wmissing-field-initializer (3 occurences)	
116	Comment	Paul-Licameli	MEMBER		Reviewed by me
117	Title	yurchor	CONTRIBUTOR	Update Ukrainian translation	Tested with msgfmt -vc, synced with code.Many thanks for merging.

118	Title	tip2tail	CONTRIBUTOR	Timer Recording Enhancements	To allow Automatic Save and Export of a recording. Also allows for additional options to be carried out after a successful timer recording such as Exit, Retsrat and Shutdown.Development by Mark Young over multiple commits squashed into one.
118	Comment	JamesCrook	OWNER		These are now in the main Audacity GitHub repo. Thanks Mark.
119	Title	tip2tail	CONTRIBUTOR	Timer Recording Disk Space Warning	As discussed on development mailing list
12	Title	SteveDaulton	MEMBER	Fix for bug 683	Also adds m prefix to some member variables for consistency.
12	Comment	Paul-Licameli	MEMBER		Steve, James has cleaned up the recent tangle in repository history, but it is still in the

					ancestry of your commit, which is why I see 36 commits above, many of them extraneous. Can you rebase your changes on the new cleaned up master branch?
12	Comment	SteveDaulton	MEMBER		Are there instructions for how to do that?
12	Comment	Paul-Licameli	MEMBER		I am not exactly sure how to advise Steve to do rebase in detail, because Ihave not succeeded with it yet. We are all learning this...I think what you want to do is make sure you have donegit remote add audacity git@github.com:audacity/audacity.gitgit fetch audacity/masterThat mirrors the cleaned up branch for you locally. And then, to take

					justyour last three commits that I see in your master,git checkout mastergit rebase --onto remotes/audacity/master HEAD~3(I am not sure whether you really need the remotes/ part)If that fails, you may need git rebase --abort to get back to a good state,but if it works, you may need -f to push to your master and lose the oldtangled up master:git push -f origin masterReference:http://git-scm.com/docs/git-rebaseOn Tue, Apr 7, 2015 at 6:44 PM, Steve Daulton notifications@github.comwrote:> Are there instructions for how to do that?> > —> Reply to this email directly or view it on GitHub>

					https://github.com/audacity/audacity/pull/12#issuecomment-90751764.
12	Comment	lllucius	CONTRIBUTOR		Be careful though. Somewhere along the line I wound up rebasing my entire repo and lost the 3 branches I had there. Wasnt a biggie...I just recreated them (well, all but 1 so far).
12	Comment	Paul-Licameli	MEMBER		I understand git reflog may help you rediscover lost branches, ordangling commits, which are not reachable from any branch head.On Tue, Apr 7, 2015 at 11:20 PM, Leland Lucius notifications@github.comwrote:> Be careful though. Somewhere along the line I wound up rebasing my entire>

					repo and lost the 3 branches I had there. Wasnt a biggie...I just> recreated them (well, all but 1 so far).> > —> Reply to this email directly or view it on GitHub> https://github.com/audacity/audacity/pull/12#issuecomment-90792414.
12	Comment	Paul-Licameli	MEMBER		An alternative to rebase is the cherry-pick command which lets youre-apply one diff at a time elsewhere in the graph.On Wed, Apr 8, 2015 at 12:18 AM, Paul Licameli paul.licameli@gmail.comwrote:> I understand git reflog may help you rediscover lost branches, or> dangling commits, which are not reachable from any

					branch head.> > On Tue, Apr 7, 2015 at 11:20 PM, Leland Lucius notifications@github.com> wrote:> > > Be careful though. Somewhere along the line I wound up rebasing my entire> > repo and lost the 3 branches I had there. Wasnt a biggie...I just> > recreated them (well, all but 1 so far).> > > > —> > Reply to this email directly or view it on GitHub> > https://github.com/audacity/audacity/pull/12#issuecomment-90792414.
12	Comment	lllucius	CONTRIBUTOR		Well, both of those options would have been handy. :-) But, I just went with recreating the branches as it was pretty easy to do.Thanks,Leland

12	Comment	SteveDaulton	MEMBER		On 8 April 2015 at 04:20, Leland Lucius notifications@github.com wrote:> Be careful though. Somewhere along the line I wound up rebasing my entire> repo and lost the 3 branches I had there. Wasnt a biggie...I just> recreated them (well, all but 1 so far).> > Thanks for the warning but its hard to be careful when I dont know what> Im doing :-)What happened here?Was it that:The audacity/audacity repository became messed up, causing all other forksto be messed up, and now the audacity/audacity master has been cleaned,commits from forks will

					mess it up again?Would it be better if I deleted my fork and used SVN for now?Steve> —> Reply to this email directly or view it on GitHub> https://github.com/audacity/audacity/pull/12#issuecomment-90792414.
12	Comment	lllucius	CONTRIBUTOR		If its easy to redo your changes, then do what I did. It was an accident for me, but it worked. :-) You could delete your fork:https://help.github.com/articles/deleting-a-repository/Then just fork again and reapply your change.
12	Comment	SteveDaulton	MEMBER		Ive deleted my repository and will start again.
12	Comment	Paul-Licameli	MEMBER		Sorry for the grief.The problem was that

					audacity/master got messy, and the mess was revertedand replaced with equivalent code changes but a neater history. We need tobase further changes on the new history to avoid merging the messy partsback in again.git rebase is supposed to make that sort of thing easy -- picking up abranch you have not yet pushed, and moving it elsewhere in the graph. Imade my first successful use of that command today.On Wed, Apr 8, 2015 at 8:08 AM, Steve Daulton notifications@github.comwrote:> Ive deleted my repository and will start again.> > —> Reply to this email directly or view it

					on GitHub> https://github.com/audacity/audacity/pull/12#issuecomment-90893711.
12	Comment	lllucius	CONTRIBUTOR		Oh no...not a big deal. The more we play/learn at the beginning, the better off well be later.
12	Comment	Paul-Licameli	MEMBER		Heres another discovery: if I rebase a banch that I already pushed to myremote, public sandbox (not to audacity), then I need the -f (force) optionto push it again. Else it complains that the branch on the remote is notupstream of the branch I am pushing.On Wed, Apr 8, 2015 at 10:41 AM, Leland Lucius notifications@github.comwrote:> Oh no...not a big deal.

					The more we play/learn at the beginning, the> better off well be later.> > —> Reply to this email directly or view it on GitHub> https://github.com/audacity/audacity/pull/12#issuecomment-90936390.
12	Comment	JamesCrook	OWNER		Paul, yes. After a rebase you have rewritten history and a push of the local branch that is tracking a remote branch will need a -f. We should avoid doing push -f to audacity/audacity as that could/will overwrite changes already made by someone else. In your own repo it is just fine.Also if you have done a push -f, and then make further changes on your branch and

					push again, you dont need a -f the second time (unless youve also rebased again). The history matches, so there is no rewriting of history to do.
120	Title	tip2tail	CONTRIBUTOR	Further Timer Recording Enhancements	Introduce further enhancement of the Timer Recording process:- Disk space warning if the recording potentially will not fit in disk space available.- ProgressDialog enhancements that allow the Stop/Cancel button to be confirmed and the elapsed time to be hidden.- Messages enhanced to clearly show the actions being taken.
120	Comment	JamesCrook	OWNER		Thanks. I rebased this off master and pushed to master. Mark, you will now need

					to branch from master to make further changes to timer record.
121	Title	ThomasFeher	CONTRIBUTOR	Fix unused parameter warnings	Fixes all occurences of unused parameter warnings in src folder.Ignores warnings in lib-src folder.Remaining warnings of unused parameter are due to usage of macros that conditionally use those parameters, depending on whether being build in debug or release mode.
121	Comment	JamesCrook	OWNER		Thanks. Ive updated and applied these.Note that: wxUSE_ACCESSIBILITYMeant that some parameters WERE used in other builds.
122	Title	tip2tail	CONTRIBUTOR	Timer Record: Ensure that Stop/Cance	As per discussion on audacity-devel

				l in progress dialog always works	
123	Title	ShanghaiTimes	NONE	First pull request	First pull request of VS2015
123	Comment	JamesCrook	OWNER		Closed. We use VS2013 and will for release of 2.1.3, so we cant accept a pull request directly in to master - which we use to build releases. There is some interest in the team in updating the toolchain. Discussion on audacity-devel email list is needed. You need to talk with the people there who want to update the toolchain.
123	Comment	Paul-Licameli	MEMBER		Will we move to VS2015 at some time?The implementation of C++11 on VS2013 is incompete in a

					few things. (Noconstexpr, no nothrow, no inherited constructors, and some other things.) But not yet in anything I find most important.PRLOn Thu, Apr 21, 2016 at 7:43 AM, James Crook notifications@github.comwrote:> Closed #123 https://github.com/audacity/audacity/pull/123.> > —> You are receiving this because you are subscribed to this thread.> Reply to this email directly or view it on GitHub> https://github.com/audacity/audacity/pull/123#event-636027729
123	Comment	JamesCrook	OWNER		We might skip VS2015 entirely and jump to a later version. Best

					discussed on audacity-devel.
125	Title	henricj	CONTRIBUTOR	VS2013 needs <functional> for mem_fun_ref	The VS2013 build is broken with an undefined mem_fun_ref. It needs <functional>.
125	Comment	Paul-Licameli	MEMBER		Sorry, thanks for the alert, it built all right on Mac. I fixed itmyself. Please close the pull request.PRLOn Sun, Apr 24, 2016 at 8:26 PM, Henric Jungheim notifications@github.comwrote:> ## The VS2013 build is broken with an undefined mem_fun_ref. It needs .> > You can view, comment on, or merge this pull request online at:> > https://github.com/audacity/audacity/pull/125> Commit Summary> - VS2013 needs

					<functional> for mem_fun_ref> > File Changes> - _M_ src/tracks/ui/Scrubbing.cpp> https://github.com/audacity/audacity/pull/125/files#diff-0 (1)> > Patch Links:> - https://github.com/audacity/audacity/pull/125.patch> - https://github.com/audacity/audacity/pull/125.diff> > —> You are receiving this because you are subscribed to this thread.> Reply to this email directly or view it on GitHub> https://github.com/audacity/audacity/pull/125
125	Comment	henricj	CONTRIBUTOR		Already fixed.
126	Title	henricj	CONTRIBUTOR	Ignore Visual Studio	Visual Studio creates .vcxproj.user files to store local, per-

				.vcxproj.user files.	user settings. These are not under source control, but do clutter the git status.
129	Title	trebmuh	CONTRIBUTOR	Adds FR comment to audacity.desktop.in	... and remove unnecessary duplication of the name (no needs to override the original name if it doesnt change when translated).
129	Comment	trebmuh	CONTRIBUTOR		ping ?
129	Comment	trebmuh	CONTRIBUTOR		Ping, any issue with this pull request maybe ?
129	Comment	trebmuh	CONTRIBUTOR		ping ?
129	Comment	windinthew	CONTRIBUTOR		This seems a good change and in accordance with the Desktop Entry Specification at https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html .Anyone think not?
129	Comment	windinthew	CONTRIBUTOR		Committed at https://github.com/audacity/audacity/commit/145a54d .

129	Comment	trebmuh	CONTRIBUTOR		Sweet, thanks.
13	Title	lllucius	CONTRIBUTOR	This will change the autosave file to a binary representation of the XML	file. By doing this, all of the processing required to convert values toa textual format (float, doubles, int, strings) is now moved to recoveryinstead. This gets it out of the autosave path and improves responsivenessquite a bit.Edits in large projects are considerably faster after this change.
130	Title	MaxKellermann	CONTRIBUTOR	Several build failure fixes	
130	Comment	Paul-Licameli	MEMBER		Please explain which build failures you are fixing with each of these commits.Please explain more about how the wxFileNameWrapper kludge causes a

					crash. I would like to keep that kludge for the compilers that let us get away with it.
130	Comment	MaxKellermann	CONTRIBUTOR		Ok, Ill add error messages to the commit messages.
130	Comment	MaxKellermann	CONTRIBUTOR		In any case, the wxFileNameWrapper kludge causes severe double free bugs. Valgrind is screaming loudly, and sometimes, even glibc notices heap corruption. Ill post details.
130	Comment	Paul-Licameli	MEMBER		According to the travis output linked herein, commit 7acc599cd7999020bec9d92f9a99cbc55429b863 is causing build failures.
130	Comment	MaxKellermann	CONTRIBUTOR		Apparently, Travis builds with a very old FFmpeg version which

					doesnt have the `const` yet. Without that commit, build fails here (FFmpeg 3.0 and 2.8.6).
130	Comment	Paul-Licameli	MEMBER		I cherry-picked the fix for Track.cpp
130	Comment	MaxKellermann	CONTRIBUTOR		So Im using my system FFmpeg, because I hate projects which ship (outdated) copies of other libraries. The FFmpeg version in `lib-src/ffmpeg/` is 2.2.2 according to 14b47b46ab51f31f43f73017aac55e2789aaa96e. Would you agree to update those headers again? (Id remove them completely, but thats just my opinion.)
130	Comment	MaxKellermann	CONTRIBUTOR		Ive added a valgrind log to the wxFileNameWrapper commit. Do you need to know anything else?

130	Comment	Paul-Licameli	MEMBER		I am not qualified to make the decision about FFmpeg. I suggest you ask the question also at audacity-devel@lists.sourceforge.netCan you figure out conditional compilation that could make the build work with either version?
130	Comment	MaxKellermann	CONTRIBUTOR		Ok, wrapped in preprocessor version checks.
130	Comment	Paul-Licameli	MEMBER		Please see my commit https://github.com/audacity/audacity/commit/e09f62031156dbe792d98a161eaf7267095e94a5which implements a different cheat for wxFileNameWrapper which might not crash you.
130	Comment	MaxKellermann	CONTRIBUTOR		This will fall apart as well any day. I wouldnt do that. I understand

					your desire for move operations, but in this case Id wait for WX to support it. Everything else is a kludge thats just waiting to crash (or corrupt data randomly).
130	Comment	Paul-Licameli	MEMBER		Its ugly either way, but humor me and see if it compiles and runs and lets you load and save a project without apparent trouble.
130	Comment	MaxKellermann	CONTRIBUTOR		Compiles and doesnt crash, no valgrind warning.
130	Comment	Paul-Licameli	MEMBER		Good, I enabled the less evil swap function. It should be safe so long as we do not change version of wxWidgets.This discussion page is still telling me you did not satisfy the Travis build for FFmpeg functions.

130	Comment	MaxKellermann	CONTRIBUTOR		Thats because your FFmpeg version numbers are inconsistent!In upstream FFmpeg, the `const` was added in commit https://github.com/FFmpeg/FFmpeg/commit/ec4f04da1 and `version.h` said:```#define LIBAVFORMAT_VERSION_MAJOR 55#define LIBAVFORMAT_VERSION_MINOR 20#define LIBAVFORMAT_VERSION_MICRO 0```Now your `version.h` without the `const` says:```#define LIBAVFORMAT_VERSION_MAJOR 55#define LIBAVFORMAT_VERSION_MINOR 33#define LIBAVFORMAT_VERSION_MICRO 100```.. which is

					the more recent version number set by commit https://github.com/FFmpeg/FFmpeg/commit/db3c9701f46d20fd7e94c3222cf4fd4524a16414 . The real problem is that FFmpeg applied those two changes in different branches, and your copy is a newer branch but without the `const` change.So now Ive changed the minimum version expected for `const` to 55.33.101, one more than Audacitys FFmpeg copy. I hope this covers all relevant versions.
130	Comment	walisser	CONTRIBUTOR		What needs to be done to get the FFmpeg patch through?
130	Comment	windinthew	CONTRIBUTOR		For a start, the commit has conflicts. And when

					that is solved, we have to be sure that the currently recommended FFmpeg 2.2.3 maximum is still supported (so e.g. the recommended Windows/Mac FFmpeg downloads at http://manual.audacityteam.org/man/faq_installation_and_plug_ins.html#ffdown still work).Or upgrade Audacity to support later FFmpeg/libav (what range of versions?)Or migrate to gstreamer and use their FFmpeg support rather than hardcoding our own. The last two are major undertakings but I am not an expert in this. I doubt we will want to make any changes now before

					2.1.3 release but I suggest taking Pauls recommendation to ask on the -devel list.
130	Comment	walisser	CONTRIBUTOR		I was hoping to give Max a chance to do this, if he doesnt I will submit a new pull request and this one can be closed.I cherry picked the FFmpeg commit only, and it doesnt break the current master branch. I dont know about Travis yet. I dont expect any change in behavior (or supported versions) as a result.FWICT this patch is about fixing a build failure since the function prototypes changed in the new headers. But since theyre only adding a const where there wasnt one, that

					wont break anything (const isnt part of the C calling conventions AFAIK). So functionally, nothing has changed, the same symbols are still loaded from the libraries. This can all be verified of course without much effort.
130	Comment	MaxKellermann	CONTRIBUTOR		As of yet, I have no feedback whether my changes will be merged. I dont want to waste time on something that will go to the trash can. When I submitted this PR, there was no conflict - the Audacity project ignored my PR for so long, and **after** that merged changed which conflicted with this PR. Yes, I will do you the

					favor and resolve the conflicts - if you want me to.About the `const`: the function pointers are not compatible if the constness of pointer targets is not the same. This has nothing to do with calling conventions - calling conventions describe how calls are made on the machine language level. Thus, calling conventions have little to do with the C language, theyre a lower level, and semantic API declarations like constness do not matter on that level.
130	Comment	Paul-Licameli	MEMBER		As I said to Max before, I do not consider myself competent to decide whichversion of the library should be used in our released binaries for Mac

					andWindows.However I reexamined commit 41de5b385b2ca721a71871fcdb00ec3fa4441b5band I saw that all changes are conditionally compiled, so that it leavesour source code compatible with use of either version of the library.Therefore I have cherry-picked and commited that. Alone, it wasunconflicting.The other commits are not relevant to upgrading the library version.Updates of Makefile.in are periodically done by other people, and simplycommit the results of an automated tool, which doesnt demand realprogramming effort.And from

					discussions earlier with Max, I found other changes towxFileNameWrapper that do not give up the performance improvements Iintended, while also fixing the complaints from valgrind.Therefore I do not intend to take those commits.The changes to .gitignore give developers some convenience while notaffecting the build at all. No development really depends on them.Therefore I intend to close the commit request, having selected the onemost important part.PRLOn Wed, Sep 7, 2016 at 2:25 AM, Max Kellermann notifications@githu

					b.comwrote:> As of yet, I have no feedback whether my changes will be merged. I dont> want to waste time on something that will go to the trash can. When I> submitted this PR, there was no conflict - the Audacity project ignored my> PR for so long, and _after_ that merged changed which conflicted with> this PR. Yes, I will do you the favor and resolve the conflicts - if you> want me to.> > About the const: the function pointers are not compatible if the> constness of pointer targets is not the same. This has nothing to do with> calling conventions - calling conventions

					describe how calls are made on> the machine language level. Thus, calling conventions have little to do> with the C language, theyre a lower level, and semantic API declarations> like constness do not matter on that level.> > —> You are receiving this because you commented.> Reply to this email directly, view it on GitHub> https://github.com/audacity/audacity/pull/130#issuecomment-245186932,> or mute the thread> https://github.com/notifications/unsubscribe-auth/ALITYQlVb91C0qUBtdgxHX_i0fQ_dK_Kks5qnlj3gaJpZM4IRpzz> .
130	Comment	Paul-Licameli	MEMBER		... And the Travis build has

					succeeded.PRLOn Wed, Sep 7, 2016 at 6:39 AM, Paul Licameli paul.licameli@gmail.comwrote:> As I said to Max before, I do not consider myself competent to decide> which version of the library should be used in our released binaries for> Mac and Windows.> > However I reexamined commit 41de5b385b2ca721a71871fcdb00ec3fa4441b5b> > and I saw that all changes are conditionally compiled, so that it leaves> our source code compatible with use of either version of the library.> > Therefore I have cherry-picked and commited that. Alone, it was> unconflicting.> >

					The other commits are not relevant to upgrading the library version.> Updates of Makefile.in are periodically done by other people, and simply> commit the results of an automated tool, which doesnt demand real> programming effort.> > And from discussions earlier with Max, I found other changes to> wxFileNameWrapper that do not give up the performance improvements I> intended, while also fixing the complaints from valgrind.> > Therefore I do not intend to take those commits.> > The changes to .gitignore give developers some convenience while

					not> affecting the build at all. No development really depends on them.> > Therefore I intend to close the commit request, having selected the one> most important part.> > PRL> > On Wed, Sep 7, 2016 at 2:25 AM, Max Kellermann notifications@github.com> wrote:> > > As of yet, I have no feedback whether my changes will be merged. I dont> > want to waste time on something that will go to the trash can. When I> > submitted this PR, there was no conflict - the Audacity project ignored my> > PR for so long, and _after_ that merged changed which conflicted with> > this PR. Yes, I will

					do you the favor and resolve the conflicts - if you> > want me to.> > > > About the const: the function pointers are not compatible if the> > constness of pointer targets is not the same. This has nothing to do with> > calling conventions - calling conventions describe how calls are made on> > the machine language level. Thus, calling conventions have little to do> > with the C language, theyre a lower level, and semantic API declarations> > like constness do not matter on that level.> > > > —> > You are receiving this because you commented.> > Reply to this email directly, view it on GitHub> >

					https://github.com/audacity/audacity/pull/130#issuecomment-245186932,> > or mute the thread> > https://github.com/notifications/unsubscribe-auth/ALITYQlVb91C0qUBtdgxHX_i0fQ_dK_Kks5qnlj3gaJpZM4IRpzz> > .

Annex B – Alloy Analyzer code for CognitiveKiP

```
module CognitiveKip
//CognitiveKIP Constructs

//Goals
abstract sig Goal {}
sig ActiveGoal extends Goal {isUnsupportedBy: set ExternalizedBelief,
propositionalContentOf: one Desire}
abstract sig PursuableGoal extends Goal {propositionalContentOf: one Intention,
isSupportedBy: some ExternalizedBelief}
abstract sig ActivityGoal extends PursuableGoal {}
sig ConditionedGoal extends ActivityGoal {dependsOn: some ActivityGoal}
sig ExecutiveGoal extends ActivityGoal {}

//Agents
abstract sig Participant {performs : one Activity, participates: some
CommunicativeInteraction}
abstract sig IndividualParticipant extends Participant {}
abstract sig CollectiveParticipant extends Participant {}
sig InnovationAgent extends Participant{ }
sig ImpactAgent extends Participant{ }
//sig KiAParticipants { involvedAt: one KnowledgeIntensiveActivity}

//Resources
sig Resource {participates: some Activity}

//IntentionalStates
abstract sig IntentionalState {belongsTo: one Participant}
sig Belief extends IntentionalState {}
sig Desire extends IntentionalState {}
```

sig Intention extends IntentionalState { }

sig Feeling extends IntentionalState {}

//Situations

sig Situation { triggers: some Event}

//Knowledge Intensive Activities

sig Event {

bringsAbout: one Situation,

//triggeredBy: one Situation,

compose: set Event}

sig Activity extends Event {

//performedBy: one Participant,

fulfills : one ActivityGoal

}

sig KnowledgeIntensiveActivity extends Activity {}

sig CommunicativeInteraction extends Activity {occursAt: one KnowledgeIntensiveActivity}

//X1: All Activity fulfills one and only one Goal

fact X1 { some act1,act2:Activity | one gl:ActivityGoal | not (act1.fulfills = gl) && (act2.fulfills = gl) }

//X2: All Activity is performed by a Participant

fact X2 { all act:Activity | one pt:Participant | act = pt.performs }

//X2: A KiA is performed by an Impact Agent

fact X3 { all kia:KnowledgeIntensiveActivity | one ia:ImpactAgent | kia = ia.performs }

//X4: A ConditionedGoal cannot depend of itself

*fact X4 { all cdg:ConditionedGoal | not cdg in cdg.*dependsOn }*

//X5: A Situation brought about an Event cannot be the same Situation that triggers the Event

fact X5 {all ev:Event | one st:Situation | not (ev.bringsAbout = st) && (st.triggers=ev) }

//X6: An Event cannot compose itself

fact X6 {all ev:Event | not ev in ev.^compose }

//X7: A KiA cannot compose a CommunicativeInteraction

fact X7 {all kia:KnowledgeIntensiveActivity | all ci:CommunicativeInteraction| not ci in kia.compose}

//X8: A CommunicativeInteraction cannot compose a KiA

fact X8 {all kia:KnowledgeIntensiveActivity | all ci:CommunicativeInteraction| not kia in ci.compose}

sig Question extends Situation {}

sig Decision extends KnowledgeIntensiveActivity {

composes: one KnowledgeIntensiveActivity,

postState:some Alternative}

abstract sig Alternative extends Situation { }

sig ChosenAlternative extends Alternative {}

sig DiscardedAlternative extends Alternative {propositionalContent: one ActiveGoal}

//X9:All Decision is triggered by a Question

fact X9{ all ds:Decision | one qs:Question | qs.triggers = ds}

fact X10 { all qs:Question | one ds:Decision | qs.triggers = ds}

//X11: All Chosen Alternative is bringsAbout by a Decision

fact X11 {all cl:ChosenAlternative | one ds:Decision | ds.bringsAbout = cl}

//X12: All Alternative is postState of a Decision

fact X12 {all al:Alternative | one ds:Decision | al in ds.postState }

// X13: All ChosenAlternative triggers an Activity

fact X13 { all cl:ChosenAlternative | one act:Activity | cl.triggers = act }

//X14: All Decision bringsAbout a ChosenAlternative

fact X14 {all ds:Decision| one cl:ChosenAlternative | ds.bringsAbout = cl}

//Externalized Intentional States

abstract sig ExternalizedIntentionalState {}

sig ExternalizedBelief extends ExternalizedIntentionalState {

representationOf: one Belief}

sig ExternalizedDesire extends ExternalizedIntentionalState {

representationOf: one Desire}

sig ExternalizedIntention extends ExternalizedIntentionalState {

representationOf: one Intention}

sig ExternalizedFeeling extends ExternalizedIntentionalState {

representationOf: one Feeling}

//Speech acts

abstract sig SpeechAct {performedBy: one Participant,

compose: one CommunicativeInteraction,

modifies: set CommonGround}

sig AssertiveSpeechAct extends SpeechAct { expressionOf: one ExternalizedBelief}

sig DirectiveSpeechAct extends SpeechAct { expressionOf: one ExternalizedDesire}

sig CommissiveSpeechAct extends SpeechAct { expressionOf: one ExternalizedIntention }

sig ExpressiveSpeechAct extends SpeechAct { expressionOf: one ExternalizedFeeling }

//Common Ground
sig CommonGround extends Situation {
contains: some ExternalizedBelief,
occursAt: one CommunicativeInteraction,
presupposedBy: some Participant}
sig ManifestEvent extends Event {modifies: some CommonGround}

//X15: "The Common Ground is composed by the Externalized Beliefs
//pressuposed to be shared by all Agents participating at a Communicative
//Interaction"
fact X15 { all cg:CommonGround | all pt:Participant |
one ci:CommunicativeInteraction | ci in pt.participates &&
cg in ci.bringsAbout && pt in cg.presupposedBy }

fact X16 { all cg:CommonGround | one ci:CommunicativeInteraction |
ci.bringsAbout = cg}

fact X17 { all ci:CommunicativeInteraction | one cg:CommonGround |
ci.bringsAbout = cg}

fact X18 { all sp:SpeechAct | one cg:CommonGround | one pt:Participant | one ci:CommunicativeInteraction |
(pt.participates = ci) && (cg.occursAt = ci) && (sp.modifies = cg) }

//Social Concepts
abstract sig SocialConcept { recognizedBy: some Participant }
sig DeclarativeSpeechAct extends SpeechAct { }
sig Commitment extends SocialConcept{counterpartOf: one Claim,
committedBy: one Participant,
compose: set SocialRelation,

createdBy: lone CommissiveSpeechAct}

*sig Claim extends SocialConcept {/*counterpartOf: one Commitment,*/*

claimedBy: one Participant, compose: set SocialRelation}

sig DischargeCondition extends Event{discharges: some Commitment}

sig SocialObject extends SocialConcept {imposedUpon: one Resource,

createdBy: one DeclarativeSpeechAct}

sig SocialRelation extends SocialConcept {commitments: some Commitment,

receivedBy: some Claim,

createdBy: one DeclarativeSpeechAct}

sig NormativeDescription extends SocialObject {defines: some SocialRelation}

//X19: An Agent cannot be the recipient of a Claim and a Commitment, the Commitment being the counterpartOf the Claim

fact X19 { one co:Commitment | one cl:Claim | one pt:Participant |

not (cl.claimedBy = pt) && (co.committedBy = pt) && (co.counterpartOf = cl)}

//X20: A Decision cannot compose itself

fact X20{ all ds:Decision | not ds.composes = ds}

//X21: All ActivityGoal is propositional content of an Intention of an Impact Agent

fact X21 { all ag:ActivityGoal | one it:Intention | one ia:ImpactAgent |

*ia in it.*belongsTo && it in ag.*propositionalContentOf }*

//X22: All Ext. Belief composing CommonGround is pressuposedBy KiA Participants

fact X22 { all cg:CommonGround | all pt:Participant | one ci:CommunicativeInteraction |

pt.participates = ci && cg.occursAt = ci && pt in cg.presupposedBy }