# Collaborative Diagnosis of Exceptions to Contracts (Extended Abstract)

Özgür Kafalı*
Dept of Computer Engineering
Boğaziçi University
İstanbul, Turkey
ozgurkafali@gmail.com

Francesca Toni
Department of Computing
Imperial College London
London, UK
ft@imperial.ac.uk

Paolo Torroni
DEIS
University of Bologna
Bologna, Italy
paolo.torroni@unibo.it

## ABSTRACT

Exceptions constitute a great deal of autonomous process execution. In order to resolve an exception, several participants should collaborate and exchange knowledge. We believe that argumentation technologies lend themselves very well to be used in this context, both for elaborating on possible causes of exceptions, and for exchanging the result of such elaboration. We propose an open and modular multi-agent framework for handling exceptions using agent dialogues and assumption-based argumentation as the underlying logic.

## Categories and Subject Descriptors

I.2.1 [**Distributed Artificial Intelligence**]: Multiagent Systems

## General Terms

Verification

## Keywords

Agent commitments, Distributed problem solving, Argumentation, Judgment aggregation and belief merging, Agent Reasoning (single and multiagent)

## 1. INTRODUCTION

Open multi-agent systems enable distributed process execution using autonomous agents. Each agent executes a different part of the process. While this provides some advantages (e.g., privacy), it also makes the process vulnerable to *exceptions*. For example, if a buyer does not receive a merchandise that was scheduled for delivery, it can conclude that there must have been an exception in the workings of the entire process. Clearly, an agent's misbehavior affects others. Thus when such an exception occurs, the agent facing the exception needs to identify the problem behind it, so as to handle it properly and get back to normal execution. However, this is a hard and complicated task, usually because the handling of an exception requires significant information exchange among a group of agents.

*The first author is supported by Boğaziçi University Research Fund under grant BAP5694, and the Turkish State Planning Organization (DPT) under the TAM Project, number 2007K120610.

**Cite as:** Collaborative Diagnosis of Exceptions to Contracts (Extended Abstract), Özgür Kafalı, Francesca Toni, and Paolo Torroni, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1167-1168.


We propose a distributed framework targeted for handling exceptions in open multi-agent systems. Contracts are expressed by way of social commitments [5]. We propose a form of collaborative diagnosis as a part of exception handling procedures, which takes place when an exception occurs, such as the violation of a commitment.

The diagnosis activities are embedded in an agent execution cycle, and they are performed whenever necessary. That is, when an exception is detected, the agents switch from normal process execution to diagnosis mode. When the exception is diagnosed (and possibly resolved with some sort of compensation), the agents go back to normal process execution.

Dialogues provide the information exchange among the agents to enable diagnostic activities to step from agent to agent until the reason of the exception is found. Reasoning uses the assumption-based argumentation (ABA) framework [1]. Thanks to its grounding on a consolidated argumentation theory, we are able to describe the diagnosis process in a high-level, declarative way, we can enable agents to construct hypotheses (arguments) about what went wrong and exchange such hypotheses between them, and we can ensure that the overall process is deterministic.

## 2. DIAGNOSIS FRAMEWORK

The proposed framework comprises agents reasoning and interacting for process execution and exception diagnosis.

A process begins execution as soon as it is initialized (e.g., the contracts between the agents are created). The process continues normal execution until an *exception* condition is detected. Then, the process enters the exception state where the agent detecting the exception starts investigating the cause of the exception. This initiates the *diagnosis process*, which is carried out by way of dialogues. When a valid justification is produced and agreed upon by the agents involved in the diagnosis, the process enters the recovery state. Ideally, if a reasonable compensation is found for the exception (e.g., by way of negotiation), the process goes back to the execution state, where it resumes its normal operation.

Agents act in an *environment*, as process entities and as diagnosis entities. As process entities, they perform actions such as paying for and delivering goods. As diagnosis entities they can gather evidence from the environment, and engage in dialogues with one another. In particular, *request explanation dialogues* correspond to delegation of diagnosis from one agent to another. That is, the agent requests an explanation from another agent about a property of interest that it believes the other agent knows more about. The other agent responds by either providing an explanation why the property holds, or by rebutting with an explanation why the property does not hold.

The agent *execution model* is in charge of recording observations, identifying (communicative and physical) actions to be performed, and executing such actions.

We propose the following dialogue utterances:

- *explain*($A_i$, $A_j$, $P$): agent $A_i$ sends a diagnosis request to $A_j$, asking for a justification for a given property $P$.
- *justify*($A_i$, $A_j$, $Q$, $P$): agent $A_i$ provides agent $A_j$ with a justification $Q$ to why $P$ holds.
- *rebut*($A_i$, $A_j$, $Q$, $\neg P$): agent $A_i$ provides agent $A_j$ with a justification $Q$ to why $P$ does *not* hold.

A request explanation dialogue commences with an utterance of the form *explain*($A_i$, $A_j$, $P$). It then continues with either a *justify*($A_j$, $A_i$, $Q$, $P$) or a *rebut*($A_j$, $A_i$, $Q$, $\neg P$), at which points it ends. The form of the property $P$ and of the justification $Q$ depends on the domain. For example, if $P$ means "the book has not been delivered", a possible justification $Q$ for $P$, if privacy limitations allow, may include a deliverer's commitment to deliver the book, indicating that the reason for $P$ is the deliverer's misbehaviour.

As an example, a request explanation dialogue may be:

$c \to b$) *explain*(*customer*,*bookstore*,¬*delivered*(*book*))

$b \to d$) *explain*(*bookstore*,*deliverer*,¬*delivered*(*book*))

$d \to E_d$) *question*(*deliverer*,$E_d$,¬*delivered*(*book*))

$E_d \to d$) *answer*($E_d$,*deliverer*,*delivered*(*book*))

$d \to b$) *rebut*(*deliverer*,*bookstore*, *answer*($E_d$,*deliverer*, *delivered*(*book*)), *delivered*(*book*))

$b \to c$) *rebut*(*bookstore*,*customer*,*answer*($E_d$,*deliverer*, *delivered*(*book*)), *delivered*(*book*))

where $E_d$ represents the environment of the deliverer, and the utterance *answer*($E_d$,*deliverer*,*delivered*(*book*)) indicates the result of the deliverer's observation from $E_d$ that the book has in fact been delivered, e.g., the delivery chart had been signed.

# 3. REASONING

For agent knowledge representation and reasoning we propose ABA [1], because of its strong theoretical properties, its proven capability of dealing with inconsistency and decision-making, and the fact that it is equipped with provably correct computational mechanisms, that will support any future deployment of our proposed representation.

In ABA, we define both domain-specific and general knowledge. Examples of **domain-specific** knowledge are the following two rules:

- $\textit{by\_contract}(\textit{cc}(\textit{bookstore}, \textit{customer}, \textit{paid}(\textit{book}), \textit{delivered}(\textit{book})))$.
- $\textit{justification}(\neg\textit{paid\_delivery}(\textit{book}), \neg\textit{delivered}(\textit{book})) \leftarrow \neg\textit{paid\_delivery}(\textit{book}), \neg\textit{delivered}(\textit{book})$.

The first rule is a fact, which models a contract between customer and bookstore. The second one represents that a problem in the delivery payment may be the reason for no delivery.

*General-purpose reasoning rules* consist of belief rules, commitment rules and action rules.

**Belief rules** allow to "internalise" beliefs drawn from observations and expected effects of actions, unless there are reasons not to do so.

**Commitment rules** model the evolution of commitments during the agent's life-cycle. For example,

- $\textit{fulfilled}(c(X,Y,P)) \leftarrow \textit{by\_contract}(c(X,Y,P)), P, \textit{asm}(\textit{fulfilled}(c(X,Y,P)))$.

is a *defeasible* rule (as commitments change during the agent's life-cycle) saying that we can assume a commitment about $P$ to be fulfilled if $P$ holds, and this assumption is feasible. To prevent unconstrained assumption making, $\textit{asm}(\textit{fulfilled}(c(X,Y,P)))$ will be subject to restrictions. For example, the same commitment cannot be assumed to be fulfilled and violated at the same time, or an agent cannot ask a question that has already been answered.

**Action rules** are of two types: for determining whether and how to consult the environment (action *question*) or for determining whether and how to conduct a *request explanation* dialogue.

For example,

- $\textit{explain}(X,Y,\neg P) \leftarrow \textit{violated}(c(Y,X,P)), \textit{by\_contract}(\textit{cc}(Y,X,Q,P)), \textit{answer}(E_X,X,\neg P), \textit{answer}(E_X,X,Q), \textit{asm}(\textit{explain}(X,Y,\neg P))$.
- $\textit{rebut}(X,Y,R,P) \leftarrow \textit{explain}(Y,X,\neg P), \textit{justification}(R,P), \textit{asm}(\textit{justification}(R,P))$.

tell under which conditions to communicate possible explanations of exceptions, by way of *explain* and *rebut* utterances. Thus agents can produce dialogues such as the one illustrated above by way of ABA reasoning. For instance, the 5th utterance ($d \to b$) is a conclusion of $d$'s ABA framework supported by rules such as the above for *rebut*, plus all legitimate assumptions that $b$ can make based on the current dialogue and its interaction with the environment.

# 4. RELATED AND FUTURE WORK

Related research on handling commitment exceptions has been carried out by Kafalı et al. [2, 3], but without integrating the diagnosis process with agent reasoning and control cycle. Such an integration is enabled here by the underlying ABA argumentation logic. In this way we can express knowledge and reasoning in a declarative and modular way, and study properties about the overall diagnosis process. A complete definition of the diagnosis framework in ABA and the definition of its properties is ongoing work.

In the future we plan to address time, which has been recognized to be a very important aspect of commitment specification and handling [6]. To fill this gap, we plan to exploit the temporal reasoning capabilities of the KGP agent model [4], which we identified as a potential candidate for the embedding of this work.

# 5. REFERENCES


[1] P. Dung, R. Kowalski, and F. Toni. Assumption-based argumentation. In I. Rahwan and G. Simari, editors, *Argumentation in AI*, pages 199–218. Springer, 2009.

[2] Ö. Kafalı, F. Chesani, and P. Torroni. What happened to my commitment? Exception diagnosis among misalignment and misbehavior. In *Proc. CLIMA XI*, LNCS 6245:82–98, 2010.

[3] Ö. Kafalı and P. Yolum. Detecting exceptions in commitment protocols: Discovering hidden states. In *Proc. LADS*, LNCS 6039:112–127. Springer, 2009.

[4] A. C. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. The KGP model of agency. In *Proc. ECAI*, pages 33–37. IOS Press, 2004.

[5] M. P. Singh. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7:97–113, 1999.

[6] P. Torroni, F. Chesani, P. Mello, and M. Montali. Social commitments in time: Satisfied or compensated. In *Proc. DALT*, LNCS 5948:228–243. Springer, 2009.