The Role of Ontology in Agent-Oriented Requirements Analysis

Massimo Cossentino¹, Carmelo Lodato¹, Salvatore Lopes¹, Patrizia Ribino¹, Luca Sabatucci¹, and Valeria Seidita²,

¹ Icar Institute, National Research Council, Viale delle Scienze Ed.11, 90128 Palermo, Italy
{cossentino, lopes, c.lodato, ribino, sabatucci}@pa.icar.cnr.it
² Department of Chemical, Management, Computer, Mechanical Engineering Università degli Studi di Palermo, Italy
valeria.seidita@unipa.it

Abstract. Goals identification is an open issue in the Requirement Analysis field for Multi-Agent System design. Stakeholders commonly don’t have full and exhaustive awareness about their real objectives in the domain problem. Goals are implicitly expressed during the inquiring phase. Thus, the goal identification task is commonly ascribed to the expertise of the analyst. In this work we propose a new methodological approach to formalize the problem domain in terms of its ontological representation. We also introduce a new heuristic for extraction and validation of goals from the problem description. In this approach the domain can provide evidences for the identification of unvoiced goals and for their validation according to the contextual data.

Keywords: Ontology, Goals, Goal-Oriented Analysis, Requirement Analysis, Multi-Agent System

1 Introduction

Today, the growing complexity of software systems demands to consistently take into consideration different aspects, such as technical concerns, as well as application domain’s stakeholders (humans and organization). The multi-agent systems seem to provide an adequate level of abstraction to deal with complex, dynamic and distribute architectures, and for an intensive human involvement. Multi-agent systems are made by a number of interacting software agents that, individually, exhibit autonomous and proactive behaviors (trying to achieve their design goals), and are also able to organize for achieving global goals.

The connection between Goal-Oriented methodologies and Multi-Agent Systems design is a mainstream research area in AOSE. The importance of explicitly take into consideration the concept of goal is underlined by many methodologies for designing multi-agent systems; in such methodologies the goal is the element to be model and to be reasoned out. A Goal is typically defined as “a condition or state of the world that the system wants to achieve”. It allows to depict a
state of the world that does not exist, but that is desired. After system goals are identified they represent a richness for the designer and the developer, who can directly implement them by using appropriate constructs such as those provided by BDI (Belief-Desire-Intention) architecture.

Close to system goals, there are users’ goals. Goal-Oriented methodologies indicate that system goals directly derive from the domain. Stakeholders’ needs and desires are candidate to generate system goals [1, 2]. Therefore, user goals identification become an instrument for reducing the gap between analyst’s and stakeholders’ knowledge.

The aim of the work we illustrate in this paper is to provide a methodological approach for identifying and detailing goals starting from a quite informal description of the problem domain and the user goals. The work is part of a more complex one where we exploit the powerful of multi-agent paradigm in order to model complex systems where agents are organized and behave as humans do in their society [3]. The final aim is a complete methodological approach from analysis to implementation and the creation of a CASE tool with a specific notation for designing multi-agent system organizations.

In this paper we focus on the goal-oriented requirements analysis and we try to find a solution to the problems related to the formalization of the system goals starting from an informal representation of the problem. Normally, goals identification success is due to how the problem is described, to the analyst’s knowledge on the problem domain and to his expertise. We want to overcome this limit by employing the powerful and the features of ontology for formally representing knowledge on the problem domain and by creating rules for identifying and formalizing goals from the ontology.

Thus we propose two main steps in the goal-oriented analysis:

– using ontology as an artefact that we call Problem Ontology for formalizing knowledge during the contextual inquiry activity;
– extracting goals from the ontology description following some specific heuristics.

As an artefact, the ontology is the result of a design process activity for which we identified specific guidelines. The main advantage in this part of work is to consider the Problem Ontology building activity as an evolutionary activity; from a free textual document, knowledge is transformed in a structured form allowing to reduce uncertainty, redundancies and missing elements inevitably due to the informal description of the initial textual document.

The Problem Ontology is the input document for the goal identification activity; the heuristics we propose allow to extract goal in an objective fashion only basing on specific guidelines; this fact avoids to be bound by the analyst experience and knowledge.

The paper is organized as follows: in section 2 our motivations and objectives are related to existing works, in sections 3 we deeply detail respectively how to build the problem ontology and how to extract goals from it, then in section 4 a case study is used to better illustrate the proposed work and finally in section 5 some discussions and conclusions are drawn.
2 Motivation and Objectives

The concept of Goal is widely used in requirements engineering methods since it allows for reducing the gap between analyst’s and stakeholders’ knowledge. Many definitions exist for Goal. In this paper, when talking about Goal, we mean “a condition or state of the world that the system wants to achieve”. This definition highlights a very interesting connection with the agent oriented paradigm since it recalls the aspects of autonomy and proactiveness, typical of software agents.

The goal identification is the preliminary activity of the agent goal modeling phase in which analyst makes the strategic objectives of the system (intended as the sum of the software and its environment) explicit. User goal identification is typically conducted in the early phases of requirement analysis, in which the core task is to conquer a significant and transferable understanding of the portion of the world where to introduce the software. This knowledge influences subsequent design decisions. It is frequent that agents’ behavior derives from the goals and needs of stakeholders.

So far, research has mainly focused on the development of methods for modeling and reasoning on goals, for optimizing their achievement and for solving possible conflicts. There is room for novel methods for systematically identifying goals from the domain and from the users. One of the most effective ways to identify user goals is to move as closest as possible to where the work is, to collect concrete data about people and their behavior. To this aim, Contextual Inquiry [4] is a collection of user-centered techniques to conduct qualitative and ethnographic investigation of the domain. The focus is on the concrete work practices of professionals, on the understanding of both physical and social contexts in which actors behave.

Several experiences [5, 1, 6] exist in literature concerning the possibility to integrate contextual inquiry techniques into requirement engineering methods. At the best of our knowledge, the integration is still to be improved in order to generate a seamless sharing of objectives, knowledge and principles [7]. This is not trivial, since the two disciplines ground on different philosophical ideas and approaches. Kaindl [8] poses the issue concerning the usefulness of common representations for requirements engineering and interaction design and the possibility and the convenience in developing a combined process. Nebe et al. [9] also argue that a fruitful cooperation is possible if we firstly identify integration points between disciplines. Leonardi et al. [10] proposes a dialogic approach to bridge these different disciplines, rules by different methods, principles and criteria. The approach encourages to maintain the individuality of the two analysis threads, contextual inquiry and requirement engineering, and to improve the dialogue among teams. Nevertheless the proposed common, shared dictionary is built by using the natural language, in a qualitative study fashion.

The work we present changes the point of view in [10], by considering an engineering asset. We propose the use of a formal ontology as an intermediate step between domain and goals. In a dialogic fashion, the domain ontology is an interchange artifact in order to share and consolidate the knowledge acquired during the contextual inquiry by different persons. The challenge of moving from
domain to goals is declined into (i) moving from domain to domain ontology and (ii) moving from domain ontology to goals.

Many works exist in literature about the domain ontology construction and the identification of incompleteness and inconsistencies [11, 12]. Following the observation that goals exist in the domain and therefore they are implicitly captured by a classical ontology [13, 14], we want to base their identification on two ontology elements: actions and positions. Domain actions are done (by someone) with some purpose. Making these ontology elements explicitly visible in the ontology triggers the heuristic reasoning for goal discovery.

The next sections explain the details of the Domain Ontology where actions and positions are explicit, and introduce the heuristic reasoning that is possible to conduct for managing the ontology complexity and extracting useful information on domain goals.

3 From Ontology to Goals

In this paper, we propose a new methodological approach for goal identification. The goal identification is based on heuristics for goal extraction starting from an ontological formalization of the problem domain, named Problem Ontology. In this approach the ontological representation of the domain may provide evidences for the identification of unvoiced goals and for their validation according to the contextual data. The description of the entire requirement analysis process is out of the scope of this paper. Our approach wants only to cope the goal identification phase of the classical agent-oriented requirements analysis.

In our approach, the goal identification phase consists of two activities: the Problem Ontology Description and the Goal Identification activity.

3.1 The Problem Ontology Description

The Problem Ontology Description activity allows to describe the problem domain elements and their relations in a quite formal way. This formalization is obtained by means of a Problem Ontology. Several definitions of the term ontology are proposed in literature, several refer to the philosophical perspective. From our point of view, instead, we are more close to Artificial Intelligence oriented definitions of ontology [15][16][17][18][19]. In this work, we adopt the definition proposed by van Heijst et.al[19]:

An ontology is an explicit knowledge-level specification of a conceptualization, (...). The conceptualization, and therefore the ontology, may be affected by the particular domain and the particular task it is intended for.

The proposed Problem Ontology wants to empathize what are the elements of interest, who are the stakeholders and how they act in the domain. To do this, we founded our ontology on the metamodel shown in Fig.1, where the core elements are Concept, Position and Action.

Commonly, the term Concept is used in a broad sense to identify "anything about which something is said"[20]. In this work, we use this term for
The Role of Ontology in Agent-Oriented Requirements Analysis

representing categories of entities in the domain. Such entities may be either, of course, physical or logical (abstract). A Concept (see Fig.1) can be specialized in Position or Object. A Position is a concept performing Actions. An Object represents all the concepts not performing Intentional Actions. Both a Position and an Object can represent the target of an action, that is to say the result toward which efforts are directed. We use the term Action to represent “the cause of an event by an acting concept” (adapted from [21]). An Action can also take a Concept as input. According to [22], we classify Actions as Intentional and Unintentional. Finally, a Predicate is the expression of a property, a state or more generally a clarification to specify a Concept.

As regards the relations among the ontology elements, the possible kinds are: (i) is-a that is the relationship between an ontological element and one or more refined versions of it; (ii) part-of that is the part-whole relationship, in which ontological elements representing the components of something are associated with the ontological element representing the entire assembly; (iii) association that is used in order to express all types of relation that can occur between two ontological elements.

As we previously said, the aim of the Problem Ontology Description activity is to realize a formal and structured representation of the information about the problem domain. We assume that a set of textual documents about the problem description is available and contains information concerning:

1. a glossary referred to common terms of the specific domain;
2. professional figures engaged in the specific activities of the domain.
3. relations among domain entities.
4. scenarios.

This kind of documents are typically affected by ambiguities due to the human interpretations or to implicit knowledge unarticulated by the domain ex-
perts. This activity consists of an iterative procedure that allows us to highlight ambiguities and lacks in the textual descriptions. During knowledge elicitation, in fact, the problem is commonly described by different perspectives and by several stakeholders. This could generate overlapping terms and redundancies that could be misinterpreted. Moreover, the unconscious tendency of the stakeholder to neglect some of his implicit knowledge may cause lacks in the problem description. The transition from a textual description to a formal and structured representation of the knowledge could make this gap visible and could simplify the design phase in the solution domain.

We decomposed the Problem Ontology Description activity in two different tasks: *Textual Extraction* and *Problem Ontology Diagram Drawing*.

The aim of the *Textual Extraction* task is the identification and the extraction of a collection of representative domain entities from the available textual documents; each domain entity becomes a candidate ontological element. The guidelines adopted to perform this task are adapted from some well known guidelines proposed in literature [23][24]:

1. Firstly, we extracted the nouns from text and we created a list of items grouped in different clusters according to their grammatical function within the sentence. Thus, a noun used as: (i) “subject” followed by a verb is a candidate *Position* if the verb describes an *Action* of the domain; (ii) “adjective” is a candidate *Predicate* of the noun it describes; (iii) “direct object” is a candidate *Object*.

2. Secondly, we identified the verbs. In a sentence, a verb may indicate: (i) an action performed by the subject of the sentence becoming a candidate *Action* in our Problem Ontology; (ii) a relation among nouns such as aggregation (PART-OF), inheritance (IS-A) or generic association.

3. Finally, we identified the adjective to be candidate *Predicate*.

The *Problem Ontology Diagram Drawing* task is devoted to introduce the elements detected in the previous task in a diagram and to represent them in a more straightforward graphic fashion. We use a UML class diagram in order to graphically represent an instance of our *Problem Ontology*. In this diagram, the *Ontological Elements* are depicted by means of rectangles labeled with the appropriate stereotype (Position, Object, Action, etc . . .). The *Actions* are moreover expressed by means of verbs. The *is-a*, *part-of* and *association* relationships are represented by means of the UML generalization, aggregation and association notation respectively. The association can be also characterized by means of a label that specifies the semantic of the relation.

Therefore, the elements coming from the previous task are firstly graphically represented by means of the appropriate ontological construct in the diagram and then they are related each other by means of the identified relations. If the problem domain is characterized by a specific organizational pattern, it is obviously useful to highlight this structure in a graphical arrangement of the diagram. This design choice may be helpful to the goal extraction due to the functional information emerging at a glance.
Fig. 2. Example of a Problem Ontology instance

Fig. 2 shows a minimal example related to a specific self-explanatory domain just to introduce a Problem Ontology instance with the adopted notation.

The results of this process are preparatory for the Goal Identification activity.

3.2 Goal Identification from Ontology

The main contribution of our paper concerns a new heuristic for goal extraction starting from a formal representation of the problem domain. The proposed method is intended to be used both as aid in order to find a set of preliminary goals and as a practice in order to validate goals discovered by means of traditional methods.

Before explaining the proposed heuristic, in the following list we introduce some definitions:

- **DEF. 1** - A **Domain Cluster** is a set of ontological elements related to a domain section representing an organizational area or a portion of a workflow.

- **DEF. 2** - A **Objective Element** is the rationale of a Domain Cluster. That is to say, the raison d’être of the domain cluster without which the portion of domain or workflow at issue is senseless.

- **DEF. 2** - A **Root Goal** is the achievement or maintenance of a final state of interest related to a Domain Cluster.

- **DEF. 3** - A **Sub Goal** contributes to the achievement or a maintenance of a Root Goal of a Domain Cluster.

Our heuristic (see Algorithm 1) is an iterative procedure applied to a Problem Ontology diagram. At the step 1⃝, we start to identify all probable Domain Clusters in the diagram at different levels of granularity. To do this, we isolate portions of ontology under the responsibility of the same Position identifying the domain clusters at the lowest level of granularity, and then we compose them into higher domain clusters. In order to complete the step 1⃝ of Algorithm 1 we have to follow this guideline: (i) Identify all the Positions. (ii) For each Position, isolate all the Actions and the related Concepts. (iii) Aggregate these
Algorithm 1: Goal Extraction

Input Data: POD Diagram;
Output: $G = \{g : g = \text{RootGoal} \land g = \text{subGoal}\}$;

1. Search Domain Cluster;
   while $\exists$ Domain Cluster do
     2. Identify Objective Elements;
        while $\exists$ Objective Element do
          foreach Objective Element do
            3. Identify Candidate Goal;
            4. Prune;
            5. Identify Objective Elements;
          end
        end
    end

ontology elements in a new domain cluster. (iv) Iteratively aggregate the lower level domain cluster into higher level new ones, by considering is-a and part-of relationships among ontology elements.

Then, the step 2 of Algorithm 1 concerns the identification of the Objective Elements for each Domain Cluster. Following the patterns shown in Fig. 3, an Objective Element is identified as:

**CASE 1** - A Concept that is only target.
**CASE 2** - A Concept that is target in a Domain Cluster DC1 and input for an Action of another Domain Cluster DC2.
**CASE 3** - A Concept that is target and input for an Action of its Domain Cluster that has itself as target.
**CASE 4** - A Concept that is target in a Domain Cluster DC1 and input for an Action of another Domain Cluster DC2 with a specific Predicate value.
**CASE 5** - An Action that has an input coming from the Domain Cluster DC1 or from another Domain Cluster DC $\neq$ DC1.
**CASE 6** - An Action that has an input coming from the Domain Cluster DC1 or from another Domain Cluster DC $\neq$ DC1 and target in another Domain Cluster DC2.

The step 3 of the Algorithm 1 consists in identifying goals according to the type of the Objective Element identified in the previous step:

- If the Objective Element is an Action then the Goal is assignable to the action itself;
- If Objective Element is a Concept, then the Goal can be inferred by the actions whose the concept is target.

These goals are documented by means of the following template that gives further information about the goals.
Goal Template:

**NAME**: label for referring to the goal; it also encapsulates the goal semantics.  
**STATE**: it is the desired condition or state of the world related to the goal.  
**WHO**: it indicates who is responsible for achieving the goal. It may differ from who has the real interest in achieving that state of the world.  
**DEPENDENCY**: it expresses a relationship between the current and another goal. The achievement of the other goal depends on the achievement of the current goal.

In Section 4 we introduce some examples in order to explain how fill the template’s fields.

The step 4 of the Algorithm 1 allows to cut from the current *Domain Cluster* all the elements that do not give more useful information for the extraction of new goals. The step 4 consists in an iterative pruning algorithm (see Algorithm 2) applied to the current *Domain Cluster* configuration in order to delete (under specific conditions) the *Objective Elements* previously identified along with some elements of the graph related to it. Following the Algorithm 2, it is possible to remove: (i) the *Objective Element* (along with its relations) when it is not the *target* or *input* of the *Actions* of another *Domain Cluster*; (ii) all those *Actions* that have only the *Objective Element* as *target*. 

---

**Fig. 3.** Patterns of Objective Elements
Algorithm 2: Pruning

**Input Data:** Domain Cluster $\text{DC}$, Objective Element;  
**Output:** Pruned Domain Cluster;  

if $\text{Objective Element} \notin \text{other Domain Clusters}$ then  
   Remove $\text{Objective Element}$;  
   Remove all its relations;  
else  
   Remove the input/target relations with elements $\in \text{DC}$;  
end  

Select the Actions $\in \text{DC}$ whose the $\text{Objective Element}$ is target;  

foreach $a \in \text{Actions}$ do  
   if $a$ has not other target element then  
      Remove $a$;  
   end  
end

The step 5⃝ of the Algorithm 1 allows to identify new $\text{Objective Elements}$ from the pruned $\text{Domain Cluster}$. If new $\text{Objective Elements}$ exist, then they are candidate $\text{Sub-Goals}$ of the current $\text{Domain Cluster}$.

We repeat the steps 3⃝, 4⃝ and 5⃝ of the Algorithm 1 until there are $\text{Objective Elements}$ in the current $\text{Domain Cluster}$. Otherwise, we select another $\text{Domain Cluster}$ from the set of the $\text{Domain Clusters}$ determined at the step 1⃝, and we start again from the step 2⃝.

We want to point out that the candidate goals are $\text{Sub-Goals}$ when they come from a pruned $\text{Domain Cluster}$. Otherwise they are $\text{Root-Goals}$.

4 Case Study

The proposed approach has been applied to a case study related to an enterprise that produces bicycles. This enterprise is composed of two main organizational areas: production and administration. The production area is devoted to produce bikes whilst the administration manages the financial and organizational aspects of the enterprise. For the sake of clarity, we model only the portion of the problem domain related to the production area.

The production area of the bike enterprise is composed of two factories, named North and South factory according to their geographic position with respect to the location of the administrative offices. Each factory is specialized in manufacturing different bike models. The North and South factories produce the standard and the tandem model respectively. The employees of a factory have different responsibilities and competencies according to the role they play. In this particular case study there are two different roles inside each factory: (i) the $\text{Production Worker}$ that is engaged in the assembly of the bicycles and in inventory (with label) of the produced articles and (ii) the $\text{Warehouse Manager}$
Fig. 4. The Problem Ontology diagram related to Bike Enterprise domain with its Domain Clusters.

that deals with the procurement of raw materials needed to build a specific bike model characterized by a production form.

The ontological description of the problem domain shown in Fig. 4 has been made for a visionary scenario in which the employees of the factories interact with a software system in order to complete their tasks. In this scenario, the system performs autonomously, two actions: it records customer information and data about the produced bikes; it creates daily labels containing information about the production date and the lot number related to the scheduled bikes.

Fig. 4 shows the diagram coming from the Problem Ontology Description activity with all detected Domain Clusters for space concerns. Domain Clusters are surrounded with different line styles. For each Domain Clusters, we identified the related goals. The complete list of goals is reported in Table 1. Among them, we show in detail (according to the previous template) the goals relating to different patterns (Cases) of Objectives Elements (see Fig. 3).

Case 1 - NAME: it is obtained by composing “To” plus the action name plus the target concept name. For instance: To Plan Raw Material Supply. It comes from the action “Plan” plus the concept “Raw Material Supply”. STATE: it is the existence of the target concept instance. For instance: ∃ Instance_of(Raw Material Supply). WHO: it is the position that is the actor of the action related by a
target relationship with the current concept. For instance: Warehouse Manager. 

**DEPENDENCY**: no information comes from this pattern.

<table>
<thead>
<tr>
<th>Domain Cluster</th>
<th>Objective Element</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory</td>
<td>Bike</td>
<td>To Produce Bikes</td>
</tr>
<tr>
<td>South Factory</td>
<td>Tandem Model</td>
<td>To produce Tandem Bike</td>
</tr>
<tr>
<td>North Factory</td>
<td>Standard Model</td>
<td>To produce Standard Bike</td>
</tr>
<tr>
<td>Production</td>
<td>Bike</td>
<td>To Assemble Bikes</td>
</tr>
<tr>
<td>Production</td>
<td>Prints Label</td>
<td>To Print Labels</td>
</tr>
<tr>
<td>Warehouse</td>
<td>Sends</td>
<td>To send Supply Orders</td>
</tr>
<tr>
<td>Warehouse</td>
<td>Raw Materials Supply</td>
<td>To Plan raw Materials Supply</td>
</tr>
<tr>
<td>Software System</td>
<td>Creates</td>
<td>To Record Customer/Bike</td>
</tr>
<tr>
<td>Software System</td>
<td>Label</td>
<td>To Create Label</td>
</tr>
</tbody>
</table>

**Table 1**: List of goals extracted from Bike Enterprise Problem Ontology

**Case 2 - NAME**: as Case 1. For instance: **To Create Label**. It comes from the action “Create” plus the concept “Label”. **STATE**: it is the existence of the target concept instance. For instance: $\exists$ Instance_of(Label). **WHO**: it is the position that is the actor of the action related by a target relationship with the current concept. For instance: Software System. **DEPENDENCY**: with goal derived from Act1. For instance: To Print Label.

**Case 6 - NAME**: it is obtained by composing “To” plus the action plus the concept. For instance: **To Use Software System**. **STATE**: it is the existence of the target concept instance AND the concept is available for access from the position AND the composition of the action plus “-ing” plus the concept. For instance: ($\exists$ Instance_of(Software System) AND (Software System is available) AND (Using Software System)). **WHO**: the position that is responsible for the action. For instance: Employee. **DEPENDENCY**: (If any) Dependency with the goal derived from the target concept.

Because the addressed case study does not present all types of goal deriving from patterns shown in Fig.3, in the following subsection we report some ad-hoc examples in order to explain the remaining cases.

### 4.1 Other Examples

Fig.5 shows two portions of the problem ontology related to the Conference Management System in which we can identify two Objective Elements related to Case 3 and Case 4 of Fig.3) respectively.
Case 3 See Fig. 5.a - NAME: as Case 1 selecting the action that generates the final state of the concept. For instance: To Write Paper. STATE: (If any) Predicate has the expected value. For instance: (if there were a Completed Predicate) Value of Completed is true. WHO: it is the position responsible for the action that generates the final state of the concept. For instance: Author. DEPENDENCY: no information comes from this pattern.

Case 4. See Fig. 5.b - NAME: it is obtained by composing “To” plus the verb representing the predicate plus the concept. For instance: To Complete Paper. It comes from the predicate “Completed” plus the concept “Paper”. STATE: the constraint specified in the input relationship. For instance: [Completed=True]. WHO: as Case 3. For instance: Author. DEPENDENCY: with the goal derived from Act1. For instance: To Review Paper.

Finally in order to explain the last Case, we report a funny example taken from a famous phrase of Forrest Gump “I just felt like running”.

Case 5. See Fig.6 - NAME: it is obtained by composing “To” plus the action. For instance: for the action “Run”, the name is To Run. STATE: it is the composition of “To be” plus action plus “-ing”. For instance: To be running. WHO: the position that is responsible for the action. For instance: Forrest Gump. DEPENDENCY: no information comes from this pattern.

5 Conclusion

The goal identification is one of the critical activities of the agent goal modeling phase in which analyst makes the strategic objectives of the system-to-be ex-
plicit. The issue is how to find a formalization of the system goals starting from an informal representation of the problem. The classic instrument to formalize the problem is the use of an ontology. But, whereas the ontology construction of the domain has been recently explored in literature, the procedure for extracting goals from the domain ontology is still an open problem.

In this paper we focused on this challenge. The solution, we proposed, consists of two tasks. The first step is to build an ontology where concepts such as Action and Position are made explicit in the metamodel. These elements are the core for identifying intentional entities in the domain and how they achieve their objectives. The second step is to explore the dimension of consequentiality among actions. These actions provide qualitative hints for the extraction of the goals.

We are also planning to conduct a controlled experiment to validate the completeness of the proposed heuristics for the goal identification, and the degree of dependency on the documents on which the analyst grounds her work.

We are aware the approach is still preliminary. For instance, so far, the Problem Ontology is unable to represent relationships between two actions. We are investigating how these relationships could help in the goal extraction. The current work represents an early step of a more ambitious plan to cover the whole agent-oriented requirement analysis, from the exploration of the domain towards the definition of agents’ goals. This will include also guidelines to (i) analyze and disambiguate the informality of the domain, to (ii) translate it into a formal document that is the Problem Ontology and to (iii) validate ontology and goal models with respect to the domain.

References