Requirement meta model of a cooperative information system oriented viewpoints

Kahina KESSIa*, Zaia ALIMAZIGHIa, Mourad OUSSALAHb

akkessi, zalimazighi@usthb.dz, LSI laboratory, USTHB university, BP 32 AL ALIA, BAB EZZOUIAR ALGIERS 16111, ALGERIA
bMourad.Oussalah@univ-nantes.fr,  LINA Laboratory, Nantes university, 2, rue la Houssinière BP 92208, Nantes cedex 3, 44322, FRANCE

Abstract

A cooperative information system (CIS) development process includes two main steps: Requirements Engineering (RE), that includes the following two phases: (organizational reality and the conception solution) and the step of Systems Engineering that involves the implementation phase of the solution. In this paper we focus on one of the critical steps of the development process of a CIS which is RE. CIS is a complex system; it involves the cooperation of many stakeholders in a common purpose and each with its own viewpoint. This complexity makes more difficult the design; RE phase is therefore a crucial phase of the development life cycle of a CIS. This leads us to propose a meta-model which allows defining the RE phase of a CIS. We used for that a notion of software engineering: the viewpoints, in order to decompose and partition the needs of a CIS according to the viewpoint of each stakeholder, to simplify its modeling and overcome their complexity. This model-driven approach which is based on a requirements meta-model of a CIS using viewpoints, will allow us to instance the necessary viewpoints to define the needs and requirements of a CIS.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: viewpoints, needs analysis, requirements engineering, cooperative information system.

* Corresponding author.
E-mail address: kakessi@hotmail.fr
1. Introduction

Technological avalanche such as networks and SOA approaches for the architecture aspect, the object approach for the aspect of analysis and modelling methods, "Groupware" and "Workflow" for the exchange and cooperation aspect, created difficulties for developing reliable and modern Information Systems (IS) in the business strategy of service. Also in the context of globalization and the globalization of markets, companies find themselves increasingly confronted with numerous challenges posed by increased market competition of the one part and the requirements of potential customers on the other. These changes have an impact on company processes, and let it to change its organization and reconfigure these processes to be more profitable. From this, arose a new generation of IS, Cooperative Information System (CIS), which includes in the definition and elaboration of development process: the "cooperation" component. The cooperation can exist at two levels: intra-company and inter-companies when they agree to collaborate on common goals. In our case, we are interested in intra-company level.

A Cooperative Information System (CIS) is "a set of more or less autonomous components, often pre-existing working synergistically by exchanging information, expertise and coordinating their activities"\(^1\). In \(^2\), they explained that a central entity may manage a collaborative process by organizing the intervention of the partners. In \(^3\), they gave a common definition that considers cooperation as "links that build together agents to achieve voluntarily a common task."

However, we note some problem with the design of the current cooperative information systems such as:
- The size and complexity of the CIS is therefore growing, their complexity makes their design more difficult.
- In a complex environment of cooperation, the work is divided up according to objective and/or roles defined and distributed throughout teams. We then have different stakeholders and each stakeholder may have a different view on the system. We must therefore to find a set of requirements that reflect the needs of these different stakeholders and relationships that may exist between them.
- All information about the system requirements cannot be discovered by considering the system from a single perspective. Rather, we need to collect and organize the requirements from a number of different viewpoints.
- Sometimes, non-functional concerns are secondary treated compared to the central issue of functional concerns.
- Different types of information are needed to specify systems including information about the application domain, information about the systems environment and engineering information about the systems development. We find difficulties to collect and classify this information.

Therefore it is very important to understand the needs and requirements of the system, which leads us to interest and define the first step of development of a CIS, i.e. the step of Requirements Engineering (RE).

RE is the basis phase of the life cycle development of every project, defining what the stakeholders (users, customers, suppliers, developers, and businesses) in a potential new system need from it, and also what the system must do in order to satisfy that need. RE refers therefore to all activities intended to discover, analyze, validate and develop a set of requirements related to a system. It shows the satisfaction of needs and commitments throughout the life cycle of a system\(^4\). In other words, it is a process that aims to establish and maintain an agreement with stakeholders on the system requirements to be built.

For the analysis needs phase of a cooperative information system, other factors should be considered. It will be necessary to determine who does what, on what, when, after what and before what, we must therefore define the actor systems, relationships and activities that may exist between them. This implies the involvement of different stakeholders involving several levels of modelling and multiple domains.

The methods that exist in the domain of RE in software engineering does not allow to address the complex needs of a CIS that involves the cooperation of several stakeholders.

In order to try to address the needs of CIS and to remedy their complexity, we propose a meta-model that highlights the essential components of the analysis phase of the needs of a CIS as well as the concepts of cooperation, namely: (the team concept, actor, data, activity and interaction), using viewpoints, which will allow to decompose the system according to the viewpoints of each actor and team of the system, in order to avoid the design of a large system and to have a tool that will allow to have a common formalism for each concept. We based our work on our precedent papers where we proposed first a meta-model defining viewpoints for the analysis needs phase of a CIS\(^32\) and after we proposed the idea of integrating the viewpoints in a requirements meta-model of a
CIS, we focus in this paper on this idea, which is based on a meta-model of the requirements of a CIS incorporating viewpoints, we named it VpCIS (Viewpoints for Cooperative Information System).

This paper is organized as follows: In the second part we present the interest we have about the proposal of an approach oriented viewpoints for the needs analysis phase of a CIS. In the third section, we propose our meta-model oriented viewpoints: VpCIS. In the fourth part present a related work: some methods that have dealt with the same problem. Finally, we conclude with our prospects.

2. Motivation and interest of viewpoints in requirements engineering (RE)

In software engineering, the motivation of viewpoints is the separation of concerns. Thus, the viewpoints are introduced as construction elements for managing the complexities of the products artefacts (such as requirements specifications, design patterns and programs).

The end users of the system are those who will use the system individually to perform tasks in order to accomplish the objectives that the organization has assigned to them. Each user has its own viewpoint on what he expects from the system. Considering these viewpoints should help to build a system closer to the expectations of its users. The approaches based on viewpoints have favoured this perspective.

A viewpoint-based approach of requirements engineering recognizes that all information about the system requirements cannot be discovered by considering the system from a single viewpoint. Rather, we need to collect and organize requirements from a number of different viewpoints.

There is several methods oriented viewpoints:

- In the domain of software architecture we quote: Kruchten 1995 model 4+1 views: 4 + 1, ISO/IEC/IEEE 42010 standard.
- In the domain of modelling of systems, we quote: SADT, UML, VUML, VPOD.
- Among the methods oriented viewpoints in the domain of RE we have: CORE, PreView, VOSE, VORD.

A viewpoint is an encapsulation of partial information about a systems requirement. Information from different viewpoints must be integrated to form the final system specification.

The main arguments in favour of an approach based on viewpoints in requirements engineering are:
- Systems usage is heterogeneous there is no such thing as a typical user. Viewpoints may organize system requirements from different classes of system end-user and other system stakeholders.
- Different types of information are needed to specify systems including information about the application domain, information about the systems environment and engineering information about the systems development. Viewpoints may be used to collect and classify this information.
- Viewpoints may be used as a means of structuring the process of requirements elicitation.

3. Requirement meta-model oriented viewpoints VpCIS

3.1. Basic concepts of requirements engineering of a CIS

For the analysis needs phase of a cooperative information system, several factors should be considered. It will be necessary to determine who does what, on what, when, after what and before what, we must define the systems actors and relations or actions and activities that may exist between them.

As we are interested in the needs analysis phase of a CIS, we are focused on the methods of the RE domain, these methods have attempted to address the needs of systems during the requirements engineering phase and all converge to describe and highlight the basic elements of requirements engineering. This leads us to deduct the requirements of a CIS that we have to find in a meta-model of requirements of a CIS:

The most important notions are: activity, actor, data and relationships that may exist between them: (Actor - Actor, Actor - Activity, Actor - Data, Data - Data, Data - Activity, Activity - Activity).

- **Actor**: also called "agent", "participant" or "user", the actor is an entity (person or group of persons (team)), member of an organization responsible for carrying out the activities assigned to him, through his roles.
- **Activity**: An activity is a step in a process in which an elementary action (non-decomposable) is executed.
• **Data**: The object that is manipulated during the realization of an activity; the data can be used as input and can also be generated by the activity (a data can be a functional or non-functional requirements or output data).
• **Actor-Actor**: relationships that can exist between two actors, an actor can call another actor or delegate to him certain activities to do (intra link: between actors of the same team and inter link: between teams and actors from different teams).
• **Actor-Activity**: An actor realizes an activity.
• **Actor-Data**: An actor identifies data.
• **Data-Data**: A data of an actor can be used to complete the data of another actor. There can be multiple relationships between two data such as inheritance, composition ....
• **Data-Activity**: An activity uses or produces data.
• **Activity-Activity**: An activity of a given actor can be delegated and added to the actions of another actor. There can be multiple relationships between two activities such as inheritance, composition ....

From these elements that we consider to be the fundamental core of meta-model requirements of a CIS, we define the following meta-model (Fig 1):

![Fig1: CIS requirements meta-model]

Now, we have deduced the basis meta-model requirement of a CIS, we notice that some concepts important to the definition of RE of a CIS are not highlighted like team and interaction, this is why we propose a requirements meta-model of a CIS oriented viewpoints, also in order to decompose the requirements and simplify modelling.

In the following we present our approach on different modelling level:

**Level modelling**: the approach intervenes in these three layers of the architecture modelling of the OMG:
- M2, the meta-model (meta-model VpCIS);
- M1, the models: the case study (MCS: Medical Care System);
- M0, the real world (an application of MCS);

3.2. Description of the requirements meta-model of a CIS oriented viewpoints in the modeling level M2

We start by the level modelling M2 where we propose a requirement meta-model of a CIS oriented viewpoints: In order to try to meet the needs and requirements of a CIS, we propose to extend the meta-model of the requirements of a CIS shown in (Fig 1) by integrating the notion of viewpoint that we represent in the (fig 2) below, with the aim to highlight the concept of cooperation including the team concept and interaction between them and stakeholders with activity, action, and data also to minimize the complexity of a CIS by decomposing system requirements according to the viewpoints of stakeholders.

This meta-model allows to insatiate an actor and a group of actors (team) viewpoints in M1 modelling level and use these viewpoints in M0 to describe the requirements of a CIS.
In order to describe the actors and teams of a CIS, the VpCIS viewpoint is composed of description level. A level description is composed of a representation mode, a process and the core meta-model requirements of a CIS as follow:

**Process:** The process of construction can be for or by reuse. We can define for example new actor's viewpoints, in this case it is for reuse or having to reuse actor's viewpoints that already exist so this is by reuse.

**Representation mode:** representation mode of the level description. The user can choose the mode of representation that suits him to describe his needs (to describe the activities, data and roles).

We describe in the following the core:

- **Actor:** We distinguish two types of actors: simple actor and team, an actor has a role / group of roles and a goal:
  - **Name:** Name of the actor or team
  - **Representation mode:** representation mode of the actor's role
  - **Role:** This is the list of attributes, skills and expertise of an actor's CIS (participant) and it implements. This role defines the position of the actor in an organization.
  - **Group of roles:** set of roles on a team.
  - **Goal:** the objective that the actor must achieve. From the goal we can deduce the requirements; the goal is not yet a requirement.

- **Activity:** can be a simple activity (action) or a composite activity (group of actions) to be performed by an actor or a team.
  - **Representation mode:** Representation mode of an activity
  - **History:** activities history

- **Data:** a data belongs to a domain, can be functional, non-functional requirement or out-put data
  - **Domain:** actor’s viewpoint domain.
  - **Functional requirement:** is a requirement defining a function of the system to develop, describes the "what": i.e. what the system should do.
  - **Non-functional requirement:** is a requirement that characterizes a property or a desired quality of the system as performance, robustness, usability, maintainability, etc.…
  - **Out-put Data:** the data provided by the actor performing his activity.

- **Interaction:** represents the different interactions that may exist between components, these interactions can be of type: inter link or intra link as follows: (inter and intra link allows us to highlight the interaction between different viewpoints.)

  - **Intra link:**
    - **Actor-Acrtor:** Association, inheritance and composition link between a team and its actors
    - **Actor-Activity:** An actor realizes an activity.
    - **Actor-Data:** An actor identifies the data in order to accomplish its activity.
    - **Data-Data:** Composition, inheritance ... relationship between data.
    - **Activity-Data:** Activity uses or produces data.
Activity-Activity: Composition, inheritance ... relationship between activities.

Inter link

Actor-Actor: Between two teams or two actors.

Actor-Activity: The actor may have to perform an activity that has been delegated by another actor.

Actor-Data: An actor may have to use data from another actor.

Data-Data: A data of an actor may need a given data of another actor.

Activity -Data: An activity of an actor uses or produces data of another actor

Activity-Activity: An activity of an actor can be linked or delegated to the activity of another actor.

3.3. Case study in modeling level M1

We now turn to the M1 level, at this modelling level we propose a case study and we use the VpCIS actor and team viewpoints that we instantiate from the meta-model to describe the system requirements:

“Medical Care System” (MCS) is an attempt to provide the awareness of the “First-Aid” to the community in an easy, cheap and rapid way. People can interact with the system, simply by connecting with the internet and the system will show them “First-Aid” procedures for different situations that required medical treatment. The people can access the experiences and guidelines of the specialist and experienced doctors 24 / 7, for any emergency situation. The system will also facilitate the people by giving the facility for connecting with the specialist doctor.

The actors of the system are:

The Victim: This is the person who is endured in sudden illness or injury.

Rescuer: He is usually the key person who provides “first aid” in any emergency situation. In any sudden case, this person will interact with the system and try to find out the proper and accurate first aid for any injury or illness.

Receptionist: The receptionist who receives the call rescuer, and is responsible for establishing a conversation between the rescuer and the nurse or between the nurse and the doctor, or contact the system administrator in case of problems with the system.

The nurse: It is the nurse who is interacting with the system and facilitates the people who want to get first aid information on any emergency or to contact the doctor for more information.

Specialist doctor: He is a specialist doctor who is responsible for the medical treatment of any sudden case, or to provide online help to any “first aid” provider.

System Administrator: System Administrator is responsible for the maintenance and the technically issues related to the system, and he is responsible to make sure the availability of the system to the users, 24/7.

We define one team:

Paramedic: nurse, receptionist

We represent in what follow (Tab 1) an example of a simple actor viewpoint, we choose the rescuer:

<table>
<thead>
<tr>
<th>Tab1. simple actor viewpoint: rescuer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Viewpoint simple actor: rescuer</strong></td>
</tr>
<tr>
<td><strong>Attributes</strong></td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Description level</td>
</tr>
<tr>
<td>Process</td>
</tr>
<tr>
<td>For reuse</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>rescuer</td>
</tr>
<tr>
<td>Representation mode of the role</td>
</tr>
<tr>
<td>Use case</td>
</tr>
<tr>
<td>Role</td>
</tr>
<tr>
<td>Goal</td>
</tr>
<tr>
<td>- Help the victim by consulting the system to find the method and adequate first aid procedures in case of emergency</td>
</tr>
<tr>
<td>Representation mode of an activity</td>
</tr>
<tr>
<td>textual</td>
</tr>
</tbody>
</table>
**Activity**
- Consult the system to find first aid and assist the victim.
- Contact the paramedic if not understanding the instructions.
- Tell the doctor the problem with the victim.

**History**
- Accident
- Consult the MCS system
- Treat the victim, otherwise in case of problems call the paramedic
- Follow the doctor's instructions to treat the victim and take him to the hospital.

**Data**
- The system name to find it online
- First aid methods
- The paramedic's phone number to ask for help

**Domain**
- Required data (functional requirement)**
  - Rescuer domain

**Constraints on data (non-functional requirements)**
- System availability
- A minimum waiting time to find the first aid on the system or have the paramedical and medical specialists

**Provided data**
- First aid care to the victim

**Interactions**

**Intra link**
- Actor-Action link
  - The rescuer performs the action: consult the system, help the victim
  - The rescuer performs the action: contacts paramedic.

- Actor-Data link
  - The rescuer needs to know the link of the system and phone number to contact the paramedical

- Activity-Data link
  - In order to join the medical the paramedic, they must have a phone number
  - Once the action performed, it provides first aid

- Actor-Actor link
  - Between the rescuer and paramedical
  - Between the rescuer and the doctor

In the level modeling M0, the approach consists on an instantiation of the viewpoints from the level modeling M1.

**4. Related work**

There are several methods who proposed solution to the needs analysis phase of a CIS, like methods oriented goal (Tropos I*, KAOS28): Goal modelling is an effective way to identify requirements, but the problem with these methods is that: it is difficult for domain experts to deal with the fuzzy concept of a goal add to this, the goal process does not reflect the actual situation but an idealized environmental one and it is difficult to analyze the goal oriented models because they are too large. We note also that these methods doesn't highlight the notion of teams (group of actors) and their descriptions.

The method CREWS5,6 proposed a solution to find the system requirements by coupling each goal discovered with a scenario that illustrates the behaviour of the system to achieve the goal in order to overcome the problems related to goals and scenarios. The method has demonstrated its effectiveness in the requirements specification of a cooperative process, but we note that the method describes a meta-model in the modelling level M2 and proposes after a case study but doesn't distinguish between the level modelling M1 and M0, add to this the method doesn't decompose the requirements which make it difficult to analyze.

The method MAMIE7, the method drew inspiration from the CREWS method by coupling the goal and scenario and added viewpoints to describe non-functional requirements for the elicitation of requirements for a cooperative information system. This method lets to well describe the requirements elicitation, but as in CREWS we note that MAMIE describes a meta-model in the level modelling M2 and proposes after a case study but doesn't distinguish between the level modelling M1 and M0 and doesn't highlight explicitly certain necessary concepts to
the definition of needs and requirements of a CIS as the interactions that may exist between actors and teams for example.

Concerning the approaches oriented viewpoints in requirements engineering like (VORD or VOSE), these approaches are well used in their domain but they are principally concerned with looking at viewpoints in isolation rather than as cooperating entities and are too rigid. They are based around the idea of a single type of viewpoint and require the specification to be fitted around this concept.

In general, in these inspiring methods the authors do not explicitly distinguish between the level of M1 and M0 modelling and have not explicitly set out some concepts needed to describe the needs of a CIS as the team concept and interaction between teams and actors, add to this it is difficult to analyze the goals and scenario-oriented models because they are too big. Viewpoints oriented approaches aim to model the domain from different perspectives in order to develop a comprehensive and coherent description of the targeted system. This leads us to proposed our meta-model oriented viewpoints, which decomposes the requirements according to the viewpoint of each stakeholder (actor/team) and avoids the design of a big system, add to this, it highlights the components needed to define the requirements of a CIS and the interaction between viewpoints.

5. Conclusion

In this article we discussed our motivations for the use of viewpoints in the analysis phase of a CIS, we have subsequently proposed a meta-model of requirements of a CIS using viewpoints: VpCIS in modelling level M2 to decompose the system requirements and simplify modelling. We instantiate after that the necessary viewpoints (team, actor) to describe the needs of a CIS in level modelling M1 in order to apply them to level modelling M0. We expect to develop a tool that will allow us to support this approach in the level modelling M1 and use it to describe the requirements in level M0.

References

13. OUSSALAH M. Architectures logicielles : principes, techniques et outils ; Traité RTA, série Informatique et Systèmes d'Information. 2014.