Extended Kaos to Support Variability for Goal Oriented Requirements Reuse

Farida Semmak[1], Christophe Gnaho[1,2], Regine Laleau[1],

[1]Laboratoire Algorithmique, Complexité et Logique, Université Paris Est,
61 avenue du général de Gaulle, 94010 Créteil cedex
{semmak, laleau}@univ-paris12.fr

[2]Université Paris Descartes, 45 rue des Saints-pères, 75006 Paris
christophe.gnaho@math-info.univ-paris5.fr

Abstract. This work is done as part of the Tacos project[1] whose aims is to define a component-based approach to specify trustworthy systems from the requirements phase to the specification phase, in the Cycab transportation domain. This paper mainly deals with the improvement of requirements elicitation in the context of Cycab domain. For that purpose, we propose to extend the Kaos goal oriented metamodel in order to enable explicit representation of variability at the early-phase of requirements engineering. This extension allows specifying a requirements family model which integrates both reusable assets and a variability model. The latter expresses the relevant domain facets along with different variants to realize them. The facets allow to structure and organize domain knowledge for reusability. The variability model then enables designers to explicitly state strategic decisions for requirements model development and then choose more accurately the relevant options of the system to-be.

Keywords: Requirements engineering, Family model, Variability, Land transportation domain.

1 Introduction

This paper presents first results carried out within the framework of the TACOS (Trustworthy Assembling of Components: frOm requirements to Specification) project. This project aims at defining an engineering approach beginning with functional and non functional goals and ending with formal specifications organized as components that verify some properties such as security, efficiency, fault tolerance, etc. [1].

---

[1] The TACOS project (Ref. ANR-06-SETI-017) is partially supported by the French National Research Agency

The application domain is the land transportation domain and more precisely the Cycab domain. A Cycab is a public vehicle with fully automated driving capability [2], [3]. Our team is concerned with the phase of requirements engineering.

Nowadays, despite a wide variety of available modelling approaches, natural language remains the main way for describing requirements in the Cycab domain. Some projects in automotive domain, such as Memvatex [4] have already used a model driven engineering approach based on UML. However, it is recognized that UML based requirements engineering are better adapted to the late-phase of the requirements engineering process, during which initial problem statements are precisely reformulated and analyzed [5] [6] [7] [8]

For the Cycab domain that requires sound requirement elicitation process, we need a requirement engineering approach which addresses the early-phase of requirement engineering during which stakeholders intentions are explored and the different alternative ways to satisfy these intentions are investigated. Goal oriented approaches have been found to be effective for this purpose.

Furthermore, the embedded systems in Cycab vehicles represent a diversity of applications for which design effort could be capitalized for reuse, notably by capturing in an integrated view, the common and variable requirements they must meet.

In this paper, we are interested in managing variability at the early-phase of requirements engineering. Thus, we propose some extensions to the Kaos Goal oriented metamodel [9] [6], with variability concepts, in order to be able to specify Requirements Family Model (RFM). This requirements family model will then enable to derive different specific models according to the needs of the stakeholders.

The paper is organized as follows. Section 2 presents an overview of the proposed approach. Section 3 deals with the application of Kaos to the Cycab domain along with some related issues. Section 4 presents the extended Kaos metamodel. Related work is given in Section 5. Finally, Section 6 concludes with some remarks about the results and future works.

2 Overview of our Approach

In the proposed approach, we attempt to apply reuse based techniques at goal level, in order to improve requirements engineering in the context of Cycab domain. These techniques are inspired from the field of software product lines engineering [10] and domain engineering [11]. The aim is to provide requirements model that captures commonality and variability of domain, from which requirement models for specific systems can be derived according to some options selected by the stakeholders.
Figure 1 presents an overview of the proposed approach.

The domain level provides the *Requirements Family Model (RFM)*, which enables the description of the large diversity of applications of the same domain, by identifying and expressing the common and variable requirements at goal level

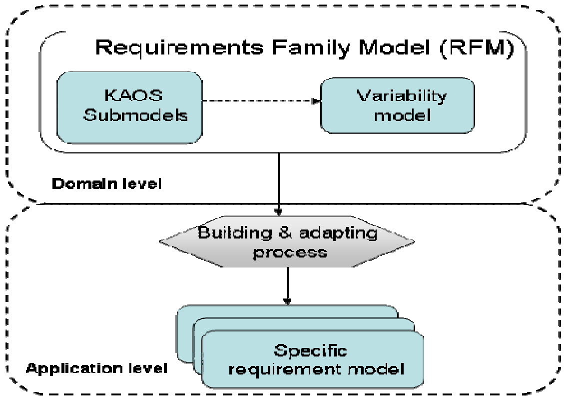

Fig. 1. *Overview of the approach*

To specify the Requirements family model, we have chosen the Kaos Goal-Oriented Requirement Engineering approach. However, the Kaos approach has not been designed to address a class of systems of a given domain and then it does not explicitly take into account variability concerns. Variability is defined as the ability of an element (component, system, model…) to be changed, personalized and configured according to a specific context. For this purpose, we propose to extend Kaos metamodel with some variability concepts. Theses extensions are described in more details in section 4.

The application level enables the building of specific requirements model. Its main component is the *Building and adapting process* that main purpose is to derive the *specific requirement model* from the RFM, according to the needs of the stakeholders.

In this paper, we mainly focus on the domain level that we illustrate by examples taken from the Cycab domain.

3 Applying Kaos Approach to the Cycab Domain

3.1 The Cycab Domain

The Cycab concept is designed by INRIA and widely exploited as an experimental platform by many researchers [2], [3]. Cycabs are small electric vehicles, designed for restricted access zones (historic city centres, airports, train stations or university campuses). They are controlled by embedded electronics which allow the automatic driving under computer control. In addition, the Cycab concept describes a set of features for a new type of automatic vehicles and thus leads to several realizations that are concrete systems.

In preliminary works within the Tacos project, we have already built with Kaos and its tool *Objectiver* [12], a requirements model for a simplified Cycab [13].

3.2 Modeling the Cycab Domain with Kaos

Based on our experience in using KAOS approach, this section seeks to show through an example from the Cycab, some lacks of using KAOS for the representation of variability at the requirements phase.

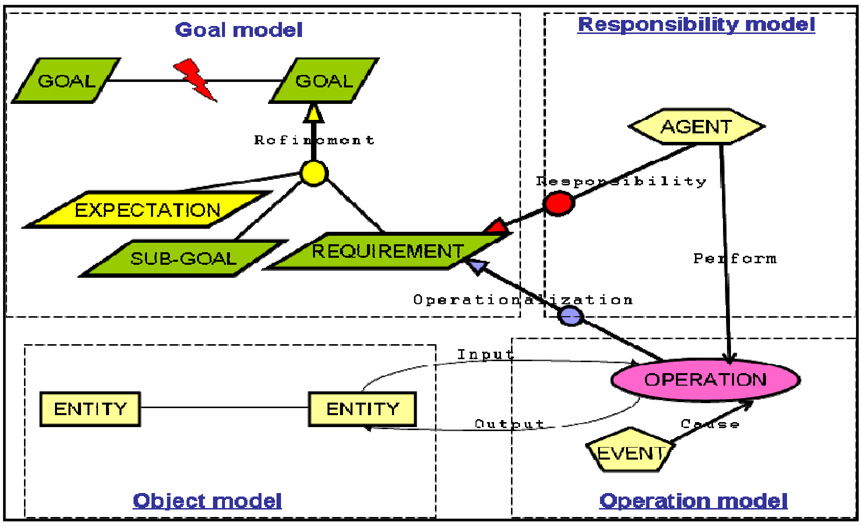

Fig. 2. *Overview of the KAOS sub-models*

As shown in Figure 2, a KAOS specification is composed of the following four models that are related through inter-model concepts and consistency rules [9], [5], [6]:

A Goal model allows capturing the objectives of the system-to-be. Goals are organized in the usual AND/OR refinement abstraction hierarchies. A goal can be refined into several alternative combinations of sub-goals. A goal that cannot be more reduced and that is assignable to an agent is a requisite. A requisite that is placed under the responsibility of an agent in the system is a *requirement*, whereas a requisite that is under the responsibility of an agent in the environment of the system is an *expectation*.

A Responsibility model captures responsibility assignment of goals to agents. Agents are either humans or automated components that are responsible for achieving goals.

An Object model is an UML model which captures the concepts of the application domain that are relevant with respect to the known requirements.

An Operation model describes all the behaviours that agents need, to fulfil the requirements they are responsible for. Behaviours are expressed in term of operations on objects which are performed by agents. In the following, we mainly focus on the goal model, which is the driving model of KAOS.

Let us consider the high level goal "Cycab transportation requests satisfied". There are many ways to fulfil this goal according to the different options considered by the stakeholders. One possible option is to have a Cycab with "on demand calling mode", which refers to a mode in which the passenger should be able to call the Cycab (for example by cellular phone or at point), another is a Cycab with "automatic calling mode" – a Cycab stopping at each station. Figure 3 presents an excerpt of the goal model which describes this situation. *Alternative refinements* provide a natural way to represent these two options. In order to visually make a difference between them, we use dashed links for the "automatic calling mode" option.

Each parallelogram in the diagram represents a goal. Thick-bordered parallelograms express *expectations.* Circles represent refinement (*AND-refinement*) of a parent goal (the one pointed to by the arrow) to a list of sub-goals. Alternatives (*OR-refinement*) are represented by distinct refinements.

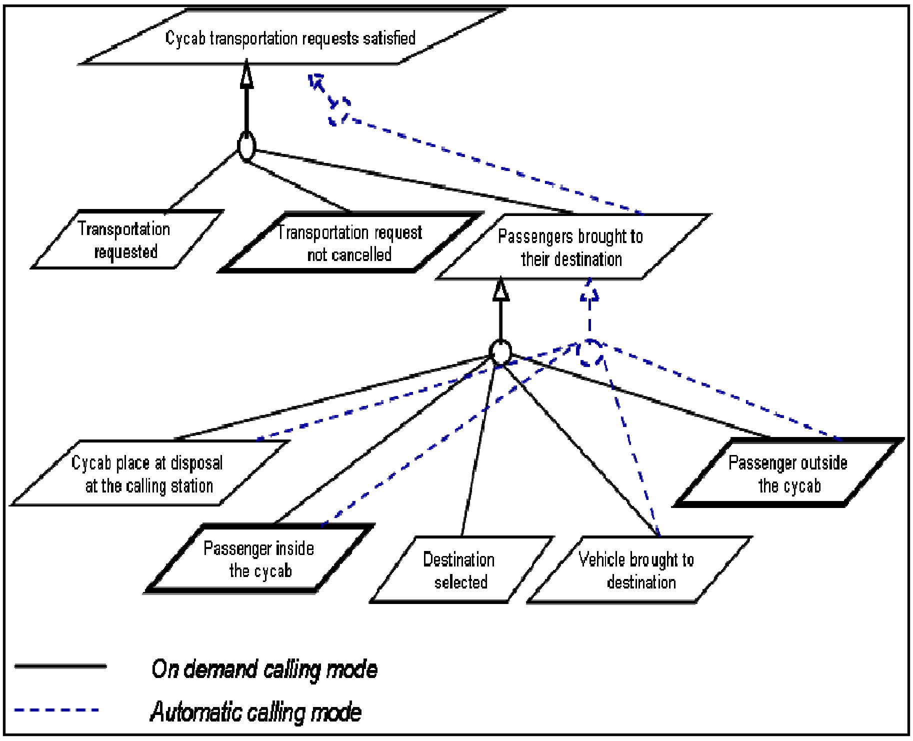

Fig. 3. *Partial Goal model of the simplified Cycab*

According to the "on demand calling mode" option (see Figure 3), the goal "Cycab transportation requests satisfied" is reduced into three sub-goals: "transportation requested", "transportation request not cancelled" and "passengers brought to their destination". On the other hand, with the "automatic calling mode" option, the same goal is only reduced in "passengers brought to their destination" sub-goal. The "passengers brought to their destination" sub-goal is then respectively refined into four or five sub-goals according to the considered option. Thus, the sub-goal "Destination selected" has not to be taken into account with the "automatic calling mode".

3.3 Issues on Representing Variability with Kaos

We have shown in the previous section that it is possible to represent variability in stakeholder goals, through the concept of *alternative refinement*. However, we can argue that due to the experience in the TACOS project, the only use of Kaos model does not enable to deal with all the issues related to variability concerns. Those issues are briefly presented below.

- **First issue**: How to deal with an important number of options?

In the Cycab domain, an important number of options may be considered by the stakeholders, resulting in an important number of alternative ways to achieve the identified goals. Thus, the question is how can these goals be refined to address all the options, while avoiding readability and combinational explosion problems? (See Figure 3)

- **Second issue**: How to deal with global variability?

Alternative refinement is better adapted to represent a kind of variability that is local to a goal in the model. However, variability may concern different parts of the goal model meaning that a given variant can have an impact on many parts of the graph. Thus, some variability in the requirements cannot be only represented by alternatives.

The rest of the paper addresses these issues. The next section presents the extensions made to Kaos.

4 Cycab Domain Modeling with Extended Kaos

This section deals with the extension of Kaos; we mainly focus on the extensions proposed in the first two sub-models which we illustrate through examples from the Cycab case study.

4.1 The Extended Metamodel

The extensions made to Kaos must show *'what does vary'* in the Kaos sub-models and *'how does it vary'*. What does vary is represented thanks to the concept of *variation point* and how does it vary is described by the concepts of *facet* and *variant*. These concepts form what we call *variability model* and the *variation point* relates this model to the Kaos Sub-models.

Figure 4 presents on the right side, the concepts (grey boxes) extending the Kaos metamodel.

A **domain** is defined as a class of similar applications which can be specified by commonalities and variability's information. The variability model focuses on the relevant domain knowledge that presents multiple options of realization. An underlying problem is then to structure and organize this knowledge for understanding and reusability. We adopt and extend the concept of facet as defined in [14]. So, a domain may be featured by many facets.

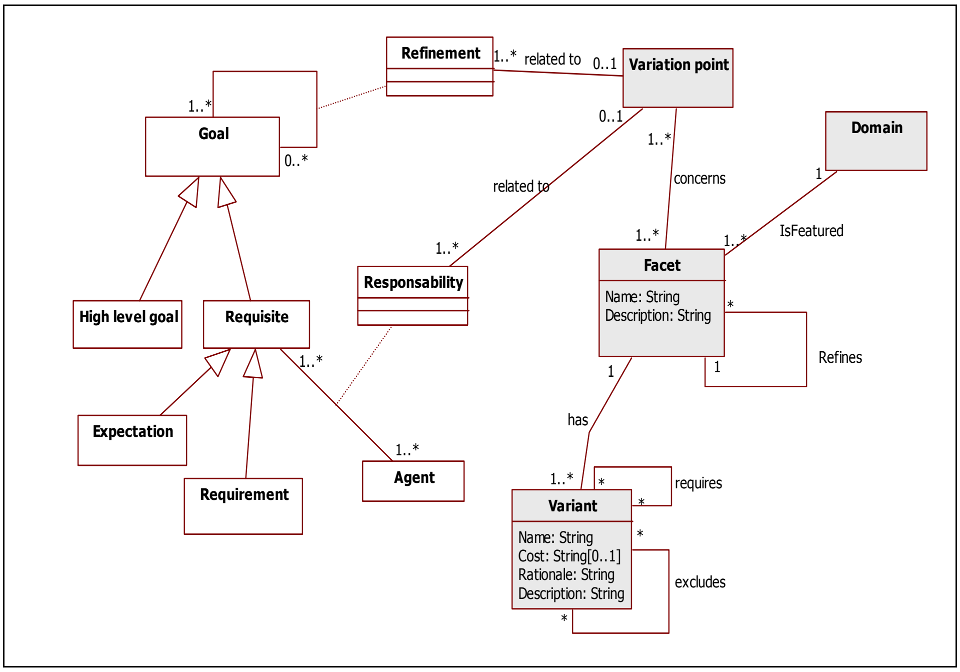

Fig. 4. *The extended KAOS metamodel*

A facet is then defined as a viewpoint or a dimension having an interest for domain. For instance, the Cycab transportation domain is characterized by several facets like: "the localization mode" which deals with the vehicle localization from external sensors, "the Road type" to precise the kind of route where the vehicle moves and so on. A facet is described by the properties: *name, description*. For instance, the facet having the name "F1: localization mode" and the description "to determine the vehicle position".

A facet has one or many **variants**. A variant is defined as a way to realize a facet. For example, a Cycab may be localized by using a GPS sensor (Global Positioning System) or a WPS sensor (Wifi Positioning System) or the internal sensor or a combination of those three variants. Thus, to the facet "F1: localization mode" are attached the following variants: {V1: GPS, V2: WPS, V3: internal sensor}. A variant is described by the following properties: *name*, *description*, *cost*, *rationale*. For instance, one of the variant of the facet F1 has a name "GPS", a description "localization by measuring signal propagation time from different satellites", a cost "Ø" and a rationale "efficient if the localized area is not surrounded by high building". The *cost* and *rationale* properties support decisions taken by the designer.

The concept of **variation point** provides a means to make explicit variability in the Kaos sub-models and by this way to precise what does vary in these models. As shown in Figure 4, one variation point is related to one (or several) Refinement association-class meaning that in this place, there are several ways to refine the goal. As shown in Figure 4, the assignment of an agent to a goal (requisite) is captured in the metamodel by the Responsibility association-class. A variation point is related to one (or several) Responsibility association-class meaning that in this place, there are several ways to assign an agent responsible of the requisite.

Finally, as shown in Figure 4, different dependencies between facet and variant may exist. These dependencies are detailed in [1].

4.2 A Partial Version of Cycab Requirement Domain

- Variability in Goal model

Let's take again the example of Figure 3 and illustrate the explicit representation of the variation points at different levels of goal hierarchy. This example deals with the high level goal "Cycab transportation requests satisfied". Figure 5 shows variation points attached to refinement links.

These variation points concern the facet named "F1: Cycab calling mode" realizable in the following two variants: "V1: automatic" (a Cycab stopping at each station) and "V2: on demand" (it stops at a station only if there is an external or internal demand). To represent the two refinement alternative as described in Figure 3, the refinement link is annotated by the couple <Facet-Variant> (see Figure 5). A refinement link without variation point means that the reduced goal is common to any application of domain.

Moreover, the same facet-variant can impact on other parts of the graph. For instance, as shown in Figure 5, the variation point <F1, V1> has an impact on two parts of the graph. Consequently, the sub-goal "passengers brought to their destination" is refined into five low-level sub-goals: the four mandatory sub-goals (see links without variation points) and the sub-goal "Destination selected" (with the variation point <F1, V1>).

On the other hand, if the calling mode is manual <F1, V1>, the goal "Cycab transportation requests satisfied" is refined into one sub-goal: "Passengers brought to their destination". This latter is only refined into four mandatory sub-goals. So the sub-goal "Destination selected" has not to be taken into account.

In this example, we have presented how the variation points attached to refinement links may specify many alternatives of goal refinement. It is a way to deal with the important number of options and then to reduce the combinatory explosion. Moreover, the same variation point can have an impact on several parts of the graph.

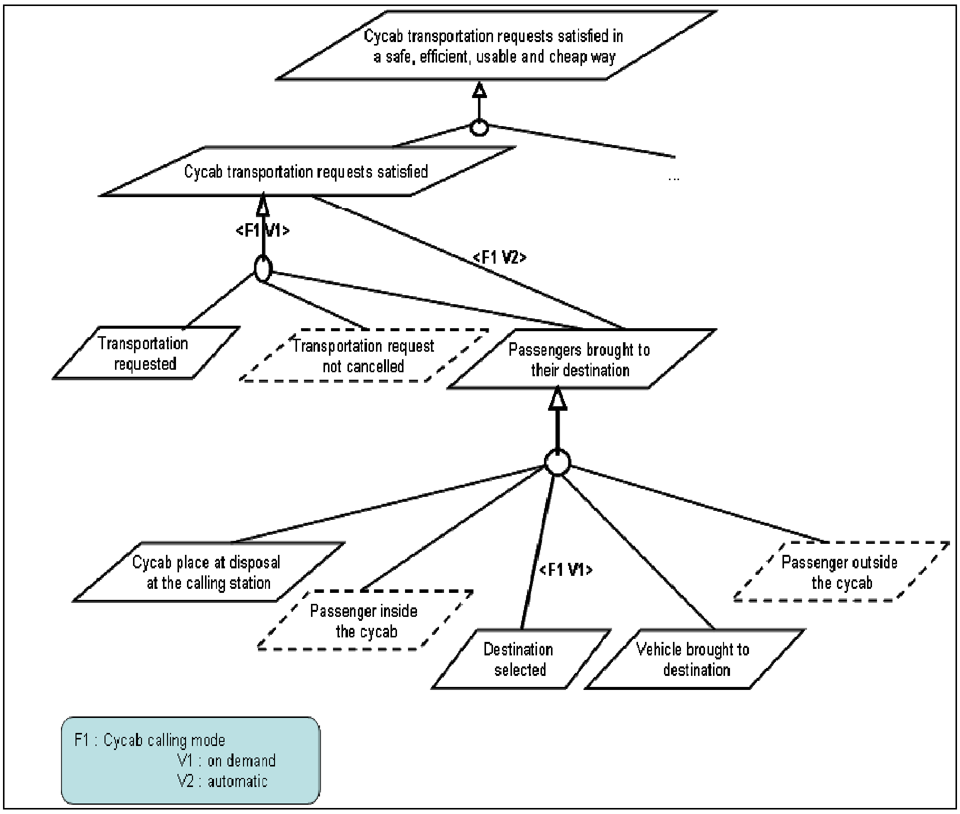

Fig. 5. An instance of the extended KAOS goal metamodel

- Variability in responsibility model

Expression of variability in this model is based on the same principle as in the goal model. The assignment of an agent to a requisite can be referred by variation points as shown Figure 4.

Let us consider others facets like "F2: Cycab doors opening/closing mode" with the variants "V1: manual" or "V2: automatic"; and "F3: Cycab driving mode" that can be either "V1: manual" (human driver) or "V2: automatic" (control system driver). Agents are represented by hexagonal boxes and requisites are represented as thick-bordered parallelograms.

Figure 6 thus shows the variation points related to responsibility link assigning agent to requisite. The agent is determined according to the facet-variant <F3, V1>, <F3, V2>. Indeed, the requisite "Cycab in movement towards the calling station" is under the responsibility of "driving agent". The decision to consider this requisite either as an expectation or as requirement will be taken only when building a specific model.

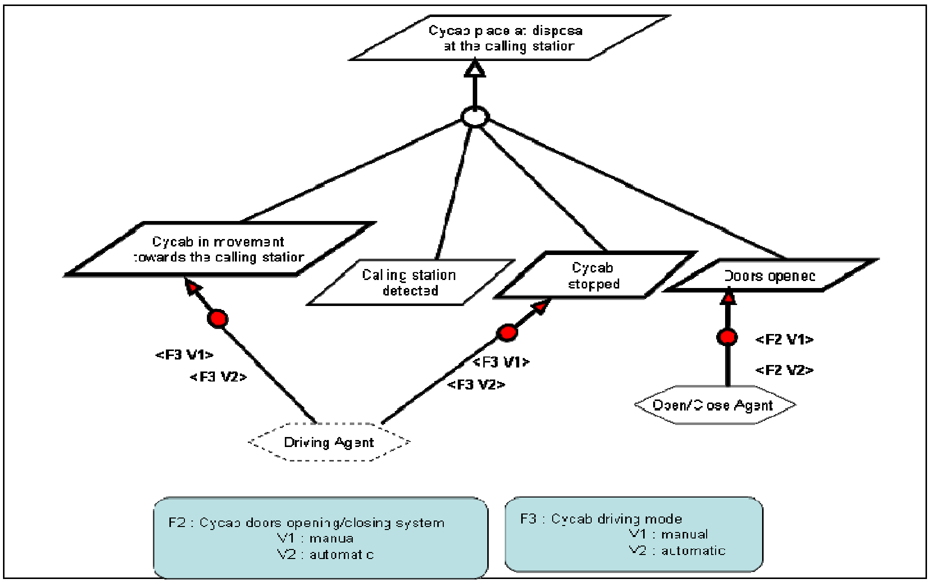

Fig. 6. *Reduction of the goal Cycab place at disposal at the calling station*

For instance, in Figure 6, the requisite "doors opened" can be specialized in a requirement or in an expectation depending on the choice done by the designer. If the facet-variant <F2, V2> ("automatic doors opening/closing mode") is selected then this requisite will be placed under the responsibility of the system agent "Driving system" and consequently will be specialized in a requirement. On the other hand, if the facet-variant <F2, V1> ("manual doors opening/closing mode") is chosen, this requisite will become an expectation because it will be under the responsibility of the "Passenger" which is an agent in the system environment.

By considering the first issue, the proposed concepts allow to deal with an important number of options avoiding by this way a combinatory explosion. According to the second issue, they allow to represent the same variation point at different places of goal model or any Kaos model. In addition, properties attached to facet and variant help taking decision and then strengthen building and adapting process of domain applications.

5 Related Work

Variability is the key challenge in building reusable infrastructure. It has already been widely studied more particularly at the software level [15], [14]. In [16], the authors explore the dimensions of Variability from a requirements engineering perspective. At domain analysis level, the Foda method [11], the most popular, has been the first one to propose the concept of *Feature*, as characteristics that discriminate systems in a domain. A Feature is defined as *a prominent or distinctive user-visible aspect, quality or characteristic of a software system or systems*. It allows building a feature model in the form of hierarchy sets. The concept of feature has been widely used and extended. The method Rseb which proposed the concepts of *variation point* and *variants* has also integrated the concept of Feature [17]. Intensive works have been done in the domain of Software Product Lines (SPL) engineering [10], [18]. The SPL approaches have used both the concepts of *Feature* and *variation point* with *variants*. They are the first ones to propose the specification of a variability model which is independent from reusable assets, while related to them. In Foda, a reusable asset is a feature which is mandatory, optional or alternative whereas in SPL approaches, reusable assets are essentially use cases or object classes related to variant, related itself to variation point.

We adopt a variability model like in the SPL approaches. We believe that it is necessary to differentiate the variability model from reusable assets because, first, it enables to represent a richer variability while reducing combinatory explosion and second, variability model becomes an effective support to designers in building an application of domain.

On the other hand, variability can be analyzed at different abstraction levels, the sooner the better. The approaches like Foda or SPL do not deal with variability at goal-oriented requirement level. Some approaches have studied variability at an early requirement engineering step [19] [20]. The approach of Liaskos is based on the semantic characterization of Or-decompositions of goals. In our approach, the And-refinement link is related to variation point indicating through this, the multiple alternatives of goal refinement.

Moreover, we have proposed the notion of facets in order to enhance and represent the discriminatory elements between domain applications. Indeed, we think that the concepts of variation point and variant are not sufficient to represent the complexity underlying the domain applications. The variation point represents only what does vary, and not the nature of which vary. The concept of facet represents a domain viewpoint or dimension. The facets allow classifying and organizing domain knowledge. The notion of facet has been used in library science to classify library domain [14] and we use it in order to make easier understanding and organizing domain knowledge.

6 Conclusion and Future Works

In this paper, we have proposed to extend the Kaos metamodel with variability model in order to provide Requirements Family Model. The variability model captures the relevant domain facets and variants that realize them. This model therefore emphasizes differences between applications of a same domain. It enables designers to explicitly state strategic decisions for requirements model development and then to choose more accurately the relevant options of the system-to-be.

Short-term activities consist in applying the variability concept to object and operational Kaos models, and defining the process which allow a designer to build specific requirement models from a Requirements Family Model. More long-term concerns will be to develop a tool to support the approach and to build formal models that take into account variability.

References

1. TACOS Project, ANR-06-SETIN-017, programme SETIN'06, http://tacos.loria.fr, 2007
2. Fraichard, T., « Cybercar: l'alternative à la voiture particulière », *Navigation* (Paris), Volume 53, N° 29, Janvier 2005
3. Parent, M.: « Automated public vehicle: a first step towards the automatic highway », *Proc. Of the World Congress on Intelligent* (1998)
4. Albinet, A., Boulanger, J-L., Dubois, H., Peraldi-Frati, M.A., Sorel, Y., Van Q.D.: « Model-based Methodology for Requirements Traceability in Embedded Systems », Proc*. of the 3rd ECMDA Traceability Workshop* 2006
5. Lamsweerde, A.: « From Systems Goals to Software Architecture », In Formal Methods for Software Architectures, *LNCS vol. 2804, Springer*, 2003
6. Lettier, L., Reasoning about agents in Goal-oriented R. E., PhD Thesis, 2001
7. Rolland, C., Souveyet, C., Ben Achour, C.: « Guiding Goal Modelling Using Scenarios», *IEEE T.S.E, Special Issue on Scenario Management*, 1998
8. Yu, E.: «Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering », *In Proc. of the 3rd IEEE Int. Symposium on R.E.* 1997
9. Dardenne, A., van Lamsweerde, A., Fickas S.: « Goal-oriented Requirements Acquisition », *Science of Computer*, 1993
10. Pohl, K., Bockle, G., van der Linden, F., *Software product line engineering: Foundations, Principles, and Techniques*, Springer 2005
11. Kang, S., Cohen, J., Hess, W., Novak, Peterson S.: « Feature-oriented domain analysis (FODA) feasibility study », CMU/SEI-90-TR-21, 1990
12. Objectiver Requirement Engineering tool, http://www.objectiver.com/
13. Brunet, J., Gnaho, C., Laleau, R., Semmak, F.: Identification of domain-related properties. Technical report, TR-2007-1, Lacl, 2007
14. Prieto-Diaz, R.: « Implementing Faceted Classification for software reuse », *Communications of the ACM*, Vol. 34, N°5, 1991
15. Frakes, W., Pole T.: «An empirical study of representational methods for reusable software component », In *IEEE TSE, Vol. 20, N°8*, (1994)
16. Liaskos S,. & Al.: « Exploring the Dimensions of Variability: a R.E Perspective », 1st *Workshop on Variability Modelling of Software-intensive Systems*, 2007
17. Griss, M., Favaro, J., d'Alessandro, M., « Integrating feature modeling with the Rseb », In *Proc. Of the 5th int. Conference of Software Reuse*, 1998
18. Bachmann, F., Goedicke, M., Leite, J., Nord, R., Pohl, K., Ramesh, B., Vilbig, A.: « A meta-model for representing variability in product family development », *PFE wokshop, LNCS 3014* 2004
19. Liaskos S., Lapouchnian A., Yu Y., Yu E., Mylopoulos J.: « On Goal-based Variability Acquisition and Analysis », *Proc. Of the 14th IEEE Int. Conf. on R.E.*, 2006
20. Semmak, F., Brunet, J.: «Variability in Goal-oriented Domain Requirement », 9th Int. *Conference on Software Reuse (ICSR), LNCS vol. 4039*, (2006)