VAODA
A Viewpoint and Aspect-Oriented Domain Analysis Approach

António Pedro Rodrigues and João Araújo
CITI/Departamento de Informática, Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa, Quinta da Torre, Caparica, Portugal

Keywords: Aspect-Oriented Requirements Engineering, Domain Analysis, Feature Modelling, Viewpoints.

Abstract: Requirements approaches can be integrated to specify domain requirements. Among them, viewpoint-oriented approaches stand out by their simplicity, and efficiency in organizing requirements. However, none of them deals with modularization of crosscutting concerns. Aspect-Oriented Domain Analysis (AODA) is a growing area of interest as it addresses the problem of specifying crosscutting properties at domain analysis level. The goal is to obtain a better reuse at this abstraction level through the advantages of aspect orientation. This work proposes an AODA approach that integrates feature modeling and viewpoints.

1 INTRODUCTION

Software reusability is key to significant productivity gains and its success is granted mostly by the use of Domain Analysis (DA). According to the Software Engineering Institute, DA is "*the process of identifying, collecting, organizing and representing the relevant information in a domain, based upon the study of existing systems and their development histories, knowledge captured from domain experts, underlying theory, and emerging technology within a domain*".

DA is a means to build reusable infrastructures to support the specification and implementation of restricted classes of applications. Some domain requirements can be referenced as crosscutting as they "cut across" other domain requirements, and that can be a problem when changes in the domain requirements occur, causing problems to system evolution. To overcome this issue, some approaches have been developed within the software engineering community, like Object-Oriented Domain Analysis (OODA) (Shlaer and Mellor, 1989), and Feature-Oriented Domain Analysis (FODA) (Kang et al., 1990).

But none of these approaches are enough to specify crosscutting properties of the domain. Aspect-Oriented Requirements Engineering (AORE) techniques can be easily adapted to do this such as Arcade (Rashid et al. 2003) which combines aspects with viewpoints. In this paper we will propose VAODA: a new approach which extends the Arcade approach with feature modelling.

This paper is organized as follows. Section 2 presents and applies our approach to using a case study. Section 3 draws some conclusions and highlights some future work.

2 THE VAODA APPROACH

Most of feature modelling approaches have little integration with other domain specification approaches. Their integration should give a broader vision of the domain. Crosscutting concerns are not specifically addressed, so the approach we propose in this paper tries to fulfil that gap. This approach integrates domain analysis with aspects, feature models and viewpoints and it should specify models at both domain and application engineering. To illustrate the approach it will be used a car park system. The requirements are stated as follows.

A car parking system may be accessed through a ticket, a gizmo or a card. In the first case a client has to get a ticket from a machine after pressing a button. Afterwards, the car is allowed to enter and park in an available place. When s/he wants to leave the parking place, s/he has to pay the ticket. The amount depends on the time spent. After paying the client can leave by inserting the ticket in a machine.

In the second case a gizmo can be purchased through a simple process where the client gives

Rodrigues A. and Araújo J. (2009).
VAODA - A Viewpoint and Aspect-Oriented Domain Analysis Approach.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 351-354
DOI: 10.5220/0001960203510354

her/his personal info, her/his debit card and the vehicle which will be associated to the gizmo. To enter the park it is necessary to have the gizmo placed in the vehicle's windscreen, and as the driver chooses to enter the park pressing a button, the gizmo is read and the gate is opened. To exit the park the process is similar. The amount to pay is shown in a display. This amount is sent to the client's bank in order to debit. In the third case, the client can use a pre-purchased card that is used to enter the park and will be debited every time he leaves the car park. The system has to control if the car parking is full or not and if it is closed or not.

2.1 VAODA in Domain Engineering

VAODA at Domain Engineering (DE) level is composed by six main activities: Identify and specify domain viewpoints and concerns; Specify a feature model; Identify crosscutting requirements; Specify aspectual viewpoints; Compose aspectual viewpoints; Integrate aspect-oriented viewpoints with feature model (see Figure 1). They will be described in detail and illustrated with the case study next.

Identify and Specify Domain Viewpoints and Concerns. Viewpoints represent an encapsulation of partial information about the system's requirements, of a particular perspective. They can be associated to stakeholders, environment and system domain. A viewpoint-oriented approach to requirements elicitation recognizes that all information cannot be obtained only considering one perspective. The viewpoints identified in this example are: ATM, Bank, Driver, Entry Machine, Exit Machine, Gizmo, Park, Park Attendant, Paying Machine, Vehicle and Vehicle Owner. Examples of relevant concerns are: Availability, Multi-Access, Response Time, Security, Usability. Templates of the viewpoints Entry Machine and Exit Machine are shown in tables 1 and 2.

Specify a Feature Model. A feature model represents a hierarchy of properties of domain concepts. Feature modeling was proposed as part of the Feature-Oriented Domain Analysis (FODA) (Kang et al., 1990) method, and since then, it has been applied in a number of domains. Based on the information provided by the viewpoint requirements and the domain experts we can derive a feature model. In our example, figure 2 shows a feature model for the case study. In our approach this is done manually, by analyzing the viewpoint requirements.

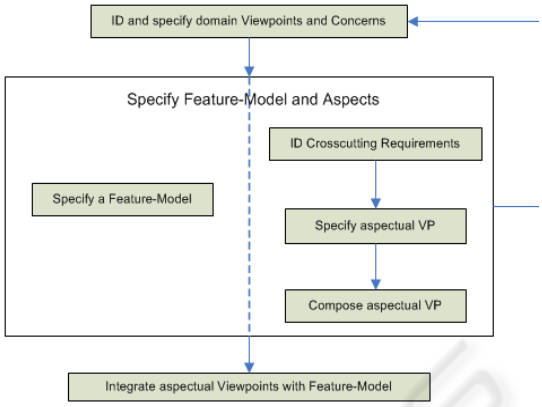

Figure 1: VAODA process in DE.

Table 1: Template for the Entry Machine viewpoint.

Name	Entry Machine
Focus	Registers vehicle entry in the park
Concerns	Availability, Correctness, Multi-Access, Response Time, Security
Source	Assignment
Requirements	1 - Entry Machine detects vehicle. 2 - Entry Machine detects entrance mode. 2.1 - Entry Machine gives ticket if ticket button is pressed. 2.2 - Entry Machine detects and reads Gizmo if Green Lane button is pressed. 2.3 - Entry Machine receives, reads and gives card back if Card is inserted. 3 - Entry Machine opens gate if entrance mode is valid. 4 - Entry Machine closes gate when the passage of the vehicle was detected. 5 - Entry Machine calls Park Attendant if Park Attendant button is pressed.
Change History	
Type	Non-Aspectual

Table 2: Template for the Exit Machine viewpoint.

Name	Exit Machine
Focus	Registers vehicle exit from the park
Concerns	Availability, Correctness, Multi-Access, Response Time, Security
Source	Assignment
Requirements	1 - Exit Machine detects vehicle. 2 - Exit Machine detects exit mode. 2.1 - Exit Machine receives ticket. 2.2 - Exit Machine detects and reads Gizmo if vehicle entered using Green Lane. 2.2.1 - Exit Machine retrieves system info on the debit value. 2.2.2 - Exit Machine Displays the debit value. 2.3 - Exit Machine receives, reads and gives card back if Card is inserted. 3 - Exit Machine opens gate if exit mode is valid. 4 - Exit Machine closes gate when the passage of the vehicle was detected. 5 - Exit Machine calls Park Attendant if Park Attendant button is pressed.
Change History	
Type	Non-Aspectual

For example, Entry Machine can be defined as a feature decomposed into several obligatory features (e.g. display, gate, intercom, park attendant button and vehicle sensor), and some variable features, in this case the kind of access to the park (e.g. by card, by ticket or using Green Lane gizmo).

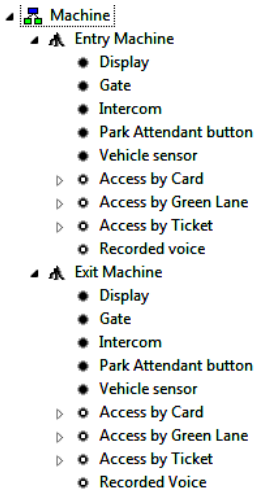

Figure 2: FMP model for the case study.

Identify Crosscutting Requirements, Specifying and Composing Aspectual Viewpoints. When there are requirements scattered in several viewpoints, those are called "Aspectual Requirements" and are modularized in a new "Aspectual Viewpoint" that will now be defined. Looking at the viewpoints above one can notice that some requirements are very *similar*, they only defer in the kind of machine involved. For example, we have:

- "Entry machine detects the vehicle", in Entry Machine viewpoint;
- "Exit machine detects the vehicle", in Exit machine viewpoint.

By removing these requirements from the original viewpoints we define aspectual viewpoints.

Thus, for a requirement to be considered an "aspectual" one, all his sub-requirements have to be similar too. Note that the requirements are similar not equal, as they differ on the subject. Thus, we will encapsulate this kind of requirement into an aspectual viewpoint and will be redefined using "Roles". Roles work as variables that can be instantiated and deal with similar requirements. This technique has been successfully applied to specify crosscutting scenarios (Whittle and Araújo, 2004).

Table 3 shows the aspectual viewpoint Machine. The crosscutting requirements include roles (represented by the names preceded with a "|") that are instantiable parts of the requirements, i.e. during composition they will be instantiated to a concrete value. In the example |Machine will be instantiated to Entry or Exit Machine.

Aspects can be also defined for the non-functional concerns, which are often crosscutting. We will not focus on this here as the work in (Rashid et al., 2003) deals with this in detail. Re-writing the viewpoint Entry Machine by removing the aspect we obtain the new templates in table 4.

Table 3: Template for the Machine aspectual viewpoint.

Name	Machine
Focus	Registers vehicle entry/exit from the park
Concerns	Availability, Correctness, Multi-Access, Response Time, Security
Source	Assignment
Requirements	AR1 - |Machine detects vehicle. AR2 - |Machine opens gate if entrance/exit mode is valid. AR3 - |Machine closes gate when the passage of the vehicle was detected. AR4 - |Machine calls Park Attendant if Park Attendant button is pressed.
Change History	
Type	Aspectual

Table 4: New version of Entry Machine viewpoint.

Name	Entry Machine
Focus	Registers vehicle entry in the park
Concerns	Availability, Correctness, Multi-Access, Response Time, Security
Source	Assignment
Requirements	1 - Entry Machine detects entrance mode. 1.1 - Entry Machine gives ticket if ticket button is pressed. 1.2 - Entry Machine detects and reads Gizmo if Green Lane button is pressed. 1.3 - Entry Machine receives, reads and gives card back if Card is inserted.
Change History	
Type	Non-Aspectual

To compose aspectual and non-aspectual viewpoints, firstly, we need to instantiate the roles in the aspectual viewpoint and then we compose them. Intuitive composition rules are shown below:

```
Compose Entry Machine with Machine
Bind |Machine to Entry Machine
   Add requirements AR1 to AR4 in Entry
   Machine.
```

Integrate the Feature Model with the Viewpoint Model. Table 5 shows the relationships between features and viewpoints. Fields marked with an "x" are considered mandatory whereas marked with "o" are considered optional.

2.2 VAODA in Application Engineering

The application engineering process defines how the system assets are used for the development of applications in the domain. In VAODA, the application engineering process consists of three steps: Configure the feature model; Reuse Viewpoints; Reuse composition rules.

Configure the Feature Model. It consists of defining an application model within the domain. In this case, the car park considered will only be accessed by card, as it is a residential one, only granting access to residents of that specific building. Based on the information provided by the viewpoint requirements, a configured feature model of the case study for application engineering can be made, (figure 3).

Table 5: Features vs viewpoints at Domain Engineering.

		F\VP	Entry Machine	Exit Machine	Machine
Machine	Entry Machine	Display	x	x	x
		Gate	x	x	x
		Intercom	x	x	x
		PA button	x	x	x
		Vehicle sensor	x	x	x
		Card	o	o	o
		Card Slot	o	o	o
		GL button	o		o
		GL sensor	o	o	o
		Ticket	o	o	o
		Ticket button	o		o
		Ticket slot	o	o	o
		Recorded Voice	o	o	o
	Exit Machine	Display	x	x	x
		Gate	x	x	x
		Intercom	x	x	x
		PA button	x	x	x
		Vehicle sensor	x	x	x
		Card	o	o	o
		Card Slot	o	o	o
		GL sensor	o	o	o
		Ticket	o	o	o
		Ticket slot	o	o	o
		Recorded Voice	o	o	o

Reuse Viewpoints and Aspects According to the Configured Feature Model. The reused viewpoints declared in domain engineering are adapted to the specific model configured before. There are several differences between the Car Park at the Domain level, and the Car Park at the Application level, as at the domain level all possibilities have to be taken into account: at application level there will be no ATM, Bank, Gizmo, Paying Machine. The concerns are the same as the ones for the Domain Engineering level. The template for the viewpoint Entry Machine is shown in table 6. The aspectual viewpoints also have to be simplified to just express the access by card.

Table 6: The configured Entry machine viewpoint.

Name	Entry Machine
Focus	Registers vehicle entry in the park
Concerns	Availability, Correctness, Multi-Access, Response Time, Security
Source	Assignment
Requirements	1 – Entry Machine receives, reads and gives card back if Card is inserted.
Change History	
Type	Non-Aspectual

Reuse Composition Rules According to Configured Feature Model. They are also reused according to the application model. The composition rules are very similar to the ones already specified in the DE.

```
Compose Entry Machine with Machine
Bind |Machine to Entry Machine
   Add requirements AR1 in Entry Machine.
   Add requirements AR2 to AR4 in Entry
   Machine.
```

In addition to the approach, a tool was developed in order to find the candidates to aspectual requirements among the viewpoints of a given system. This tool was developed in Java, also using the Swing widget toolkit (an API for providing a graphical user interface (GUI) for Java programs).

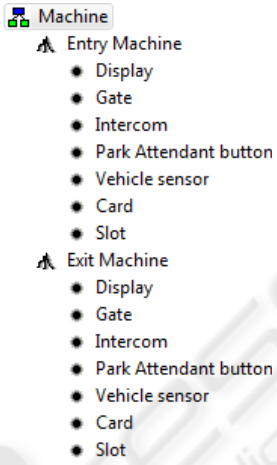

Figure 3: Configured feature model for the Car Park.

3 CONCLUSIONS

This paper proposed an approach to handle crosscutting requirements (i.e. candidate aspects) in domain analysis, called VAODA. The approach is composed of two main parts, the domain and application engineering parts. A tool was developed, in order to help identifying the aspectual functional requirements within a set of domain viewpoints.

For future work a more complete composition language is also a matter to take into account, when dealing with aspectual requirements. Finally feature interaction should also be contemplated.

REFERENCES

Czarnecki, K., Antkiewicz, M., Kim, C. H. P., Lau, S. & Pietroszek, K. (2005a) Fmp And Fmp2rsm: Eclipse Plug-Ins For Modeling Features Using Model Templates. Oopsla'05, Acm Press, , Usa.

Kang, K., Cohen, S., Hess, J., Novak, W. & Peterson, A. S. (1990) Feature-Oriented Domain Analysis (Foda) Feasibility Study. Sei, Cmu, Cmu/Sei-90-Tr-21, Nov.

Rashid, A., Moreira, A., Araújo, J. (2003) Modularisation And Composition Of Aspectual Requirements. Aosd, Acm Press, Boston, Usa.

Shlaer, S. & Mellor, S. J. (1989) An Object-Oriented Approach To Domain Analysis. Acm Sigsoft Software Engineering Notes, 14, 66-77.

Whittle, J. & Araujo, J., (2004) "Scenario Modeling With Aspects", Iee Proceedings.