A Preliminary Comparative Study Using UML-AOF - A UML Profile for Aspect-Oriented Frameworks

José Uetanabara Júnior  
Rogério Lazanja  
Centro Universitário Eurípedes de Marília - Univem  
Marília, São Paulo, Brasil  
CEP 17.525-901  
imiklotovx@gmail.com  
rolazanja@tdkom.com.br

Paulo Afonso Parreira Júnior  
Rosângela Aparecida Delloso Penteado  
Valter Vieira de Camargo  
Departamento de Computação da Universidade Federal de São Carlos (UFSCar)  
Caixa Postal 676 – 13565-905 – São Carlos - Brasil  
{valter, paulo_junior}@dc.ufscar.br

Abstract. UML-AOF is a UML profile that aims to make the characteristics of aspect-oriented frameworks (AOFs) more evident in models than standard profiles for aspect-oriented programming (AOP). UML-AOF groups several stereotypes and, tagged values which represent design and architectural details commonly found in AOFs, such as idioms, patterns and extension mechanisms. In order to evaluate the applicability of this profile, a comparative study was conducted to analyze the number of errors and the time spent by software engineers using both UML-AOF and a profile proposed by Evermann, that is a conventional profile for AOP. Based on the collected data, we have observed that the number of errors as well as the time spent analyzing the models were significantly better than using the conventional profile.

Categories and Subject Descriptors

General Terms
Design and Documentation

Keywords
Aspect-Oriented Modeling, Crosscutting Frameworks, Comparison among Aspect-Oriented Modeling Approaches

1. INTRODUCTION

After the emerging of AOP [10] several researchers started to study the impact of this new programming technique along the software lifecycle. Many of them concentrate their researches in aspect-oriented modeling, providing some contributions by proposing UML profiles for modeling the new AOP characteristics [17][3][6][7][5].

In other research field, some authors also started to study the impact of AOP in the development of software frameworks [2][13][16][4][15][9][12][1][14] making appear the term “aspect-oriented framework”, but without a common consensus about what this term really means. In the context of this work, we are using the term “Crosscutting Framework” (CF) as a kind of AOF that encapsulates just one crosscutting concern, such as persistence, distribution, concurrency or security [2]. We believe that CFs is the most common kind of AOF found in literature.

Although it is possible to model AOFs with profiles for standard AOP, many of their characteristics are not evident when using these profiles. Frameworks have a complex architecture and many of the specific characteristics are not made evident in their design models. In a previous work, Uetabanara and Camargo [1] identified the common AOFs characteristics and proposed a UML profile, called UML-AOF, for designing AOFs. This profile includes stereotypes and tagged values that aids software engineers in identifying some those specific characteristics.

Our focus here is to show an empirical study [18] with two aims. The first one is to compare the time spent by software engineers using UML-AOF and using Evermann’s profile [5]. The second one is to show the number of errors commited by software engineers using UML-AOF and using Evermann’s profile [5]. Through the collected data we observed that number of errors obtained by the subjects showed that our profile is more effective than Everman’s profile. Besides, we also have a slightly benefit related the productivity of the analysts.

2. CROSSCUTTING FRAMEWORKS AND UML-AOF PROFILE

CF is an AOF that has abstract mechanisms and variabilities, both related to a unique crosscutting concern, such as persistence or concurrency [2]. CFs are used to support the development of non-functional parts of an application and their biggest benefit occurs when several of them are available in a repository, as each one deals with a unique crosscutting concern.

Despite all of the well-known reuse benefits promoted by framework-based development, applications that are developed with their support have a complex three-layer architecture, which results in complex design models. The first layer (framework layer) encapsulates abstract and concrete modular units of the framework; the second layer (instantiation layer) is responsible for putting together the first and the third layers by creating concrete modular units extending the abstract ones of the framework; and the third layer - application layer – is the application itself. In the context of CFs, the problem is worse than conventional frameworks, as the development of an application may be based on “several” CFs, as each of them addresses just one crosscutting concern. The final design models are complex and make comprehension, reuse, and maintenance difficult tasks[2].

CFs have a lot of architectural and design characteristics that are not present in standard object-oriented frameworks and that deserves to be evident in design models. In a previous work we brought up these characteristics and create a profile to make them evident [1], which can be seen in the first column of Table 1. The characteristics we have identified are related to design details, architecture details and points of extensions.
Regarding to design details category, we have found that the idioms proposed by Hanenberg [8] and the Data Catcher Pattern [2] are common design decisions employed by several aspect-oriented frameworks, even if the framework engineers do not use these same nomenclature. Each idiom, pattern and the other categories were represented by stereotypes, tagged values of enumeration types for representing CFs characteristic, as can be seen in the last column of Table 1. By the usage of stereotypes and tagged values, UML-AOF provides a way of modeling these common characteristics so that they remain expressive in the model, supporting software and framework engineers in maintenance and reuse tasks. So, the stereotypes, tagged values and enumerations can be divided into three categories: 1) Design 2) Points of Extension and 3) Architecture, as can be seen in Table 1. Each category has a specific set of stereotypes that belongs to the common characteristic identified, for example in the Design category, the container introduction idiom has three stereotypes, Container, Container Connector and Container Loader – each stereotype represents a role employed by this idiom [8]. Therefore, when these stereotypes are found, the software engineers know that the Container Introduction idiom is being used in the CF. The same reasoning is valid with all the other stereotypes.

### 3. THE COMPARATIVE STUDY

#### 3.1 Study Definition

The study aims at verifying if a design model modeled with the UML-AOF profile facilitates the identification of common CFs characteristics when compared with a model designed with the Evermann’s [5] profile. The word “facilitates” means that i) someone must commit fewer errors using UML-AOF than using a conventional profile when analyzing a design model and ii) someone must be faster to identify a specific CF characteristic using UML-AOF than using a conventional profile when analyzing a design model. Besides, we are also interested in improve the productivity when analyzing models.

#### 3.2 Study Planning

a) **Selection of Context** The study was conducted with post graduate students of Computer Science, using two design models from two CFs - a persistence and a security CF and two applications: Abaco application and a Hotel reservation system - Abaco application consists in a system which manages the reservations of a hotel. Therefore, there were four complete design models available for the comparative study. The first one is the persistence framework with the Abaco application modeled with UML-AOF profile; the second one was the security framework with the Abaco application modeled with the UML-AOF profile, the third one was a persistence framework with the Hotel reservation system modeled with Evermann’s profile and the fourth one was the security framework with the Hotel reservation system modeled with Evermann’s profile [5].

<table>
<thead>
<tr>
<th>Category (CF characteristics)</th>
<th>UML-AOF Stereotypes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container Introduction Idiom</td>
<td>&lt;&lt;Container&gt;&gt;, &lt;&lt;ContainerConnector&gt;&gt; and &lt;&lt;ContainerLoader&gt;&gt;</td>
</tr>
<tr>
<td>Marker Interface Idiom</td>
<td>&lt;&lt;MarkerInterface&gt;&gt;, &lt;&lt;AspectSpecification&gt;&gt; and &lt;&lt;MarkerSticker&gt;&gt;</td>
</tr>
<tr>
<td>Data Catcher Pattern</td>
<td>&lt;&lt;DataCatcherPattern&gt;&gt;</td>
</tr>
<tr>
<td>Points of Extension</td>
<td>&lt;&lt;PointCutHook&gt;&gt;, &lt;&lt;Hook&gt;&gt;</td>
</tr>
<tr>
<td>Architecture</td>
<td>&lt;&lt;Interface-F&gt;&gt;, &lt;&lt;InstantiationInterface&gt;&gt;, &lt;&lt;Aspect-F&gt;&gt;, &lt;&lt;InstantiationAspect&gt;&gt;, &lt;&lt;Class-F&gt;&gt; and &lt;&lt;InstantiationClass&gt;&gt;</td>
</tr>
</tbody>
</table>

### Table 1 - UML-AOF Stereotypes

b) **Formulation of Hypotheses** Two types of hypotheses were elaborated: the hypothesis for the effect of the use of different applications and frameworks in the results (Ho e Ha) and the hypothesis for the effect of the use of different UML profiles in the results (Hb e Hc).

Ho: There are no differences among developers using different applications and frameworks according to the time of identification of common CFs characteristics and number of errors committed during the analysis of the model.

Ha: There are differences among developers using different applications and frameworks according to the time of identification of the common CFs characteristics and the number of errors committed during the analysis of the model.

Hb: There are no differences among developers that use the UML-AOF profile and the developers that use Evermann’s profile according to the time of identification of common CFs characteristics and number of mistakes committed during the analysis of the model.

Hc: There are differences among developers using UML-AOF profile and the developers using Evermann’s profile according to the time of identification of the common CFs characteristics and the number of mistakes committed during the analysis of the model.

c) **Variable Selection.** There are two kinds of variables: dependent and independent. The dependent variables are those that are under analysis. In our case are the “time spent analyzing the models” and the “number of errors” committed during the study, like the “applications” and the “profiles”.

d) **Design of the study** The study was designed in blocks (Table 2) and balanced to ensure that the treatments had an equal number of subjects and to provide a comparison of the effects by the independent variables. In the first phase, the subjects worked with different case studies modeled with the UML-AOF profile. In the second phase, the CFs and applications were changed and the subjects worked with the case studies modeled with Evermann’s profile. Notice that we decided not to benefit our profile by putting it in the second phase, as at this phase the subjects were more experienced in analyzing the models and this knowledge could benefited the profile used at the second phase.

e) **Type of project:** A factor with two treatments (2 * 1 factorial).

The distribution of subjects in the groups was done by trying to balance the number of the subjects and the skills of the subjects equally. The subjects’ skills have been evaluated through characterization form. It was assigned a numeric value for each answer, so that the more skilled the subject is, the more points he scored. With the subjects’ skills identified, it was possible to define which subjects could be considered more skilled than others. That was made through the sum of points of each subject according to the scores explained previously. The subjects P1, P2, P3 and P4 formed the group 1 and P5, P6, P7 and P8 formed the group 2. The subjects P3 and P5 were considered expert.
f) Instrumentation

During the comparative study, the following documents were used: 1) a consent form for the subjects registration; 2) a profile characterization form; 3) four questionnaires, where two of them had to be answered during the pilot experiment and the others had to be answered during the real experiment.

The questionnaires applied in the comparative study had four categories of questions: 1) questions about idioms and patterns, 2) question about points of extension, 3) questions about layer differences and 4) questions about normal characteristics of an AOP. The reason for the last category of question was to evaluate if the UML-AOF profile was not obscuring the concepts defined by Evermann’s profile [5].

About the design models, two of them were used during the pilot experiment (numbers 1 and 2) and four of them were used during the real experiment (numbers 3, 4, 5, and 6). The case studies are shown below.

- a case study modeled with the UML-AOF profile using a Department Management application;
- a case study modeled with Evermann’s profile using a Department Management application;
- a case study having the persistence framework with the abaco application modeled with the UML-AOF profile;
- a case study having the security framework with the abaco application modeled with the UML-AOF profile;
- a case study having the persistence framework with the Hotel reservation system modeled with the Evermann’s profile;
- a case study having the security framework with the Hotel reservation system modeled with the Evermann’s profile;

The CFs used were a persistence and a security frameworks, both from Camargo and Masiero [2].

3.3) Execution

The study was conducted in six steps: 1) subjects filled the consent form; 2) subjects filled the profile characterization form; 3) we conducted the training about essential and basic concepts (AOP, AOFs, CFs, UML, UML-AOF, Evermann’s profile). 4) we created the groups: four people in each group; 5) the pilot experiment was conducted and 6) the real experiment.

The aim of the pilot experiment was to simulate the comparative study, by showing design models and questionnaires similar to the real ones. So, the subjects would be more familiar with the experiment dynamic. The pilot experiment consisted in a case study modeled with Evermann’s profile [5] and also with the UML-AOF profile [1]. First, the subjects received the case study modeled with the UML-AOF profile along with its questionnaire. After done, the subjects received the same case study modeled with Evermann’s profile. During this process, the subjects’ questions were answered.

The real comparative study was made in two phases. In the first phase, the group 1 answered questionnaire 1 according to case study 1 (see Table 2) and group 2 answered questionnaire 2 according to case study 2. In the second phase, group 1 answered questionnaire 2 according to case study 3 and group 2 answered questionnaire 1 according to case study 4.

4) COLLECTED DATA

In Tables 3 and 4 the spreadsheets containing the data collected during each phase of the comparative study are shown. In the first spreadsheet the subject (who worked with the case studies modeled with the UML-AOF profile) score is shown and in the second spreadsheet the subject score using Evermann’s profile is shown [Evermann, 2007].

For each phase the subjects received one questionnaire to be answered according to the case study. Each questionnaire has three questions that were used to evaluate the architectural stereotypes, four questions were used for design stereotypes, and one question was used for points of extension stereotypes. Each question had a maximum value. Numeric values for each question have been given to make the data analysis more precise. In question 1, for example, three items were asked. If the subject answered all of them correctly he scored 3 points, if he answered two of them he scored 2 points, if he answered 1 of them he scored 1 point and if all the items were incorrect or blank no points were scored. Using a simple “right” or “wrong” instead of the point system would have been difficult because a subject could answer 2 of 3 items correctly making his response to the question neither completely right nor completely wrong.

5) DATA ANALYSIS

On the stereotypes for idioms, there was a small amount of errors with UML-AOF on the marker interface idiom. In the Evermann’s profile all subjects from group 1 got the question about the marker interface idiom wrong, and one subject got the question about the container introduction idiom wrong. The subjects succeed using UML-AOF profile because it provided a way (by using stereotypes) for making these characteristics expressive in the model.

The subjects answered all the questions about the “design stereotypes” correctly in both profiles – Evermann and UML-AOF – i.e., they easily identified the modular units relevant to the framework and software engineers. We suppose the reason for that is that the case studies were small and rather easy to understand. So, we consider the stereotypes for “idioms” more relevant than the “architectural” stereotypes.

About the stereotype for the “data catcher” pattern, almost all subjects succeeded at identifying the modular units that had the data catcher pattern using UML-AOF profile. However, the same did not happen with the Evermann’s profile [Evermann, 2007]; three subjects did not identify the modular unit that had the data catcher, and so they couldn’t answer the questions about the

<table>
<thead>
<tr>
<th>Table 2 – Design of the study</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Group 1</strong></td>
</tr>
<tr>
<td>Training and application of the pilot experiment.</td>
</tr>
<tr>
<td>1st Phase UML-AOF profile</td>
</tr>
<tr>
<td>2nd Phase Evermann’s profile</td>
</tr>
</tbody>
</table>
pattern (questions 6 and 7). So, we consider the stereotypes for the data catcher pattern relevant in this comparative study.

About the stereotypes created for “points of extension”, it was observed that they were not so relevant in this study. The number of errors in question 8 – the question that evaluated these stereotypes – was small: with the UML-AOF profile the subject P2 got one of the two items asked correct and with the Evermann’s profile subject P2 got the question wrong. All of the other subjects answered it right.

Finally, the last two questions of each questionnaire (questions 9 and 10), evaluated if the UML-AOF dos not prejudice the understanding of the concepts already defined by Evermann.

Taking in consideration the percentage of score from both groups, it was observed that with the UML-AOF profile the groups got 95% of correct answers and only 5% of errors. With Evermann’s the groups got 79% of correct answers and 21% of errors, a difference of 16% among them. During the first phase, group 1 got an average score of 20.25 and in the second phase their score was 17.75 a difference of 12% among them. The same group got 1.75 errors during the first phase and 5.25 in the second phase a difference of 66%. During the first phase, group 2 got an average score of 23 and in the second phase their score was 18.25 a difference of 20% between them. The same group got 0 mistakes during the first phase and 3.75 in the second phase a difference of 100% among them. So, it is concluded that the UML-AOF profile aids framework and application engineers at identifying the common CFs characteristics in the model.

Taking in consideration the time spent analyzing the models, subject P2 took 15 minutes to answer all the questions with the UML-AOF profile and 21 minutes to answer all the questions with the Evermann’s profile. So, with our profile this subject was 27% faster. Also it was observed that the subject P4 took 21 minutes to answer all the questions with the UML-AOF profile and 16 minutes to answer all the questions with the Evermann’s profile. So, with Evermann’s profile this subject was 23% faster. So, we concluded that time was not a relevant comparative factor, since the analysis with the fastest subjects with each profile showed that UML-AOF was only 4% faster.

Table 4 – Second Phase Collected Data

<table>
<thead>
<tr>
<th>PHASE 1 - EVERMANN’S PROFILE</th>
<th>GROUP 1</th>
<th>GROUP 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time (min):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Score:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>% Score:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Score:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mistakes:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Mistakes:</td>
<td>0.3</td>
<td>0.0</td>
</tr>
<tr>
<td>Total score:</td>
<td>81</td>
<td>92</td>
</tr>
<tr>
<td>Total mistakes:</td>
<td>7</td>
<td>0</td>
</tr>
</tbody>
</table>

The graphics shown in Figure 1 have been generated from the collected data. Through these graphics it is possible to visualize the number of errors, the time spent analyzing the models and the score of each subject on each phase.
5.1) Threats of Validity

The first threat of validity is about the subjects’ situation. Someone can argue that they felt pressed to benefit the UML-AOF profile. However, neither of them were students in a course taught by the authors. So, there was no reason for them to have felt pressured.

The second is about the training. Someone could argue that training was more focused on UML-AOF characteristics and neglected important characteristics from Evermann’s profile, which would give some advantage to UML-AOF. However, we showed the subjects how to model the specific CF characteristics in both profiles during the training.

Another is about impartiality. We tried to conduct the experiment being impartial. For example, we decide to use the UML-AOF profile in the first phase, so the Everman’s profile could have been benefit due to this decision. Another point is that, during the training we showed very clearly how to model some characteristic using both profiles. So, the subjects knew how to identify the specific characteristics using both profiles.

Another point is about the size of the case studies. Our results are based on small case studies. Although we believe that the same behavior will be represented with bigger case studies, we can not ensure that.

Someone could also argue that since the aim of this comparative study was to evaluate if the UML-AOF profile makes the common CF characteristics evident in design models, the questions were easier to answer in the case studies modeled with the UML-AOF profile. However, we tried to balance the question’s difficulty and also to include some questions specific for Everman’s profile.

The last one is about the profiles. Comparing a profile that is specific for modeling CFs with a profile that is used for modeling conventional OA programs may not be fair. Someone could argue that what should be done is comparing two similar profiles for modeling CFs. However, to the best of our knowledge there is no in the literature another useful profile for AOFs. Rausch proposed a profile for AOF, but it is a preliminary work [13]. Besides, it is perfectly possible to model a CF using a standard profile for AOP.

6) RELATED WORK

A lot of researchers have proposed profiles for AOP [17][3][6][7][5], however only Rausch [13] has proposed a profile specific for AOFs, like what is done in this work. Rausch proposed models to be used in requirements and design levels for AOFs. Besides, a technique for gluing models was developed. This technique uses bidirectional arrows and notes with OCL (Object Constraint Language). Each hook is specified inside the notes. Through the OCL, the application elements are linked with the hooks in the framework. Rausch’s work has some deficiencies: it does not have a meta-model and it has large and complex OCL statements because there is not a specific language for the aspect binding.

An important point is that we do not find in literature a comparative study among models. So, we believe that the planning of our study was done in a suitable way and that the results are confident.

7) CONCLUSION AND FUTURE WORK

In this paper we show a comparative study about the applicability of UML-AOF in terms of number of errors and time spent analyzing a model. The primary objective of UML-AOF is to make some specific CF characteristics evident in design models. Our argument is that design models modeled with conventional profiles for AOP makes the job of framework and application engineers harder than using a specific profile. We also argue that such a profile does not prejudice the conventional comprehension of the standard abstractions of the AOP in models.

According to the results obtained we made some conclusions. The number of errors was a relevant factor during this comparative study: the groups committed several errors using Evermann’s profile. So we concluded that by using the UML-AOF profile the number of errors committed is significantly small. The stereotypes used for representing idioms and patterns were more relevant than the stereotypes used for architectural details and for representing points of extension. That could be observed by the data sheets that showed which question the subjects got right and which they got wrong with each profile. For the idioms and pattern stereotypes, many subjects got the questions with the UML-AOF profile correct and got the questions wrong with Evermann’s profile justifying its relevance. With the documentation and points of
extension stereotypes, the subjects got the questions right in both profiles justifying the non relevance of these stereotypes in this comparative study. So we concluded that in this comparative study, the architectural and points of extension stereotypes could be omitted. In this comparative study “time” was not a relevant factor since the difference between the total time that the subjects took to answer all the question with both profiles were only 8 minutes.;

The usage of different CFs and applications in the case studies, did not affect the comparative study results. This can be observed by the average of the results that are found in the spreadsheets: both groups worked with different CFs modeled with different applications during each phase yet the data gathered during the respective phases was not so different. In Table 3, it is possible to see that both groups got a nearly even number of correct answers (a difference of only 7 points from each group) using different case studies and in Table 4 the difference of correct answers was only 2 points. The same reasoning is valid with the number of mistakes which was close in both phases of both groups.

Although the UML-AOF profile were created based on our experience with CFs, we believe that it could be used for modeling Aspect-Oriented Application Frameworks [12][11]. Some of these frameworks have some additional characteristics, such as “extension join points for example”, which could be easily incorporated in our profile by means of stereotypes and tagged values.

As a future work we aim to conduct this comparative study with a bigger comparative study with more subjects. Besides we also intend to use different CFs and more case studies. Also different profiles shall be used for the comparison. By doing so, the conclusion about the documentation and points of extension stereotypes might changed.

All the documentation of the study and also the collected data are available in www.dc.ufscar.br/~valter/UML-AOF.rar. We appreciate if our laboratory package could be used by other researches for replicating our study.

ACKNOWLEDGEMENTS

The authors would like to thank CNPq for the financial support (Projeto Latin – AOSD – PROSUL – Proc. no. 490478/2006-9 and Proc no. 133140/2009-1).

REFERENCES


