A Tool for Modeling Software Development Contexts in Small Software Organizations

Daniel Ortega, Luis Silvestre, María Cecilia Bastarrica, Sergio F. Ochoa
Department of Computer Science, Universidad de Chile, Santiago, Chile
{dortega, lsilvest, cecilia, sochoa}@dcc.uchile.cl

Abstract - Characterizing the software development context has been recognized as relevant for a variety of purposes, particularly for tailoring the organizational software processes according to the features of each project to be addressed. That tailoring activity makes the software processes more efficient and effective. However, formally modeling software development contexts, which is usually required to perform this tailoring activity, represents a challenge for many organizations; and particularly for small software companies, which usually have low expertise and resources to deal with formal methods and notations. This paper presents a usable software tool for defining the software development context for an organization and also for particular projects. Preliminary evaluations of the tool usability and usefulness have been performed. The obtained results indicate that the tool provides all the services required to model development context, and potential users found it easy to use.

Keywords - Software development context, model-driven engineering, modeling tool, software process tailoring.

I. INTRODUCTION

Software processes have been recognized as a critical piece for developing software systems [14]. However, defining and applying suitable software processes demands a great effort.

Since there is no unique software process able to deal with various organizational, project and product characteristics, software companies require either to have a series of process or even better, to have an adaptable process model.

Typically the characteristics of a software company are considered in its organizational software process; therefore they do not need to be considered for each particular project. However the project and product features must be considered in every development, because they usually change from one project to another. These aspects are usually known as the project context. This project context varies according to different attributes such as: product size, project duration, product complexity, team size, application domain knowledge, and familiarity with the involved technology, among others.

Several researchers have identified the need to specify the project context to determine which software process is the most suitable to support each development [3][11][13]. This process selection can be done following different approaches, but all of them consider the project context as a mandatory input. For instance, the template-based approach of the Crystal process family [10] chooses a different process flavor according to the characteristics of the project at hand. The Unified Process (UP) [17] follows a different strategy. It utilizes a framework-based approach where all projects share a basic process framework that is extended according to the project characteristics.

Although the existing proposals focus on the same problem, they require manual activities and personal decisions made by the process engineer and the project manager for each and every project, making it time and resource-consuming. These features make the software process selection an error-prone, unrepeatable and unpredictable activity.

In a previous work the authors proposed a tailoring approach based on model transformations [15], which receives as input the organizational software process model and the project context model, and it produces a software process model that is particularly adapted to address the considered development project (Figure 1).

This approach is repeatable and it avoids the typically mistakes made due to the use of manual activities and personal decisions of process engineers or project managers.
This tailoring process is based on a Model Driven Engineering (MDE) strategy. The MDE-based tailoring strategy that we have been developing for the last couple of years [15] formally defines the organizational process and the context as models, and it takes them as the input of a tailoring transformation whose output is the adapted project process. Consequently, all tailoring approaches need a description of the project context in order to obtain the appropriate process.

However the model transformation that tailors the organizational software process needs, as input, a project context specification in XMI format. This model specification can be done just manually using the Eclipse Modeling Framework1. This activity is not difficult, but it is not user friendly and therefore it cannot be afforded by most software organizations. Particularly, small companies do not usually count on people experienced in working on formal specifications.

Currently most of these small organizations do not invest effort in tailoring their software process at all, because it is not possible to ensure a minimal usefulness of the obtained tailored process.

This paper presents a Web-based tool that eases the definition of the software development context, making this activity affordable for small organizations. The tool allows process engineers and the project managers to define the software development contexts in two stages: (1) definition of the organizational context model, and (2) configuration of that model to obtain the project context model.

The tool was particularly designed to be used in small software organizations, so it is able to manage only a few characteristics for software projects as well as several characteristics, depending the particular needs. Another essential issue for our target users, i.e., the process engineer and the project manager, is usability. This avoids them to deal with low level details related to context models representations. The usefulness and usability of the proposed tool was evaluated, and the obtained results are highly encouraging.

The paper is structured as follows. Next section presents the related work. Section III describes the context modeling tool and the considered context dimensions. Section IV discusses the preliminary results on the usefulness and usability of the tool. Section V presents the conclusions and the future work.

II. RELATED WORK

This section presents and discusses the related work concerning to the software process tailoring approaches considering how context information is used in each of them. Different ways of modeling project context are also presented. The following two subsections describe the proposals in these research areas.

A. Software Process Tailoring Approaches

The approaches proposed to perform process tailoring are diverse and they mainly differ in terms of expected process formality, company size and available tool support [21]. For instance, the assemble approach [12] allows a process engineer to tailor an organizational software process using a set of predefined actions (e.g. add, delete, split and merge activities). These processes are specified as Petri nets, and therefore they do not specify roles and work products. In the assemble approach the tailoring process is performed manually, consequently it has the previously mentioned limitations. The context information is implicitly managed and the process engineer is in charge of the assembling process.

Another well-known approach is the situational method engineering (SME) [22], where pre-existing pieces of an organizational process are selected and combined properly to produce an adapted process for a particular project. In most cases the effort for tailoring processes using SME is still huge, because it involves manual work and requires experienced people participating in the tailoring. This jeopardizes the use of this technique particularly in small software companies.

Some reference processes, like the Unified Process [17], use an adjustment guide approach. The tailoring rules are specified as general guidelines to adapt phases and activities depending on the context of the project to be addressed. The success of this approach depends mainly on the experience and skills of the process engineer to consistently apply these guidelines to the different projects and to wisely recognize the project context as it is not explicitly defined.

Commercial processes, such as RUP [20], use a configuration approach [5]. This strategy is focused on defining a general process, and based on it, specifying a set of process configurations that are created for each particular project. RUP-based processes tend to be large and complex, and the development of the configurations involves important effort and skills. Moreover, defining an adequate set of configurations that satisfies all the possible project scenarios represents a challenge difficult to address [9].

In contrast with the previous proposal, agile methods (such as XP [4]) use an auto-adaptable approach, where the project- and team-adapted process emerge from the set of principles, values and practices that define the agile methodology; i.e. the resulting process is not based on a conscious decision about the context.

Other proposals, such as the Crystal Methodology [10], follow a template-based approach depending on the project. The Crystal family specifies four methodology templates (Clear, Yellow, Orange and Red), where each of them is more detailed than the last one. This tailoring approach is not time consuming, but it does not provide accurate processes to deal with particular projects, only project size and complexity are considered for deciding the color of the process to be followed.

1 http://www.eclipse.org/modeling/emf/
Killisperger et al. [18] propose a tailoring strategy where a general process is instantiated for each particular project, up to the enactment level. This approach seems to be useful, but it requires formalizing the general process, which represents an enormous effort that is not affordable for most small software companies. This approach also requires the specification of the execution context.

B. Modeling Project Context

The literature reports several definitions for context in computer science [7][25]. The context for software development has been defined as the set of attribute instances that characterizes the project, the product to be developed, the participating resources, the tools to be used and the environmental conditions [7].

It has been considered for software development with diverse purposes: effort estimation [6], risk management [23], choosing the appropriate development methodology [24], or tailoring the development process [20].

Concerning the modeling of project contexts for software process tailoring, it is a topic that has gained relevance during the last years. Perez et al. [21] present a set of context characteristics that are used to customize software process models. Similarly, Armbrust et al. [2] define three dimensions of context variables when defining a software process line: product, project and process. Claudepierre et al. [8] describe the context dimensions and variables that are relevant to tailor processes in the information systems domain. Although these approaches seem to be appropriate to address tailoring of complex processes, they are not suitable for simple processes that are usually present in small software organizations.

Koolmanojwong and Boehm propose four process patterns for rapid process deployment, using contextualized information [20]. The software process for every particular project is derived from these patterns.

Finally, Hurtado et al. [16] present a survey of context representations that can be used for tailoring software processes. Such a work, that is part of the authors’ previous research, is focused on small software organizations; therefore the project context model considers a few context dimensions and variables. The tool described in this paper is based on this work. Next section describes the software modeling tool, its context model and the main user interfaces.

III. SOFTWARE CONTEXT MODELING TOOL

The modeling process supported by the proposed tool involves two sequential activities (Fig. 2). The first one is performed by the process engineer, and its goal is to define the relevant dimensions, attributes and values of the organizational context model.

The context dimensions represent groups of related contextual attributes. The context attributes describe particular contextual elements inside a dimension. Finally, the context attribute values express a specific value to be assumed by a context attribute.

The tool provides a particular user interface specifically intended to allow the process engineer to define the organizational context model.

The second activity in this process is performed by the project manager (Fig. 2). This person uses the tool to configure the previously defined organizational context model, according to the features of each particular project. This configuration is performed by assigning values to the attributes considered in the organizational context model. The result of this configuration is the project context model, represented through a xmi file, that is used as input for the process tailoring.

These two activities are coordinated so that only those attributes defined as part of the organizational context model can be configured for a project. Next subsections describe in more detail these two modeling activities.

A. Organizational Context Model Definition

As mentioned before, the authors defined an organizational context model as a way to help mainly the process engineers or even perhaps project managers to specify the context model for his/her company [16]. This model includes dimensions, attributes in each dimension, and the potential values for the attributes that influence the selection of the most appropriate process to be applied.

The tool for specifying the organizational context model provides the user with canonical specification that has been obtained from empirical experience of the authors and consultants in software process improvement and definition. Based on that, the tool presents this canonical context model as a starting point, and the process engineer can adjust it to the needs of his organization. The context dimensions, attributes and values considered in this specification are presented next.
Software Development Project Context Canonical Case

Dimension Project
Project Type: {newDevelopment, extension, maintenance, tuning}
Duration: {short, medium, large}
Client Involvement: {high, medium, low, known}
Problem Knowledge: {clear, ambiguous, unclear}
Time Constraints: {veryConstrained, typical, unconstrained}
Budget Constraints: {veryConstrained, typical, unconstrained}

Dimension Team
Team Size: {veryRestricted, typical, unrestricted}
Team Expertise: {high, regular, low}
Business Knowledge: {know, affordable, unknown}
Product Knowledge: {know, affordable, unknown}

Dimension Product
Technical Complexity: {high, medium, low}
Quality Relevance: {high, regular, minimum}

Dimension Process
Process Focus: {finalProduct, everyProduct}

Every software company can interpret each context value differently, because different companies can use the same attribute or value name, but with a different meaning. Moreover, they have to determine which context variables and dimensions are relevant for them. For instance, a certain company may get engaged in different types of project, such as development, maintenance or bug-fixing. In that case, the extension software project, as established by the canonical context model, does not make sense for them.

Some other companies may also distinguish between a corrective maintenance from an evolution or extension; therefore they should adjust the pre-defined semantic for those project types.

It can also happen that the same kind of project may be commonly known as bug-fixing in a company and incident in another. Therefore the canonical context model (and also the proposed tool) does not only act as a guideline for process engineers, but also it contributes to homogenize the jargon used for project dimensions, variables and values within each company.

Figure 3 shows part of the user interface of the proposed tool, which helps defining the organizational context model. The tool allows defining each dimension, all attributes and their potential values. Finally, the context model is generated and a named xml file is created: the organizational context model.

The tool is a Web-based application, and this first part is intended for one specific kind of user role: process engineer.

B. Project Context Model Configuration

Each particular project in the any company has its own characteristics. These particular characteristics determine the concrete process that best fits it, therefore they need to be identified.

Let us suppose that we have to model the project context for a software project that involves a Project Type (i.e. the context variable defined in the context model) that is maintenance-correction, the Provider is outsourced, the Customer Type is known, and Project Duration is small, as shown in Fig. 4.

The tool presents the user (i.e. the project manager) the configuration options established by the organizational context model obtained in the previous stage, and he can select the appropriate value in each case.
Once all the context variables have been set, the concrete context model can be generated as an xmi file. The particular result of the previous example is shown in Fig. 5. We can see that the value for Project Type (the highlighted item) is shown to be Maintenance-Correction.

![Fig. 5. Project’s Context Model in EMF](image)

This context representation, which is visualized using the Eclipse Modeling Framework (EMF), is exactly the one we have used for the example in [15]. Therefore we can assume that it could be directly used as input for the tailoring transformation we presented in that paper. In that case we had built the context model directly, but the company’s users found it was cumbersome to deal with low abstraction level tools. The tool described in this paper hides these low level representations to the end-user, and therefore it has a real potential to make the project context definitions much easier, providing thus a direct benefit for small software organizations.

C. Design of the Tool

The architecture of the tool adheres to the model-view-controlled pattern. Figure 6 shows the class diagram that represents the design of the context modeling tool. It involves nine classes which are explained next.

Most of these classes have just getters and setters methods. The Context class has a method named createCanonicalContext, which allows creating a project context based on the context model defined in section III.A.

That context is a data structure, represented in xml, which considers several ContextDimensions, ContextAttributes and ContextAttributeValue.

Every project context (represented by the class ProjectSpecificContext) is an instance of the canonical context model, which is managed through the Context class.

The specific context elements for a particular project are managed by the ContextElement and ProjectSpecificElement classes.

The Link class establishes priorities between every couple of context attributes. This information is specified by the process engineer when he/she defines the organizational context model. After that, this information does not need to be set by the project manager when defining every new project context.

![Fig. 6. Class Diagram of the Tool](image)

IV. PRELIMINARY RESULTS

Three types of evaluations were performed to determine the capabilities and limitations of the proposed tool. The first evaluation was focused on determining the correctness of the xmi file generated by the tool, which represents the project context. The second one was used to identify the usability level of the tool according to the end-users. Finally we evaluated the usefulness of the web application as an instrument that supports end-users during the organizational and projects context definition. These evaluations are described more in detail in the next sections.

A. Correctness of the xmi file

In order to determine the correctness of the project context representations generated by the application, we have used the tool to create the two already known organizational contexts of companies participating in the ADAPTE project [1]. The obtained results were compared with the organizational contexts already defined by these companies, and also with the canonical model specified in section III.A.

The comparison was done through a manual work that considered each dimension, attribute and value specified in the xmi file. The obtained results were completely consistent with those expected; i.e. they adhered to the canonical context model and matched with the already defined organizational context model of the involved software companies. Although these results are still not enough to get strong conclusions, they indicate that the proposed application has the potential to create consistent organizational context models that adhere to the canonical model.
Moreover we created two project contexts using the previously defined organizational context models. The obtained results were compared using the procedure described before.

After performing such an activity, we can say that the project contexts generated by the tool adhered to the canonical model and also to the project context definitions made by the software companies involved in this analysis. Similar to the previous case, the results obtained in the project context definitions were correct and consistent.

B. Evaluation of the tool usability

This activity was focused on determining the users’ satisfaction when they are using the application, and also the effort required to understand the role of the various user interfaces implemented in the tool. Four people that usually act as process engineers and project managers participated in this evaluations process. Although the number of participants in the evaluation is small, the features of these evaluators are so specific that it was not possible to count on more people for this activity.

We asked the participants to perform particular activities using the tool. The process engineers performed twelve activities, e.g. to log into the system and then to access the Web form to modify the organizational context model or to modify the name and description of a specific context dimension. The project managers performed five activities, e.g. to create a specific context for a particular project and generate the resulting xmi file.

After performing those activities the process engineers and the project managers filled a survey and also they were then interviewed. The survey considered eleven items that were rated by the users utilizing a 5-point Likert scale, where “0” indicated “completely disagree” and “5” means “completely agree”. Figure 7 shows the evaluated items and also the obtained results.

The results show that the proposed application is usable for the end-users. The results obtained from the interview confirm this assumption. The functionality available for the process engineers got the maximum score and the functionality for the project managers got a 4.75 score in the Likert scale.

C. Evaluation of the tool usefulness

The interview conducted after the usability evaluation was focused on determining the usefulness of the tool, as an instrument that supports the organizational and project context definition in small software companies.

During that interview several questions were asked to the participants. Some of them focused on confirming the usability results obtained in the previous evaluation stage, but most of them focused on determining the tool usefulness.

The participants’ answers were highly positive in all cases. Particularly we asked them if there is any service that is usually required to define the organizational context or the project context, which is not implemented in the proposed tool. All of participants answered that the functionality of the Web application is complete from their point of view.

We also asked them about the effort and expertise required to perform these context definition activities. All of them said that both, the users’ required expertise and effort are low. Therefore we could preliminary assume that the proposed tool is usable and useful to address the stated problem. As a conclusion, we can say that this proposal could help small software organizations to use their organizational and project contexts for tailoring their software processes, and thus to perform more efficient and focused developments.

V. CONCLUSIONS AND FUTURE WORK

The use of project context information has gained relevance during the last years as a driver to tailor software processes. Although this is a transversal concern of the software industry, software tailoring is particularly relevant in small software organizations, because they usually are more vulnerable and therefore they have less room to make mistakes.

The process adaptation for small software companies should be fast and accurate, because they usually have little time and expertise to perform such an activity. Counting on a more suitable process enables these companies to build better products in a more efficient way, allowing the whole software industry to improve its results.

The current alternatives reported in the literature to tailor software process to particular project contexts, involve manual work and personal decisions, or the use of complex information representation, which is (as discussed in section II) not appropriate for small software organizations. Trying to deal with that problem, this article extends a previous proposal of the authors [15] and presents a Web-based tool for modeling software development contexts. This tool hides the inherent complexity of specifying the organizational context and also the context of each particular project.
Although the formal representation of the context model is specified in XML format, the end-users (i.e., the process engineer and the project manager) utilize just typical Web user interfaces to create, adjust and configure this model. This eases considerably that activity and makes it more affordable for small software organizations.

This tool is freely available at [1], and it was designed to support both, the process engineer during the definition of the attributes that may affect the process to be applied, and also the project manager when assigning the particular values to these attributes depending on the project at hand. This tool has been successfully verified using already defined and pre-validated formal context models.

The usability of the proposed tool has also been preliminary validated with process engineers. However we have still pending to perform additional usability and usefulness tests, involving the engineers participating in the ADAPTE project [1].

ACKNOWLEDGEMENTS
This work has been partly funded by project Fondef D0911171 (ADAPTE) of Conicyt, Chile.

REFERENCES