**A Methodology for Model-Driven Software Configuration Management Implementation and Support**

Arturs Bartusevics

*Faculty of Computer Science and Information Technology, RTU, Kaļķu str. 1, Rīga LV-1658, Latvia arturik16@inbox.lv*

**Abstract:** The paper presents the methodology how to implement model-driven software configuration process in projects of developing applied computer systems and software. A methodology consists few dependent blocks that together allows decrease manual works of implementation and support configuration management process. The most important blocks of the methodology are problems and actions monitoring systems in project and database of knowledge. Monitoring systems can fix all problems in project. Knowledge database consist many rules. Using forward chaining procedure become possible to generate new rules in knowledge database that helps determine is some problems depends on configuration management area and find solution of this problem. New rules generation process help to make corrections in configuration management model. Configuration management model presents all actions and relations needed for implementation of configuration process. Monitoring systems and knowledge database can change current model according to changes in project. The paper describes also risks and limitations of using this methodology and necessary researches to make this methodology useable in projects.

**Keywords:** configuration management, model-driven process, knowledge base, monitoring systems.

**Introduction**

Recent development trends of information technologies show attempts of transforming processes into reuse-oriented and model-driven ones(Osis and Asnina, 2011). The time when success of an information technology project was determined by a number individuals that were masters in their field who would use certain tools to solve particular problems, has come to an end.

Software configuration management is no exception to this. Vast development is taking place in this field. Software configuration management is a discipline that ensures product identification and control of their versions in the course of an information technology project. Another task completed by configuration management is creation of a product and it reaching the customer’s operating environment correctly(Hass and Mette, 2003).

Although configuration management has long been defined by various standards of information technologies and various tools have been created to solve particular configuration management tasks, problems still persist. When software is developed, it can be installed in test mode, however in operation or production mode unexpected errors emerge upon installation. An information system is supported by two developer teams placed in different countries. One of the teams corrects errors and the other team develops new functionality. There are situations when both teams need to make changes in the same source code file. The error must be corrected as soon as possible, yet the new functionality will be put in operation only within a month. It is unclear how the changes should be made and stored. The product to be developed successfully and correctly compiles when both teams need to make changes in the same source code file. The time when success of an information technology project was determined by a number individuals that were masters in their field who would use certain tools to solve particular problems, has come to an end.

Software configuration management is no exception to this. Vast development is taking place in this field. Software configuration management is a discipline that ensures product identification and control of their versions in the course of an information technology project. Another task completed by configuration management is creation of a product and it reaching the customer’s operating environment correctly(Hass and Mette, 2003).

Listed here are only a few of the problems that reveal shortcomings in the field of configuration management. As mentioned above, despite the vast amount of configuration management tools, problems continue to persist. There can certainly be various problem causes in each separate project, however the author believes that the following major causes are usually present:

a. Lack of time and resources to view and introduce configuration management as a discipline, which is particularly characteristic to smaller projects. Often development is started, yet practical issues that emerge due to poor configuration management are only solved when the issue stands in the way to functioning;

b. Absence of a broadly recognized universal plan or methodology that would allow introducing a full configuration management process that complies with IT standards within an acceptable period of time. Often there are people or a single person in the project or company who is able to solve the particular configuration management problems within the project, yet there is no general methodology or a model prescribing how to do it. As a result either the person introduces configuration management only partially and for the purposes of the particular situation or the person is drawn in only when the problem appears.
Configuration management often is not perceived as a targeted process, but as concrete tools that “have to be in the project”. When asked about the essence of configuration management, some well-experienced IT specialists state a particular version of the control system, continuous integration server, application processing system or another tool this person uses in the project. Without a joint and targeted process new practical problems continue to emerge, and often new problems appear when the old ones are solved (http://experience.openquality.ru/software-configuration-management). This is because there should initially be a targeted process. The author believes at the start of the process goals must be defined, which will prevent concrete problems, and only then can the tools that reach these aims be installed. For instance, before installing a version control system, one should think of how and in which structure the source code will be stored, who will have access to it and whether several parallel branches will have to be sustained, etc.

d. Another problem lies in the project development, namely, the project develops and changes so rapidly that configuration management can no longer solve all the problems and complete all the tasks. The initial configuration management plan then becomes irrelevant, yet there is no time or human resources to bring it up and sustain it. For instance, a company sustains a certain information system and the everyday work is organized in a way that users e-mail their problems to the support team of programmers. While the number of problems does not exceed 100, everything runs well, but what to do when e-mails with issues reach into thousands, and it is found that there are several problems concerning one and the same issue or cases when two or more problems flagged are in conflict with each other? Rapidly introduce one of the application support systems? If so, then how to transfer all the raised problems from the e-mail and how to detect the conflicting ones? In this case there can be many questions, yet it will not be possible to obtain answers rapidly. This is due to the fact that initially the application processing system was not well planned out and it was not taken into account that the project can develop and number of issues may grow considerably (http://experience.openquality.ru/software-configuration-management).

Given these problems that persist in information technology projects, it can be concluded that a methodology is needed that would allow an effective and targeted introduction of configuration management processes into the project. It is needed to advise on more suitable tools for achieving the goals of the particular process and to support configuration management process in accordance with the current needs of the project throughout its course. In other words, the methodology should implement the development of configuration management process and its adaptation to the changing circumstances of the project.

General Description of the Configuration Management Implementation and Support Methodology

Configuration management as a discipline is regulated by several information technology standards, such as ITIL or CMM/CMMI (Hass, 2003). According to these standards a configuration management plan and model can be compiled, which would be suitable for a particular project at a certain time. Afterwards technologies and tools would have to be chosen in accordance with the plan, which would help solve and automate configuration management tasks. When technologies and tools are chosen, these can be installed and the configuration management process can be regarded as implemented within the project. Is that really so? Before answering this question I will mention a few problems that persist in practice and which the author of this article has encountered:

1. When a new project is started, there is lack of human resources and time to study configuration management standards and to compile a configuration management plan and a model for the particular project. Within the actual situation such activities are rarely given enough resources. Most often definition of requirements is started, along with setting up the project and initial development, yet problems that emerge due to poor configuration management, are solved locally without giving much thought to their causes.

2. Even if there are attempts to initially introduce a joint and targeted process of configuration management, the project develops too quickly and soon the initially defined process does not match the actual project situation. As a result the process is defined, however new problems continue to emerge regarding version control, change management, product compilation and installation, and other problems linked to configuration management.

What follows is that an initially thought out configuration management process plan does not yet mean that we can speak of a full-fledged and targeted process that would help avoid problems and obtain a qualitative product. This is also because configuration management is perceived as a necessity to solve several local problems or a process is introduced that quickly looses its effectiveness due to the vast pace of project development. In this case the project often contains a document that kind of describes configuration management, yet it does not fit the actual situation.

All of the above shows that a method is required that would allow doing the following:

1. By using only the available resources, to sufficiently quickly set up a configuration management process, which would comply with a concrete project at a particular period of time.
2. To ensure process development in accordance with the development of the project. Development must be automated within the scope of possibility. That means that the process must change along with the project, and there should be a continuous link between configuration management and other project processes.

By developing the methodology, which could solve the mentioned tasks, two assumptions had to be defined, upon which the configuration management implementation and support methodology will be based, as described in this article. First, each information technology project contains particular actions. The project takes place while there are particular actions. As an example of these actions we could mention development of the specification, writing of the source code and testing of memory, module compilation, preparation of a project report, etc. Second, as a result of actions, certain problems can emerge. For instance, a module has not been compiled, a mistake has been made when testing memory, a programmer has overwritten the changes made by another programmer when modifying the source code. Therefore each project contains actions and problems that emerge when certain actions are performed. Whereas the aim of configuration management is independent of the type of project and technologies used in the project. The aim of configuration management is to ensure product identifications, version control and changes reaching the end product correctly.

If we imagine that there are actions and problems in each project, then the task of the configuration management process is as follows: to minimize the number of problems that emerge by taking actions to reach configuration management goals.

Given the just formulated configuration management task, the following methodology for implementing and supporting configuration management is offered. In the initial stage of the methodology, monitoring of actions and problems is performed. There are two systems that record all the actions and problems. During the action monitoring only those actions are obtained form the set of actions that are oriented towards a solution of configuration management tasks. When actions related to configuration management are obtained, the link between these actions is defined. In the second step of the methodology actions and links between them are transformed into a model. In the following step the model is supplemented with action attributes, which describe actions from the aspect of the chosen tools and technologies. Parallel to action monitoring and transformation, problem monitoring takes place in the model. Each problem is registered in the storage. Later on it is determined due to which actions this problem has emerged. If this action belongs to a set of configuration management actions, a problem solution is sought. In order to find a solution, a base of configuration management rules is required. This base finds solutions to concrete problems and then introduces these changes to the model. Action and problem monitoring takes place continuously. Changes to the configuration management model are made in two cases: if a new action is defined that concerns configuration management and if a solution to one of the configuration management problems is found in the configuration management base of rules. Fig. 1 shows a schematic representation of the methodology, showing components of the methodology and links between them.

![Fig. 1. Methodology for Developing and Supporting the Configuration Management Model.](http://aict.itf.llu.lv)
Methodology consists of five elements and eight data flows, which ensure a link between methodology elements. A configuration management model is one of the components of the methodology. The model changes and develops while the methodology is in action. The general goal of the methodology is to sustain the configuration management model in line with the current situation of the project. Whereas the goals of the model is to considerably show to all project participants the configuration management process in a clear way. Another goal of the model is to show in what ways each of the configuration management tasks is solved and to what extent this solution contributes to development of a correct end product, which is one of the most important configuration management goals.

**Description of Methodology Elements**

**Action Monitoring**

This system continuously analyzes all actions taking place in the project. The action monitoring system receives information when a certain event has taken place within the project. Afterwards the obtained information is analyzed. As a result of information analysis action is identified. One action in the context of this system is the vector D< I, A, P, S>, which describes one particular action and where:

- I – identifier of action,
- A – actor that performs this action. This can be a concrete person from a project, for example, a tester, a programmer or a system analyst. The actor in this case can also be one of the systems used by the project, for instance, a continuous integration server, in which the automatic project tests are running,
- P – action performed by the actor. This is a concrete verb that clearly characterizes a particular action, for instance, to test, to compile, to write, to register, to generate, etc.
- S – a set of action attributes. Attributes help better understand the meaning of a particular action. Attributes contain information about what exactly the actor carries out, which tools it uses, and which systems it works with. At the moment when the action monitoring system contains several actions, the S set can also contain an attribute that shows links with other actions.

When information about the project is processed and the attempt to transform it into a formal definition of action is successful, action-monitoring system sends information about the actions to the configuration management rule base, from which it receives a response whether the action concerns the configuration management process. In case of a positive answer, information about actions is sent further to the transformer – an element that transforms action vector D into an element of the configuration management model. The action monitoring system functions on an ongoing basis.

**Problem Monitoring**

The principle of problem monitoring system activity is very similar to the action monitoring system. The system continuously analyzes events in the project with an aim to identify errors and problems. When a problem emerges in the project, for instance, changes have disappeared, tests work with errors, source code cannot be compiled, then the problem monitoring system is triggered. The obtained information is analyzed and information about the problem is formalized. Within the given methodology a problem is a vector P< I, A, D, R, R1>, where:

- I – identifier of a problem,
- A – a noun that characterizes what the problem concerns, e.g., a source code file, a compiler, or a specification. This can also be a person who caused the problem, for example, the programmer.
- D – a verb or verbs that characterize the essence of the problem, e.g., did not compile, lost, did not work, etc.
- R – reference to an action, as a result of which the mentioned problem emerged. At the moment when the problem emerges, the value of R is empty. It is only filled when a response is obtained from the action monitoring system.
- R1 – possible solution of the problem. Initially the value is empty. It is only filled when a response is obtained from the action monitoring system.

When the problem is formalized, information about it is sent to the action processing system. That is where all actions are searched with the aim to identify the action due to which the problem emerged. At this point the attribute of vector R is filled. If the action cannot be identified, the problem is recorded into a special list of the problem monitoring system, which will only be analyzed once the set of actions within the action monitoring system is changed.

After a response is received from the action monitoring system and the problem vector attribute R is filled, the problem is sent to the configuration management rule base, where solution to the problem is sought. If the solution is found, the problem monitoring system receives a response that the problem has been solved and it is marked as resolved. At this moment attribute R1 is filled for the problem. Whereas if no solution is found, the problem is sent from the configuration management rule base with an empty R1. In this case the problem is added to a special list by the monitoring system.
Configuration and Management Rule Base

This element of methodology is of great importance. It determines whether the configuration management model will be able to correctly develop according to the new project requirements and the changing situation.

The configuration management rule base consists of the following elements:

- Action mapping – a map that contains information about actions required to solve one of the configuration management tasks. The map consists of several elements. Each of these elements has two attributes. The first attribute is one of the configuration management tasks, for instance, ensuring version control, yet the second element are the actions necessary for solving this task as well as a list of potential tools and technologies to be used when solving the given task. It is by following this map that allows determining whether action D from the action monitoring system concerns the configuration management discipline or not.
- Problem solution mapping – a map that contains problems and their potential solutions and tools to implement these solutions. This is the map on which search takes place when looking for a solution for a concrete problem P.

The configuration management rule system is hosted by an individual with theoretical and practical knowledge on configuration management. For instance, it can be a configuration manager in the project. The question in what ways to implement the ability to process the mentioned map elements with the help of software remains yet to be answered. It needs to be acknowledged that regardless of map realization, the development and hosting of a qualitative system will not be possible without the intervention of a specialist.

Transformer

This is a tool configuration model building. It receives actions D with attributes and transforms the acquired vector into the model. Realization of the transformer depends on what type of model is used. In any case the model has certain elements and links between them. The transformer receives one action D<I, A, P, S> and it has to be an algorithm that transforms this vector into a model element. Realization of the transformer will not be reviewed in this paper. In order to realize the transformer, additional research and experiments are required.

Configuration Management Model

This model is the main component of the methodology. The general goals of the model are as follows:

- To demonstrate the solution of configuration management tasks as a targeted process;
- To demonstrate what tools and technical means solve which of the problems and in what ways the tools interexchange information.
- To help the configuration manager avoid a situation, in which the solution of one problem is the cause of another;
- To help the configuration manager avoid the emergence of similar problems once a certain problem is solved.

In the context of the described methodology any project problem that refers to one of the configuration management tasks is a shortcoming or an error within the configuration management process model. Therefore by fixing this shortcoming or this mistake, first of all the particular problem is solved and second of all theoretically a situation is achieved, in which a similar problem will not repeat itself. It is therefore crucial for the model to be as precise as possible, while the process itself should fit the particular project as a much as possible.

This paper will offer a configuration management process model, which is based on actions from the action monitoring system. Upon receiving the action vector D, the transformer develops the following model components:

1. Action – the main element of the model, which reflects a particular action. This element reflects attribute P from the vector D.
2. Object (actor) – an element that reflects attribute A of vector D. It is a person or an object that performs action P.
3. Action attributes – elements that reflect attribute S of vector D. These are attributes characterizing action, which clarify the meaning of action, show in what ways the tools are used when performing this action, as well as what kinds of interfaces exist for communication with other tools (attributes).
4. Report – an element that contains information or a set of transfers that emerge after actions are completed. It is a report that is sent to another object (actor), which belongs to a different action.

Therefore the model of the configuration management process consists of four basic elements. The main element is action. Action can have one or several points of entry, in which the object (actor) activates this action by giving all necessary information. Action has a characterizing set of attributes, which gives full understanding on the course of action and its tools, helping to complete or to automize this action. Action also has one or several points of exit, which contain the report that, in itself, can either be a transfer of a particular action or a set of...
information for another object (actor). This information may be necessary for another object (actor) in order to activate the next action of the model. Fig. 2 shows a fragment of the model in the context of a single action.

Fig. 2. Fragment of a Configuration Management Model.

Tools and Obstacles to Introducing Methodology

When discussing the possibility of making use of the described methodology in current practice, the author detects several risks and obstacles to implementing the practical approbation of the methodology. There is potential of the following risks:

- Monitoring systems can be imprecise or may not be suitable for the particular project. Therefore the model development process in line with the methodology may be imprecise or contain errors. If monitoring cannot identify certain actions or problems significant from the aspect of configuration management, it will not be possible to develop a precise model corresponding to the real-life situation.
- The configuration management rule base may have shortcomings and it may be imprecise, which may lead to inability to classify the actions registered in the project. As a result the model may contain a certain action that does not at all correspond to configuration management and rather is related to a completely different project discipline. The configuration management rule base may also create obstacles to problem solution, for it may not contain a solution for a particular problem. As a result the problem will remain to exist within the project and the model within a model will not detect a possible solution. The configuration management rule base may also contain a faulty fact, which is based on a subjective assessment of a particular specialist, yet which in reality is untrue.
- There is a risk that the formalization of the problem solution will not be of sufficient quality in order to correctly introduce changes to the configuration management model. The idea is that it is not enough to find a solution to the problem, for one must also present this solution to the model transformer in a clear and unambiguous way in order to make correct changes to the model. If a solution is found in the rule base, but in the course of solution formalization mistakes are made, the changes in the model may not solve the problem and it may repeat itself in the future.

Another risk group has not just to do with methodology, but with the current situation within the discipline of configuration management. IT specialists, instead of perceiving configuration management as a targeted process that solves tasks of a certain kind, may sometimes perceive it as a total of tools and technologies, without which the "project will not work", according to them. Such specialists are not likely to try something new and view configuration management as process- and model-oriented. The author believes that sometimes due to psychological reasons it is difficult to choose something completely new and unknown to this point, for in the precious years of the existence of the IT field there has been a tendency to perceive configuration management in the context of tools. Yet taking into account the modern tendency for the field of information technologies to establish itself as a science and processes to become model-driven, to perceive configuration management solely as a set of several tools is incorrect and not in line with the current tendencies (Frakes, 2005).

Conclusion

Given that the methodology is new and there is no detailed description of its elements and implementation, the author believes that we still have a long way to go to achieve practical approbation of methodology and result assessment. Therefore several tasks must be completed, which should not just strengthen the methodology, but also create possibilities for applying it in practice. Given the current situation, the following future tasks are at place:
To work on possibilities of implementing and introducing a monitoring system. It is necessary to study the existing monitoring systems and to draw conclusions which of them would be the most suitable for action and problem monitoring. One problem to be solved is signal generation, and the other problem – what information must be in the signal in order to formalize it into an action or problem vector.

Working with the configuration management rule base. As mentioned before, the rule base is an essential component of the described methodology. For the rule base to be able to solve various problems, it should contain many rules, facts and a summary of best practice examples. Given the fact that configuration management is currently being described by many sources of information and is implemented in practice by numerous tools, development of such a rule base manually would be a lengthy and irrational process. As noted before, one of the advantages of the described methodology is the ability to apply it from the very start of the project, without investing additional resources into preparatory events. And if an extensive configuration management rule base is to be created manually, the methodology looses its meaning. From here we can conclude that we need a sufficiently fast and automatic way to create a configuration management rule base. We therefore need to search for ways how to achieve this. The author allows the possibility of having to use one of the data acquisition or artificial intellect methods, or possibly even several methods at once. Some thought should be given to depiction of rules in order for the monitoring systems to work with the rule base.

Given the existing tendency to make the IT processes reuse-oriented and model-driven, it can be concluded that the attempt to make the configuration management process into a model-driven one is modern and corresponds with the tendency of the field of information technology to reinforce itself as a scientific field.

References