The CESAR RTP as Product Line: The Configuration of an Integrated Tool Chain

Peter Graubmann, Michael C Jaeger, Reiner Schmid

THE RESEARCH LEADING TO THESE RESULTS HAS RECEIVED FUNDING FROM THE ARTEMIS JOINT UNDERTAKING UNDER GRANT AGREEMENT N° 100016 AND FROM SPECIFIC NATIONAL PROGRAMS AND/OR FUNDING AUTHORITIES.

The CESAR RTP as Product Line: The Configuration of an Integrated Tool Chain

Peter Graubmann, Michael C. Jaeger and Reiner Schmid
Siemens AG, CT T DE IT 1
peter.graubmann@siemens.com, michael.c.jaeger@siemens.com, reiner.schmid@siemens.com

Abstract. Industrial development usually combines different methodical approaches and, in more or less all cases, does employ a variety of different tools in its development process. The CESAR EU project aims at building a platform for the integration of these tools. This integration platform is a highly customizable asset that must be adapted to the organisational and technical characteristics of the partners' projects. To solve the configuration problem, the platform is considered as a product line with the integrated tool chains specifically tailored for the partners as its product line members.

Keywords: integrated tool chain, tool chain configuration, variability model, product line engineering.

Abbreviations: CESAR, RTP, OVM, PLE, CVO.

1 Introduction

The CESAR project addresses problems faced by the transportation (automotive, aerospace, and rail) and automation industries that share a common need to develop reliable embedded systems in a competitive global market. A solution to these problems is to identify processes, methods and tools able to increase the productivity during system development and validation phases, while maintaining very low risks of failures.

The CESAR project is a place where such tools and methods should be identified, improved and integrated to provide a technical platform (together with associated development processes) that cover the entire product life cycle, encompassing: requirements management, architecture design, verification and validation, as well as diagnosis in operation. The goal of this CESAR Reference Technology Platform (RTP) is to provide a flexible, highly adaptable reference platform from which specific integrated tool chains can be derived for the development of embedded systems in the automotive, aerospace, rail and automation domains which are tailored to the particular requirements of the users.

From its nature, the RTP can be understood as the core asset base of a "product line" that produces integrated tool chains that have to meet the respective embedded system development needs as specified by the different stakeholders in the CESAR project. The configured tool chain is essentially determined by the following constituents:

- The development process which the tool chain is envisaged to support.
- The domain-specific regulations and standards.
- The company-specific requirements, settings and guidelines.

These three constituents allow for a rather large variation in the resulting tool chains: Adapting the product line paradigm in order to configure the tool chains according to the stakeholders' demands means to describe the possible variation with an appropriate variability model. In this variability model, the information addressed by the above listed three constituents, however, has to become tangible, that is: it has to be properly modelled in sufficiently great detail. Therefore we explain to what extent configuration options apply for the RTP and how this fits into a product line engineering approach for creating RTP instances.

The paper will present our approach in the next section 2. The following section 3 shows more details about the CESAR RTP. Section 4 explains a relevant related field where the configurability has already been established: the V-Model XT which is a process model for software projects. Section 5 describes the concrete application to the RTP. The paper ends with our conclusions in Section 6.

2 Approach

To outline slightly more precise what kinds of models are behind our approach we will first discuss the following three main categories:

(a) The development processes followed in CESAR have to be described as a collection of development process fragments that can be easily composed (the models are "fragments" in the sense that they partition CESAR relevant development processes, like, for instance the V-model or development in a product line engineering effort, and they have to allow for a deliberate composition according to the stakeholders' wishes). The development process fragments reference the respective methods and tools that are to be used in their context.

(b) Models of the relevant regulations and standards, expressing their respective requirements to development steps, methods, tooling, etc. (if, for instance, a governmental regulation requests the documentation for certification purposes in a certain format, then the configuration of a tool chain is influenced by this if the certification is relevant in the respective business; or see for instance safety standards like IEC 61508 (industry automation) [IEC] or ISO/DIS 26262 (automotive) [ISO] and their requirements to the development process).

(c) Models of company-specific definitions and settings, that is, proprietary development process steps, specific collaboration requirements (a distributed development, spread over several sites all over the world, results in other requirements to a tool chain as a development in a small collocated team), or, for instance, infrastructure and technology requirements (some tools pose certain requirements on the available technology; some require a certain infrastructure or a certain type or amount of repositories, etc.).

Besides the models according to the above described main categories, which identify the main features of a tool chain, additional information appears to be necessary to configure the details of a tool chain. This kind of information is provided within the following additional models:

- Models of the available/existing infrastructure technology,
- Models of the collaboration structure,
- Models of development methods,
- Models of management activities,
- Models of the available tool chain tools.

Models that express the domain and the product influences are not explicitly mentioned since they are assumed to be mainly expressed by the combination and selection of the above listed models. To have a name for them and to allow for an easier way of referencing them, the collection of all the models listed above is referred to as *RTP base models* (since they form the core asset base of the RTP for tool chain integration). Meta-model languages may be used as a common language to define these RTP base models.

The above list of RTP base models may seem to be rather lengthy and there may be the question, whether all these descriptions are necessary. But undoubtedly, the respective information may influence a tool chain generation. However, which one of the above indicated RTP base models is really needed and what the proper abstraction level will be, has to be elaborated on concrete trials and validated by practical experiences.[1] An example of what kind of information to what extent is needed is provided in section 4 by a discussion of the systems development model "V-model XT" and the CESAR Variability Ontology (CVO) which provides for a product line engineering variability model that can be used for the derivation of tool chains in the product line development context. For development process fragments, for instance, a model using a workflow modelling language seems to be appropriate. In any case, it is essential, to focus on the abstractions that allow to express the relations between the respective model elements that are necessary to configure accurately a tool chain as needed by a CESAR stakeholder.

Several CESAR partners use the system development model "V-Model XT" as development paradigm. It provides tailoring based on the project's type, variant and characteristics and ensures that for a particular instance, all mandatory activities are performed. Then, the project manager must create a project plan which defines the schedule of the activities and the decision points in the project. After everything has been setup, the project can start.

A similar concept can be followed in a case where a PLE approach is chosen for the development of applications. Within the CESAR project, a product line ontology, the CESAR Common Variability Ontology (CCVO) is developed that describes the development process for product lines in detail. Variation points are given here in particular with respect to the needed domain and application engineering sub-processes and the thereby employed methods and tools.

[1] The list of RTP base models given here tries to be rather exhaustive; this is mainly thought to stimulate a fruitful discussion.

When configuring a tool chain, the basic questions to be answered are what methods and tools should be used and what are the consequences with respect to other tools already incorporated in the tool chain. These essentially are the kind of questions that are asked during a resolution of a feature tree. Thus, the models, listed above, have to provide the relations that determine how the model elements can be used in combination. The basic relations of this kind are:

- "model entity X *requires* model entity Y"
  (as, for instance, the use of a requirements management method is required as soon as a requirements analysis step occurs in a development process), and
- "model entity X *excludes* model entity Y"
  (as, for instance, the use of Linux as platform excludes a tool that exclusively runs on Windows).

In more complex situations, it may happen that a certain set of model entities *requires/excludes* a certain set of (other) model entities. Actually, in concrete cases, there may be rather complex relations based on this *requires/excludes* relation (as, for instance, "if model entities U and V are present, then also model element Y has to be present but model element Z must not be present"). And, of course, the *requires* relation can be weakened to the request that a certain model element may entail the presence of one (or more) elements of a given set of other model elements. Or the presence of the model elements may be optional.

Actually, with the above relation, we aim at a structured "feature" or variability model as a coherent and concise representation of the variation with respect to the complete set of all RTP base models. This RTP variability model is expected to be orthogonal to the RTP base models. Here, "orthogonal" means that the description of the variability is not intermingled with the models themselves but is represented by a separate model. This facilitates management and exploitation, and in particular the evolution of the variability model as well as the other models to a considerable extent. Usually, in a "classical" Orthogonal Variability Model (OVM), the base models refer to the models of the "classical" engineering disciplines like requirements models, architectural models, realisation models, or test models, and so on. Figure 1 shows the relations within the OVM according to a diagram taken from the MoSiS Project [MoSiS]. In our context, the OVM "Base (Meta) Models" are replaced by our, the tool chain constituting models we have dubbed "RTP base models".

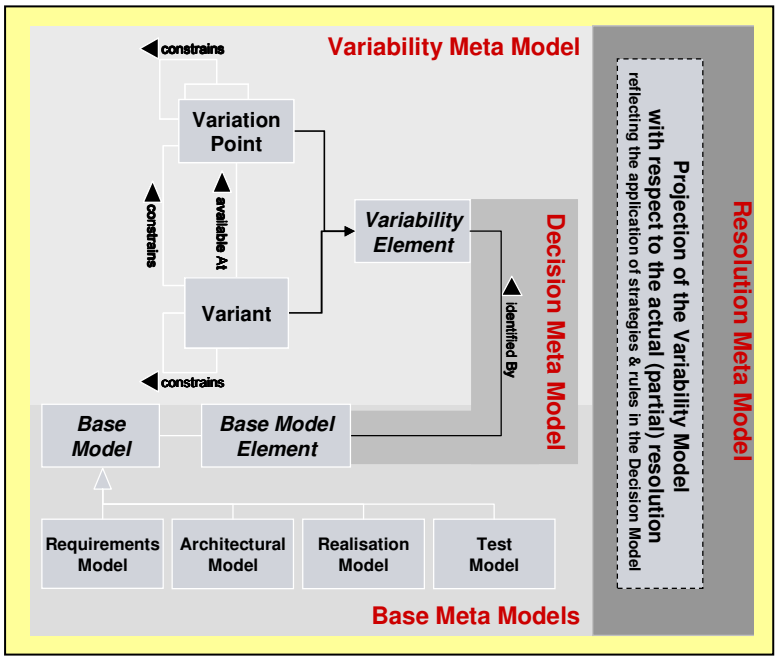

Figure 1 Orthogonal Variability Meta Model OVM [MoSiS]

The variability model describes the variation possible in these models including the relations (constraints) between variants, essentially based upon the above mentioned *requires/excludes* relation. The associated decision model then describes the strategies and rules according to which the variability is resolved and the variants are selected. The resolution model eventually describes the respective configuration according to the resolved variability. The variability may be completely resolved. Then the final configuration is described by the resolution model. The variability, however, may be only partially resolved; that means, there is still some variability left in the model which can – and actually has to– be resolved later.

As already stated above, in the case of RTP tool chain configuration, the base models of Figure 1 correspond to the RTP base models described above; and the RTP variability model is based on the RTP base model elements. Configuring an integrated tool chain consequently means to resolve the variability in the RTP variability model according to the needs of the envisaged development process.

Starting point for a particular tool chain configuration process is the specific development process selected for a particular product development for which a proper tool chain has to be configured. Thus, before the tool chain can be derived, it becomes necessary to determine the development process.

According to this dichotomy, the various RTP base models are separated into:

- The *process models* which are the models, relevant for the definition/derivation of the company specific development process:
  (a) the development process fragments, the regulations and standards, the company specific definitions – these models describe the building blocks from which a company/business specific development process can be composed;
  (b) the methods, the management activities, and the collaborative structure – these models describe the means that are addressed and utilized by the development process: and
- The *tooling models* which are the models describing the tools and their infrastructure technology and thus relevant to derive the tool chain.

The orthogonal RTP variability model connects all related model elements, of course also across the boundaries of the separation between process and tooling models (see Figure 1).

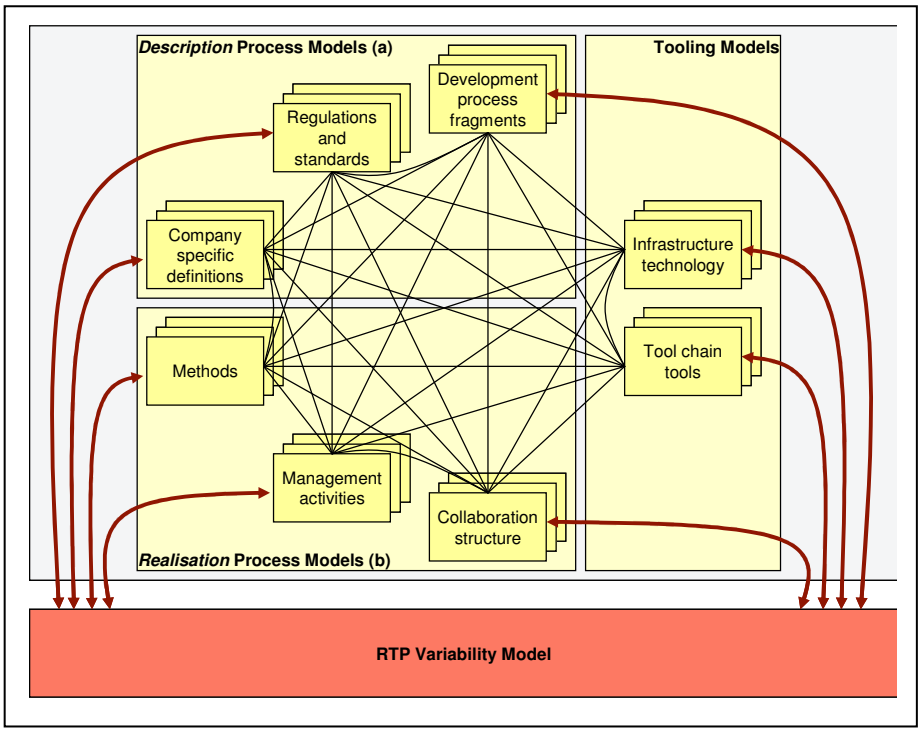

Figure 2 The RTP base models and the RTP Variability Model: their elements are potentially closely related (but, of course, there is not necessarily a relation between elements of each base model, as the lines could possibly suggest)

The descriptions of the development variations (the choice between the "development fragments"), the variations caused by domain-specific regulations and standards, the variability introduced by company specific definitions, as well as the peculiarities of the methods, processes and the collaboration structure have to be formed into a RTP variability model. This part of the RTP variability model is called the RTP Process variability model.

The variability model behind the tooling models is the basis of the tool integration, that is, the selection of the best fitting infrastructure technology together with the tool chain tool peculiarities and adapters. This variability model is called the RTP Tooling variability model.

Together, the RTP process variability model and the RTP tooling variability model form the RTP variability model which describes all the possible variations allowed for the configuration of a CESAR RTP compliant integrated tool chain. Its resolution eventually provides the blue-print of the tool chain.

3 Application to the RTP

Following the separation of RTP base models in the previous section, configuring a tool chain for the RTP is an activity that has to be executed on two levels:

- There is the "upper level," on which the appropriate (domain-, company-, and team-specific) development characteristics have to be chosen. This means the resolution of the variability in both the description and the realisation process models (as defined above in Figure 2). The resolution of the variability given on this level leads to a detailed determination of the development process.
- And there is the "lower level" of tool integration where the interoperability of the CESAR tools has to be ensured (see the tooling base models in Figure 2). The configuration of the right interoperable tools (and probably the needed adapters for model exchange) is done on this level.

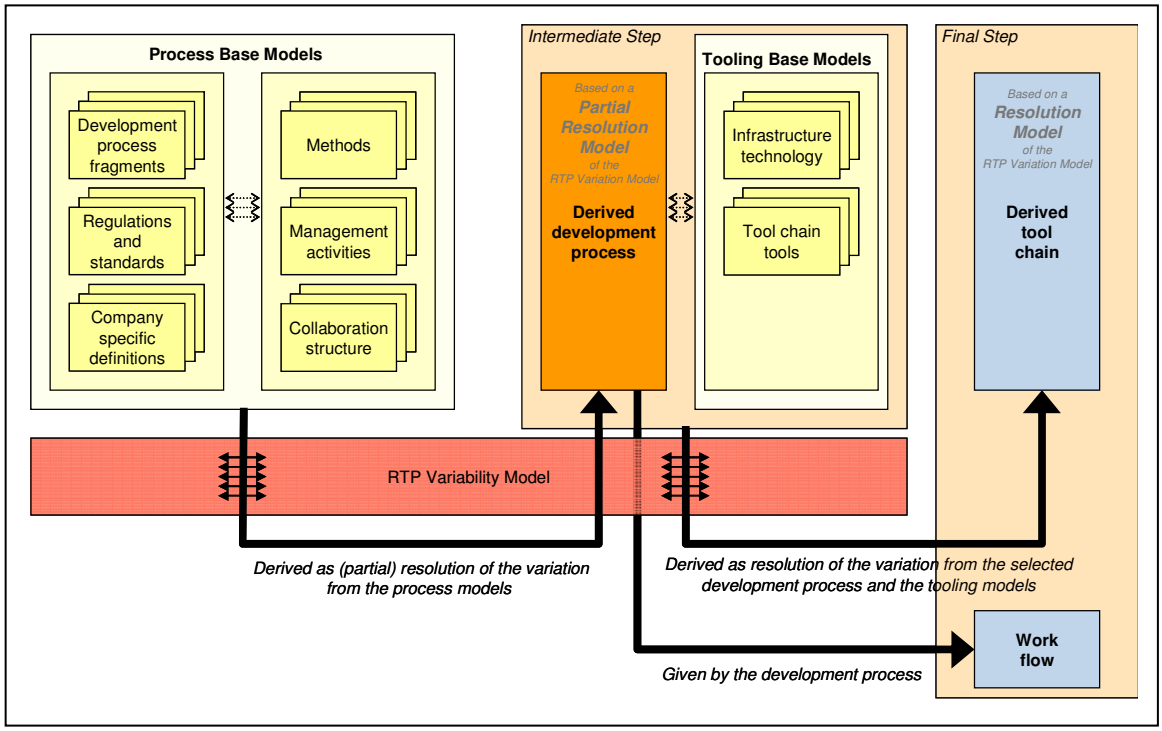

Figure 3 Tool Chain Configuration Process Overview

Configuring a tool chain on the "upper level" in fact means to select from the variation provided within such models as the V-model or the product line development model. The approach pursued here tries to define fragments of the development processes like the V-model or the product line development process that can be identified and selected (always referring to the given context). However, selecting and configuring development process fragments is not yet enough. The regulations and standards that are to be used in a certain domain have to be considered as well; they influence the selection of the possible methods and tools and they determine additional activities and adaptations to the processes. Furthermore, there are usually also company-specific definitions and internal standards which influence the tool chain configuration steps in addition. This is the first part of the resolution process in the "upper level"; it corresponds to the resolution of the development process variability described in the description process models (see Figure 2).

The second part of the resolution process in the "upper level" is concerned with the so called realisation process models (see again Figure 2), that is, with the selection of the appropriate methods (development process fragments may allow the use of different methods), (product) management activities and collaboration structures (that is, the team management structures).

Configuring the tool chain on the "lower level" is based on a decision for a definite development process. It adds the tooling to the development process, that is, it resembles the selection of the proper infrastructure technology and the adequate tool chain tools with their peculiarities and adapters. The separation of the resolution process in the upper and lower level is intended to decouple the determination of the development process from the technology constraints. Practical experiences have to show to what extent this separation is really possible.

Figure 3 gives an overview of the RTP tool chain configuration process: The RTP Process variability model built from the variations in the process base models is resolved according to the given requirements. This leads to a partial resolved RTP Process variability model which essentially corresponds to the chosen development process in detail. A workflow can be derived from it. Based on the RTP Tooling variability model, the resolution can be completed and provides the blueprint for the tool configuration.

The orthogonal RTP variability model connects all related model elements, possibly also across the boundaries of the separation between process and tooling models.

The different RTP base models may have various origins. Most important and widely used models are defined by the RTP owner (an open source like model can be envisaged to maintain and evolve the RTP base model base) who is also in charge of evolving the corresponding RTP variability model. In contrast, company-specific definitions have to be brought in by the companies themselves. It seems however advantageous if the company-specific definitions do not add their own model elements but restrict themselves to rules and selection strategies (being maintained in the decision model and applied during the resolution process).

Keeping the RTP variability model up to date with the RTP base models is presumably no simple task and some support should be developed. Semantic techniques, like ontologies to identify related model elements, might be of help. Probably, web services orchestration techniques can be exploited, too. IBM's JAZZ platform [JAZZ] for collaborative work may be of help. In this context, it however remains important that the development process fragments are related as variants and thus, a consistent selection of the development process fragments is configured into a suitable development process.

The derivation of the development process thus becomes a selection of the appropriate development process fragments – and the variability model ensures that constraints, introduced by the selection of a certain fragment will be consistently maintained during the entire selection process. Thus, for instance, the decision for a "safety development process V-model" (see [VModelXT] and [ToolChain]) leads to the choice between safety-oriented requirements processes. The selection of one of them leads to the choice of suitable methods and selecting one means that the development process is defined at this point. Methods entail suitable tools; end eventually, the identification of technological constraints defines the tool chain. Thus, resolving step by step the variability model with respect to the needs of the specific product development model in mind and employing the automated resolution support provides for a consistent development process and an adjusted tool chain. (This, of course, does only work provided that the RTP base models and the RTP variability model are defined correctly and sufficiently interrelated.)

If a rich base of RTP base models is available, project leaders (architects, etc.) can define a fine-tuned development process and the associated tool chain. The description of the development process as intermediate result provides a workflow (and not all workflow elements have to be necessarily related to tools).

The difficult part is to maintain and evolve the RTP base models and the RTP variability model. The relations and language elements defined in [ToolChain] have to be integrated, in particular into the tool chain tool and the infrastructure technology models. Integrating tools means thoroughly consider the tool properties and it becomes especially crucial that the elements of the tool chain can only interoperate by exchanging adequate documents/data which makes appropriate transformations necessary.

The envisaged work will elaborate on this tool chain configuration approach. It will take, as a starting point, the V-model and the PLE paradigm as starting points for the derivation of the development process fragments. A selection of tools in the RTP will provide the targets that can be associated and integrated with the development methods in order to form the RTP variability model which can be used as configuration basis.

4 V-Model XT: An Example for Configurable Project Setup

The idea of an adaptable tool chain refers directly to customisable development processes. A prominent example for a customisable development process is the German V-Model XT, which represents the latest development of the V-Model from the early nineties. The V-Model is mandatory for federal administration and defence projects, and a similar approach, the US V-Model has been adapted by the US National Council on Systems Engineering (INCOSE). The V-Model is guidance for planning and running projects with an emphasis on verification and validation.

The latest development of the V-Model, the V-Model XT puts tailoring of the project structure as the fundamental step when setting up the project, which is called static tailoring. Opposed to that, the V-Model

supports also dynamic tailoring which allows the project manager to react to changes that occur during the project. The static tailoring performs as follows:

1. Selection of a project type (e.g. running this project in an owner or contractor role)
2. Selection of a project variant (e.g. software development or maintenance project)
3. Identification of project characteristics (e.g. involving hardware development, or requiring special dependability features)

By this three-step determination of the software project, the V-Model provides a defined set of project activities that a project must perform. For example, a development project from a project owner perspective must involve the project activity requirements engineering. Opposed to that, a system development project run by a contractor does not have requirements engineering as mandatory element.

The V-Model provides tailoring based on the project's type, variant and characteristics and ensures that for a particular instance, all mandatory activities are performed. Then, the project manager must create a project plan which defines the schedule of the activities and the decision points in the project. After everything has been setup, the project can start.

The tailoring today is supported by tools. A project manager can perform the tailoring by a GUI and continue with the project plan and further documentation including tools like Microsoft Project. Referring to a customisable RTP, the V-Model resides on a higher level of abstraction. While the V-Model covers the organisational view, the RTP covers the development tool view. Accordingly, the idea of a configurable RTP is the natural consequence of an adaptable development process like the V-Model. As mentioned above, the development process is considered one of the direct inputs for the configuration of the RTP.

4.1 The Transition to Tools

The V-Model XT is about the development process. As a standard V-Model XT project setup, the project manager arranges the resulting activities of the tailoring effort to a project roadmap. Tools exist that also support this step. The definition of activities in the V-Model XT does not contain the "How?", an activity is just defined by its input and output, its responsibilities and what to do (the tasks inside the activity). However, it does not define how to reach the product, which tools are used, how many persons are involved or which technique should be applied. The V-Model XT does not describe "how products are created".

At this point, the CESAR RTP comes into play. The CESAR RTP does not focus on the tailoring of the project process model. Rather this is input for configuring the RTP. In fact, it provides the consequent next step for implementing a development process on the tool level; the CESAR RTP configuration results in an instance that represents a particular tool chain. Referring to the V-Model, the result from the tailoring are decisions for the tool chain. Table 1 lists examples for such decisions for the tool chain. This example refers to the project example, presented in the V-Model XT reference manual (cf. Part II, V-Model XT Doc). This example outlines a project that has the goal to develop a retrieval and access system for intellectual property assets.

The presented example project has the following settings as determined in the tailoring phase: It is a "System Engineering Project" with a single contractor. Thus, there are just two parties involved, the contractor and the customer. The set project profile has no positive decisions referring to the use of COTS products, requiring the capturing of the project performance or special accounting demands. Neither, special issues of date security are demanded for this project. As a result, the tailoring defines a set of project jobs as given on the left column of Table 1 in the section "Selected Elements".

The transition from the tailoring result in the V-Model XT result is as following: each of the selected elements or project profile results in implication for tool use or the establishment of a process. As a consequence, each of the selected elements implies the use of tools. Examples for possible tools are given in the right column of Table 1.

The next logical step is to narrow down the list of tool candidates in order to establish a concrete tool chain. As the list of tool candidates show, the outcome of the V-Model XT tailoring cannot define concrete tools, but a list of tool candidates. The consequence is that additional criteria are needed to narrow the list of tools down. So, as the introduction has pointed out, the development process is just one criterion which influences one instance of the CESAR RTP, there are several other factors, like the domain in which the development takes place, which influence the instance configuration. The section on the RTP background and the scope outlines the extent of possibilities here.

Table 1 Potential Tool Options for V-Model XT Example Project

Project Type	
System Engineering (Contractor)	Integrated Development Environment - Java-technology oriented projects: Eclipse - Microsoft-technology oriented project: VisualStudio
Project Type Variant	
Single-Contractor-Project	Influence: No collaboration platform required outside organisation
Project Profile	
System Security (AG): No	(accordingly no influence on tooling)
Accounting: No	(accordingly no influence on tooling)
Measurement/Analysis: No	(accordingly no influence on tooling)
Use of COTS Products: No	(accordingly no influence on tooling)
Selected Elements	
Project Management	Productivity Tools: - Collaboration Tools: Sharepoint, Wiki or trac - Document Management Tools: Trac, LIveLInk - Calendars and Groupware: Lotus Notes - Build Automation: CruiseControl, Anthill, Maven, Ant, VisualStudio
Quality Assurance	Test Tools, Automated Testing - Test Automation: Rational Robot, Selenium, HP QuickTest - Reporting: Clover
Change-/Issue Management	Issue Tracker - Issue Tracker: Bugzilla, Rational ClearQuest, JIRA, Microsoft Dynamics CRM, Team Foundation Server, Trac
Configuration Management	Concurrent Versioning System - Source Repositories: ClearCase, SVN, Team Foundation Server - Source Repository Replication: ClearCase
Requirements Engineering	Requirements Engineering Tool - Rational Doors, Requisite Pro
Delivery and Approval	Automated Product Packaging - Domain dependent solutions - IDE-integrated solutions: Microsoft Team Build
Contract Conclusion	(no influence on tooling)

5 The PLE Process Model: An Example

The CESAR Variability Ontology (CVO) is based on three main models. This allows distinguishing different concerns and thus facilitates the understanding. These three main models are:

- The *Product Line Engineering Model* describes the “big picture”, composed out of the activities, necessary for successful PLE, and the entities produced by these activities. The model relates the activities and provides insights in the product line specific engineering processes.
- The *Product Line Model* describes the elements, a product line consists of, and their relations. This model adds a view on the relations of the entities produced by the product line engineering activities, presented in the product line engineering model.
- The *Variability Model* focuses on the central product line aspect, namely the description and management of variability. It covers the entities and describes how they are linked up.

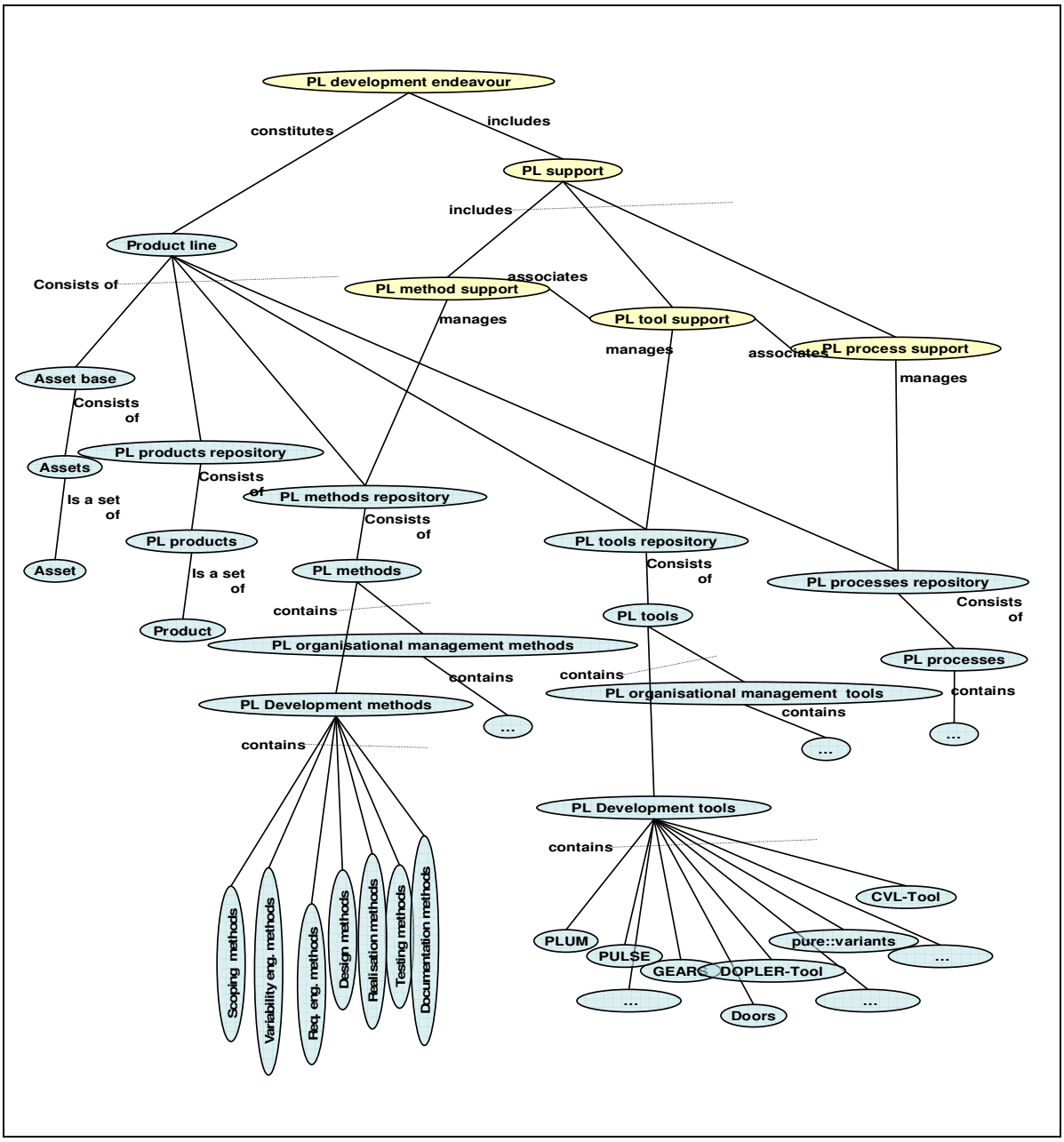

Figure 4 Cutting from the CESAR CVO description which forms a basis for a PLE process model.

Figure 4 is a cutting from the CESAR CVO and describes the product line engineering support and the main repository structure of the product line artefacts.

Product line support is structured as method support, process support and associated tool support. Methods, processes and tools are considered collected in product line repositories, respectively. This separation was chosen because usually, there is no one-to-one relation between methods or processes and tools. Usually, a product line method or process can be supported by different tools, and often by more than one. And also the other way round, a tool may be utilized by more than one method or process.

Product line methods are structured into product line development methods, into organisational and managerial methods for product line engineering, and possibly into others (probably, separating safety engineering methods for product lines from PL development methods is helpful since they, in a sense, are themselves concerned with all product line development steps in domain as well as in application engineering). Thus, the structure indicated here is "tentative" and has to show its convenience if it is used for the composition of a company-individual CESAR product line. In this respect, it will become essential to identify appropriate "method fragments" which allow for a convenient composition of the individual approaches.

The same eventually holds for the product line processes that should be partitioned into such fragments which can be conveniently supported by tools and are appropriate to be composed to an entire individual PL process for a CESAR partner. Again this composition of the process can then be used to adapt the configurable tool chain.

The product line tools are structured here according to the same criteria as the methods. It will become important, to identify the tools characteristics (both, from a usage point of view and with respect to their integration into the RTP tool chain) to allow a proper association of methods and processes with the tools listed in the ontology. The appropriate reasoning could then show which tool can be used in a given context defined by selected methods, followed processes and technology/organisational settings of a company.

Figure 4 describes also the top level product line data repositories, that is, the asset base, containing all product line development artefacts, and the product repository, collecting the products derived in the product line and their complete documentation, needed for maintenance and other purposes. As a reference for that, the CESAR project also develops a tool chain for the engineering of product lines as a part of the CESAR RTP, version 3 in conjunction with the PLUM tool [PLUM].

6 Conclusions

Today's development environments consist of a tool chain involving a set of tools for different purposes. The CESAR project conducts research about tool chain configuration and about ensuring interoperability between the involved tools and the exchanged development artefacts. As the CESAR project explains, the configuration of such tool chain is influenced by various aspects such as the organisations' development process or the set software standards or the required level of safety.

In this work, we provide a different view on the configuration of such tool chain: From our point of view, such tool chain can be seen as set of base assets that are configured for a special development effort. From this point of view, the set of base assets are used to create a tool chain that represents a product. A common set of concepts for software development and PLE is required in addition to provide the interoperability between the involved tools. The presented ontology provides a step towards this direction. However, the ontology has also outlined that the scope of such product line can be considered very large – and the scope of a product line represents in general a characteristic to keep small in order to ensure the maintainability of the product line. As such, the biggest challenge for future work will be the testability and maintainability of the 'large-scoped' RTP as a product line.

References

1. [ToolChain] Roland Mader, Gerhard Grießnig, Eric Armengaud, Christian Hein, Christian Steger, Reinhold Weiß: Model Based Tailoring of Integrated Tool Chains for Embedded System Development. Workshop MDTPI / MBSDI of ECMDA 2011, Berlin, June 2011.
2. [VModelXT] V-Modell® XT, Version 1.3, 2008 – http://www.cio.bund.de/DE/IT-Methoden/V-Modell_XT/v-modell_xt_node.html
3. [IEC] IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems. Part 1-7, 2002
4. [ISO] ISO/DIS 26262: Road Vehicles – Functional Safety, part 1-10, 2009
5. [JAZZ] IBM Rational JAZZ Technology Platform, http://www-01.ibm.com/software/rational/jazz/
6. [CESAR] Survey Report on Modeling Languages, Components Technologies and Validation Technologies for Real-Time Safety Critical Systems, Public CESAR Project Deliverable, D_SP3_R1.1_M1
7. [MoSiS] ITEA MoSiS Project: http://www.itea2.org/call1_mosis
8. [PLUM] PLUM – Product Line Unified Modeller, European Software Institute (ESI) Tecnalia, Spain, project website at http://www.esi.es/plum/ *(accessed in November 2011)*