ToolNet – Domain and Tool Connector

Manuel Reis Monteiro
EXTESSY AG
D-38442 Wolfsburg, Germany
m.monteiro@extessy.com

Patrick Maeder
Technical University of Ilmenau
Software Systems/Process Informatics Group
D-98693 Ilmenau, Germany
patrick.maeder@tu-ilmenau.de

ToolNet has been designed to provide a technical infrastructure for integrated system development. It was developed by DaimlerChrysler Research Berlin in cooperation with the EADS and commercialised by EXTESSY.

The vision of ToolNet is to integrate different development disciplines: functional development and mechanical development. This is necessary as these two strands are currently only loosely coupled. It should be emphasized that the ToolNet approach goes beyond the technical integration of tools towards a unified development philosophy that overarches the different disciplines.

In the context of this text mechanical development mainly deals with the dimensions and geometry of a component and the definition of its assembly structure, while functional development means the design of the basic structure as well the corresponding software. Both strands contribute to the product in the end, yet the development of the components is isolated from each other. Each side cannot access the information necessary to foresee integration problems and changes in the development process cannot be propagated to the other team. This lack of interconnectedness can result in difficulties and increased cost when the products of both strands have to be combined. With current state-of-the-art technologies this problem cannot be solved. There are two aspects why this is the case: Firstly, most of the tools use vendor-specific formats to store the data and even if imports and exports exist, the semantic context is lost in this transformation. Secondly, the tools have grown more and more complex and allow to model a limited range of the problem space very accurately, but require experts to handle them. The number of persons who overlook the whole project declines, so that few developers are aware of relations between the different stages of development. ToolNet is meant to deliver the technical infrastructure that bridges the gap between the two strands and reflects their natural interconnectedness. The strategy of the ToolNet developers is to create prototypes to prove the design at various stages of the project. This approach is prone to result in waste of design and code because some decisions might prove dead-ends and without proper documentation the efforts may be lost. To counter these effects, the abstract model of the ToolNet software should be expressive, precise and consistent with the code. The prototypes will be fully documented using the UML and a specialized UML profile, Refactoring is used to keep design and code consistent. This work refers to the development of the first prototype of ToolNet, therefore only a selected number of features will be found in the actual implementation. Discussions with developers from the system development sector as well as the prototype-centric approach lead to a number of non-functional requirements for the tool integration. The project plan envisages a series of prototypes that build upon each other and embrace an increasing amount of domains of the system development process. A domain identifies one step in the development process and is marked by a particular activity and a set of specialized tools (see Table 1). In order to build ToolNet incrementally the integration of two arbitrary domains have to be independent of each other in the code. The basic idea behind ToolNet is that tools are connected through one adapter to a common core and not through multi-lateral couplings.

Domain	**Representative**
Requirements Management	Doors
Modeling	MatLab, Artisan Studio, ...
Hardware Design	HDL Author
3D Modeling	Catia
Testing	CTE, Imbus
Product Data Management	Metaphase

Table 1: Excerpt of a tool environment in the automotive sector

This approach has a number of advantages: first, it alleviates the impact of version changes because only one connection has to be modified, secondly it makes deployment easier, because the configuration can focus on one connection instead of keeping track of many dependencies and third, the communication between tools is reduced to one standard method, which increases maintainability.

The discussions with the potential ToolNet users also revealed a number of top level functional requirements. Process integration was identified as the pre-eminent interest: the presentation should reflect the relationship of the domains and provide an adequate technical representation. It is therefore necessary to have means that allow to express the relations between the domains. Since the data of one tool is available to all other participants of ToolNet, it must be kept consistent over the whole project.