A SW-HW Integration Process for the Generation of Platform Specific Models

Shariful Islam, Technische Universität Darmstadt, Germany, ripon@informatik.tu-darmstadt.de
György Csertán, Budapest University of Technology and Economics, Hungary, csertan@mit.bme.hu
András Balogh, Budapest University of Technology and Economics, Hungary, abalogh@mit.bme.hu
Wolfgang Herzner, ARC Seibersdorf research, Austria, wolfgang.herzner@arcs.ac.at
Thierry Le Sergent, Esterel Technologies, France, Thierry.LESERGENT@esterel-technologies.com
András Pataricza, Budapest University of Technology and Economics, Hungary, pataric@mit.bme.hu
Neeraj Suri, Technische Universität Darmstadt, Germany, suri@informatik.tu-darmstadt.de

Abstract

Design of large dependable real-time embedded systems relies on integration of software elements onto a set of hardware nodes. In this paper an overall process of how to generate a Platform Specific Model (PSM) from a set of Platform Independent Models (PIM) and a hardware resource description is presented in the course of designing such embedded systems. Subsystems with different requirements are allocated and integrated onto shared hardware resources based on the constraints of the specified HW resources. The focus of this paper lies on developing the methodologies and tools for the generation of the PSM. Accordingly a dual-track approach has been proposed. First track is based on a transformational process and the second is a Multi Variable Optimization (MVO) approach.

1 Introduction

Mapping of software and hardware elements under platform resource constraints is a crucial step in the model based design of embedded real-time distributed systems. A technology invariant mapping of the desired varied system functionalities onto a targeted set of physical resources is usually addressed by the SW-HW integration process. The SW-HW integration phase in the development of embedded systems must have special methodologies and tools [1]. An integration framework is being developed in DECOS[1], where the PSM is generated from a set of PIMs and a platform description, adhering to the Model Driven Development (MDD) [2] methodology. In order to perform the technology-invariant mapping from PIMs to PSM, it is required that the system functionality is represented at a level allowing its mapping to an abstract description of the hardware resources while maintaining functional, performance and dependability requirements. Consequently, PIMs abstractly describe the system's functional and extra-functional properties, i.e., performance and dependability, without knowing any platform details. The PSM is at a level of detail where relevant application requirements are matched to specific platform concepts and technologies.

In MDD, PSM is the level where system design meets implementation concepts and languages. According to that, the PSM is a view of a system from the platform specific viewpoint and is expressed in terms of the specification model of the target platform. Techniques are required to develop a PSM. The task to develop a PSM can be very complex. To ease the designer's task, a PSM should come with guidelines, patterns and rules [2]. In this context, the generation of a PSM is performed by the guidance of the SW-HW integration. Moving from a traditional federated design paradigm to an integrated approach, we present two methods to perform the integration: Transformation approach and MVO approach. Both approaches meticulously guide the PIM-to-PSM mapping. The mapping problem is known to be NP-hard [3], which in general requires development of heuristic techniques yielding near-optimal solutions. The mapping is defined as follows: a) assigning different SW elements to suitable HW nodes such that platform resource constraints and dependability requirements are met (resource allocation) and b) ordering software executions in time (scheduling).

[1] EC Integrated Project IST DECOS www.decos.at

In this paper we focus on developing methodologies and tools in order to generate the PSM. We consider mapping of subsystems with different criticality using a transformation and mapping approach. Using a transformation approach, adhering to model-based design principles, our aim is to convert initial PIMs into a PSM. In [4] applying model driven technology a tool suite is developed that supports the development, configuration, deployment and validation of a component based distributed real-time and embedded systems. AIRES (Automatic Integration of Reusable Embedded Systems) [5] is an MDD analysis tool that evaluates whether certain timing, memory, power, and cost constraints of real-time and embedded applications are satisfied. However, in our approaches we address dependability and optimization aspects implicitly. During mapping we consider a set of constraints which have to be satisfied while dependability and real-time requirements are not compromised to render the result valid. A key part of the mapping is also to optimize the feasible mapping from dependability and real-time point of view. After this mapping process the PSM is obtained which controls the deployment of a system. Our final aim is to develop the executable PSM to run on a target platform.

The rest of the paper is organized as follows. Section 2 presents the relevant models to be considered for the developed process. Section 3 outlines the dual-track mapping process. Transformational and the MVO approaches are described in Section 4 and 5 respectively. Section 6 gives an overview of the tool suite used for the generation of the PSM.

2 Relevant Models

In this section, we present the models for the subsystems representation, the hardware platform model and give an overview of the MDD in DECOS followed by the brief description of SW development.

2.1 Subsystems Representation

The model for the representation of subsystems consists of a set of PIMs where a PIM describes a single Distributed Application Subsystem (DAS). PIMs are represented using a meta-model, the so called PIM metamodel. A DAS is defined as a self contained subsystem, which performs specified services, e.g., x-by-wire systems. Two different kinds of DASs are considered separately during the system design [6], safety critical (SC) and non-safety critical (non-SC). In case of SC DASs, which are highly dependable, an acceptably safe minimum performance level must be provided at all times.

DASs are further decomposed into smaller units called jobs. A job represents the smallest executable software fragment and the basic unit of work that employs the communication system for exchanging information with other jobs.

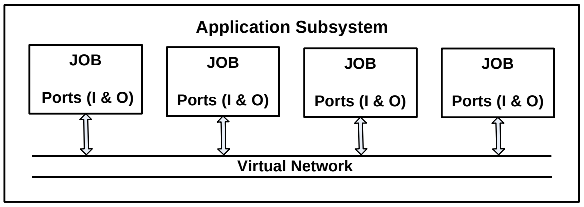


Figure 1: A single DAS

As seen in Figure 1, jobs communicate via input and output ports through a virtual network. A virtual network is an overlay network on top of the time-triggered network [7], which possesses a dedicated slot in the underlying Time Division Multiple Access (TDMA) scheme. PIMs can be generated by either using UML tools or using a Domain Specific Editor (DSE) inside VIATRA [8] (VIsual Automated model TRAnsformations).

2.2 Platform Model

We assume an inter-node network topology allowing every node to communicate with other node. A HW node is a self-contained computational element (single- or multiprocessor) connected with a communication controller. A node can also contain additional resources, such as sensors and actuators. In [9], the platform model is called Cluster Resource Description (CRD). It is represented using a meta-model called Hardware Specification Model (HSM) for capturing the resources of the hardware platforms, e.g., computational resources, communication resources, special purpose HW like sensors or actuators etc. A tool set has been developed based on this meta-model which speeds up the modeling process.

2.3 SW Development

Parallel to the PSM generation process described here, in DECOS, the PIM is used to guide application code generation using SCADE [10] as described in [11]. In addition to the good features from SCADE for safety critical embedded applications (formally defined, deterministic, etc.), two properties are interesting to get the information required for PSM: *(i)* SCADE code execution time is bounded; indeed, it is not possible to specify with SCADE language unbounded loops. *(ii)* In order to meet DO178-B constraints, generated code from SCADE qualified Code Generator involves only a small subset of the ANSI-C language, with no fancy pointer arithmetic or such. This makes the Worse Case Execution Analysis reasonably applicable. Measures derivable from that like code size or execution times should be considered during PSM generation, at least by manual feedback.

2.4 DECOS- Model Driven Design

The model-driven design starts with the application subsystem, which is modeled without considering the platform details. This abstract model is then transformed to PSM. In Figure 2, the PIM is the abstract model and the PSM is the transformed model. The constraints shown in Figure 2 have to be satisfied while mapping is done from PIM-to-PSM. The PSM development is constrained by the dependability requirements, constraints with respect to hardware resources and constraints with respect to network resources (e.g., bandwidth). Dependability is the prime driver of our mapping process as fault tolerance has to be provided through mapping specifically while performing the allocation.

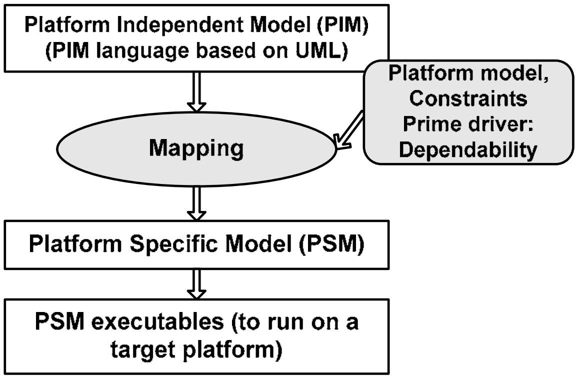

Platform Independent Model (PIM)
(PIM language based on UML)
Platform model,
Constraints
Prime driver:
Dependability
Mapping
Platform Specific Model (PSM)
PSM executables (to run on a
target platform)

Figure 2: DECOS-Model Based Design

Replication of critical jobs makes the system more dependable by tolerating fault. Fault tolerance can be enhanced by allocating replicas of jobs onto different nodes and either having recovery replicas to take over when a failure is detected, or use voting to mask the failure of a job. Therefore, the allocation of jobs onto HW nodes becomes a critical step in the mapping.

There are various ways we can perform the model transformation. According to [2], the model transformation can either be done manually, with computer assistance or automatically. In our case, the model transformation will allow designer to perform an interactive transformation of PIM-to-PSM. In the PIM-to-PSM mapping process, we use VIATRA as a model store and as a transformation tool that performs translations between models defined by a corresponding meta-model.

3 The SW-HW Integration Process

Based on the models and technology presented in the previous sections, we now present the overall mapping process. The goal of the mapping process is to assign jobs onto available HW resources. In order to facilitate the design process, we propose a dual-track approach for generating the PSM.

- Transformational approach and
- MVO approach

3.1 Outlining the Approaches

In the first phase a transformation based mapping process is developed and then extended to an MVO approach. The process of transformation handles constraints one-by-one. It finds a local solution meaning that it finds a feasible solution for resource allocation while satisfying constraints. We have developed a heuristics based systematic resource allocation approach for mapping which has been presented in [12]. In the next sections we devise the framework for both transformation and MVO approaches followed by the description of the mapping prerequisites.

3.2 Prerequisites of the Mapping Process

The SW-HW integration process assumes that a specification of the PIM and the CRD is available. The PIM is the prime input to the mapping process. Additional properties that have to be satisfied in the mapping are modeled as constraints. All constraints imposed on the jobs of a DAS are needed before mapping can take place. This includes details such as memory requirements and scheduling information. Measurements on existing code or estimations can be used to obtain this information. For SC jobs, the designer has to specify the required degree of replication in order to ensure fault-tolerance. Other types of constraints, such as the computational capability and memory capacity of the computing nodes as well as network bandwidth have to be extracted from the CRD.

Before the mapping process can start a cross model consistency check of the input models (i.e. PIMs, CRD and constraints) should be performed. This means that checking the feasibility of transforming the input models into a PSM. The feasibility test checks whether the chosen assignment is possible, i.e., enough resources exist. For example, if fault-tolerance requires that a job is replicated three times, this needs a hardware platform with at least three nodes. Likewise, it can be checked whether required sensors or actuators are available in the needed number. During feasibility check, the properties of the PIM and the constraints are matched with the feature list of the CRD. If feasibility checks are passed, the mapping process can follow. Passing this feasibility check does not guarantee the existence of a valid mapping since it only means there are no apparent inconsistencies preventing a valid mapping. However, a prior feasibility check is not mandatory. In this case the check is performed implicitly during the allocation process itself.

4 The Transformation Process

As already mentioned this process is model based which convert one model to another model of the same system. The proposed transformation process is shown in Figure 3 compatible with MDD methodology.

After considering the input models, we start by replicating the PIM sensor and actuator (S/A) elements according to the CRD. High critical jobs are also replicated. Then we perform the feasibility tests across the models outlined above. Since the input models are assumed to be self consistent we do not perform the individual model consistency checks in the PSM process. As the main target of the transformation process is to assign jobs onto available HW resources, the problem is articulated as a constrained resource allocation problem. Generally in resource allocation problems, a set of resources has to be assigned to a set of activities, satisfying specific requirements and constraints. The Platform Interface Layer (PIL) is configured through the PIM-to-PSM mapping process. The PIL provides the higher-level services [6] of the DECOS platform. As indicated in Figure 3, a number of further key steps are included in the transformation process, which will shortly be outlined below.

4.1 Marked PIM

The design starts with a PIM, in one step forward, marked PIM can be used which bears the information from the PIM meta model with some platform specific information.

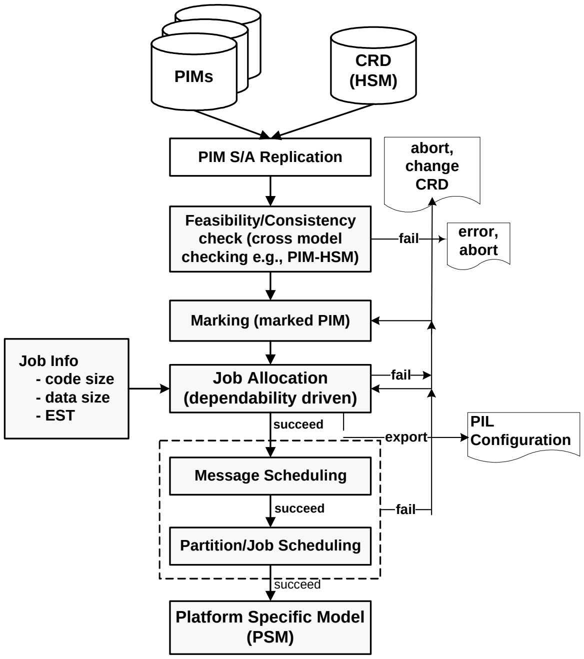


Figure 3: The Transformation Process

The PIM does not contain any references to platforms or technologies. After finalizing a PIM, the elements are marked in 'marked PIM' to give control information to the PIM-to-PSM mapping process. For example, the message type definition (embedded to CAN or LIN[2]) is provided at the PIM marking level.

4.2 Allocation

The assignment of jobs onto suitable HW nodes is done in this step. The left side box in Figure 3 with the job information allows the user to provide (estimating/assuming) the information of job code size, data size and about earliest start time (EST) which is used during allocation. Considering dependability and real-time as prime drivers, in [12] we presented a schedulable allocation algorithm for the consolidated mapping of SC and non-SC applications onto a distributed platform. The objective of our allocation technique was to come up with a feasible solution ensuring criticality partitioning, avoiding error propagation and reducing interactions across DASs.

[2] Controller Area and Local Interconnect Network

4.3 Scheduling

The scheduling is performed for the given allocation by using the scheduling tool suite by TTTech[3] (TTP-plan for message scheduling and TTP-build for job scheduling). For that purpose, the interim PSM is exported as file in a dedicated Python format and forwarded to TTP-plan and TTP-build. These tools compute feasible schedules and generate corresponding configuration files for the target platform.

4.4 Transformation in VIATRA

The VIATRA2 (release version 2) [8] model transformation framework addresses the challenges of model transformations for MDA and provides a complete system to define, develop, test, execute, debug, and maintain model transformations and code generators. For example, the input models PIM and CRD are transferred into the VIATRA model space from where they are taken as inputs into other transformations for example in job and S/A replication's transformation. This way the VIATRA model space is enriched by each transformation and at the end the PSM is generated for deployment.

5 The MVO Framework

As stated, the second dimension for mapping is the MVO approach that covers more theoretical aspects of SW-HW integration. Several resource allocation approaches deal constraints one-by-one presuming independent resource constraints. Since meeting a constraint may potentially cause violation of other constraints, it entails the fact that it is necessary to consider all constraints together. Although it is easier to consider incremental or progressive SW-HW integration, an overall solution is what is preferable. Ideally if a specific solution exists across PIM and CRD, the transformation process suffices to the mapping process. However, we strive to find a local solution in the previous process where as in generalized MVO approach the focus is on finding a global solution with respect to given objectives. The overall solution is restricted by a set of constraints that must be satisfied and multiple objectives (e.g., minimize interactions, maximize resource and bandwidth utilization) that evaluate the optimality of a solution, which lead to an MVO problem. We propose the MVO framework shown in Figure 4 followed by the details of mapping shown in Figure 5. The framework meticulously guides the mapping process and evaluates the goodness (finding the near-optimal solution) of the possible mappings. The feasible mapping can be generated by using heuristics process developed in [12].

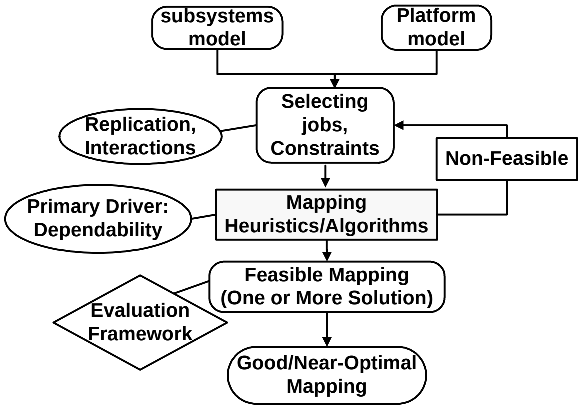

subsystems model
Platform model
Selecting jobs, Constraints
Replication, Interactions
Non-Feasible
Primary Driver: Dependability
Mapping Heuristics/Algorithms
Feasible Mapping (One or More Solution)
Evaluation Framework
Good/Near-Optimal Mapping

Figure 4: The MVO framework

The top part of Figure 5 shows the SC PIM and non-SC PIM. SC PIM contains the jobs (A1, A2, B1, B2, …) of SC DASs and non-SC PIM contains the jobs (P1,P2,Q1,Q2,…) of non-SC DASs. They are usually constrained by limited HW resources while mapping is done. The middle box of Figure 5 represents availability of the HW resources and services and the constraints, which are met during mapping. The callout box (PIL, PIM, platform) show the sources of resources, services (Middleware-MW) and constraints. The bottom part of the Figure 5 shows mapped jobs onto the HW nodes after meeting the physical and communication resources.

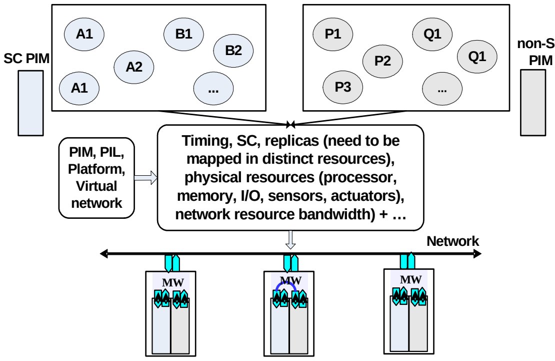

SC PIM
A1
B1
B2
A2
A1
...
P1
Q1
Q1
P2
P3
...
non-SC PIM
PIM, PIL, Platform, Virtual network
Timing, SC, replicas (need to be mapped in distinct resources), physical resources (processor, memory, I/O, sensors, actuators), network resource bandwidth) + …
Network
MW
MW
MW

Figure 5: MVO mapping

In order to develop this approach, the need is to define what constitutes a good mapping. Metrics representing a single vector with multiple variables are defined in this regard. The metrics depend strongly on the system objectives and the application model under consideration. A proper

[3] http://www.tttech.com/products/software.htm

selection of variables in metrics aids the mapping process and ascertaining trade-offs. The importance of various criteria may differ depending on the application under consideration and the number of variables also varied with. The variables are quantified by using specific variable evaluators. The variables are then formulated into an MVO function. The function is evaluated during the optimization of the mapping process.

The resulting optimized mapping is the content of the PSM, which is the optimized placement of jobs onto nodes of the hardware platform.

5.1 Constituting Integration Metrics

As mentioned in the previous section it is necessary to constitute an integration metrics covering various criteria. A first criterion is to satisfy hard constraints to make sure that the mapping is feasible. If an optimization process starts with a feasible mapping then the hard constraints can be excluded from the metrics. However, it also depends on the construction and evaluation of the optimization function. Careful attention must be taken so that these constraints are not violated during the evaluation of the optimization function.

Following criteria are identified to constitute integration metrics. Firstly, satisfaction of constraints implies the absolute constraints on behavior, whether semantic, temporal, or others. We refer this as satisfaction of hard constraints that need to be satisfied otherwise the solution is not valid. Secondly, dependability: It is clear to the system designer that the systems which are responsible for critical services must be designed to be dependable. Traditionally system designer design the system just to provide dependability, however, the system should not only provide the dependability for critical applications, it should also optimize the system for better fault-tolerance, e.g., minimize interactions in order to reduce the error propagation probabilities. Thirdly, real-time: A hard real-time system must respect all timing constraints and also should optimize the desired system design from real-time view, e.g., minimizing the scheduling length.

5.2 Variables Selection and Evaluators

There can be one or more suitable mappings satisfying above criteria. Thus, the problem entails assessing the possible mapping(s) as well as quantify/estimate the amount of each variable of a metric is necessary to fulfil the purpose. So first of all there are two points need to be studied, selection of variables and evaluators for the variables. A consistent and complete set of variables is needed to be defined so that the chosen variables are adequate in indicating the degree to which overall objectives are met [13]. Though there is no concrete way or any step-by-step procedure to select them, some points should be followed, e.g., it is necessary to have clear picture of adding variables, the variable should be operational, avoiding redundancy of variable (i.e., defining two variables which are same from an operational view but different in name) etc.

To quantify the amount of each variable in each mapping, we need evaluators for them. Obtaining variable is a straightforward process, and depends on the system designer. For example, in case of communication, one possible variable evaluator may be the number of messages per second.

5.3 MVO Function

This section formulates the MVO function as a value function by using the variables. A value function is defined as a function, which associates a scalar-valued function v (q) to each point q in an evaluation space, representing the system designer's preferences, provided that choosing a feasible alternative from a set of contenders such that v is maximized or minimized. In order to break down the assessment of a complex multi variable value function into smaller problems that can be solved independently, additive value functions can be used, which is a powerful mechanism. And usually MVO optimization problems are solved by scalarizing the overall function, meaning that the problem is converted into one single or a family of single objective optimization problems. This subproblem has a real-valued objective function that possibly depends on some parameters and can be solved using single objective optimizers. The mathematical representation of the MVO function in additive form is shown below.

$$mvo(q_1, q_2, ..., q_n) = \sum_{i=1}^{n} \lambda_i v_i(q_i)$$

$$\sum_{i=1}^{n} \lambda_i = 1, and\ 0 < \lambda_i < 1$$

λ_i is the trade-off factor.

The metrics of the variables is represented as follows:

$$M[v] \equiv M[v_1, v_2, ..., v_n]$$

Using existing metaheuristics like simulated annealing, tabu search and mixed-integer programming, the formulation of the optimization problem is simplified [14]. Currently we are developing the algorithm based on the metaheuristics. As mentioned before, the resulting optimized mapping is the best assignment of jobs onto nodes of the hardware platform.

6 The PSM Metamodel

The PSM contains the result of the SW-HW integration. Consequently, the PSM metamodel consists of mainly the packages shown in Figure 6. The main steps of the PIM-to-PSM transformation are allocation and scheduling. In addition, the representation of configuration data is needed in order to enable the parameterization of the high-level services, i.e., encapsulated execution environment, fault-tolerance layer, virtual communication and gateways, and diagnosis.

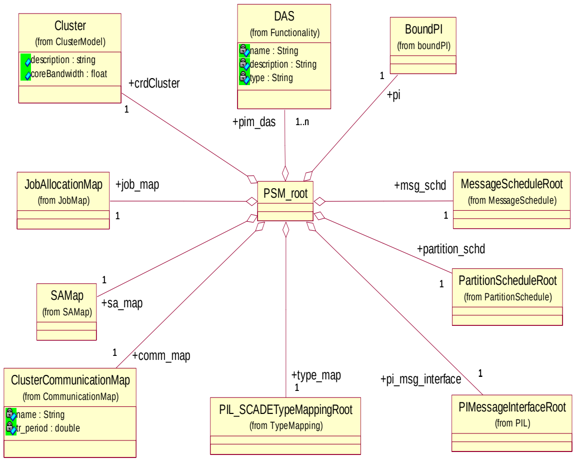


Figure 6: PSM overall view (UML notation)

PSM combines the specifications in the PIM with the details that specify how that system uses a particular type of platform and what additional services the DECOS architecture provides [6]. Allocation and scheduling bring these three models into one stage employing high-level services of the integrated architecture. It is the PSM overall view.

6.1 Content of the PSM

The PSM of the DASs is a model where the jobs have been assigned to nodes and the virtual networks to physical (time-triggered) networks. The PSM is a model repository that contains all necessary information to produce a runtime on a specific platform. In general, it may contain information in all or any combination of these types of file format like C files, Makefile, different XML (Extensible Markup Language) [15] files and even text files. A PSM environment should support the generation of all the files required by the targeted platform. In the following section we portray the PSM tool set.

6.2 PSM Tool Set

The PSM development tool is a VIATRA based framework shown in Figure 7 interfacing with TTP tools and SCADE. The toolset is used to perform the interactive PIM-to-PSM mapping which is described in Section 4. In the development of the PSM, a multitude of metamodels and inputs are needed. Consequently, a multitude of transformations are done in the course of the PSM generation by using VIATRA. The prototype is developed through appropriate techniques with supporting tool suite, e.g., the heuristics based PSM allocation algorithm can be formulated into transformation rules in VIATRA.

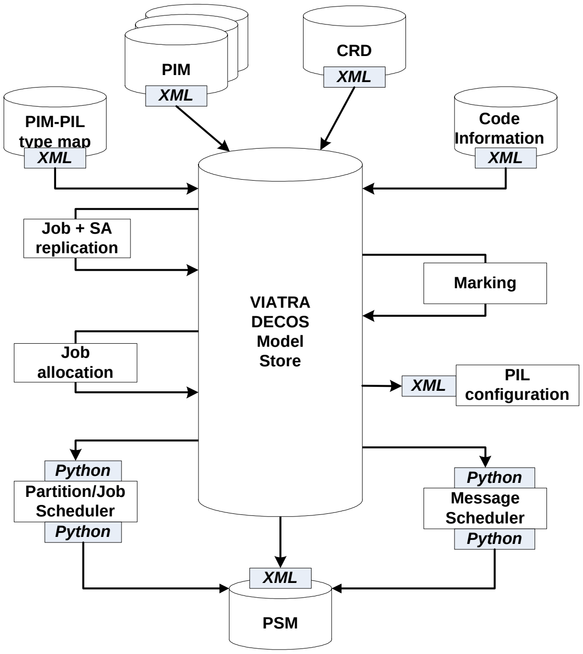


Figure 7: PSM tool suite in DECOS

The steps of different transformations have to be communicated and be usable by different users. It is therefore to be ensured that the information transported via the interfaces of the transformations is available in a form that can be processed by all involved parties. Taking these considerations into account, we have chosen to use XML documents for representing most of the information. Schemas (XSD) are used to define the structure of those XML documents.

The tool sets used for the generation of the PSM are as follows: PSM metamodel is created in UML in Rational Rose, PSM data format is XML, input models are imported into the VIATRA model store in XML format, message scheduling is done by external calls to TTP-plan, job scheduling is done by external calls to TTP-build. Python file format is used for both TTP-pan and TTP-build to interact with VIATRA. Models are exported from VIATRA model store in order to generate the XML file for PSM. All about the PIM-to-PSM mapping process can be displayed in a multi-page editor with pages to view the information about input and output models. The Eclipse Framework [16] is used in this regard.

6.3 MVO in Tool Set

We use an existing tool for the fast and reliable implementations of MVO. We intend to use CPLEX as an optimization tool. ILOG CPLEX [17] provides a variety of ways to interact with it during the development and deployment of their applications. CPLEX can be linked with a wide set of modeling languages like ILOG OPL development studio, which helps to develop optimization application quickly. CPLEX has linkable libraries for C development and it can be called in batch mode. In both cases either using CPLEX or a different MVO program (using CPLEX libraries), it can be called from VIATRA on the usual way of calling external programs. In this way, it is interfaced with PIM-to-PSM toolsets.

7 Conclusions

We have presented a two-way approach for deriving PIM-to-PSM mapping, based on model-driven technology. Besides a more conventional way, which considers one constraint at a time, a multi-variable optimization, which also allows for taking certain quality assessments, has been presented. The mapping process is supported by a generative tool suite mainly based on VIATRA framework and TTP-tools. It enhances the development, configuration and deployment of a system. Our future work includes the implementation of the mapping process onto a real HW platform by utilizing examples of automotive and avionics systems.

Acknowledgments

This work has been partly supported by the European IST project DECOS (EC IST FP6 IP DECOS) No. 511764.

References

[1] A. Berger, Embedded Systems Design: An Introduction to Processes, Tools and Techniques. CMP Books, USA, 2002.

[2] OMG. Model driven architecture, A technical perspec tive. Technical report, OMG Document No. ab/2001-02-04, Object Management Group.

[3] D. Fernandez-Baca, Allocating Modules to Processors in a Distributed System, IEEE Trans. on Soft. Eng., 15(11): pp. 1427–1436, 1989.

[4] K. Balasubramanian, A. Krishna, et al, Applying Model-Driven Development to Distributed Real-time and Embedded Avionics Systems, International Journal of Embedded Systems, 2005.

[5] S. Kodase, S. Wang, Z. Gu, and K. G. Shin, Improving Scalability of Task Allocation and Scheduling in Large Distributed Real-time Systems using Shared Buffers, In the proc. of the 9th IEEE, RTAS, May 2003.

[6] H. Kopetz, R. Obermaisser, P. Peti, N. Suri; From a Federated to an Integrated Architecture for Dependable Real-Time Embedded Systems; Technical report 22/2004, TU Vienna, July 2004.

[7] H. Kopetz, G. Grünsteidl, TTP-A Protocol for Fault-Tolerant Real-Time Systems. Computer, 27(1): pp. 14–23, 1994.

[8] A. Balogh, A. Németh, A. Schmidt, I. Rath, D. Vágó, D. Varró and A. Pataricza, The VIATRA2 Model Transfor mation Framework, In the proc. of the ECMDA, 2005.

[9] B. Huber, R. Obermaisser and P. Peti. MDA-Based De velopment in the DECOS Integrated Architecture- Modeling the Hardware Platform. In the proc. of the 9th IEEE ISORC, pp 43–52, 2006.

[10] SCADE Suite Technical and User Manuals, Version 5.0.1, June 2005, Esterel Technologies.

[11] W. Herzner, M. Schlager, Th. LeSergent, B. Huber, S. Islam, N. Suri, A. Balogh, The DECOS Tool-Chain – From Model-Based Design to Deployment of Integrated, Embedded, Real-Time Systems. In the proc. of ME'06.

[12] S. Islam, R. Lindström and N. Suri, Dependability Driven Integration of Mixed Criticality SW Components, In the proc. of the 9th IEEE ISORC, pp 485–495, 2006.

[13] R. L. Keeney and H. Raiffa, Decisions with Multiple Ob jectives: Preferences and Value Tradeoffs. Cambridge University Press, UK, 1993.

[14] J. Dréo, P. Siarry, A. Pétrowski and E. Taillard, Metaheuristics for Hard Optimization, Methods and Case Studies, Springer-Verlag Berlin Heidelberg, Germany, 2006.

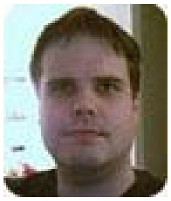
[15] OMG. XML Metadata Interchange (XMI), v2.0. http://www.omg.org/docs/formal/03-05-02.pdf.

[16] EclipseCommunity, Eclipse Modeling Framework, http://www.eclipse.org/emf, June 2005.

[17] ILOG, A ILOG optimization technology, http://www.ilog.com/products/cplex/, March 2006.

Shariful Islam received the B.Sc. degree in Electrical and Electronic Engineering from Khulna University of Engineering and Technology, Bangladesh in 2000 and the M.Sc. degree in Information Technology [Study Model: Embedded Systems Engineering] from the University of Stuttgart, Germany in 2004. He is currently a PhD candidate at the DEEDS group in the Department of Computer Science at TU Darmstadt, Germany. His research interests include design and assessment of integrated dependable real-time embedded systems, system design optimization, SW-HW integration - main issues are dependability, real-time and power. He received the 'Prime Minister Gold Medal' award in recognition of his outstanding academic performance in B.Sc. He is a student member of the IEEE. More information on his interests and activities provided at:
http://www.deeds.informatik.tu-darmstadt.de/ripon

György Csertán holds an MS in Electrical Engineering and an MS in Business Administration. He received his Ph.D. in Computer Science at the Budapest Technical University in 1998 in Information Technology. He has many years of experience in leading research and development projects in the field of information technology and its application. He was the initiator of the Central Transdanubian Research & Development Portal. His professional interest includes dependable systems, e-business systems, and dependability in embedded systems. He has a wife and three daughters. His personal interest is photography and hiking.



András Balogh holds an MS in Technical Informatics received from the Budapest University of Technology and Economics in 2004. Currently he is a Ph.D. student at the Fault Tolerant Systems Research Group at BUTE. His professional interest includes model-driven development of dependable systems and e-business systems, and formal analysis of system models. He has many years of experience in architectural system design, and in software and transformation development. He holds courses on Eclipse-based development, and UML modeling.His personal interest is railway-modeling.

Wolfgang Herzner received his doctor degree in Computer Science at the Vienna University of Technology in 1984. Initially working in the fields of computer graphics and multimedia, he was engaged in corresponding international standardisation within ISO/IEC JTC1 until 1996. After co-designing and developing large industrial systems, e.g. for digital video surveillance, he now works in the area of safe software engineering and related areas like model-driven software development or (design) patterns, with emphasis on dependable embedded real-time systems. With that focus, he is currently involved in projects like DECOS. Herzner is with the ARC Seibersdorf research since 1984. He is member of ACM, IEEE Computer Society, the Austrian Computer Society (OCG), and reviewer of IEEE Software Engineering

Thierry Le Sergent received his PhD on Computer Science from the "LAAS-CNRS" in 1993. He has been involved for 7 years in the development of software engineering methods and tools (ASA+, ObjectGEODE) at Verilog and CS-Verilog. Years 2000 and 2001, he managed the IST SafeAir project for Telelogic Technologies Toulouse. In 2002, he joined Esterel Technologies as R&D project manager leading the SCADE Gateways team. Activities of the team includes the design and development of SW tools that link SCADE environment to other formalisms, e.g. UML/SysML and Simulink/Stateflow. He is managing IST projects for Esterel Technologies: RISE (FP5), DECOS (FP6), SPEEDS (FP6).

András Pataricza is a graduate from the Technical University of Budapest (BUTE) and holds a Ph.D. degree from the Hungarian Academy of Sciences. Currently he is leader of BUTE's Research Group for Fault-Tolerant Systems. Professor Pataricza has received multiple recognition awards from scientific and industrial organizations like the Hungarian Academy of Sciences, IBM, HP and others. He has acted as technical leader and/or advisor to many international scientific projects and is very active in EU scientific research programs. He has published many technical papers, books, tutorials, magazine articles, conference papers and technical reports.

Neeraj Suri received his Ph.D. from the University of Massachusetts at Amherst. He currently holds the TU Darmstadt Chair Professorship in "Dependable Embedded Systems and Software" at TU Darmstadt, Germany. His earlier academic appointments include the Saab Endowed Professorship and faculty at Boston University. His research interests span design, analysis and assessment of dependable and secure systems/software. His group's research activities have garnered support from DARPA, NSF, ONR, EC, NASA, Boeing, Microsoft, Intel, Saab among others. He is also a recipient of the NSF CAREER award. Suri serves as an editor for IEEE Trans. on Dependable and Secure Computing, IEEE Trans. on Software Engineering, ACM Computing Surveys, IJSN, and has been an editor for IEEE Trans. on Parallel and Distributed Systems. He is a member of IFIP WG 10.4 on Dependability, and a member of Microsoft's Trustworthy Computing Academic Advisory Board. More details are available at:

http://www.deeds.informatik.tu-darmstadt.de/suri