PROCEEDINGS

55. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

13 - 17 September 2010

Crossing Borders within the ABC

Automation,

Biomedical Engineering and

Computer Science

Faculty of
Computer Science and Automation

www.tu-ilmenau.de

TECHNISCHE UNIVERSITÄT
ILMENAU

Impressum
Published by

Publisher:	Rector of the Ilmenau University of Technology Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c. Peter Scharff
Editor:	Marketing Department (Phone: +49 3677 69-2520) Andrea Schneider (conferences@tu-ilmenau.de) Faculty of Computer Science and Automation (Phone: +49 3677 69-2860) Univ.-Prof. Dr.-Ing. habil. Jens Haueisen
Editorial Deadline:	20. August 2010
Implementation:	Ilmenau University of Technology Felix Böckelmann Philipp Schmidt

USB-Flash-Version.

Publishing House:	Verlag ISLE, Betriebsstätte des ISLE e.V. Werner-von-Siemens-Str. 16 98693 Ilmenau
Production:	CDA Datenträger Albrechts GmbH, 98529 Suhl/Albrechts
Order trough:	Marketing Department (+49 3677 69-2520) Andrea Schneider (conferences@tu-ilmenau.de)

ISBN: 978-3-938843-53-6 (USB-Flash Version)

Online-Version:

Publisher:	Universitätsbibliothek Ilmenau ilmedia Postfach 10 05 65 98684 Ilmenau

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=16739

OPTIMAL MAPPING OF FUNCTIONS TO ARCHITECTURES USING MODEL BASED DESIGN

Frank Lohse

Department of System and Software Engineering
Ilmenau University of Technology
Ilmenau, Germany

E-Mail: Frank.Lohse@TU-Ilmenau.de

ABSTRACT

Today networked systems are very complex. Growing system and performance requirements, limited hardware resources and additional constraints complicate their design. Formal and informal quantities and specifications depend the overall system behavior too. Model based engineering enables system analysis and performance estimation in early design steps. Which system function should be processed by which architecture as a special problem is focused in this paper. Therefore a specific component library forms the basis of virtual prototyping. The intrasystems variable level of detail allow a simulation on different abstraction levels. The resulting performance values as input for naturally-motivated algorithms are used for optimization of bin packing problem (mapping) and shortest path problem (communication). Two prototypically examples clarify the forced methodology.

Index Terms— Model based design, Virtual prototype, Design space exploration, Naturally-based optimization

1. INTRODUCTION

Involving the sectors of automotive, avionik, controlling, networking or industrial plants, the complexity of embedded systems grows rapidly within the last years. Especially their design is often very complicated. Many critical system errors occure by incomplete or inaccurate specifications. Other additional performance requirements and quantities like resource load, communication delay, degree of networking, type of architecture, topology, device count, memory usage, execution time or energy consumption influence the overall system behavior too. Another main problem is the combination or integration of subsystems. One subsystem for itself works perfectly and is optimized, but integrated into the overall system issues of application flow or bottlenecks may occure. Taking account of certain functions, another important criterion is the optimal use of system components and ressources. The derived question, which function should be processed by which system component, is focused in this paper. Two short examples clarify this problem.

- **Automotive:** Normally a new ECU realizes a new functionality. Keeping in mind that premium cars contain over 70 ECU's integration problems may occure, resulting in system errors or higher test count. A better option relocates this functionality onto another existing ECU. Standardized automotive software architecture, jointly developed by automobile manufacturers, suppliers and tool developers, offers this opportunity like descriped in AUTOSAR (AUTomotive Open System ARchitecture).

- **Wireless sensor networks:** Network coverage, energy consumption or link quality are important performance values of wireless sensor networks. Especially message transfer is expensive. An individual mapping of functions to network devices for an optimal use of their limited hardware and software ressources is necessary. A valid and optimized combination of functions for computation, meassurement, calculation or evaluation may reduce communication costs for each device, resulting in higher network lifetime.

Keeping the named question above in mind, the model based engineering requires several system variants, high simulation speed and performance analysis, requirements and constraints. The spanning design space has to be searched for an optimum. Both NP-complete problems, mapping (similar to bin packing) and best communication (similar to shortest path), should be solved via fast naturally-motivated algorithms. An individual refinement of model components may increase

Thanks to Thomas Luetzelberger for funding.

simulation and optimization speed too.
After this short introduction some theoretical bases for model based design and optimization control are explained. Chapter 3 shows the system design flow and underlaying concept. Two prototypical examples are descriped in chapter 4, one for best mapping and communication, one for calculating simulation time of different abstraction levels using executable specification of CAN[2]. The paper closes with a summary, a conclusion and future steps.

2. THEORETICAL BASIS

In contrast to requirement engineering model based engineering supports early design steps. To solve the topics question it is necessary to transform the system specification and requirements into an executable model on abstract level of detail. The resulting *virtual prototype* enables system analysis, simulation and performance estimation for consecutive optimizations. Results feed into the development process to optimize the final system behavior. Goal is to understand, master and control complex networked systems as a whole (behavior, performance parameters, build up, configuration, etc). Finding system errors in early design steps is another goal [1]. Selection of valid and optimized design variants enhances the final system behavior. But often the specification is inaccurate or not complete. It is also unclear, which components the final system should have, how they work together and which performance is expected. The enormous diversity of components, communication protocols, function implementations, etc. spans a huge *design space*, which has to be searched for an optimum depending on system requirements [2]. Usually count of optimization parameters is very high. A *fitness function* combines all important parameters into one unique value, later defining the best solution. *Pareto optimum* defines a paramater balance, where one parameter cannot increased if another is descreased simultaneously. The huge dimension of the design space may not allow the use of numeric algorithm. The decision for naturally-motivated algorithms bases on a simple example of three architectures and eight functions resulting in 6561 different variants. Numeric algorithms may too slow and not suitable [3]. Candidates for naturally-motivated algorithms are tabu search, simulated annealing, ant colony optimization, genetic algorithm, treshhold accepting, great deluge algorithm, neuronal nets, metropolis algorithm, hill climbing or stochastic tunneling [4].

3. DESIGN CONCEPT

The top down design flow covers all layers from mission level down to code level. For the possibility of variable switching it is necessary to use standardised model components. Depending on the rating criterion some components or subsystems are less important than the others. This enables a scalability as shown in figure 1. The advantage is a better overview of important performance parameters and higher simulation speed.

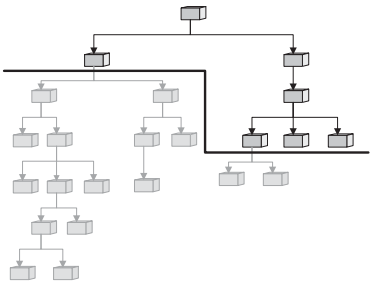

Fig. 1. Individual refinement levels of a system model

The optimization loop is shown in figure 2. A model library for architecture and function components forms the bases. Their combination results in an executable system model including a valid configuration. All possible performance values are generated by simulation. A specific fitness function combines them into one unique value, that is needed by the optimization. The simulation and optimization control stops, if the fitness function value reaches the optimization goal. Futher work depending code generation and AUTOSAR process is represented in [5].

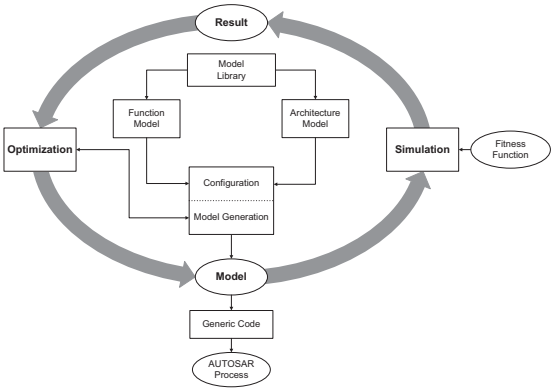

Fig. 2. Optimization Loop

For system design MLDesigner (MLDesign Technologies, Inc.[1]) was used. It supports modelling on different abstraction levels. Several modelling domains can be combined and simulated. The model generation bases on hirarchical block diagrams like in UML and covers all abtraction levels down to code- and finite state machine-layer, which enables scalability [6].
Building system variants bases on a model component

[2] www.semiconductors.bosch.de/pdf/can2spec.pdf

[1] www.mldesigner.com

library. It contains all components for architecture, function and communication. Depending on the developed systems scenario the components represent automotive (CAN [7]), avionics (fibre optics), controlling (ZigBee [8]), networking (TCP/IP [9]) etc. Verificated and validated standard components of MLDesigner are used to model other simple functionalities.

4. IMPLEMENTATION EXAMPLES

The presented methodology has several subproblems. Two of them are explained in this chapter. On the one hand a prototypically example for optimal mapping, on the other hand a simulation example for speedup by individual refinements.

4.1. Optimization Example

The optimizations point of starting is a fictitious system with three linked architectures (see Fig. 3). Each architecture contains several functionality. Communication ports allow transmissions between functions and architectures. Component level parameters for function size, execution time and communication delay reflect the high abstraction level. System level parameters like iteration count, length of tabu list, greedy variant or fitness function objectives are needed by the optimization. The initial solution is represented by matching colors of functions and architectures. The system model cannot be executed yet, because of missing data generation and collecting. Input for optimization are matrices containing initial mapping, linked functions, architecture speed and communication setting.

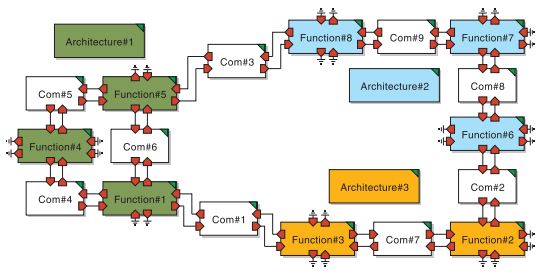

Fig. 3. Example of a fictitious System

To find an optimal mapping of functions to architectures, first the fictititious system is saved as .mml-file. An external parsing routine extracts all entire parameter information into new variable strings used by followed optimization via naturally-motivated algorithms tabu search and simulated annealing. An optional greedy algorithm is for comparison purposes. The resulting optimal system configuration is written back into new specific .mml-files for each algorithm.
A total of four fitness functions is implemented, two are shown below. Fitness function J_2 matches best to the given system requirements because of minimizing the divergence, while J_1 forces an optimum near the origin.

$$J_1 = c_p \cdot \frac{c_c}{v_{c_c}} + t_p \cdot \frac{t_{exe}}{v_{t_{exe}}}$$

$$J_2 = c_p \cdot \frac{c_c - v_{c_c}}{v_{c_c}} + t_p \cdot \frac{t_{exe} - v_{t_{exe}}}{v_{t_{exe}}}$$

(c_p - communication cost priority, t_p - execution time priority, c_c - communication costs, t_{exe} - execution time, v_{c_c} - objective communication costs, $v_{t_{exe}}$ - objective execution time)

For the optimization by tabu search all functions run once without any scheduling. The initial system configuration has a setting of (9,9), representing execution time for all functions of 9 and communication costs of 9. After optimization using fitness funtion J_2 with $v_{c_c} = 6$ and $v_{t_{exe}} = 12$ the optimum is (6,12) and fits to the given sytem requirement. Fitness function J_1 evaluates an optimum at (5,12), which is closer to the origin. A selection of evaluated system configurations during 100 iteration steps is shown in figure 4. Simulated annealing estimates same results. All results of the greedy algorithm variants deviate from the optimum, in fact 2.0 - 11.6.

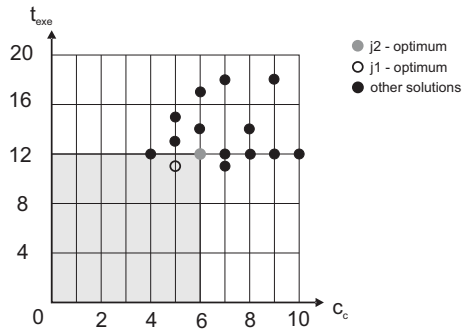

Fig. 4. Design Space of possible Solutions

4.2. Scaling Example

Simulation speed and degree of model detail influence each other. The more detailed a model is, the more its simulation speed decreases. If the forced performance value detail level is low, bstract subparts are more useful for simulation acceleration.

The second example describes the scaling. Figure 5 shows a simple peer to peer network of two CAN devices. The left one is implemented on bit level using FSM's (finite state machines), the right one contains an

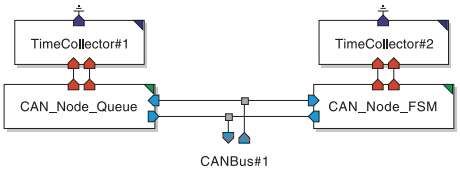

Fig. 5. System Mixture

high level queueing model. Both devices are wire connected and have the same functionality of data transmission and date receive. The variable system components allow two other system configurations; both high level and both bit level. The simulation performes a CAN bus with 1 MBit/sec and 99% bus load.
Three simulations of 10000 system seconds and their results are shown in table 1.

	High Level	Bit Level	Mixture
Simulation Time in sec	8,22	25,54	17,81

Table 1. Simulation Results

5. CONCLUSION

The first part of the paper shows an abstract model based description and design approach of complex networked systems. A work flow for their optimization under specified quality criteria using tabu search, simulated annealing and greedy algorithm variants is explained by a theoretical example, resulting in an optimized and reconfigured system. Depending on the problem size, arithmetic algorithms seem too slow. Naturally-motivated optimization heuristics are more suitable.
The second part clarifies the manipulation of simulation speed by an example of executable CAN-specification on two different abtraction levels. Both simulation results have the same quality, but different times of simulation. An adapted combination could accelerate further optimization cycles.
Parsing functionality, optimization algorithms, scaling and simulation control have to be integrated into MLDesigner to get a full executable overall system.
The overall manual work flow has disadvantages towards full automatic solutions. It is necessary to automate that.
Tests on more extensive and realistic systems are needed, followed by validations against real systems to check optimization and system behavior correctness.
Further work could address the implementation of different optimization algorithms and more optimization parameters like memory usage, maintenance costs or purchase costs. Dynamic models for operating system, CPU, scheduler, routing, order planning or deadlines replace the static performance parameters, resulting in more detailed performance values.

6. REFERENCES

[1] Tommy Baumann, "Simulation driven design of distributed systems," in *SAE 2011 World Congress, Detroit, Michigan, 2011 (accepted for publication)*, 2010.

[2] Fernando De Bernardinis and Alberto Sangiovanni-Vincentelli, "A methodology for system-level analog design space exploration," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition Volume I (DATE), S. 10677*, 2004.

[3] Bernd Hardung, *Optimisation of the Allocation of Functions in Vehicle Networks*, Ph.D. thesis, Technischen Fakultaet der Universitaet Erlangen-Nuernberg, 2006.

[4] Martin Feldmann, *Naturanaloge Verfahren - Metaheuristiken zur Reihenfolgeplanung*, Number 3-8244-6890-5. Gabler Edition Wissenschaft, 2000.

[5] Christoph Bodenstein, Frank Lohse, and Armin Zimmermann, "Executable specifications for model-based development of automotive software," in *2010 IEEE International Conference on Systems, Man, and Cybernetics (accepted for publication)*, 2010.

[6] Thomas Liebezeit, Volker Zerbe, and Tino Loeffler, "A framework for mission level design," in *12th IASTED International Conference on Applied Simulation and Modeling (ASM), 03.-05. September, Marbella, Spanien, ACTA Press, Proceedings, S. 616-621, ISBN 0-88986-384-9*, 2003.

[7] Marcus Mueller, Johannes Kloeckner, and Wolfgang Fengler, "Bit accurate timing analysis on a frame based can model," in *Proceedings of 4th IFAC Workshop on Discrete-Event System Design*, 2009.

[8] Frank Lohse, Volker Zerbe, and Thomas Luetzelberger, "Architecture analysis and optimization of reconfigurable, complex systems," in *Proceedings of 14th IEEE International Conference on Intelligent Engineering Systems (INES)*, 2010.

[9] Matthias Eck, "Entwurf und validierung des tcp/ip - ethernet protokolls auf performance - ebene," M.S. thesis, Technische Universitaet Ilmenau, 2007.