Presented by
Gert Döhmen, Andreas Lehmann, Maurice Girod

SPEEDS

Component-Based Design using a Complete Virtual System Model

Content

- What SPEEDS is
- Motivation of the project
- SPEEDS Concepts
- Deliverables and status
- Demonstrator implementations
- Next actions within Airbus

What SPEEDS is

SPEEDS stands for Speculative and Exploratory Design in Systems Engineering.

- “Speculative”: design solutions’ characteristics not known with certainty
- “Exploratory”: investigating the design space from all relevant viewpoints

SPEEDS is a European Project, involving various industrial and academic partners.

- Funding: mainly by the European Community, 6th Framework Project in Embedded Systems Development
- Budget: 16M€ (Airbus: 2M€)
- Duration: from May 2006 to April 2010
- Lead: Gert Döhmen, Airbus Operations GmbH

SPEEDS is mainly about:

- Modular (component-based) system design with multi-domains in a distributed partner environment
- Integration of domain/tool-specific design models into one tool-independent “Complete Virtual System Model”
- System analysis and simulation across domains, based on a “Complete Virtual System Model”

SPEEDS principal benefits:

- Enabling a system-integrator to manage the deliverables of multiple system developers
- Facilitating the move from local-optimisation to macro-optimisation, cross-domain
- Allowing for early design validation and reducing the need for expensive system tests on physical test rigs

Motivation of the project

Current practice in Systems Engineering:

Text-based and spreadsheet-based development is the dominating way of working.

- Advantages:
  - Commonly available, widely known and inexpensive office tools can be used
  - Specific knowledge on formal system-modelling methods is not required
- Disadvantages, especially with regard to designing complex systems:
  - Difficult to ensure model consistency and clarity → difficult to understand other peoples' models
  - Difficult to support system analyses and simulations
  - Difficult to perform system-wide (macro, in contrast to local) optimisations

Motivation of the project

Current trend in Systems Engineering:

There is a shift visible towards formal model-based development. These models consist of formal objects that are defined by a modelling language, such as SysML.

•Advantages, particularly if a "standard" modelling language is used:

- The models are clear and understandable; thus distributed teams can efficiently share these models
- System consistency and formal correctness can be automatically analysed
- System-behaviour simulations and even automatic code generation for control units are possible

•Shortcomings of the current model-based approaches, especially for multi-domain systems:

- Across domains incompatible modelling languages and tools are used. Thus, the different domains' views on the overall system cannot be integrated.
- The current models do not fully support a modular, component-based way of working. That is, the characteristics, such as performance and constraints, of individual components are not formally described within the models.

➢ Consequences:
  ➢ Automatically analysing and validating system models, across domains as well as running simulations of complex, multi-domain system behaviour are difficult.
  ➢ The independent development of components by distributed partners and component reuse are difficult.

Motivation of the project

The future of Systems Engineering:

The SPEEDS project addresses the shortcomings of the existing model-based approaches ...

- ... by providing a concept and a technology for cross-domain model-integration. That is, the individual domain & tool specific models can be integrated into one complete system model.
- ... by providing a concept and a technology for advanced component-based design, based on “contracts”. Such contract is a component-description in terms of its characteristics (promises) and constraints (assumptions).

➢ As a consequence:

   ➢ Automatic system model analysis and validation across domains as well as running simulations of complex, multi-domain system behaviour are enabled early in the design process. Thus, modifications become less expensive and costly tests based on physical test rigs may, to a certain extent, be replaced by cost-efficient virtual system simulations.

   ➢ Independent component development by distributed partners and component reuse are supported.

•Further, SPEEDS provides a high-level System-Development Process. The novelty of this process is a management mechanism for steering design iterations and controlled incremental design.

AIRBUS

SPEEDS Concepts

Complete Virtual System Model

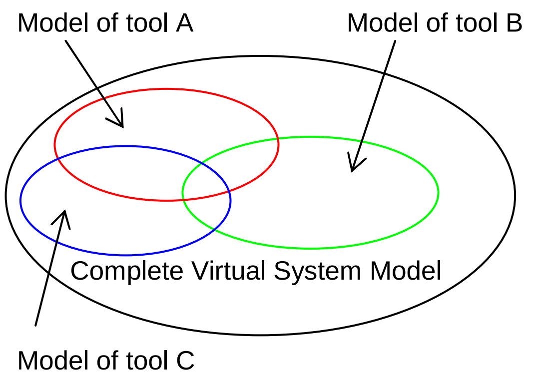
•The SPEEDS team has developed and implemented a formal meta-modelling language. This language provides the format for a Complete Virtual System Model that integrates all domain-specific and tool-specific system models.

•For any system modelling tool an adaptor can be generated that converts the tool-specific model, based on the tool-specific format, into the SPEEDS meta-model language. For a number of existing COTS modelling tools, such adaptors have already been provided by the respective tool vendors.

•After the conversion, a control mechanism adds the converted model to the Complete Virtual System Model and stores it in a repository. The model is considered "virtual" because in its completeness it cannot be loaded into any individual modelling tool. Each modelling tool is only able to load and manipulate those aspects that can be handled by the particular tool.

•The Complete Virtual System Model is used as the vehicle for communicating system modifications in one domain (e.g. System Functionality) to other, affected domains (e.g. System Costs, System Safety). Also, routines use the Complete Virtual System Model as the basis for running system-wide, cross-domain system analysis and simulation (system verification) jobs.

•The formal meta-modelling language and thus the Complete Virtual System Model is able to differentiate between different system abstraction layers. Thus, system analysis, validation and simulation activities can be performed very early in the design process when the system is only available as a rather abstract model. Also, consistency checks between different abstraction layers may be run.



SPEEDS Concepts

Contract-based design, analysis and simulation

•Within the Complete Virtual System Model, for each model component a “contract” is defined. Such contract is the combination of “promises”, i.e. what the component promises to do, and “assumptions” that define under what environmental conditions the promises are valid, i.e. what the component needs for being able to deliver the promise.

•These contracts facilitate a component-based, modular way of working. As the components are well defined by the contracts, they may be independently developed in a distributed team and partner environment. Thanks to the integrating nature of the Complete Virtual System Model, the development partners may just as well use different modelling tools.

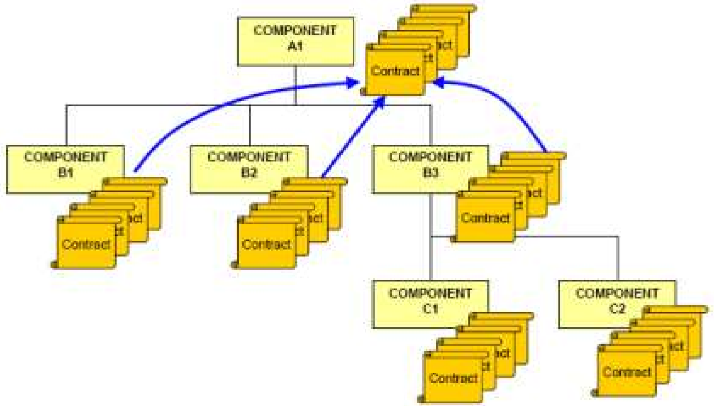
•The attachment of contracts to the components in the Complete Virtual System Model allows for applying analysis techniques, which facilitate early virtual system integration tests. Based on the contracts it can be checked if the local component-contracts together with the components' interconnections are consistent with the global system contracts.

•These analysis techniques provide the designers with information on if the components are correctly specified such that the composition can satisfy the global system requirements. Thus, system validations are possible during early design phases.

•The contract-based design concept also allows for component replacement, such as component reuse from libraries. If a system component-contract is dominated by a library component-contract, the system component can be replaced by the library component without having to re-validate the entire system.

•The overall system analysis is based on *relations*: an implementation has to *satisfy* the related component's contract, a component's contract has to be *compatible* with the overall system (the assumptions are feasible within the system) and *consistent* within itself (within the set of promises and assumptions there are no contradictions).

•In the SPEEDS context, a system analysis is managed via the Process Advisor Tool, which has been developed by the SPEEDS team. On top of these basic model analysis approaches, the Complete Virtual System Model may be subjected to further checks run by specific analysis and simulation tools.



SPEEDS Concepts

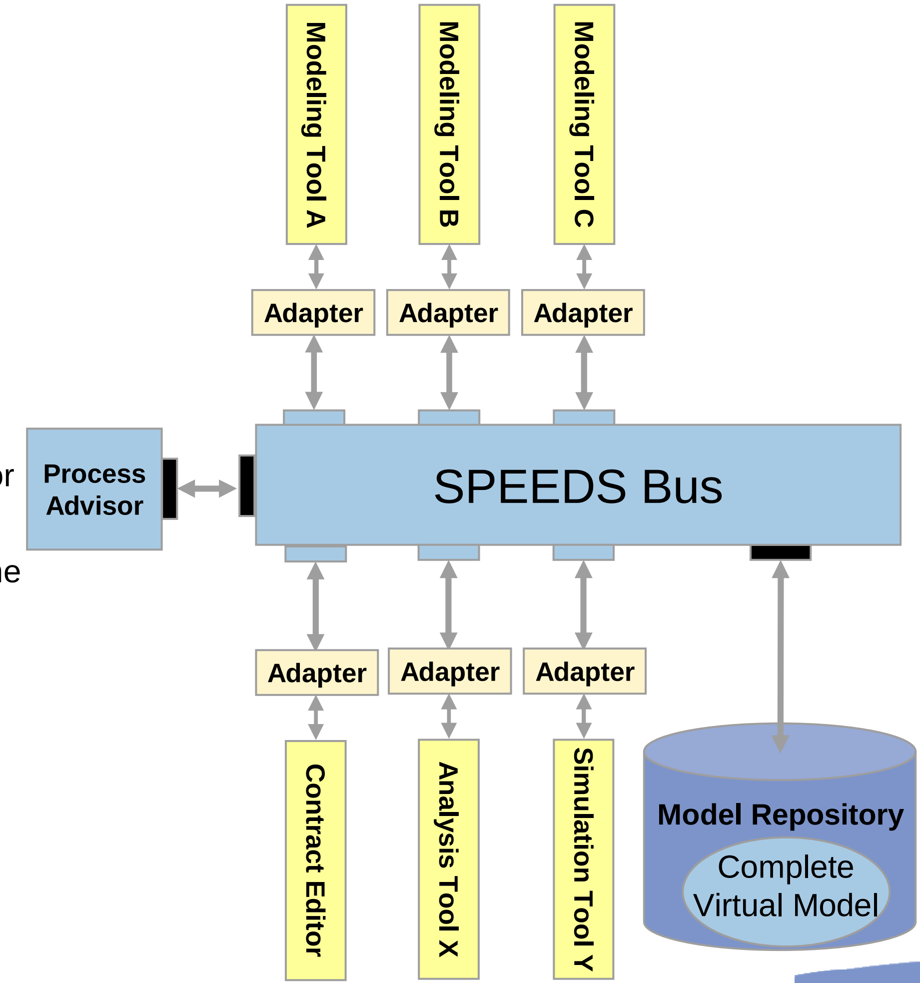
Tool integration architecture

•The SPEEDS environment integrates the various system modelling, analysis and simulation tools. It is based on a service notion. Each tool participating is a service provider as it solves a certain task for the user.

•The SPEEDS environment provides the infrastructure and description mechanism to realise a multi-server architecture where each communicating partner may be server and user of services. This is a typical setup of a bus infrastructure and thus the glue of SPEEDS environment is the SPEEDS bus.

•The overall control mechanism, i.e. data flow control, central logging and event notification are realised by the Process Advisor Tool.

•Each participating tool is connected to the SPEEDS bus by some tool-specific adaptor.



SPEEDS Concepts

Overall Speculative Design Process-Principles

The SPEEDS team proposes some high-level process-principles that can be implemented in any system development undertaking. These principles are:

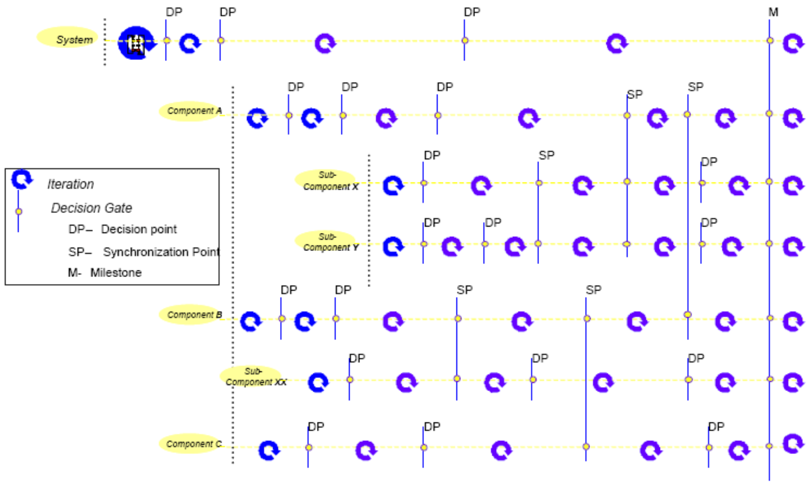
- Component-based and contract-based design (specification of promises and assumptions per system component);
- Iterative and incremental component development;
- Decision Gate after any iteration.

"Traditional" development processes focus on Milestones. A Milestone is a Decision Gate that verifies the global progress of the development at the end of a major phase. The SPEEDS team proposes to increases the control capacity by introducing Decision Gates that focus not only on the global level, but also on granularity levels, such as individual iterations.

There are 3 types of Decision Gates:

- Milestone: high level checkpoint, marks completion of major development phases.
- Synchronization Point: checkpoint between iteration results of related components.
- Decision Point: checkpoint after any individual iteration.

In addition to the Decision Gates, there is an overall and permanent design observation activity. A principal aspect of this observation activity is design component and system maturity monitoring.



Deliverables and current status

Deliverables:

- Complete Virtual System Model
  - ✓ Definition of the formal meta-modelling language for creating a Complete Virtual System Model (document)
  - ✓ Implementation of the formal modelling language (software)
- Contract Editor
  - ✓ Definition of the contract specification language (document)
  - ✓ Implementation of the Contract Editor (software)
- Analysis
  - ✓ Definition of a general system analysis methodology based on a Complete Virtual System Model (document)
  - ✓ Implementation of prototype analysis and validation tools (software)
- Simulations
  - ✓ Definition of a hosted simulation approach (document)
  - ✓ Implementation of adaptor for a specific COTS simulation tool (software)
- Tool integration architecture
  - ✓ Implementation of SPEEDS-bus (software)
  - ✓ Implementation of Process Advisor tool (software)
  - ✓ Implementation of adaptors for some specific COTS system design and simulation tools (software)
- Overall Speculative Design Process-Principles
  - ✓ Description of the overall Speculative Design Process (document)
  - ✓ Description of how to generate system models using the SPEEDS concepts and deliverables (document)

Current status:

- Basically, all the above mentioned deliverables are available. However, particularly with respect to the software deliverables there is still room for improvement. This is regarding usability and robustness.

Demonstrator implementations

- General concept evaluation studies (public)
  - Water-tank-control system (focus of Airbus)
  - Autonomous passenger-taxi operations system (focus of another industrial partner)
- Project partner-specific pilot studies (confidential). Airbus pilots:
  - High-Lift system
  - IMA
- Conclusions from the demonstrator implementations:

  The demonstrator implementations have shown that overall the SPEEDS concepts are feasible and beneficial. Within Airbus, the Technology Readiness Level 4 (TRL4) was granted. TRL4 means, the concepts have been successfully tested in a laboratory environment.

AIRBUS

AN EADS COMPANY