Proceedings of the 14th European Simulation Symposium (ESS 2002).
October 2002. Dresden. ISBN 3-936150-21-4

A SIMULATION ARCHITECTURE FOR COMPLEX DESIGN PROJECTS

Hans P.M. Veeke
Delft University of Technology, Sub Faculty of
Mechanical Engineering and Marine Technology,
Industrial Engineering Group
Mekelweg 2, 2628 CD Delft
The Netherlands
H.P.M.Veeke@wbmt.tudelft.nl

Jaap A. Ottjes
Delft University of Technology, Sub Faculty of
Mechanical Engineering and Marine Technology
Transport Technology Group
Mekelweg 2, 2628 CD Delft
The Netherlands
J.A.Ottjes@wbmt.tudelft.nl

Alexander Verbraeck
Delft University of Technology
Faculty of Technology, Policy and Management
Systems Engineering Department
P.O. Box 5105, 2600 GA Delft
The Netherlands
a.verbraeck@tbm.tudelft.nl

Yvo Saanen
Delft University of Technology
Faculty of Technology, Policy and Management
Systems Engineering Department
P.O. Box 5105, 2600 GA Delft
The Netherlands
yvos@tbm.tudelft.nl

KEYWORDS
Distributed Simulation, Simulation Architecture, Transportation, Reusability, Interoperability

ABSTRACT

This paper presents the functional design of a simulation architecture supporting simulation research from the very start to the very end of a complex design project. The architecture is currently being used in the FAMAS.MV2 research program to investigate terminal structures for container handling on the planned extension of the Rotterdam port area “Maasvlakte 2”. The simulation architecture creates a harmonized simulation environment (a “backbone”) that links and synchronizes the different simulation models during the different project stages into one experimentation and demonstration environment. The result is an architecture supporting multiple simulation platforms in an almost completely transparent way for the users. The design has been successfully implemented and the first simulation experiments have already been performed with it.

INTRODUCTION

Because the space in the Rotterdam port area is running short when container volumes to be handled continue to increase, a new piece of land will be reclaimed from the North Sea near the current area of the Maasvlakte. A large scale research project is started to investigate the best ways of using this new area and to formulate detailed proposals (FAMAS, 2000).

The project can be considered a design process, so it can be characterized by a series of actions: formulation, analysis, search, decision, specification, and modification. Because it’s a long term project (several years), a changing situation may cause ongoing revisions of the requirements and intermediate results. So at every stage of the project, these actions are highly interactive as illustrated in Figure 1 (Jensen and Tonies, 1979).

Simulation will play a major role in all actions shown. It will also be used during all stages of the project, from the global definition to the final detailed studies. During the project several project groups will be formed each using simulation for decision support on a subsystem of the whole complex.

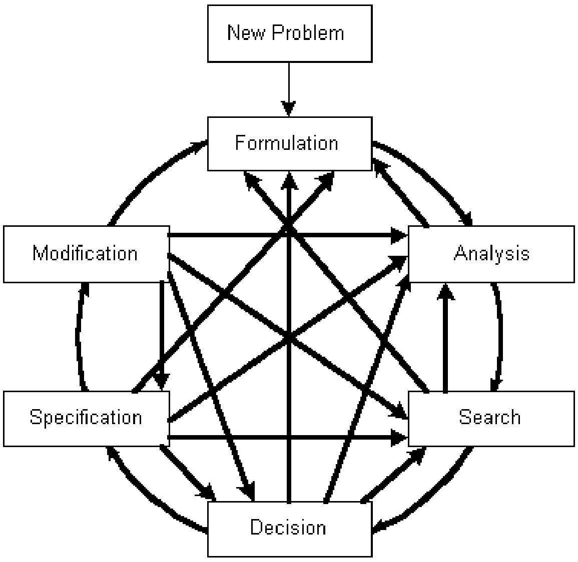


Figure 1. Conceptual design process

To preserve the project requirements throughout these stages and to support simulationists in all actions the need was felt for a simulation architecture: the *“backbone”*.

First the requirements for such a backbone will be defined. From there the needed functionality will be described and finally the first results of implementation will be shown.

REQUIREMENTS FOR THE SIMULATION BACKBONE

The FAMAS.MV2 project is a research project. The research includes the infrastructure of the intended area and feasibility studies of innovative concepts for operations and control of systems and resources. The area is considered a system of cooperating container terminals, some of them serving all transport modalities and others serving only part of the modalities. During all phases and in all subsystem studies simulation will be used to model alternatives and to 'prove' chosen concepts.

Our experience with simulation studies in similar large scale projects of the past is, that they tend to:

- be performed in 'environmental isolation'; a study is performed apart from the overall system and only considering the subsystem or resource under study
- be performed in 'hierarchical isolation'; new project groups are formed during the lifetime of the overall project and often 'forget' their study is defined as a result of earlier simulations, that were executed during a (probably) more global stage.

Besides that, the FAMAS.MV2 project is only concerned with the handling of containers, so significant overlap can be expected in types of resources to be used. Reuse of already modeled equipment will be very profitable.

Finally all research projects may have their own preferences in using commercially off-the-shelf (COTS) simulation tools. To be successful it is of great importance that the backbone is able to link and synchronize these tools in a transparent way for the user.

So the backbone must offer facilities to prevent both types of isolation, stimulate reuse of models and components and offer maximum transparency with COTS-tools.

This results in the following requirements specification of the backbone project itself. The backbone structure must support:

- *Hierarchical modeling:* Hierarchical modeling preserves the decisions made in high level projects and offers the possibility to test detailed systems in their global environment.
- *Reusability:* Reusability stands for reusing already modeled resources and control
- *Interoperability:* Interoperability means the use of different simulation and programming environments working together in a distributed way.

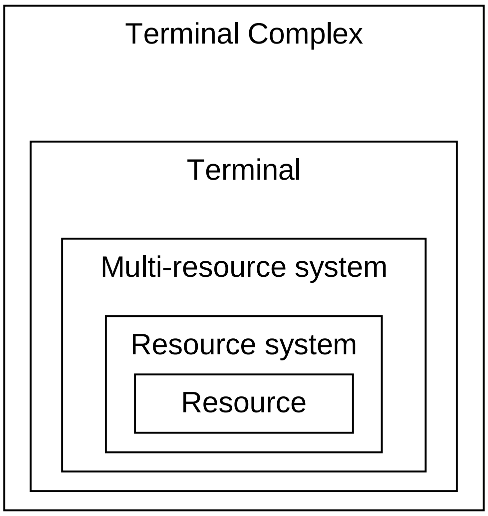
The profitability of the backbone will be a reduction in development costs of global and detailed analysis studies and the preservation of project goals during the whole project.

HIERARCHICAL MODELING

Looking at container handling in an abstract functional way, we can distinguish three basic functions acting on each container:

- Store: a change of state and a change of time (ΔS, Δt)
- Transport: adds a change of position(ΔP, ΔS, Δt).
- Transfer also includes a change of modality (ΔM, ΔP, ΔS, Δt)

Both at the level of the complete area and at the level of a single resource container handling can be expressed in terms of these functions. A function may be executed by a single equipment (e.g. an Automatic Guided Vehicle (AGV) only performs the Transport Function); Several functions may be combined in one equipment (e.g. a carrier combines all functions). A complete terminal can be modelled by a structure of functions with seaside transfer, quay transport, inbound stack store, outbound stack store, land side transport and land side transfer. In fact the complete area has already been modelled in terms of these functions ((Veeke, Ottjes, 2002).



The full structure of the area can be hierarchically described as shown in figure 2.

Fig. 2. Hierarchical structure of container terminal complex

The top-down hierarchical relation is a subsystem decomposition and can be defined by "Level N consists of one or more Levels N+1". Special notice should be given to the terminal complex level. Historically each terminal is considered an autonomous organizational unit in contrast to the terminal complex. Part of the study is to fill in the terminal complex level, ranging from a single inter terminal transport function to an autonomous organization.

Where the global model uses function descriptions until and including the terminal level and models the flow of individual containers, each subsequent model can uniquely identify the subsystem under study and model it in more detail.

Containers flow through all hierarchical levels, but are physically present only at the resource level. Containers can be part of the terminal complex level –while being exchanged between terminals- but will always be coupled to a resource (equipment or space). Resources can be exchanged at any hierarchical level, either between resource systems or between terminals. One resource belongs to one and only one resource system at a time.

Containers and resources represent the physical flows, but besides that a data flow is needed for each function at each level for control reasons. Combined with the well known control-paradigm (de Leeuw, 2000) this leads to the basic function structure for each hierarchical level (fig. 3).

Returning to the requirements, the backbone must support the preservation of results during the iterative process of zooming in and out. So results on dimensions, flow densities, infrastructure and equipment capacities should be easily attained for each model. Moreover, each model must be able to be run in the environment of the model of the next higher level. The consequence for the backbone design is that the structure must support *distributed simulation*. Having this as a conclusion, the question arises to which hierarchical level connections between distributed parts should be supported or even be prescribed. Connection to the backbone can be realized at three functionally different positions of fig. 3. But before we are able to discuss this we first return to the other requirements.

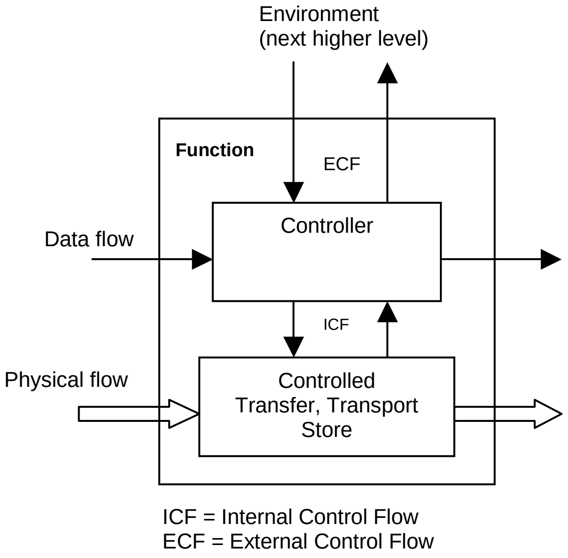


Fig. 3. Elementary control and execution structure

REUSABILITY

Reusability can be interpreted in two ways:

- As the reuse of already modeled components.
- As the reuse of experimentation environments.

As mentioned before, the FAMAS.MV2 project is only concerned with container handling. It can therefor be expected that many resource systems for different terminals use the same type of resource (based on technology and economy of scale). For example quay cranes will be used at almost all participating deep sea terminals and carriers will be involved in almost all truck handling. At the resource level of hierarchy (see fig. 2) the resource control will also be equal in most cases. However, there can (and will) be differences at higher levels of hierarchy. These levels will be the major subject of logistic research in this project.

So far, reuse is discussed from the bottom-up point of view (as if the resources are modeled in detail already). Simulation research however will be executed with a top-down approach. During the project first models will be developed at the highest levels, containing complete terminal and resource system components, albeit very global. These components can be used in connection with a model at the next levels to create an environment which generates container and job flows and restrictions in space and time (i.e. preservation of decision making).

So high level models can be reused as environments for low level models, substituting subsystem models by more detailed models.

As a conclusion the backbone must support the easy connection between (more or less) global system models and already defined detailed resource models. Not only horizontal flows (fig. 3) must be connected, also vertical control flows because they support hierarchy levels.

INTEROPERABILITY

Interoperability means the connection and communication between different simulation and/or real components. But the urgency for this heavily depends on the way simulation models are developed and used. The FAMAS.MV2 project serves two goals: logistic research to improve container handling and presentation to prove resulting concepts to stakeholders and industry.

During research modelers must be considered “individuals” not to be bothered with technical details because of the backbone. The contrary is true: the backbone should support the modelers whenever possible. So if a modeler decides to use some COTS simulation tool, he/she must be able to do so. Most COTS tools however are not constructed to operate in an environment as just one of the tools. So the backbone has to offer facilities to connect these tools in a transparent way to each other or other programs. Besides the technical aspect of communication there is one main concern to achieve this. All simulation tools used in one experiment must be synchronized to the same simulation time axis and the backbone must provide functionality to achieve synchronization: a “Time Manager”. Therefor a minimum of restrictions is put on the tools to be used. The backbone will only support discrete event simulation tools. This is not really restrictive, because simulations in container handling usually use discrete event tools. However COTS tools must provide an entry point into their event mechanism to facilitate event based synchronization. This entry point is preferably located at the moment the local simulation time of the model is about to be advanced to a ‘next event’, because maximum time compression can then be preserved (suppose this next event lies 100 time units in future and nothing else happens in the environment, then this interval can be skipped). Some tools however only offer an entry point based on fixed time intervals and the entry point is handled as a ‘normal’ event. Especially for simulation models at the highest hierarchical levels there is a danger of serious speed degradation by this (Veeke, Ottjes, 2001). Above that, interval based synchronization is not straightforward (Fuji et al., 1999), so the decision was made that the backbone will only support event based (“conservative”) synchronization for research purposes.

For verification and presentation purposes it is to be expected that during the course of the project real equipment will be connected to simulated environments and control. So the backbone time management must also support real time synchronization. In these cases speed is not an issue. So interval based synchronization is applicable and needs only to be tuned to the real communication frequencies.

CONNECTIONS

The conclusions so far for the technical specification are:

- a function approach will be used to support hierarchical modeling. A function is considered a combination of control and controlled execution (fig. 3).
- Connections to the backbone involve both horizontal and vertical flows.
- Connection to the backbone must be transparent for the users
- Synchronization of simulation clocks will be conservative event-based or real time interval based.

The connections to the backbone can be positioned at 4 different positions in relation to the function concept (fig. 4)

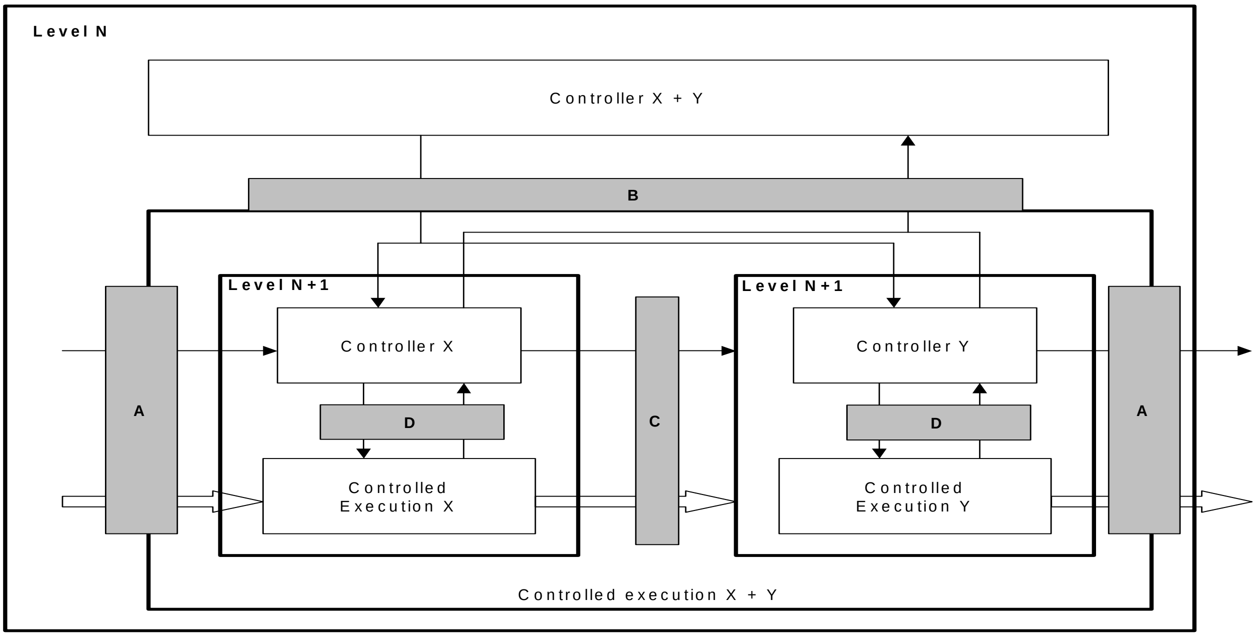


Fig. 4. Connection of functions to backbone

To illustrate the positions two hierarchical levels (N and N+1) are shown. At level N+1 two functions X and Y are detailed representations of the function X+Y at level N. The positions A and B involve a “zooming” connection: data from level N enter level N + 1 adding detail or leave level N + 1 loosing detail. Positions C and D are connections at the same hierarchical level.

Considering the vertical flows first, position D splits the function into separate control and execution parts. This separation is needed for presentation and verification purposes only. During the last stages of the project real equipment will be connected to verify simulation results. During the first stages research will only be delayed if this connection is made obligatory. Position B however serves one of the main goals of the project as a whole. Starting at the highest level control restrictions will be introduced for studies at lower levels. For example physical layout will be defined and fixed. They provide standards and service demands which may not be changed without notice. So use of the backbone in this way is obligatory form the very start.

The same holds for the flows passing position A. There are data flows and resource flows. The entities of the flows are present at both hierarchical levels, with different degrees of detail. Suppose at level N a job is generated to transport a container from T1 to T2. At level N the job is simulated by just ‘waiting’ for some transportation time, at level N+1 the same job will be handled by a complex control algorithm assigning a resource from a limited capacity system to the job and activating the resource, that will drive with specific resource characteristics to the destination. It may even be necessary to add properties to the container. After transportation the job will be ‘returned’ to the higher level by a completion message including container. Job and container are data entities, being used at different levels with a different degree of abstraction. This is even more

the case with the resource. At level N it is just one item of a system with infinite capacity, at level N+1 it's a 'physical' object with technical characteristics being part of a limited capacity system. Above that, job, container and resource are present at both levels at the same time!
In fact, this communication must be completely handled with straightforward data messaging; there's no need to transfer physical objects: it would even be incorrect from the hierarchical modeling point of view.
Position C represents the exchange of data and modeling components at the same hierarchical level. In this case a resource cannot be present in both X and Y at the same time. It's the modelers responsibility to preserve this. It's up to him/her if physical object transfer is used (e.g. by means of CORBA) or not. The backbone takes no position in this; it only facilitates mutual connection by publishing the 'addresses' of the joining models.

BACKBONE FUNCTIONALITY

Finally overall conclusions can be drawn for the technical specification of the backbone.
First of all the backbone needs a functionality to easily create an experimentation environment. This 'Run control' function is the starting point for each experiment. It's "address" should be known beforehand. When running starts, a specific experiment must be chosen. To define experiments a user friendly 'scenario manager' must be developed to specify joining modules, inputs and simulation run data.
To synchronize the connected simulation models a 'time management' function is needed.
A visualization function is needed to show the progress of the experiment as a whole. Each simulation tool will have its own visualization facilities; however to show the environment as a whole specific visualization methods are needed.
Finally, it should be supported that the progress of a simulation run can be registered. This 'logging' function can be used to trace the events of the simulation experiment but also to register data for statistics and graphs afterwards.
The resulting structure of the backbone is shown below.

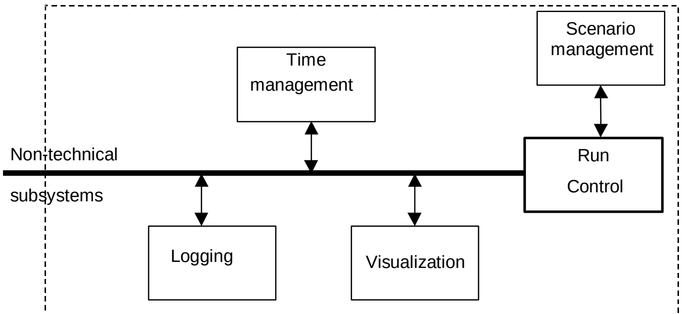


Fig. 5. Internal backbone structure

The functions shown are called the "technical subsystems" of the backbone. The non-technical subsystems may represent any simulation model, control program or real equipment, provided they use the messaging standard defined for the backbone. The messaging standard should be simple and straightforward.

CONCLUSIONS AND DEVELOPMENTS

A simulation architecture for complex design projects must take the demands into account, that are specific for iterative character of design. It has been shown that a 'multilevel' hierarchical research approach differs from a 'one level' experiment approach and includes a modeling approach as well as a technical simulation approach.
The functional description of fig. 5 has been realized now (Boer et al., 2002). A simulation model for the global level is up and running (Veeke, Ottjes, 2002) and the first experiments with a distributed transportation function are being tested. Further detailed studies will start soon using the structure developed.

REFERENCES

Boer C.A., Verbraeck A., Veeke H.P.M., 2002, "*Distributed simulation of complex systems: Application in container handling*", Accepted for SISO European Interoperability Workshop, Harrow, Middelsex UK, june 24-27.

FAMAS.MV2 2000-2002, "*Towards a new generation of automated terminals on Maasvlakte 2*", FAMAS Report, 2000

Fujii, S. et al., 1999, "*Synchronization Mechanisms for Integration of Distributed Manufacturing Simulation Systems*", Simulation 72:3, pp.187-197, ISBN 0037-5497/99

Jensen, R.W. and Tonies, C.C. (Editors),1979, "*Software Engineering*", Prentice-Hall, Inc., Englewood Cliffs, NJ.
de Leeuw, A.C.J, "*Business Management: primary process, strategy and organization*", 2000, ISBN 90-232-3582-7

Veeke, H.P.M., Ottjes, J.A., 2001,"*Applied distributed discrete process simulation*", *Proceedings of the 15th European Simulation Multiconference* (ESM 2001),Prague, ISBN 1-56555-225-3

Veeke, H.P.M., Ottjes, J.A., 2002, "A generic simulation model for systems of container terminals", ", *Proceedings of the 17th European Simulation Multiconference* (ESM 2002), Darmstadt, Germany. pp 581-587,
ISBN 90-77039-07-04.