DSOL: Distributed Simulation Object Library

An overview of service oriented simulation

15th European Simulation Symposium & Exhibition

N.A. Lang & P.H.M. Jacobs

November 4, 2003

Systems Engineering

Delft University of Technology

Agenda: why yet another simulation framework?

- Three cases explicate the need.
- What is service based simulation?
- Requirements for DSOL.
- Tomorrow's tutorial: what may you expect?

TUDelft

Case I : Department of Defense

- The F101 bomber supply chain: visualize the past simulate the future. What-if we go to war?
- Models are distributed; stakeholders are web-enabled and distributed; processing power is distributed;
- Stakeholders need do's & don'ts; reports & documents, examples, lessons, libraries, etc.

TUDelft

Case II: Frog Navigation System

- Can we achieve the required capacity for automatic guided vehicles for a parking garage in the city of Rotterdam?
- Organization has library of algorithms for congestion free routing, curved speed computation, etc. Can they be used in a simulation study?
- 7 full-time simulation and AI experts are interested in collaborative model construction. What are the development tools?
- The organization wants parts of the model to be implemented in the eventual control. How does the decision support tool cope with this?

TUDelft

Case III: Global supply chain game

- A consortium of technical universities and MBA’s join a global supply chain game.
- Goal of the game is to introduce DSS techniques and the consequences of these techniques on profit, sustainability, etc. etc.
- Distributed players, some real other fake, parts actually played, parts simulated. 24/7 asynchronous communication → each player in his own language…

TUDelft

Conclusion: the problems we aim at?

- Multiple **distributed** stakeholders.
- System conceptualization in a **system** of models
- System specification is tightly linked with **underlying IT infrastructure**.
- **Collaborative** model construction and problem solving.
- The construction of **complex** models, in terms of time, aspects and scope.

TUDelft

A vision on service based simulation

Current simulation tools	Service based simulation
1 stakeholder is locally supported by a simulation study	N stakeholders are web-enabled and simultaneously supported.
1 model formalism is allowed in a simulation study	N formalisms can be used to express parts of the model
1 simulation expert can simultaneously work on a simulation model	N simulation experts and IS-developers can collaboratively work on a model

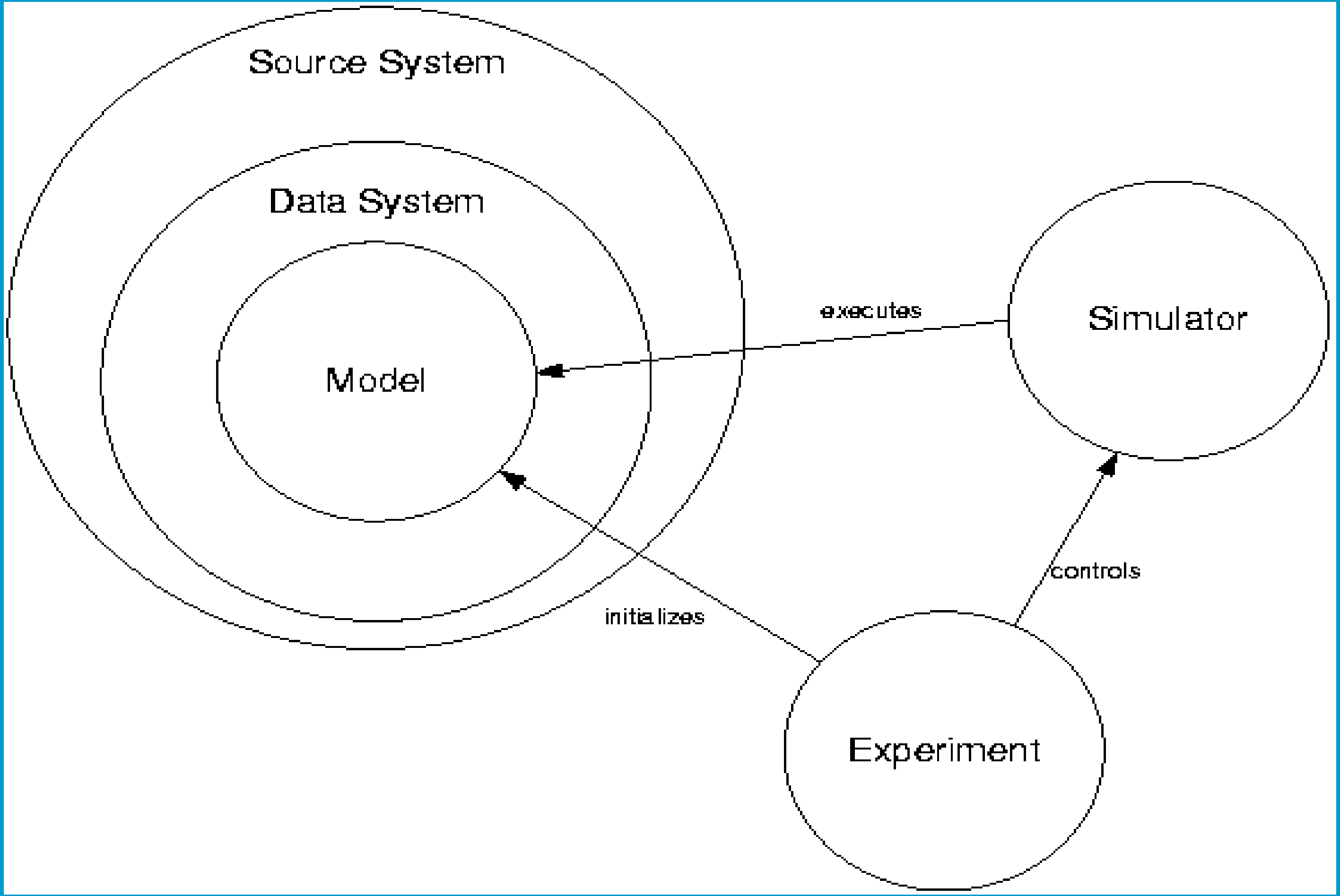
TUDelft

A vision on service based simulation (2)

Current simulation tools	Service based simulation
1 set of tools can be used for reporting, animation, etc. etc.	N services can interact to present, report, compute the study
1 processor can be used to deploy the simulation model	N computational processors can be used to deploy models, services and simulators
monolithic models are developed, globally accessible members	modular models are developed, encapsulated members

TU Delft

Framework for modeling & simulation



Aims of this presentation

- Discuss the 3 main requirements for DSOL:
- **Distributing** the framework for modeling and simulation.
- Providing enough **formalisms** for the construction of models
- Implements DSOL in a **service oriented** architecture.

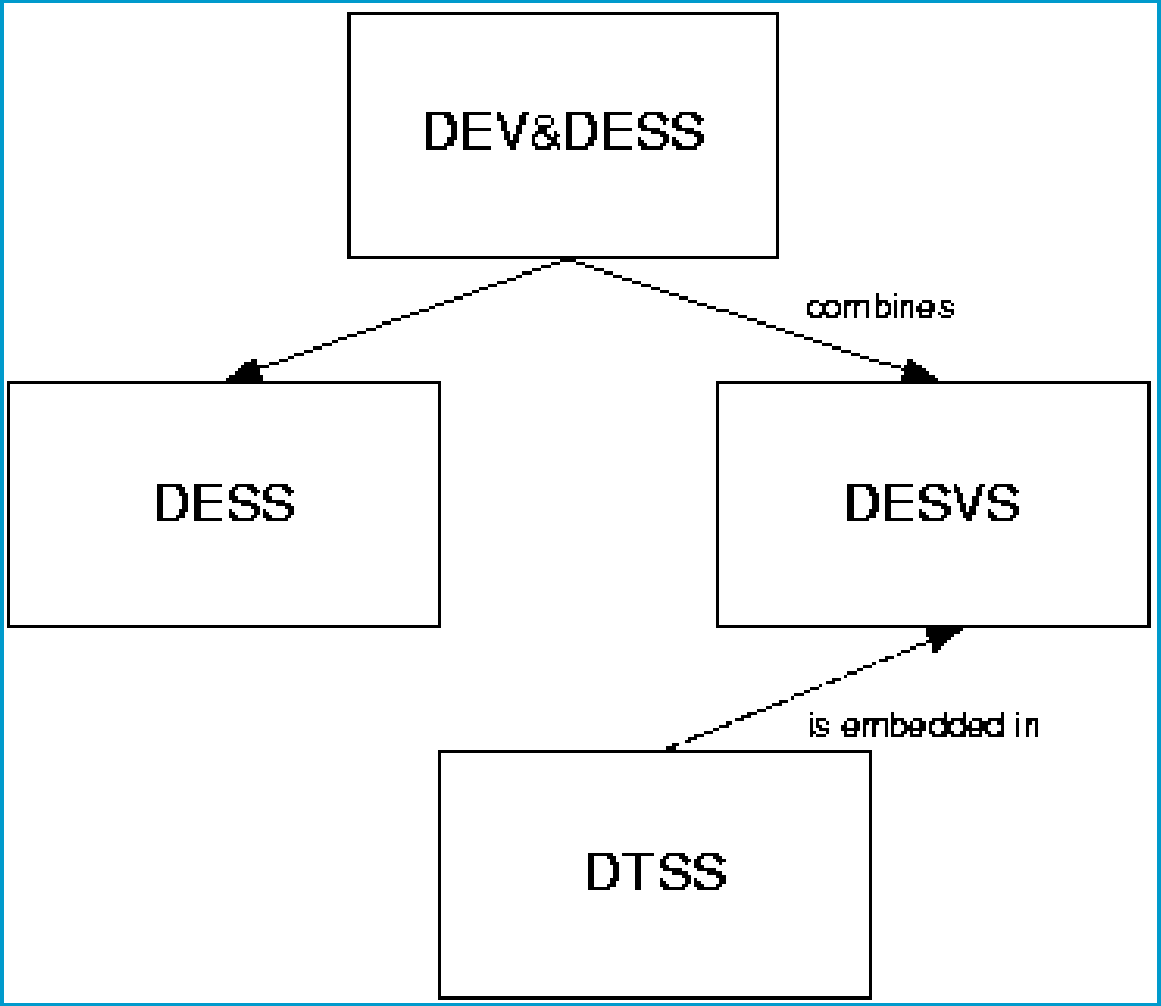
TUDelft

Distribution in the framework for M&S.

- Distributed DSOL **application**: Open remote experiments, report to the web, view animation remotely. Supported in DSOL : **YES**!
- Distributed **model** specification: Entity and simulator distributed, customer and resource distributed, etc. Supported in DSOL : **YES**!
- Distributed **simulator**: HLA, time synchronization between simulators. Supported in DSOL : **NO**!

TUDelft

Formalisms in the framework for M&S



Relations between formalisms.

- **Combining** relation : A combines B and C whenever B and C in A.
- **Embedding** relation: A embeds B whenever any model expressed in B can be expressed in A.
- **Root formalism**: there is no formalism which can be embedded by root formalism A.

Formalisms in DSOL:

- DEV&DESS root formalisms is supported by a unique simulator. This **guarantees** multiformalism!
- Several non-root formalisms are nevertheless supported with unique simulator for **performance optimization**. (DESS, DEVS)
- Several non-root formalisms are supported by libraries **embedding** the formalism (Flow is embedded in DEVS).

TUDelft

Service oriented system design (I)

- **Object orientation** is the de facto modeling and programming language for the design of information systems. (see notes!)

- A **service** is a **contractually** defined behavior that can be implemented and provided by any component for use by any other component, based solely on the contract. This contract is also referred to as the **interface** of a service.

TUDelft

Service oriented system design (II)

- The encapsulation of states. Since a service only describes methods, the state of a component providing the service is shielded to those components invoking it.
- The prevention of relations between objects or components. Service oriented design encourages objects to relate to an interface. Whenever the particular specification or embodiment of the service changes, this has no consequences for the relation.
- Collaborative system specification. Whenever the interfaces of a system are designed, several engineers can collaboratively and concurrently implement parts of the system without potential integration problems.

TUDelft

Conclusions

- DSOL is a Java based distributed application.
- DSOL is an object-oriented simulation framework for distributed modeling.
- DSOL supports several formalisms among which (DESS, DEVS, DTSS, DEV&DESS).
- DSOL is open source and published under GPL at sourceforge.

TU Delft

DSOL Tutorial tomorrow

- Hour 1: overview of packages, architecture, concepts, statistics, experiment, etc.
- Hour 2: Build small models : m/m/1 queue, timeshared computer, continues differential equation model.
- Hour 3+4: Build a complex model of a typical European sea container terminal

TU Delft