COMPARISON OF APPROACHES AND CONCEPTUAL FRAMEWORKS FOR COMPONENT-BASED MODELING AND SIMULATION (M&S)

Marko Hofmann
Institut für Technik Intelligenter Systeme (ITIS)
an der
Universität der Bundeswehr München
Bundeswehr München

Axel Lehmann
Institut für Technische Informatik
und ITIS e.V.
Universität der

e-mail: lehmann@informatik.unibw-muenchen.de
URL: http://www.informatik.unibw-muenchen.de/inst4/

Our Research Focuses and Experiences

- **Institut für Technische Informatik**
  - → Discrete event M&S methodology
  - → Parallel and distributed simulation
  - → Knowledge-based Simulation
  - → Performance / reliability analysis / diagnosis of computer and telecommunication systems

- **Institut für Technik Intelligenter Systeme (ITIS e.V.)**
  - → M&S developoment as a multiple phase engineering process
  - → VV&A (verification, validation & accreditation)
  - → M&S performance and dependability (safety)
  - → (military) analysis, planning and decision support systems

COMPARISON OF APPROACHES AND CONCEPTUAL FRAMEWORKS FOR COMPONENT-BASED MODELING AND SIMULATION (M&S)

OUTLINE

1. M&S - “State of the Art” and Demands
2. M&S-Design, Implementation and Application:
   A Multi-Phase Development Process
3. Component-Based Modeling: Vision and Reality
4. Formal & Architectural Approaches for Component-Based M&S
5. Component Architectures and Technologies
6. Component-Based M&S : Status and Future Perspectives

. M&S - “State of the Art” and Demands

Trends / Requirements of (Technical) Systems Developme

- **Rapid technological innovations**
  - → new technologies (e.g. ICT)
  - → efficient, powerful, computer-assisted . . . . Tools (e.g. CAD, CAM, . .
- **Increasing systems complexity & lifetime**
  - → embedded systems
  - → distributed systems
  - → networks of components / systems
- **Increasing productivity & cost-benefit**
- **Decreasing cycle times for system’s innovations**

Major Challenge

⇒ **“Mastering” of systems over lifetime w.r.t. multiple aspects /goals!**
⇒ **goal-oriented, efficient, effective and dependable models**

⇒ **MODEL-ENGINEERING**

Major evolving application domains for symbolic M&S:

Evaluation of e.g.:

- Technical systems
- Economical systems
- Environmental observations
- Social, political or cultural issues
- (military) conflict situations and strategies
- natural, biological or medical systems

Categories of "Evaluation":

(w.r.t. goal(s) specification & solution space)

- **Analysis** (e.g. diagnosis, selection, classification, decision support
- **Synthesis** (e.g. construction, planning)
- **Prediction** (e.g. evolvement over time, training)

Current Importance of M&S

(as a discipline / methodological approach / tool set)

→ Receives increasing acceptance by decision makers
→ Becomes more and more a “standard” method / tool set
→ Seen as a major enabling technology for innovations
→ Research relevance is well-documented, e.g. by

- *1995, US-DARPA:* *“... M&S is one of the top-10-key enabling technologies ...*
- *1998, DoD (Dr. Gansler):* *“... by the year 2000 ... Systems development in 25 % less time...”*
- *1999, US-government:* *IT² Research Initiative*
- *1999, PITAC report:* *“Fund research in ... global-scale networks and its associated information infrastructure .... including .... Modeling and simulating network behaviour (Recommendation 3.3.2)*
- *2002, EU 6th Research Framework Program*

Major Challenges for M&S applications:

- Increasing systems / M&S complexity
- Decreasing cycle times for systems / M&S innovations
- Increasing systems / M&S lifetimes
- Increasing variety of M&S-aspects / purposes
- Safety, reliability, ... cost-benefit constraints (for systems / M&S)
- “HW-/SW-/User-in-the loop” simulation

⋮

- User acceptance; ease of use & credibility
- Virtual / Augmented Reality

General Demands:

- Hierarchical modeling
- Interoperability
- Reusability
  → adaptability
  → multifacetted modeling
- Collaborative & distributed M&S
- Model credibility
  (→ VV&A)
- Model engineering

2. M&S - Design, Implementation and Application

⇒ **A Multiple-Phase Development Process**

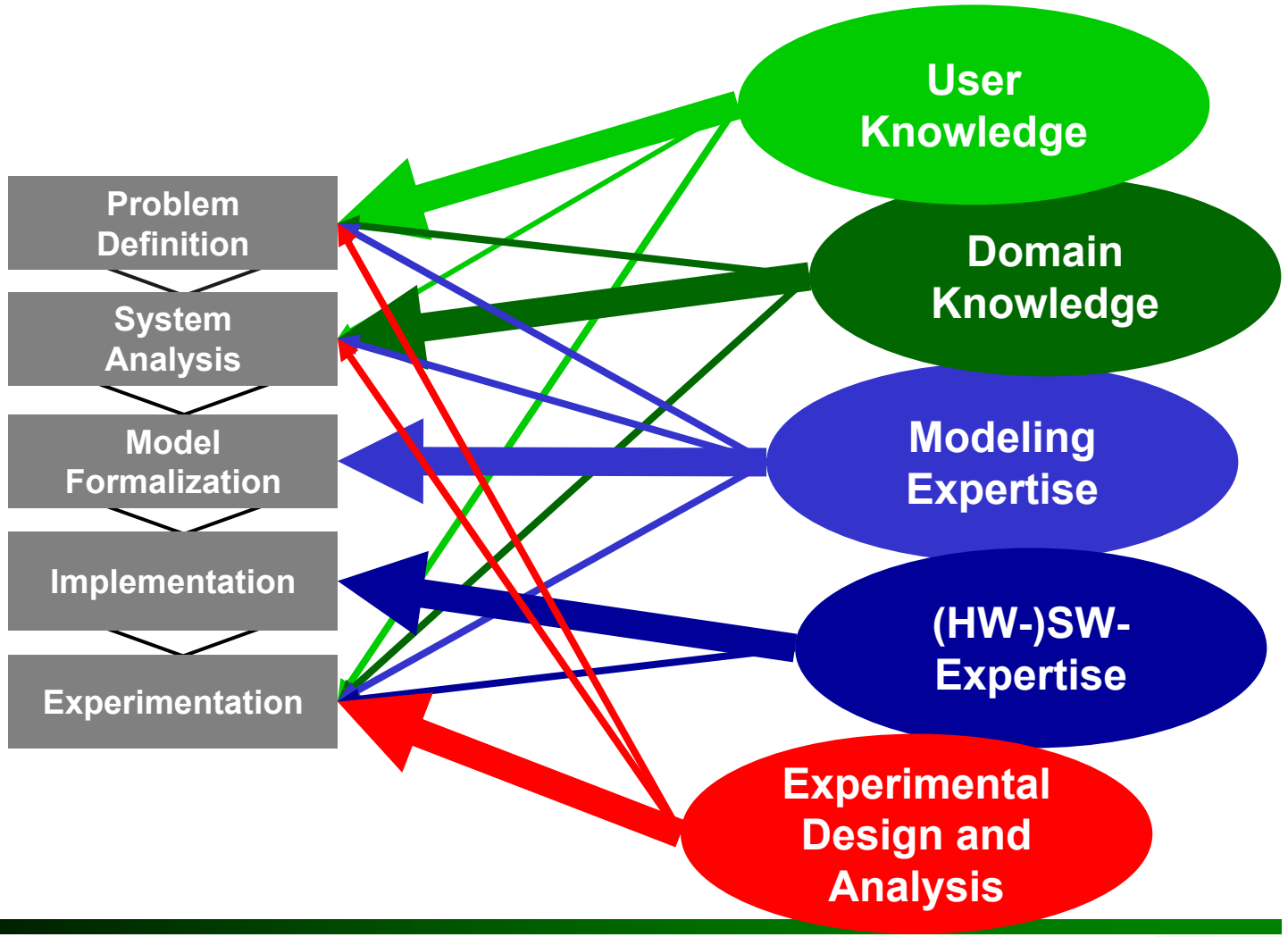
Example: **Effectiveness and efficiency of a "Booking System"**

(client-server architecture)

Goal parameters e.g.:

→ processing time per transaction
→ client/server utilization
→ queueing time, queue length

M&S-Sources of Knowledge and Expertise



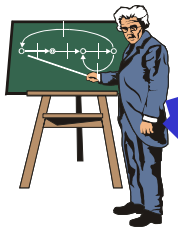
M&S-Development Team

Customer

User

Domain Expert

Project Manager (Contractor)



Modeller

Programmer

Problem Definition

System Analysis

Model Formalization

Implementation

Experimentation

User Knowledge

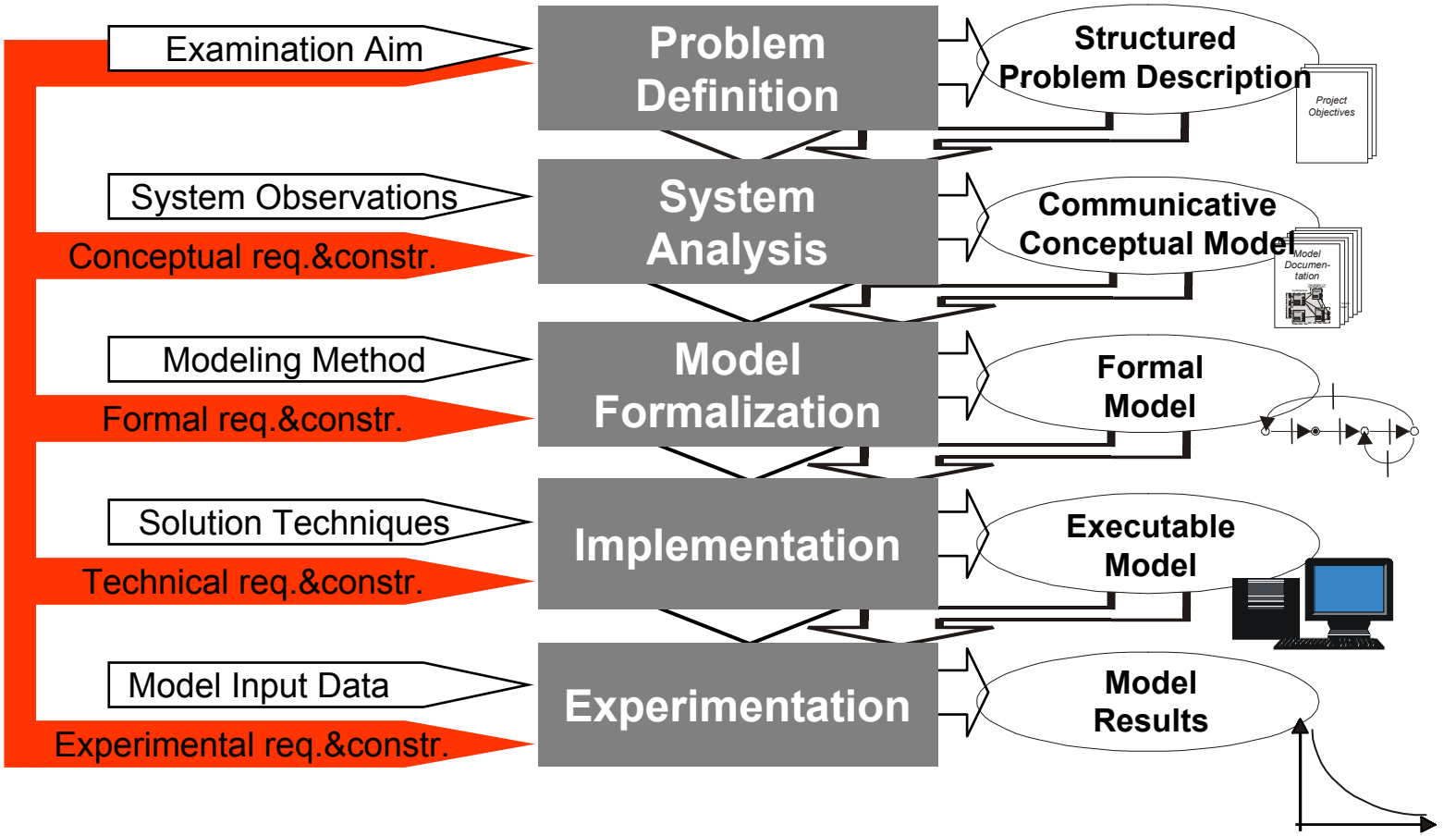
Domain Knowledge

Modeling Expertise

Experimental Design and Analysis

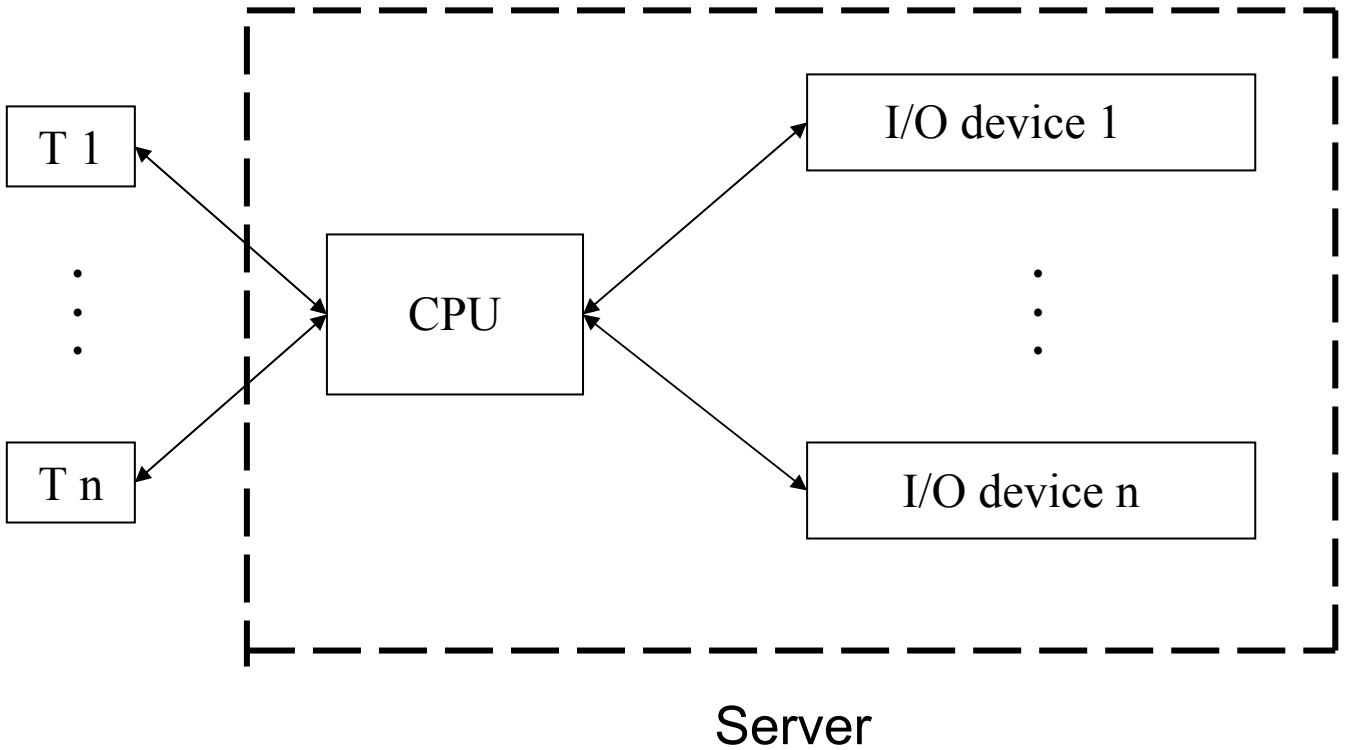
(HW-)SW-Expertise

Phases & Products in the M&S - Development Process



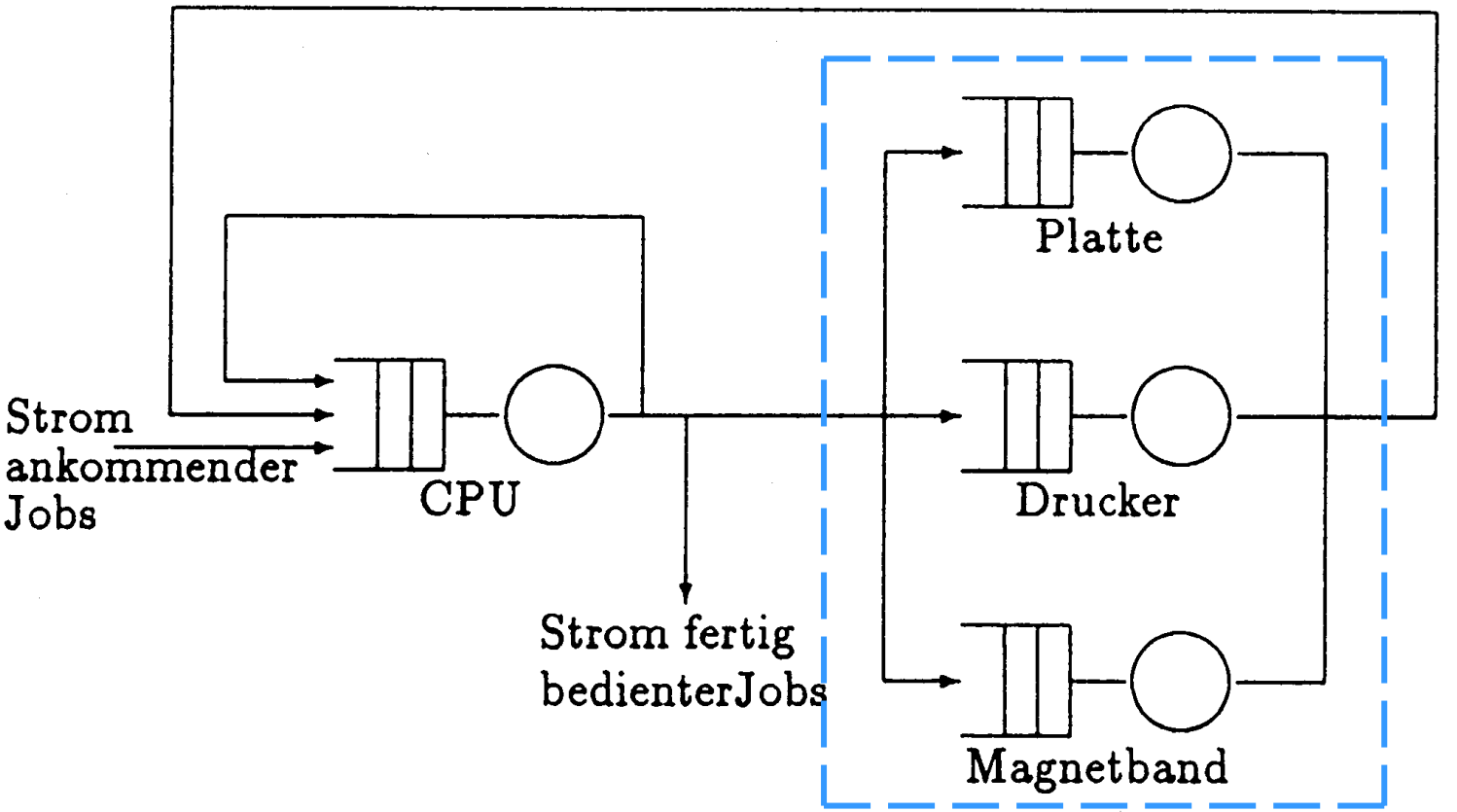
Example: „Booking System“

Conceptual Model :



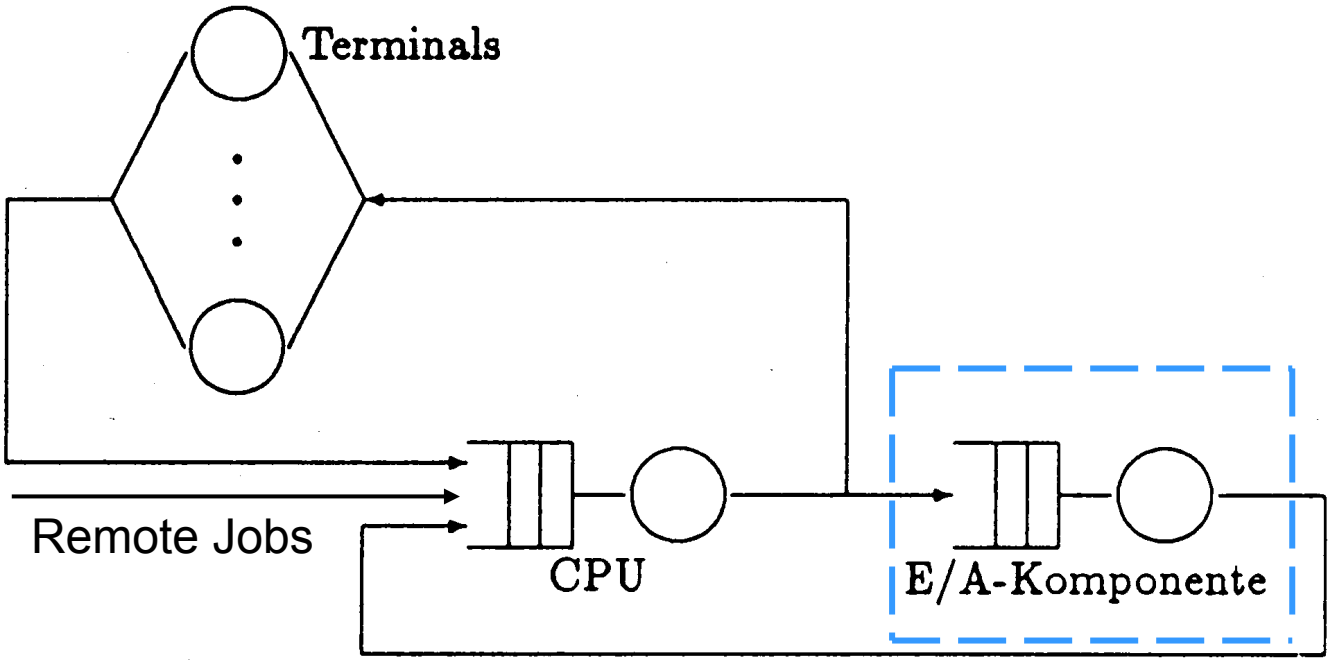
Example: Booking System

Formal Model (Version 1):



Example: Booking System

Formal Model (Version 2):



Performance measures

calculated by an analytic solution of the Formal (Queueing) Model or implemented as a discrete event simulation

- State probability **p(k)**
- Utilization ρ (m service stations)

$$\rho = \frac{arrival\ rate}{service\ rate} = \frac{\lambda}{m\mu} < 1$$

- Throughput λ

$$\lambda = m \bullet \rho \bullet \mu$$

(service rate = arrival rate, stationary state)

❍ Performance measures:

Little's law:

state/number of customers

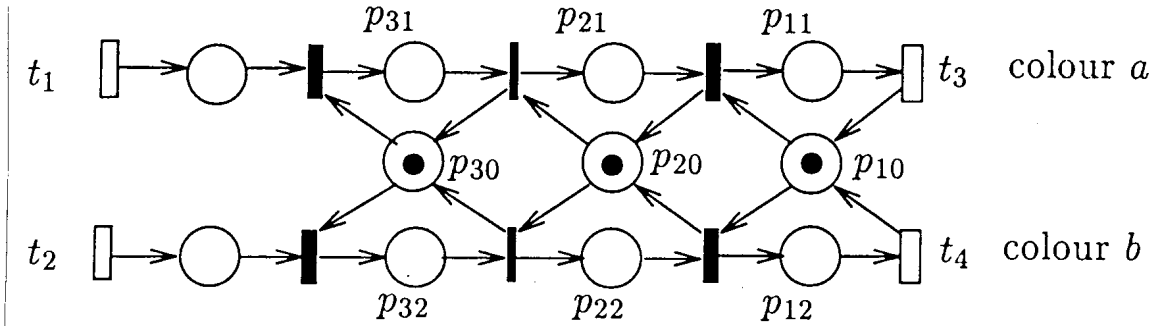
$$\overline{k} = \lambda \overline{t} \quad \text{or}$$

$$\overline{Q} = \lambda \overline{w} \quad \text{queue length}$$

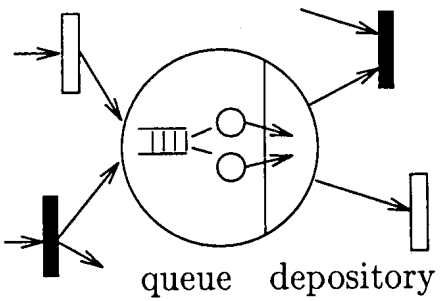
With: response time t, queueing time w, service rate μ

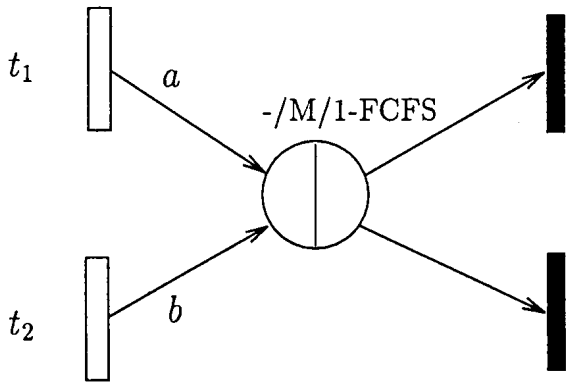
$$\overline{t} = \overline{w} + \frac{1}{\mu} \quad ; \quad \overline{k} = \Sigma \; k \bullet p(k)$$

Queueing Petri Nets (QPN)



GSPN model of a FCFS queue





QPN model of a FCFS queue with 2 colours of tokens

Conclusions

(e.g. regarding the example “Booking System”)

- **Conceptual Model:**
  → structural & functional description of “components”
  → different levels of abstraction

- **Formal Model:**
  → formal specification of “components” (↔ selected modeling paradigm(s))
  → different levels of abstraction
  → hierarchical modeling approach
  (• decomposition into submodels / “components”

- **Executable Model(s):**
  e.g. → analytic solution
  → simulation
  } reusable SW- “components”

3. Component-Based Modeling: Vision and Reality

What is a "Component" ?

- *Douglass (2000):* *"... it's whatever you want it to be"*
- *Szyperski (1997):* *"Components are for composition"*
- *Griffel (1998):* *"... they are having an autonomous functionality" (Translation)*
- *Meyer (2000):* *"... may be used by other software elements (clients) ... without the intervention of the components developers."*
- *Meyer (2000):* *"A Software component ... is a meta product: .....*

Specification
plan
(parameterized)
instance(s) . . . „ (Translation)

M&S-component: =

(pragmatism, intended purpose,

semantic,

syntax)

Properties of a (Software) Component:

- "Includes a specification of all dependencies; (HW-, SW-, etc.)
- includes a precise specification of offered functionalities; ("information hidin
- plug and play;
- is usable on the sole basis of that specification;
- is composable with other components;
- can be integrated into a system quickly and smoothly."

(Meyer, 2000)

component-class /-type: description of structure & implementation

component-objects /-instances: implementation

Differences to Classes / Objects and Modules

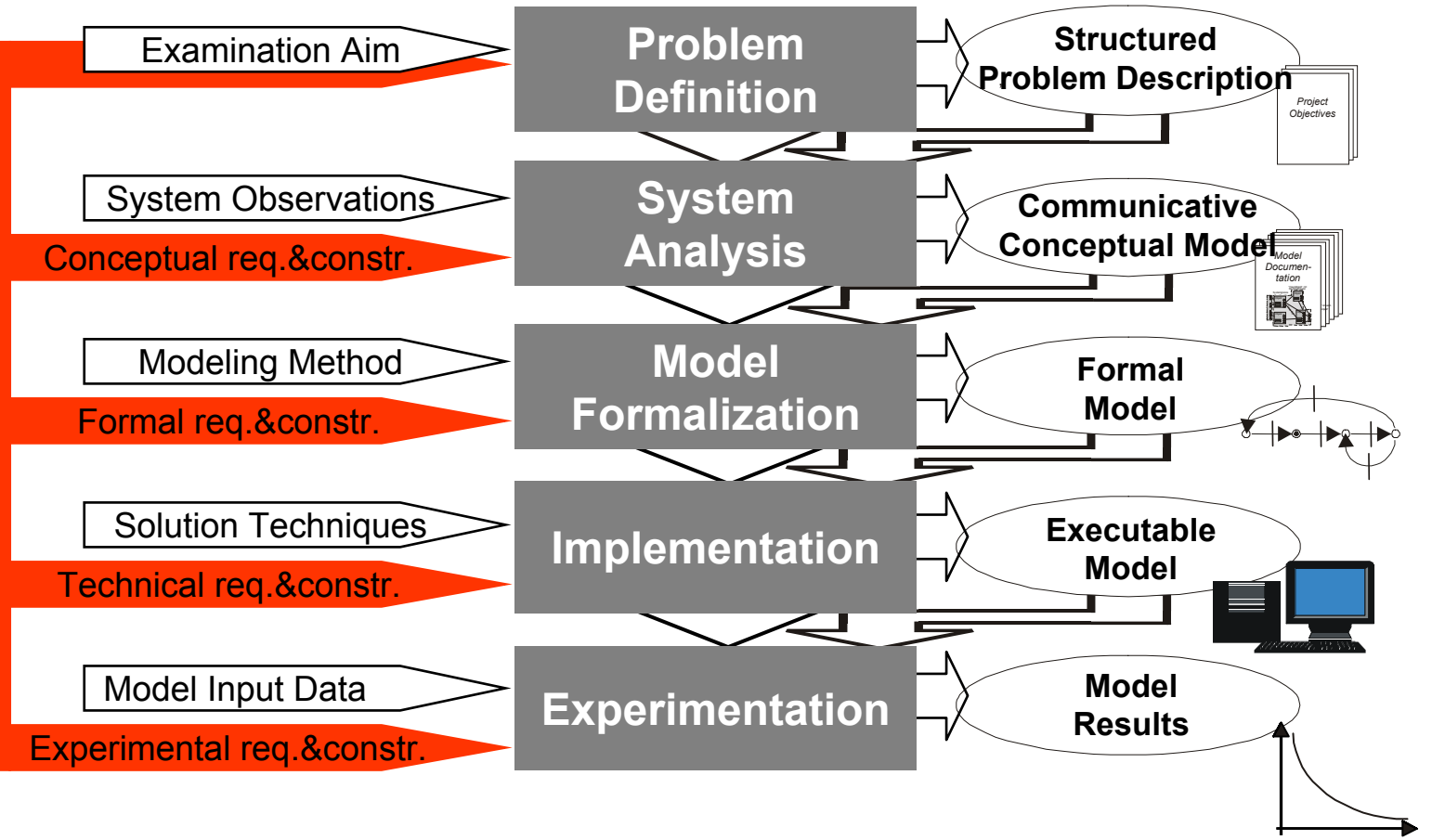
(according to Szyperski, Mayer)

- **Very vague !!**

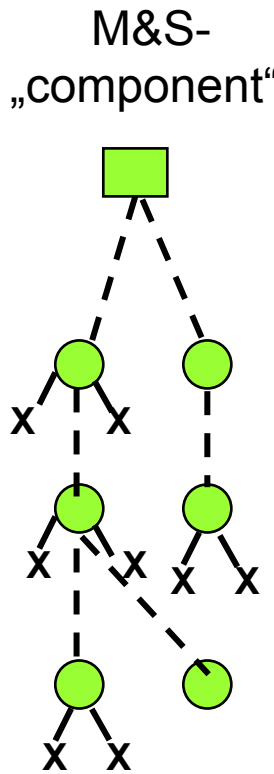
- **Objects:**
  - have a state which can be persistent
  - import / export data, methods
  - can be part of a component

- **Modules:**
  - can be seen as “minimal components”
  - can include abstract data types & object classes
  - have no resources
  - connections to other modules are known (fixed structural arrangements)
  - can contain classes (e.g. Oberon, Modula-3)

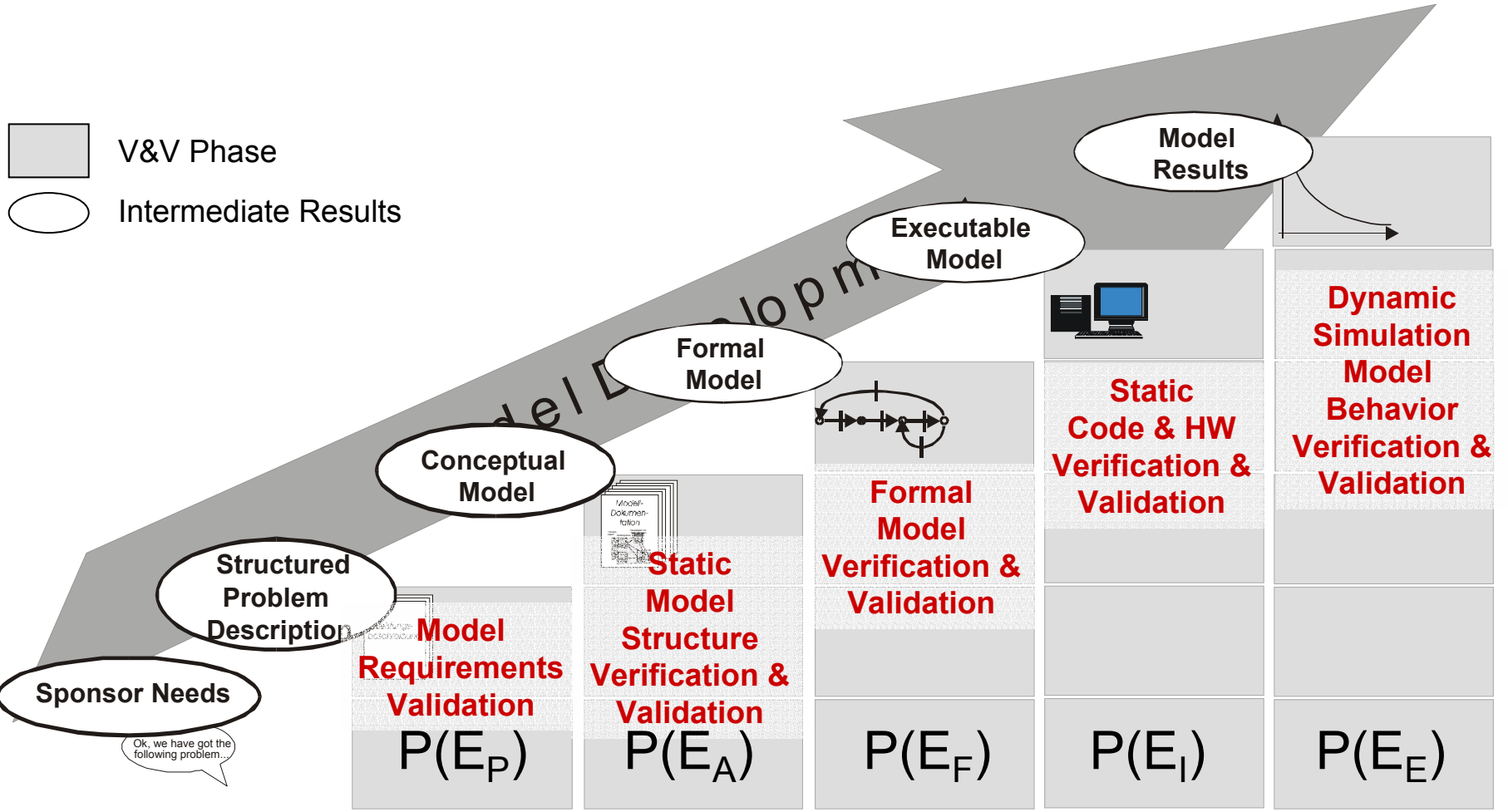
Phases & Products in the M&S - Development Process



Phases & Products in the M&S - Development Process

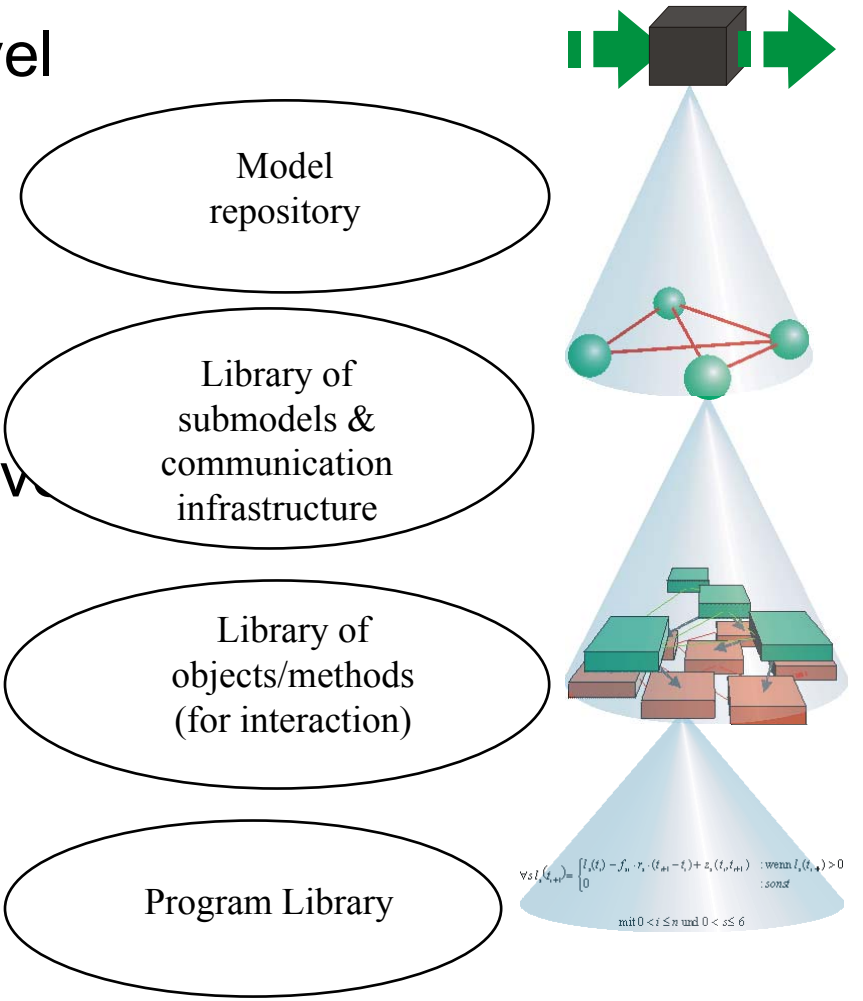


Model Development & V&V



Vision: Model („Component“) Specification Levels

- Model Federation Level
  - „Black Boxes“
- Model Level
  - Autonomous, interoperable models
- Submodel / Object Level
  - Submodels / Object structures of different modeling paradigms
- Function Level
  - Coded basic functions / algorithms



Component-Based M&S : Current Approaches

- **Hierarchical modeling via**
  - decomposition
  - aggregation
  - hybrid solution / implementation techniques
- **Generic modeling object templates (depending on the modeling paradigm), e.g.**
  - class / object libraries
- **Function / Program libraries, e.g.**
  - statistical analyses
  - random number generators
- **Coupling of monolithic models, e.g.**
  - federation of models (DIS, HLA, . . .)
  - agent-based simulation

Conclusion: **Missing comprehensive formal & methodological approaches for component-based M&S**

4. Formale Approaches for Component-Based M&S

(→ Formalisms for Component-Based M&S Specifications)

DEVS (Discrete Event System Specification, *Zeigler*)

Atomic DEVS : = (X, S, Y, in, ex, au, v)

X	:	set of input ports & input events set
S	:	sequential states set
Y	:	set of output ports & output events set
in : = $S \rightarrow S$	:	internal state transition function
ex: = $Q \times X$	:	external transition function
au: = $S \rightarrow Y$	:	output function
v : = $S \rightarrow R^+$	:	time advanced function

- **Coupled Models ↔ hierarchical modeling approach**

UML (and extensions)

→ standard notation for analysis, design and implementation of (object-oriented)-systems:

→ UML-meta model ↔ specifies abstract semantics

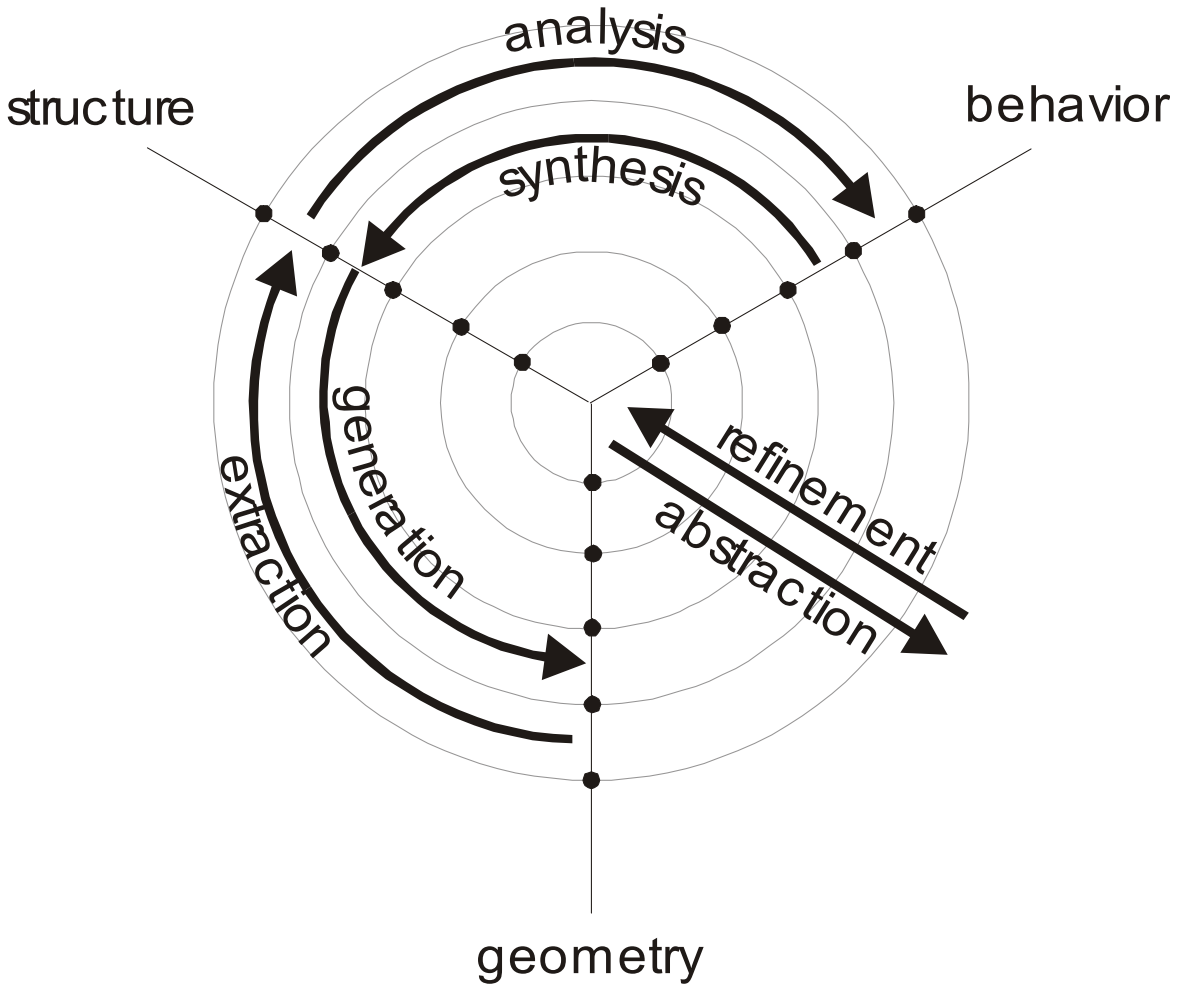
→ UML-notation ↔ describes a set of diagrams (→ syntax):

Static diagrams:	Use Cases, Class / Object Diagrams
Behavioural diagrams:	Activity ~ State ~
Interaction diagrams:	Sequence ~ Collaboration ~
Implementation diagrams:	Component ~ Deployment ~

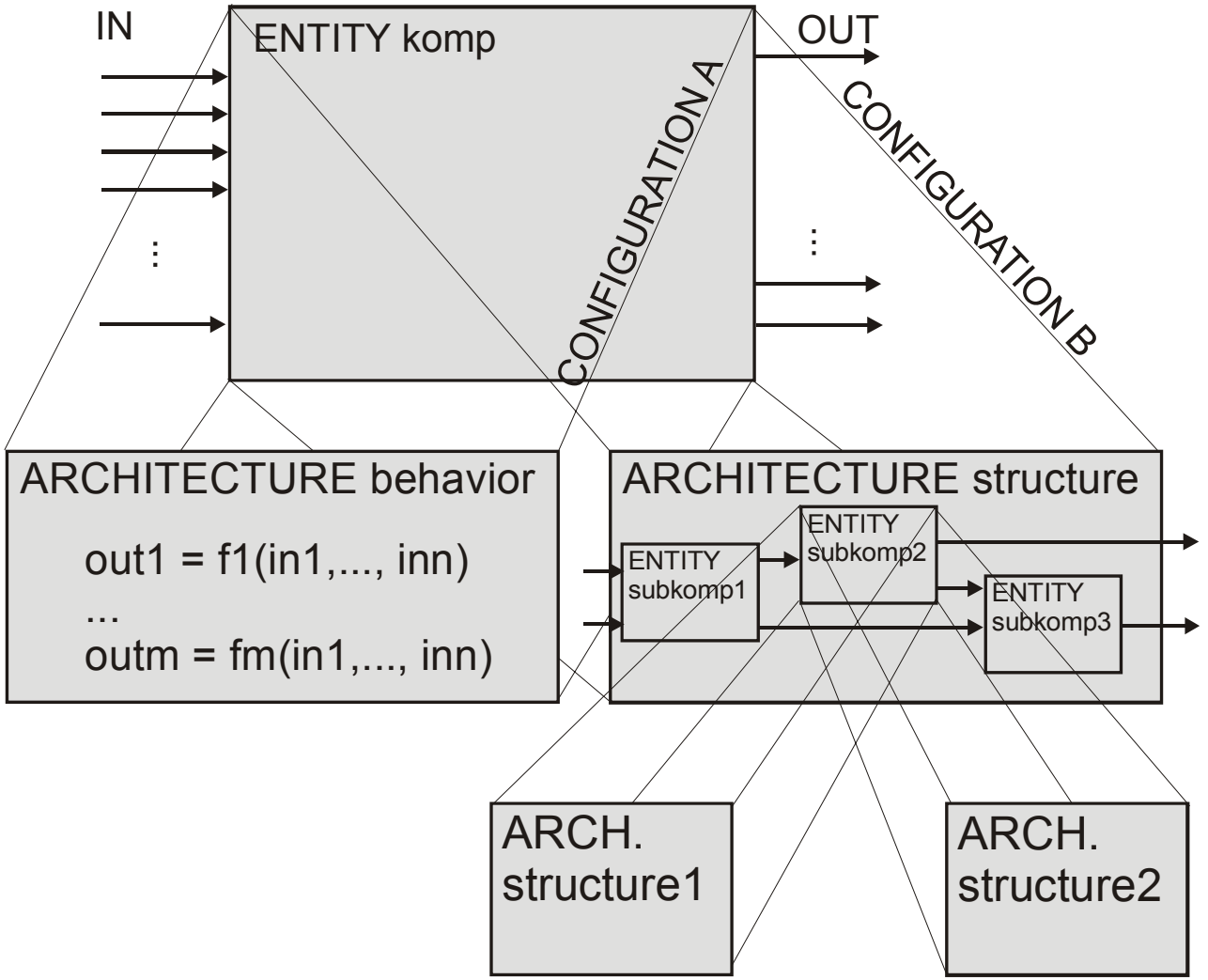
→ Extensions, e.g. for quantitative software performance M&S

VHDL: Hierarchical & Component-Based Hardware / Microelectronic Circuit Design

- **Gajski-/Y-Diagram**



• Component and its instances in VHDL

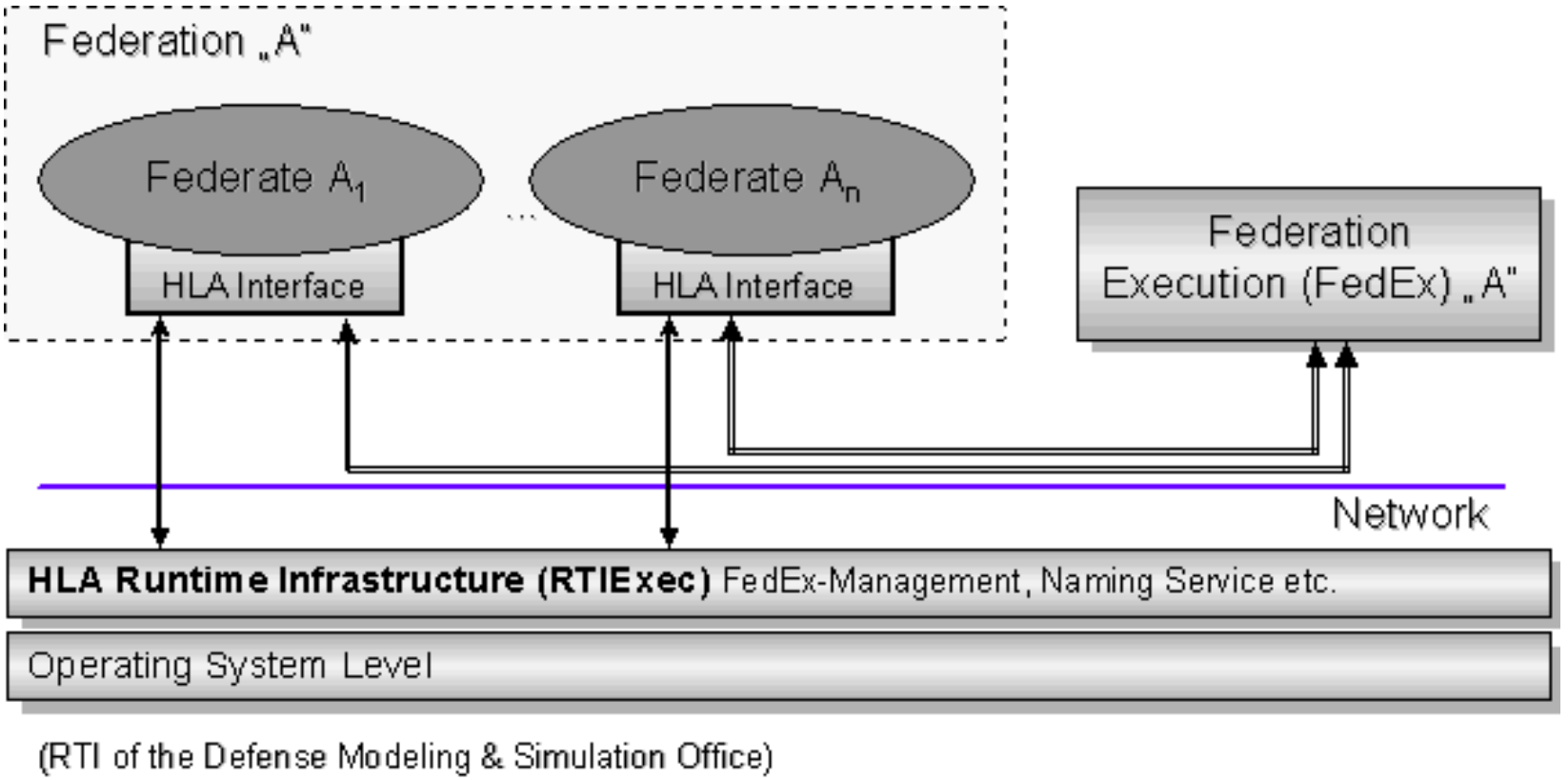


5. Component Architectures and Technologies

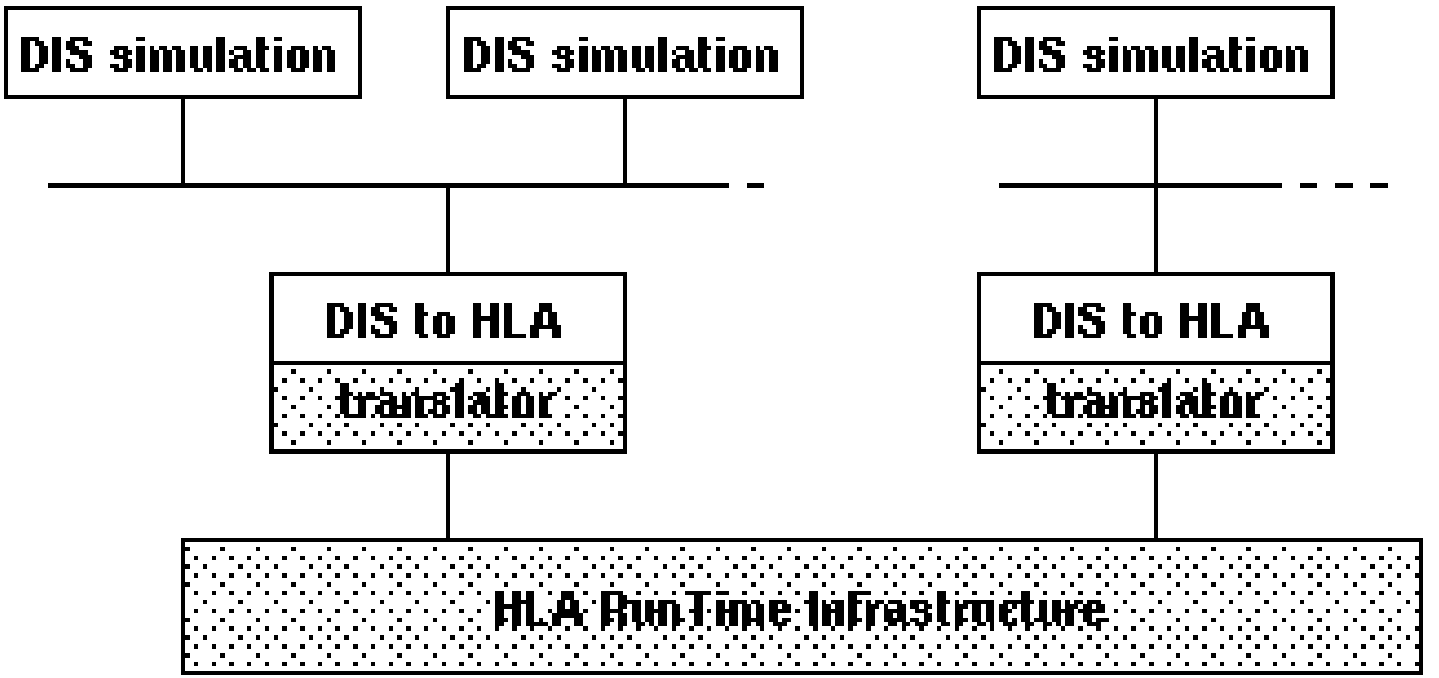
- **Architectures**
  - OMA (Object Management Architecture; basis for CCM)
  - Agent-based Simulation
  - DIS (Distributed Interactive Simulation)
  - HLA (High Level Architecture)

- **Component Technologies**
  - Component Object Model (Microsoft)
    (COM, COM+, DCOM, ActiveX)
  - Java Beans, Model Beans, EJB (Sun Microsystems)
  - CORBA Component Model (CCM; OMG)

❍ High Level Architecture (HLA)



(RTI of the Defense Modeling & Simulation Office)


DIS simulation
DIS simulation
DIS simulation
DIS to HLA
translator
DIS to HLA
translator
HLA RunTime Infrastructure

6. Component-Based M&S: Status and Future Perspectives

Demands

- **"Mastering" of systems / M&S complexity over lifetime**

  → Hierarchical M&S e.g. by

  - decomposition
  - aggregation

  → **Reusability & interoperability of M&S-components via:**

  - a semiotic approach:
    (pragmatism/intended purpose, semantic, syntax)
  - based on: a comprehensive M&S methodology system-theoretic formalisms, and on architectural & technical frameworks

- **M&S development & maintenance: engineering discipline ! ?**

Status - Perspectives

- Component technologies available
- Component architectures developed

} SW-focus !!

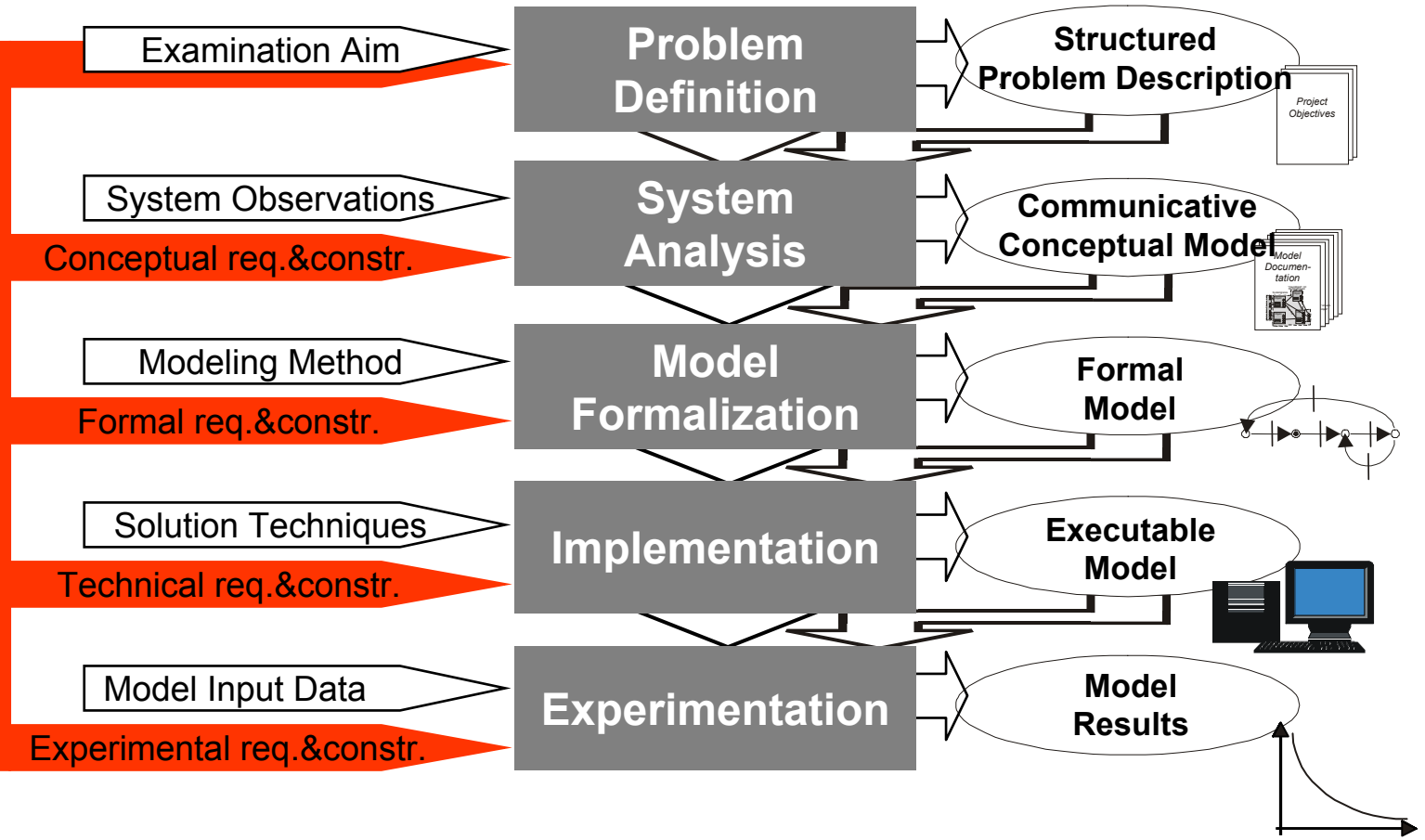
- but: M&S-component specification:
  (pragmatism/intended purpose description
  semantic (plan)
  syntax (implementation)) is almost missing !!

- **Long Term Perspectives:**

  → Repositories for: pragmatic concepts,
  formal model components (algorithms),
  executable model components (code)

  → Increasing long-term availability / reusability
  productivity & credibility (→ VV&A)

Phases & Products in the M&S - Development Process



M&S-„component“