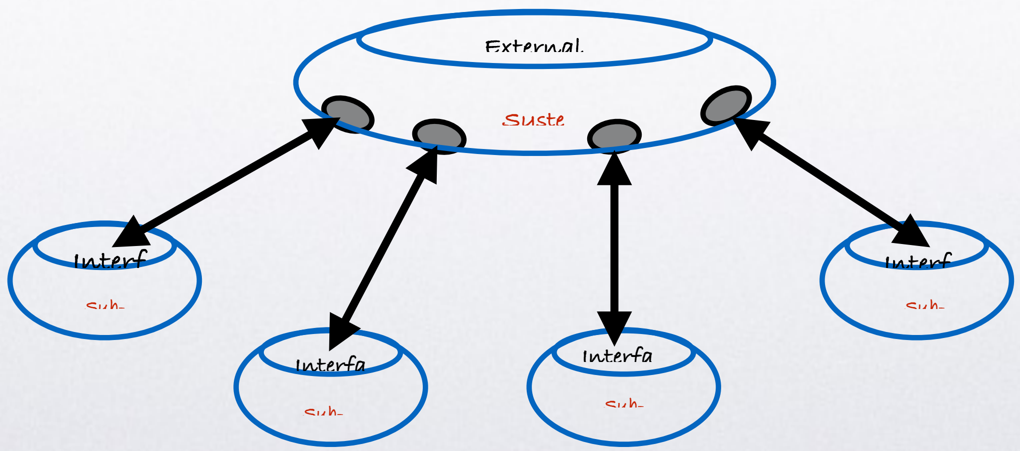
PMR 5237

Modelagem e Design de Sistemas Discretos em Redes de Petri

Aula 11: Modelagem formal de sistemas baseada em eventos



Prof. José Reinaldo Silva
Escola Politécnica da USP

Plano de Aulas

Aula	Tema	Data
Aula7	Propriedades das redes P/T e Coloridas	9/11
Aula8	Análise de Invariantes	16/11
Aula9	~~Técnicas de modelagem~~	23/11
Aula10	Métodos de Design orientados a estados	30/11
Aula11	Métodos de Design orientados a eventos	07/12
Aula12	Perspectivas de pesquisa em modelagem de sistemas discretos com RdP	14/12

	PMR3510	PMR3410	PMR5237
9/12	Entrega do EP (meia-noite)		Milestone do artigo (7/12)
14/12		Entrega do Pitch (video 3min) da Startup	Milestone do artigo (14/12)
17/12	Entrega da monografia		
21/12		Entrega da documentação da proposta de Startup	Artigo Final (21/12)

Peer Play as a Context for Identifying Profiles of Children and Examining Rates of Growth in Academic Readiness for Children Enrolled in Head Start

Elizabeth R. Bell, Daryl B. Greenfield, Rebecca J. Bulotsky-Shearer, and Tracy M. Carter
University of Miami

Research has shown that early interventions are most successful when they have a comprehensive focus that is individualized to children's needs. The present study employed a person-centered approach to identify profiles, or subgroups, of children displaying early patterns of peer play behaviors in an ethnically and linguistically diverse Head Start program, and examined the academic trajectories of these children during one school year. Four profile groups were identified, and analyses revealed that these profiles were invariant across ethnicity and dual language learner status. Most children were represented in a group who engaged in behaviors that facilitated peer interactions. These children had the highest academic skills across the preschool year. Interestingly, children in a profile characterized by a combination of play interaction skills and play disruption had the second highest academic skills throughout the year compared with children in a profile characterized by below-average play interaction skills but low disruptive behavior during play. A small number of children were represented in a profile characterized by low interactive, disconnected, and high disruptive behavior with peers and had the lowest academic skills throughout the year. The mean differences in academic skills across profiles of peer play behaviors remained the same across the year. These findings have implications for future research and educational practice surrounding the role of peer play in the Head Start classroom.

Keywords: Head Start, peer play behaviors, school readiness, latent profile analysis, whole-child approach

Children living in poverty are at increased risk for exposure to environmental stressors and limited access to adequate resources (e.g., family stress, lack of desirable housing, exposure to community violence; G. J. Duncan, Brooks-Gunn, & Klebanov, 1994). Experiencing these multiple stressors places children at additional risk for difficulties adjusting to formal schooling, often leading to poor academic achievement, particularly compared with their middle- and high-income peers (O. Lee & Buxton, 2010). Unfortunately, evidence suggests that the achievement gap between low-income and high-income students is continually increasing (Reardon, 2013). Research identifying and promoting emergent competencies, how they vary among children, and how they are associated with academic learning is needed to inform how to best protect these vulnerable children from experiencing difficulties upon entry into school (Barbarin, 2007; Kagan, Moore, & Bredekamp, 1995).

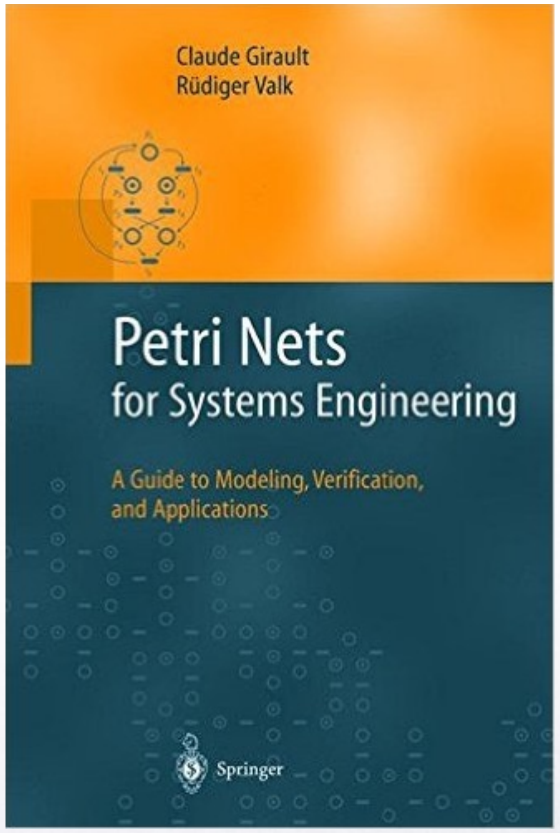
Early intervention programs, such as Head Start, have the potential to alleviate the risks of poverty associated with poor school adjustment and achievement (V. E. Lee & Burkam, 2002; Shonkoff & Phillips, 2000). Research has shown that such interventions are most successful when they are comprehensive and flexible to meet each individual child's specific needs (Ramey & Ramey, 1998). Head Start is the largest federally funded early childhood program in the United States serving predominantly low-income children. Since its inception, Head Start's comprehensive intervention approach has focused on promoting the development of the whole child (Zigler & Bishop-Josef, 2006). Head Start provides educational, health, and social services to low-income children and their families with the goal of promoting children's development across multiple domains, including cognitive, social, emotional, and physical. Specifically, Head Start performance standards mandate that classrooms must utilize social interactions to support each child's cognitive and language skills by "using various strategies including experimentation, inquiry, observation, play and exploration" (1304.21 [a] [4] [i]; U.S. Department of Health and Human Services, 2006, p. 70).

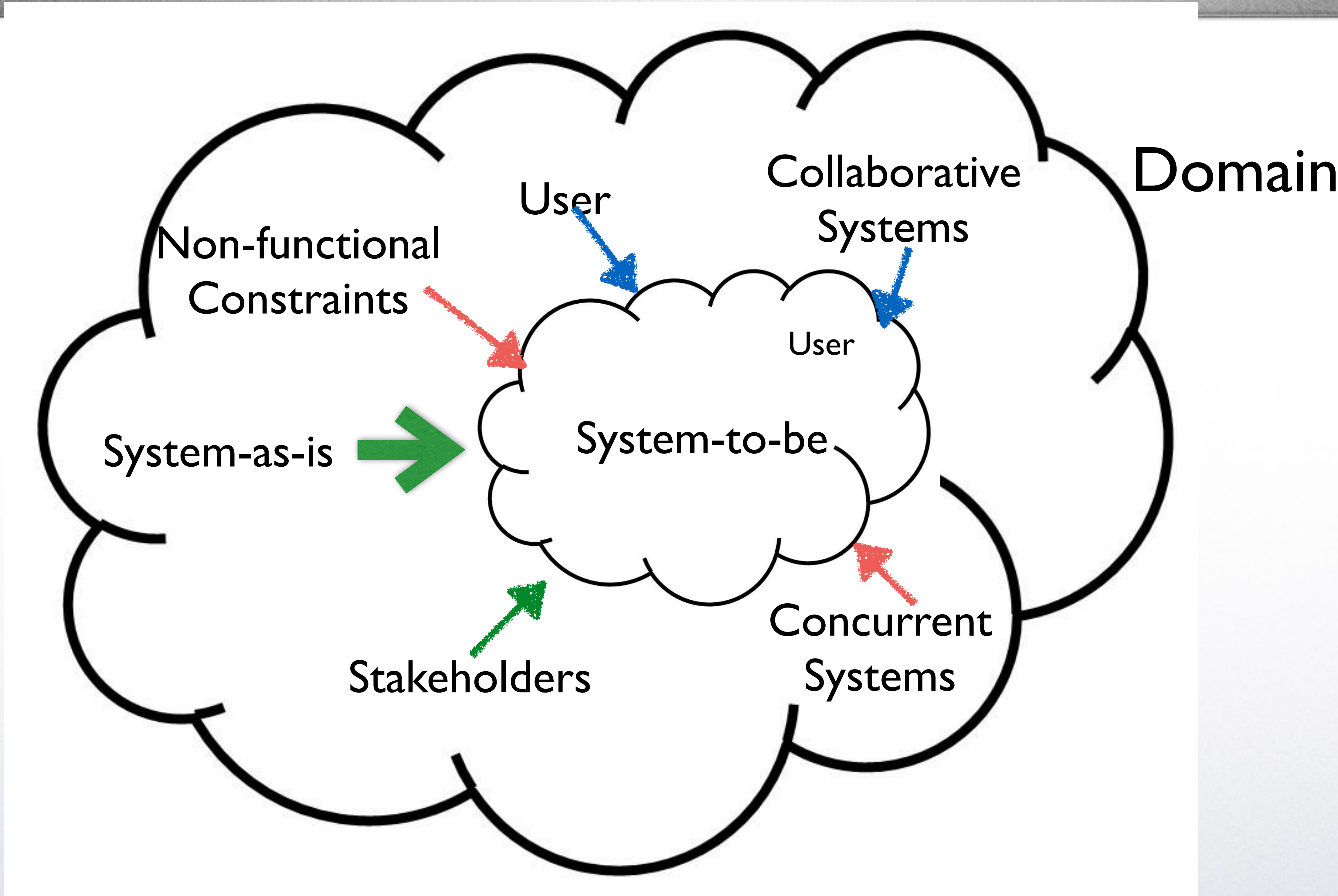
Developmental theory and research suggest that the preschool classroom is a naturally occurring context for peer play through which children acquire knowledge and skills (Coplan & Arbeau, 2009; D. G. Singer, Golinkoff, & Hirsh-Pasek, 2006). A growing body of research conducted in Head Start classrooms provides concurrent and longitudinal evidence for the positive associations between behaviors that facilitate peer play and language and mathematics skills in preschool, kindergarten, and third grade (Bulotsky-Shearer, Bell, Romero, & Carter, 2012; Fantuzzo, Sekino, & Cohen, 2004; Hampton & Fantuzzo, 2003; Sekino, 2006). In addition, research has also found that behaviors that interf[illegible] peer play are associated [illegible]th poorer academic

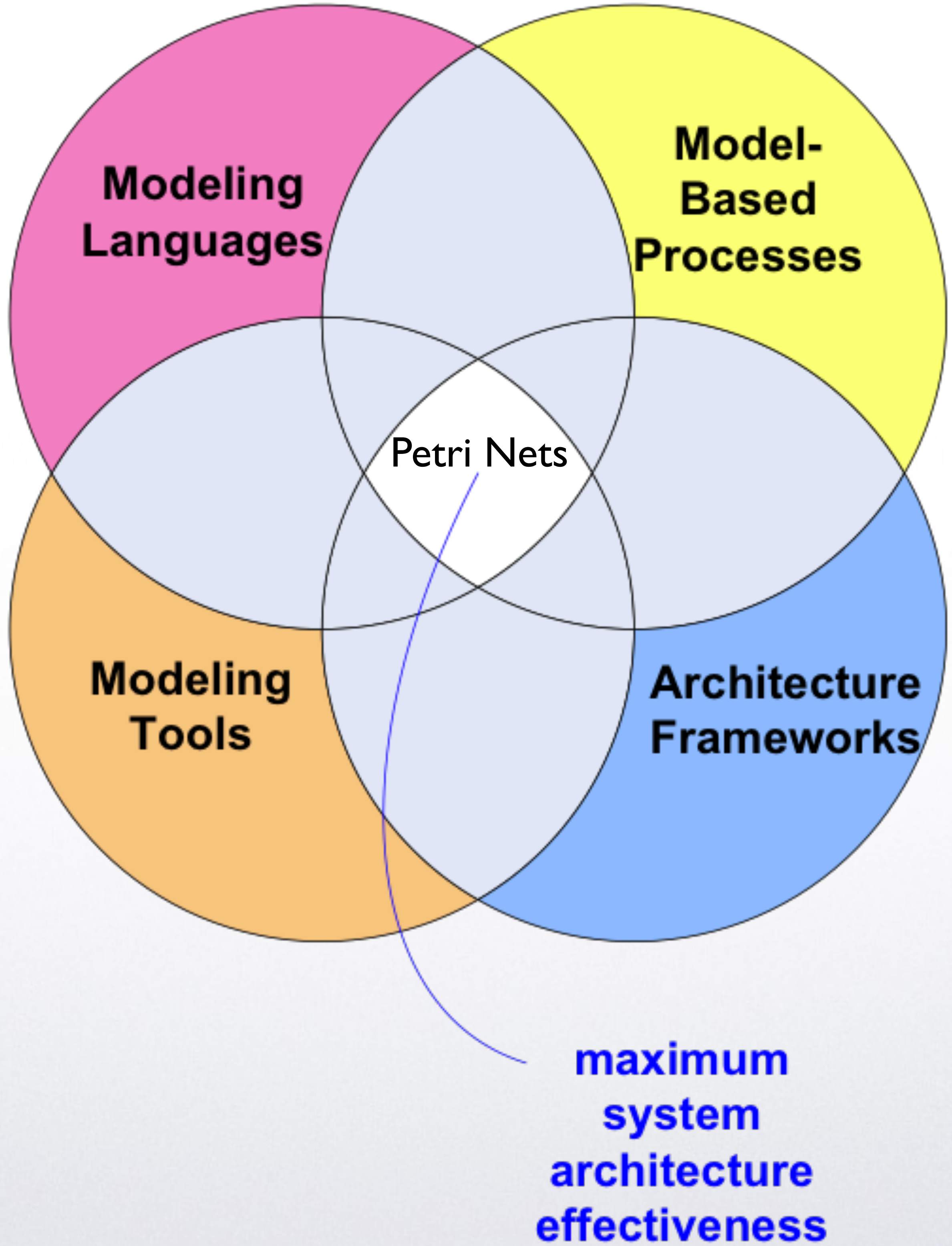
This article was published Online First January 11, 2016.
Elizabeth R. Bell, Daryl B. Greenfield, Rebecca J. Bulotsky-Shearer, and Tracy M. Carter, Department of Psychology, University of Miami.
Correspondence concerning this article should be addressed to Elizabeth R. Bell, who is now at Accelerate Learning Inc., 5700 Richmond Avenue, Suite 1025, Ho[illegible]on, TX 77056. E-mail: ebell@acceleratelearning.com

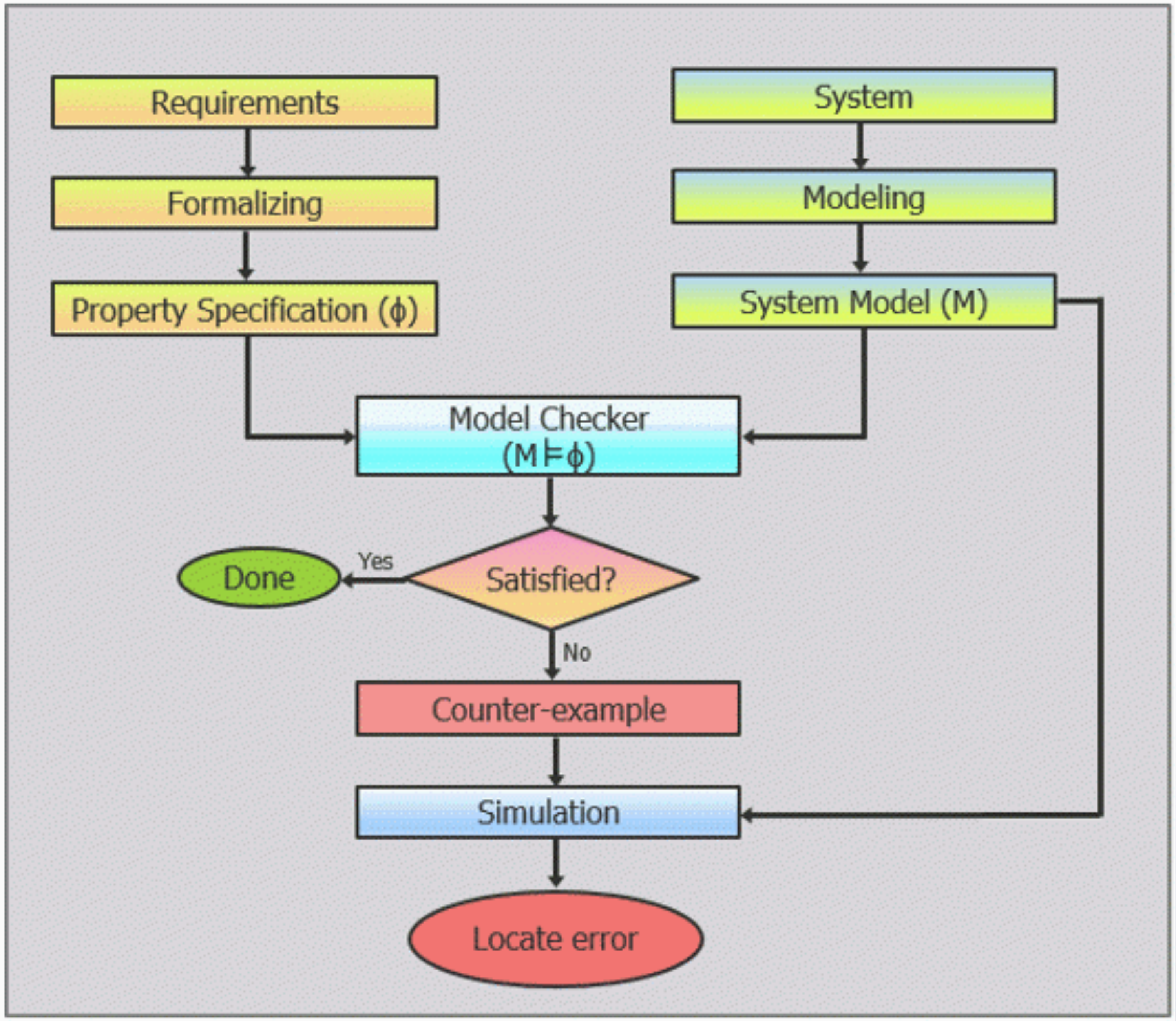
Artigo Final: teremos mais um milestone na aula que vem, onde vocês devem inserir a proposta de uso das RdP junto com a sua aplicação. A entrega está marcada para 14/12 e teremos alguns minutos no final da aula para discussão. O artigo final fica para o dia 21/12.

Part II. Modelling

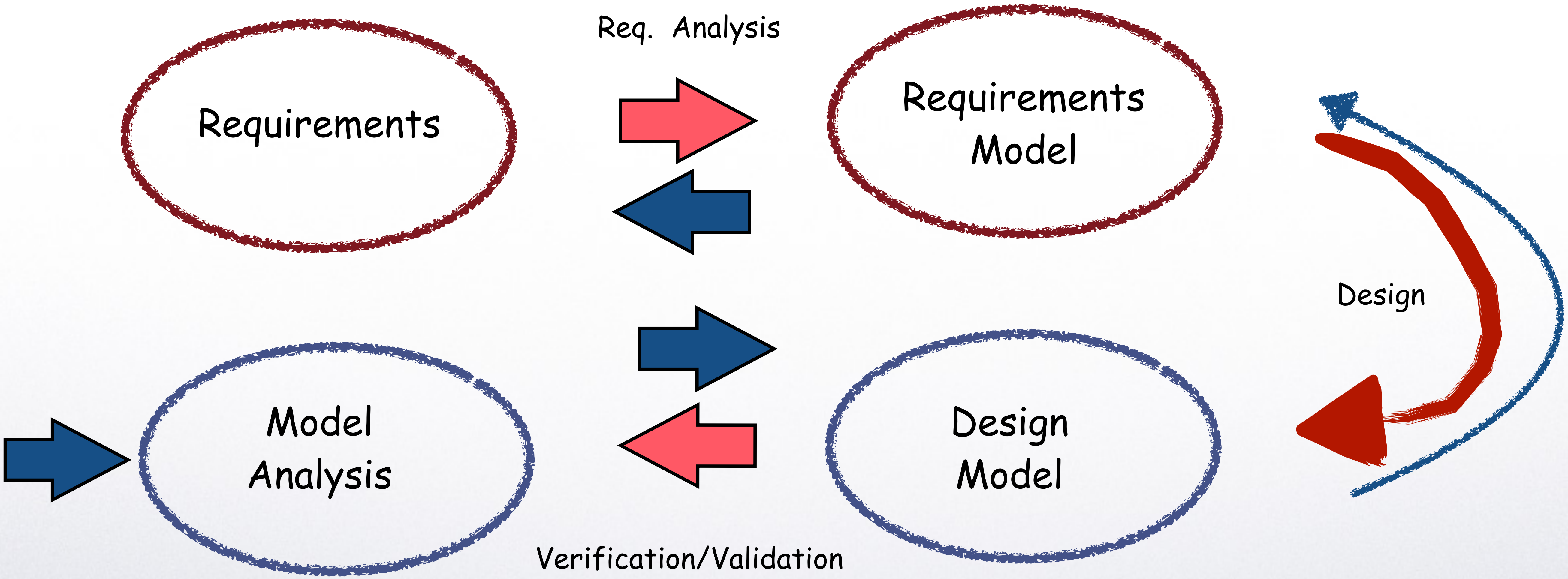


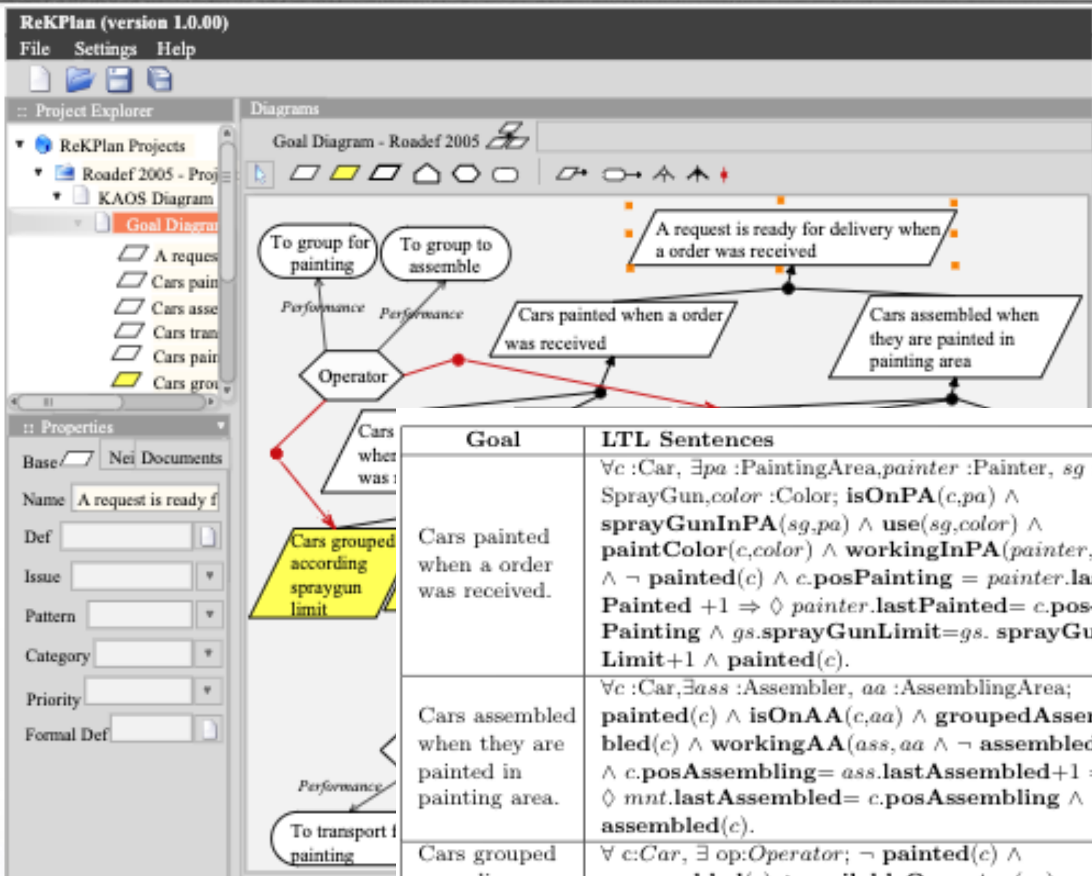

Domain
User
Collaborative Systems
Non-functional Constraints
User
System-as-is
System-to-be
Stakeholders
Concurrent Systems


Modeling Languages
Model-Based Processes
Petri Nets
Modeling Tools
Architecture Frameworks
maximum system architecture effectiveness

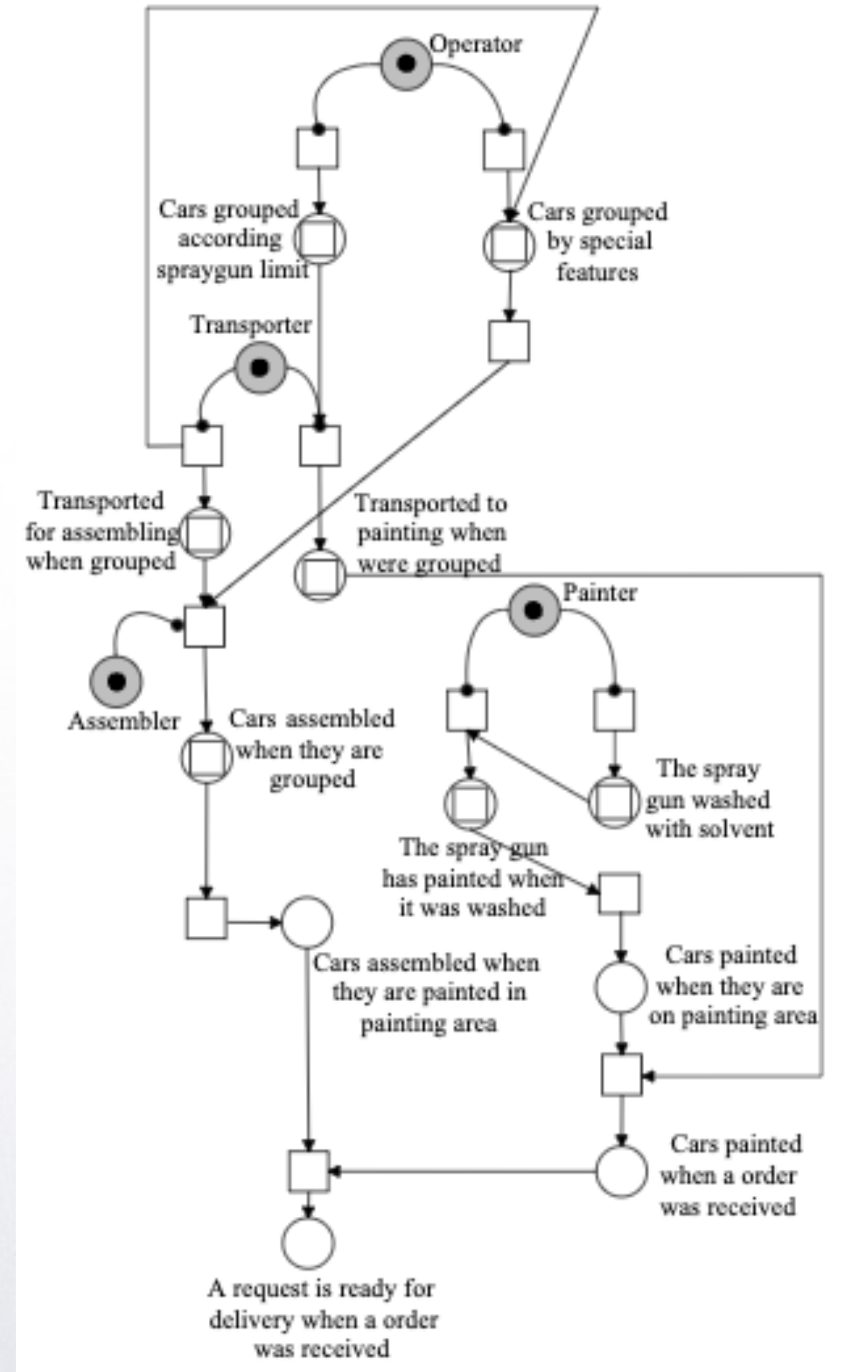

Requirements
Formalizing
Property Specification (ϕ)
System
Modeling
System Model (M)
Model Checker
(M⊨ϕ)
Done
Yes
Satisfied?
No
Counter-example
Simulation
Locate error

Use of Petri Nets in Design



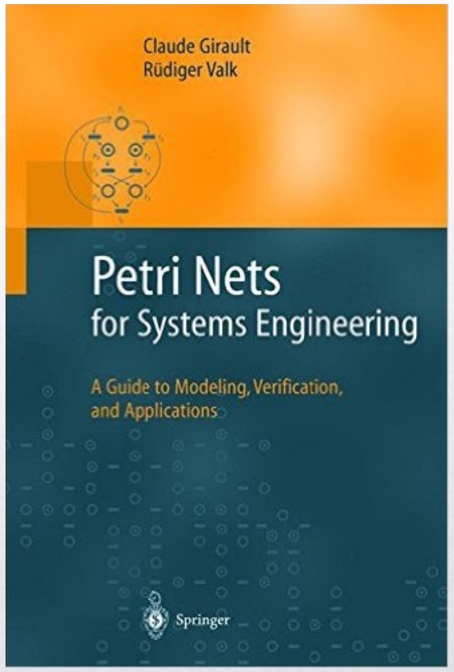


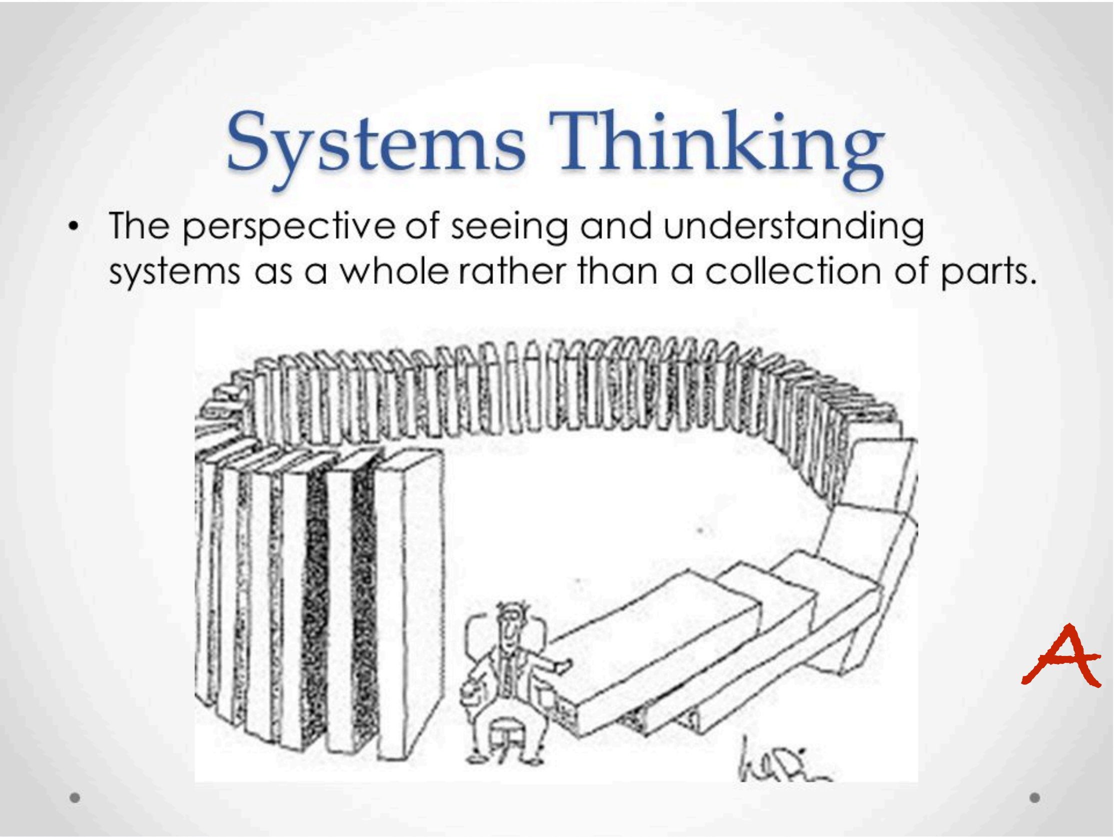
Goal	LTL Sentences
Cars painted when a order was received.	∀c :Car, ∃pa :PaintingArea, painter :Painter, sg : SprayGun, color :Color; **isOnPA**(c,pa) ∧ **sprayGunInPA**(sg,pa) ∧ **use**(sg,color) ∧ **paintColor**(c,color) ∧ **workingInPA**(painter,pa) ∧ ¬ **painted**(c) ∧ c.**posPainting** = painter.**lastPainted** +1 ⇒ ◊ painter.**lastPainted**= c.**posPainting** ∧ gs.**sprayGunLimit**=gs. **sprayGunLimit**+1 ∧ **painted**(c).
Cars assembled when they are painted in painting area.	∀c :Car, ∃ass :Assembler, aa :AssemblingArea; **painted**(c) ∧ **isOnAA**(c,aa) ∧ **groupedAssembled**(c) ∧ **workingAA**(ass, aa ∧ ¬ **assembled**(c) ∧ c.**posAssembling**= ass.**lastAssembled**+1 ⇒ ◊ mnt.**lastAssembled**= c.**posAssembling** ∧ **assembled**(c).
Cars grouped according spray gun limit.	∀ c:Car, ∃ op:Operator; ¬ **painted**(c) ∧ ¬ **assembled**(c) ∧ **availableOperator**(op) ∧ c.**posPainting** = 0 ⇒ ◊ **groupedPaint**(c).
Cars grouped by special features.	∀c :Car, ∃ op:Operator; **painted**(c) ∧ **availableOperator**(op) ∧ ¬ **groupedAssembled**(c) ⇒ ◊ **groupedAssembled**(c).
Transported to painting when were grouped.	∀c :Car, ∃tra :Transporterr, ∃pa :PaintingArea, sg :SprayGun; **groupedPaint**(c) ∧ **availableTransporter** (tra) ∧ ¬ **painted**(c) ∧ ¬ **isOnPA**(c,pa) ∧ pa.**currentPaint** < sg.**sprayGunLimit** ⇒ ◊ **isOnPA**(c,ap) ∧ pa.**currentPaint** = pa.**currentPaint** +1 ∧ c.**posPainting** = pa.**currentPaint**.
Transported for assembling when grouped.	∀ c:Car, ∃tra:Transporter, aa :AssemblingArea; ¬ **assembled**(c) ∧ ¬ **isOnAA**(c,aa) ∧ **availableTransporter**(tra) ∧ **groupedAssembled**(c) ⇒ ◊ **isOnAA**(c,aa) ∧ aa.**currentAssembled** = aa.**currentAssembled** +1 ∧ c.**posAssembling** = aa.**currentAssembled**.
The spray gun washed with solvent.	∀ sg :SprayGun, ∃ painter:Painter; ¬ **clean**(sg) ∧ **has**(painter, sg) ∧ painter.**qcarsPainted**> 0 ⇒ ◊ **clean**(sg) ∧ painter.**qcarsPainted**= 0.
The spray gun has painted when it was washed.	∀ c:Car, sg:SprayGun, ∃ painter:Painter; **has**(painter, sg) ∧ **clean**(sg) ⇒ ◊ ¬ **clean**(sg).



Matching by construction

1. Create the set of places used for the net model.
2. Design constraints describing the behaviour and the structure of the solution, which ensure at least the safety properties of the specification.
3. Add all transitions that do not violate the constraints.
4. Prove the dynamic properties.

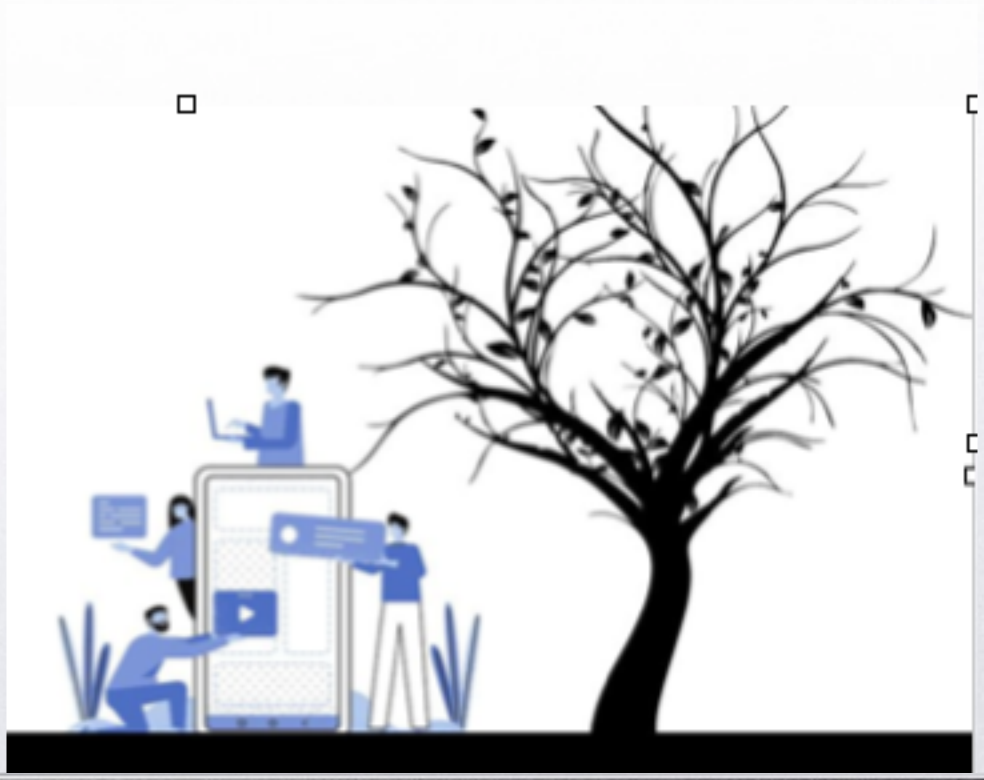




Look at the forest first
and refine,

A false dilema!?

⟷

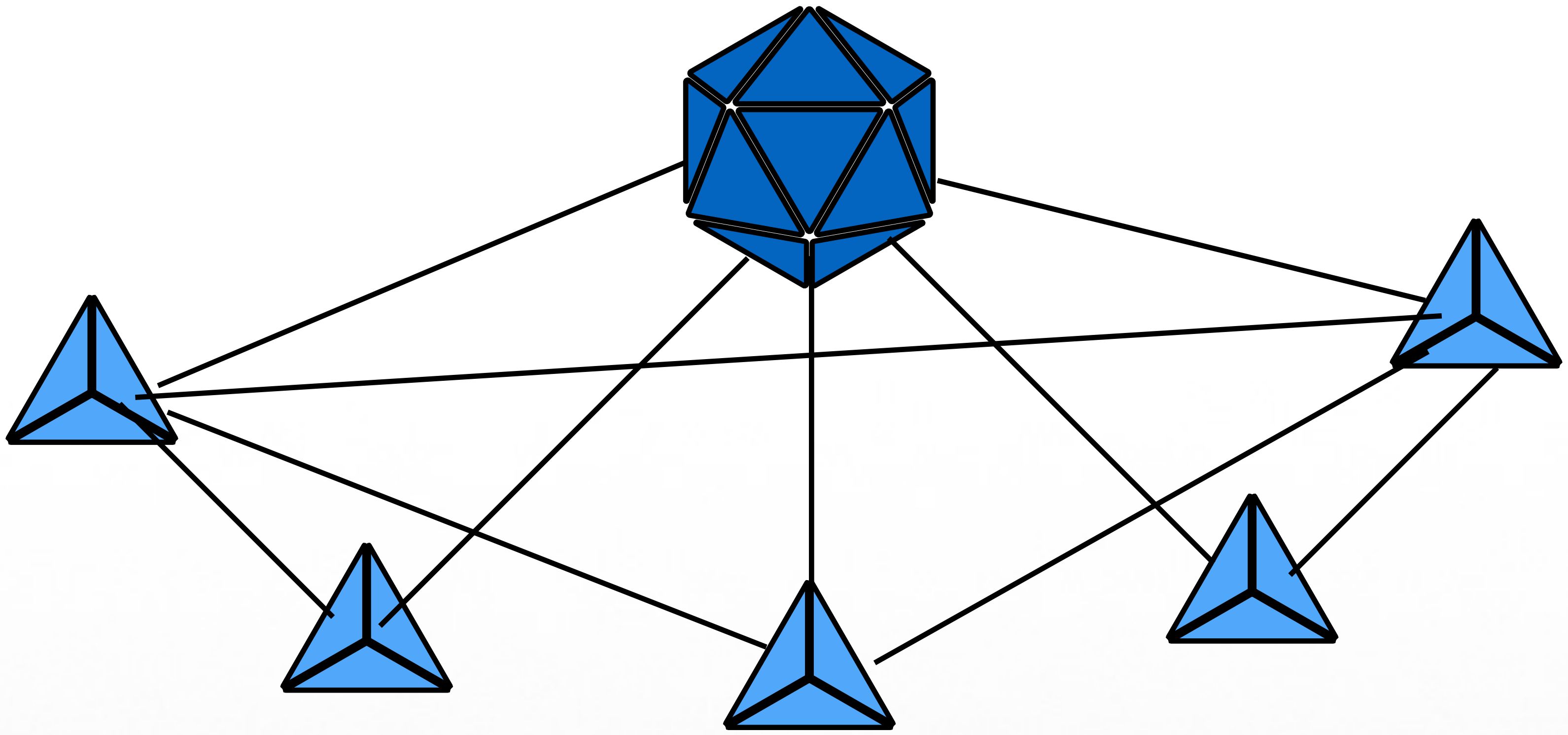


Look at the trees
and compose.

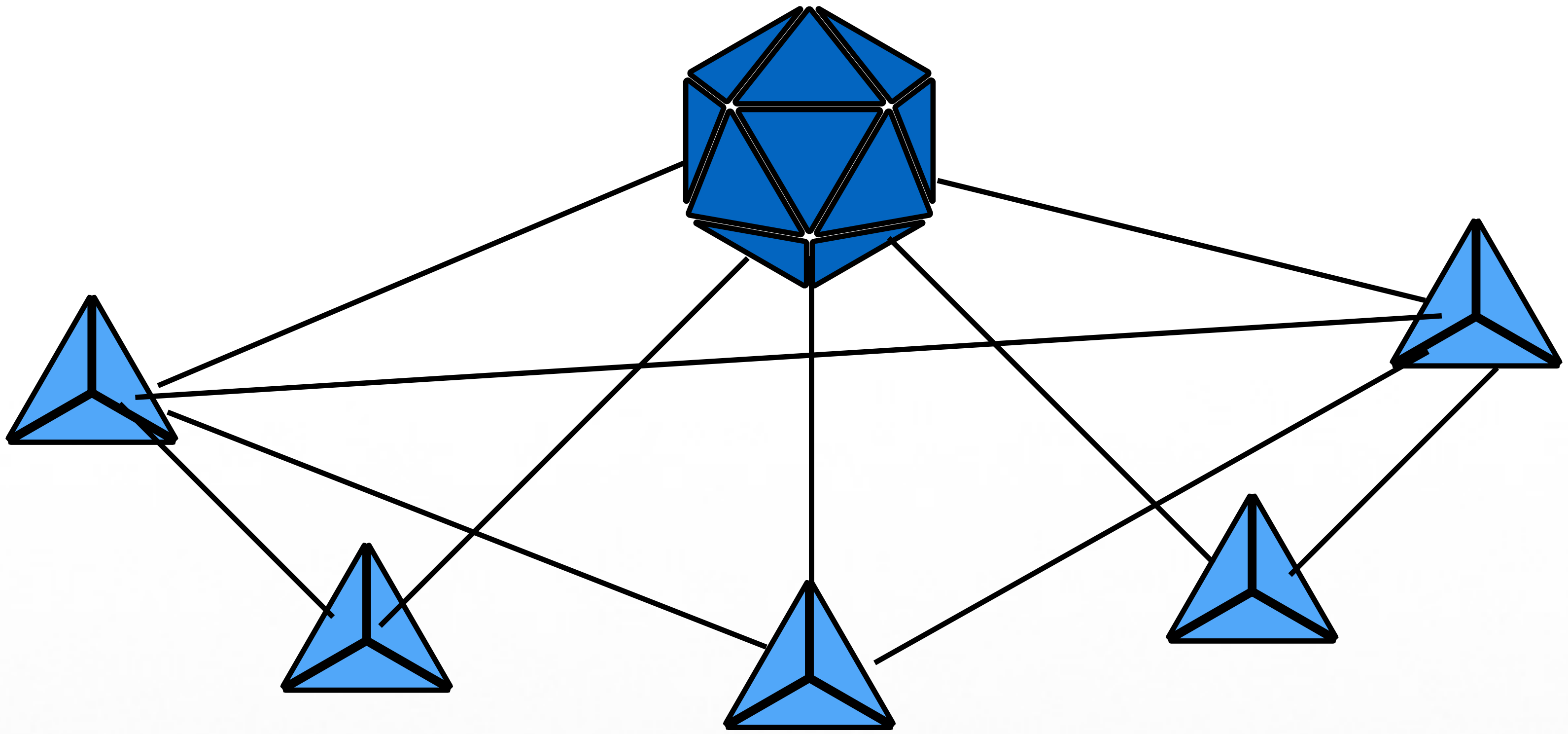
A false dilema!?

Actually, choosing a state-space or events development does not detach the process from the system approach, just orient priorities.

We advocate a top-down and structured approach, starting with high-level Modeling dividing the system into communicating subsystems, and ending with low-level modeling defining the protocol of a subsystem.

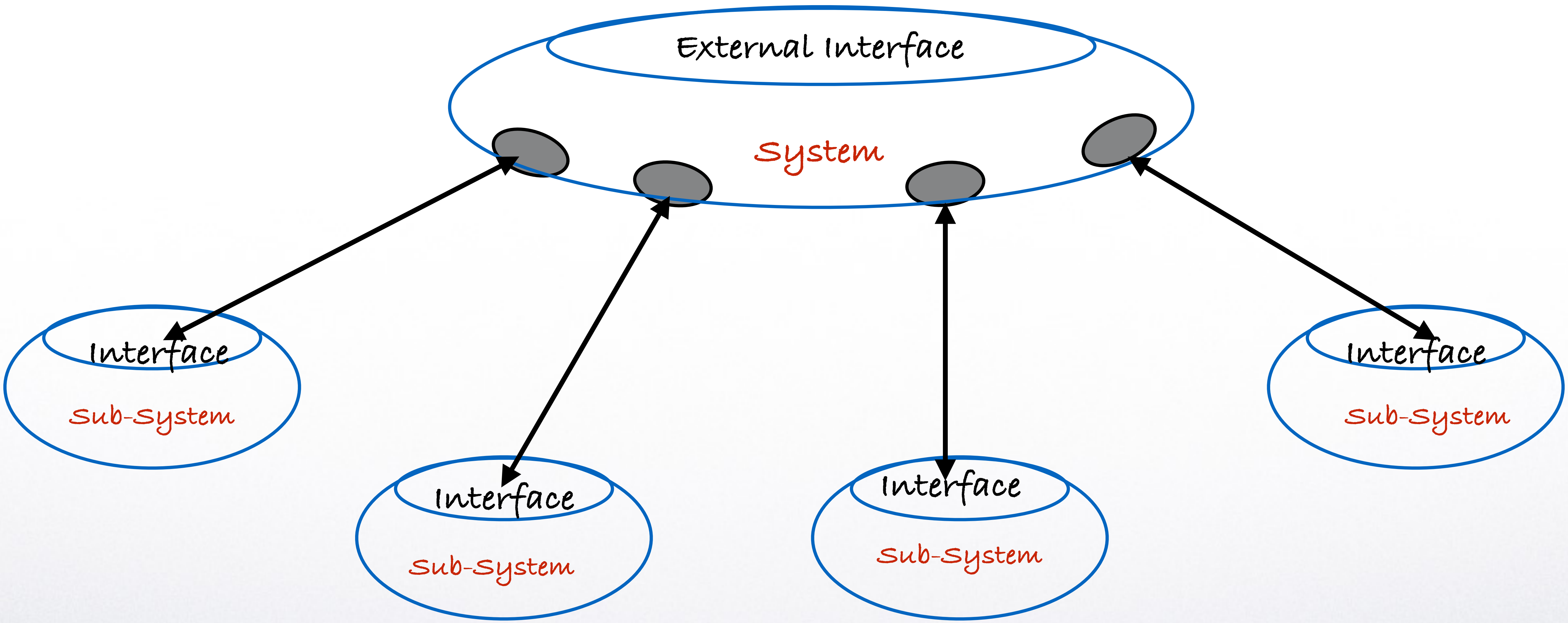


From the complete net model one can derive the protocol, which is often too complicated to be of any value. However, after abstracting from internal communication, retaining only a few essential actions, the system often has to satisfy a simple protocol. The notion of branching bisimilarity can be used to verify that the complete system obeys this protocol.



Physical X Functional Decomposition

The System of Systems (SoS) Approach



9

An Object-Oriented Approach to the Design of Flexible Manufacturing Systems *

J. Reinaldo Silva †
Computer System Group, University of Waterloo, Waterloo, Canada
e-mail: reinaldo@csg.uwaterloo.ca

H. Afsarmanesh
Computer System Department, University of Amsterdam, The Netherlands
email: hamideh@fwi.uva.nl

D.D. Cowan
Computer System Group, University of Waterloo, Waterloo, Canada
email: dcowan@csg.uwaterloo.ca

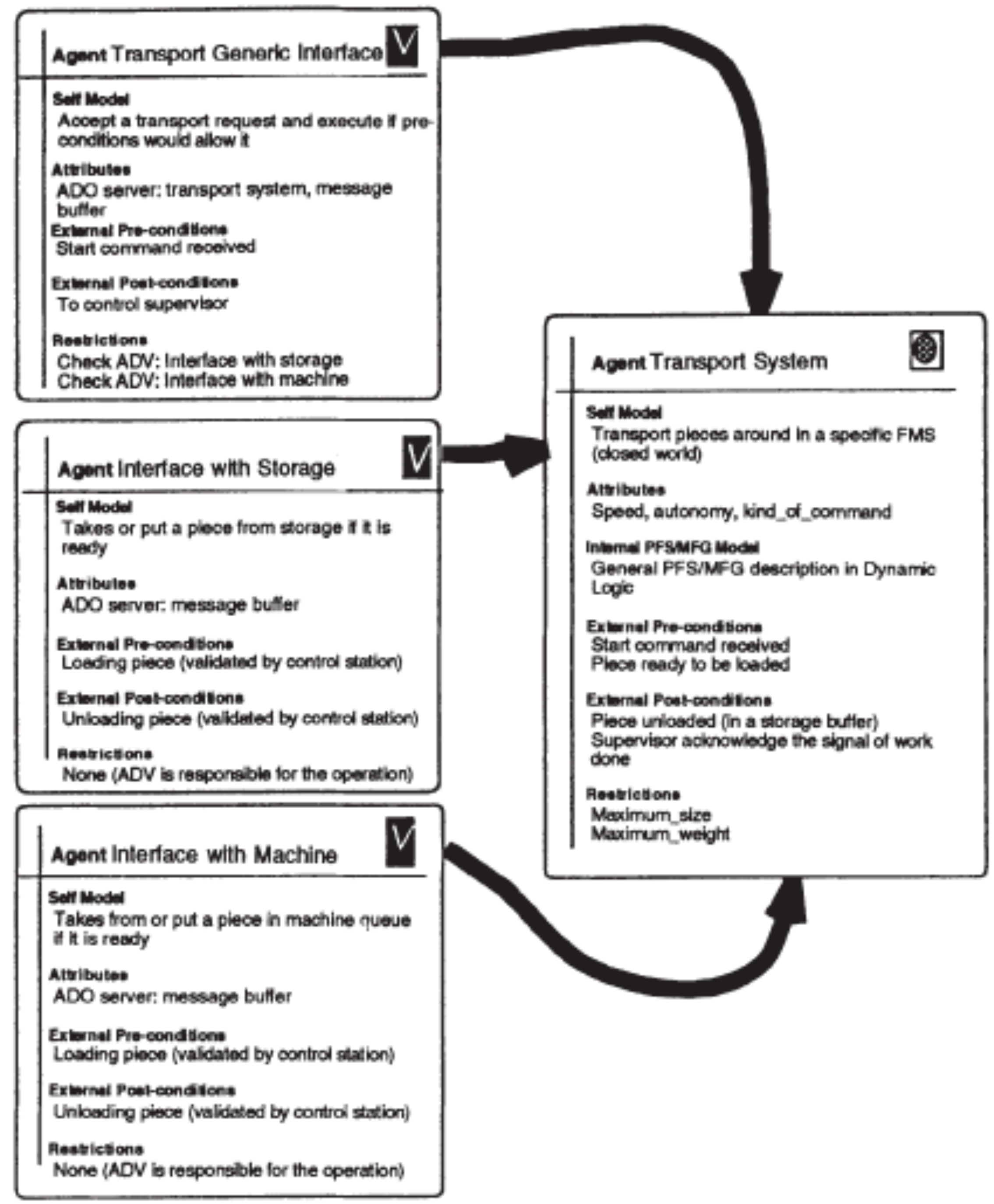
C.J P. Lucena
Computer Science Departement, PUC-Rio de Janeiro, Brazil
email: lucena@inf.puc-rio.br

Abstract

In this paper a hybrid top-down/bottom-up method that can be viewed as an extension of the traditional dynamic modeling technique using Petri Nets and Parametric Design is presented as an approach to the design of Flexible Manufacturing Systems. The resulting method supports a clear separation of functionality among the design objects by using the ADV/ADO object-oriented design framework. Thus, the designs as well as the general functional models can be reused. Comparing the method described in the paper with the object-oriented architecture introduced and employed in the PEER object-oriented database system suggests an implementation approach which can support the object clustering properties of ADV's and ADO's.

Keywords

OO-design, Flexible Manufacturing Systems, Petri Nets, Abstract Data Views, design reusability



IFIP-Balanced Automation Systems, 1995

Decomposing a system (stepwise) means defining subsystems (eventually independent systems) and (first) their interface (with the main system or with any other system).

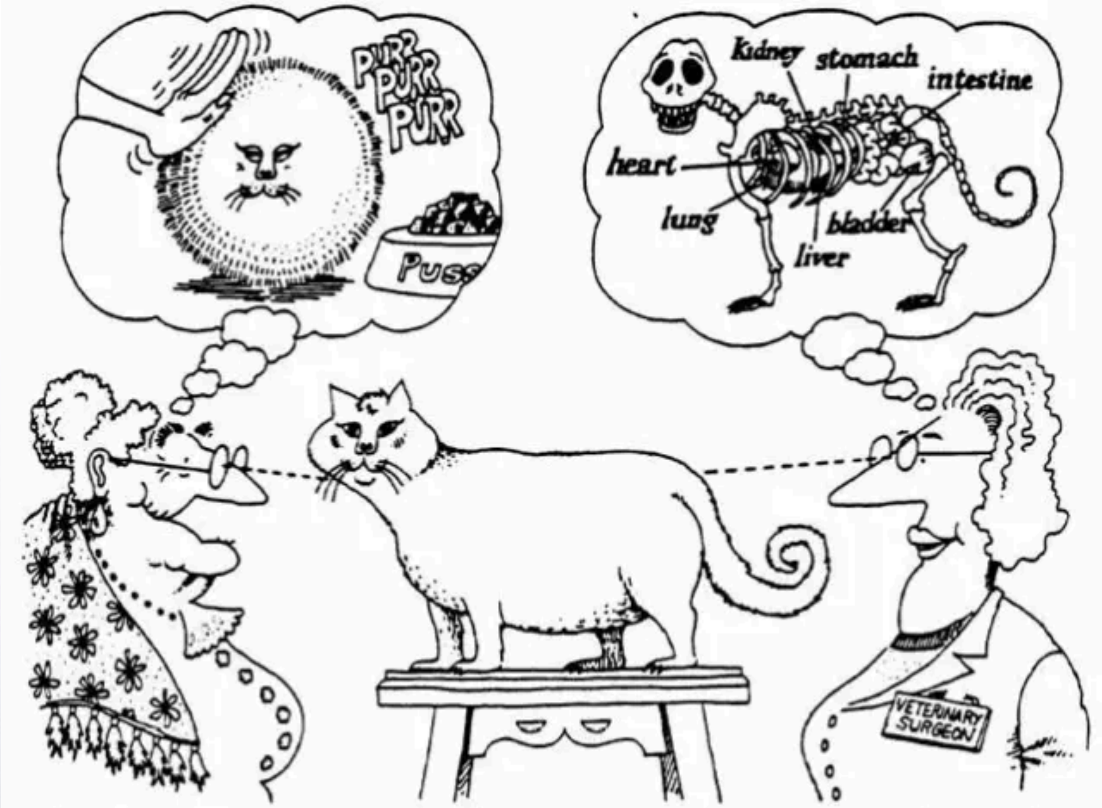
➡ Object-oriented Systems Design

A abordagem de objetos

Completeza comportamental

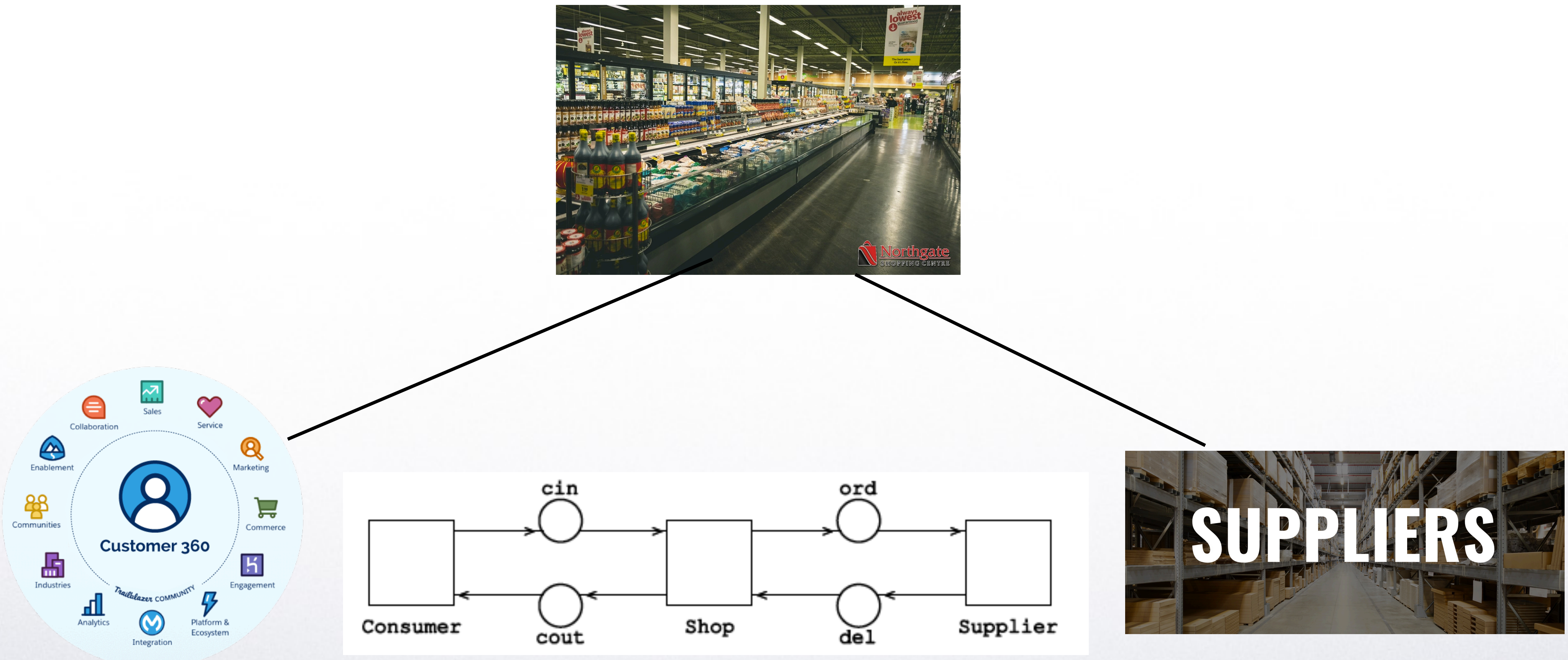
- separation of concerns
- encapsulation
- classification
- inheritance (single and multiple)
- polymorphism

It should be noticed that objects can be used indistinctly in a physical, functional, or hybrid decomposition.

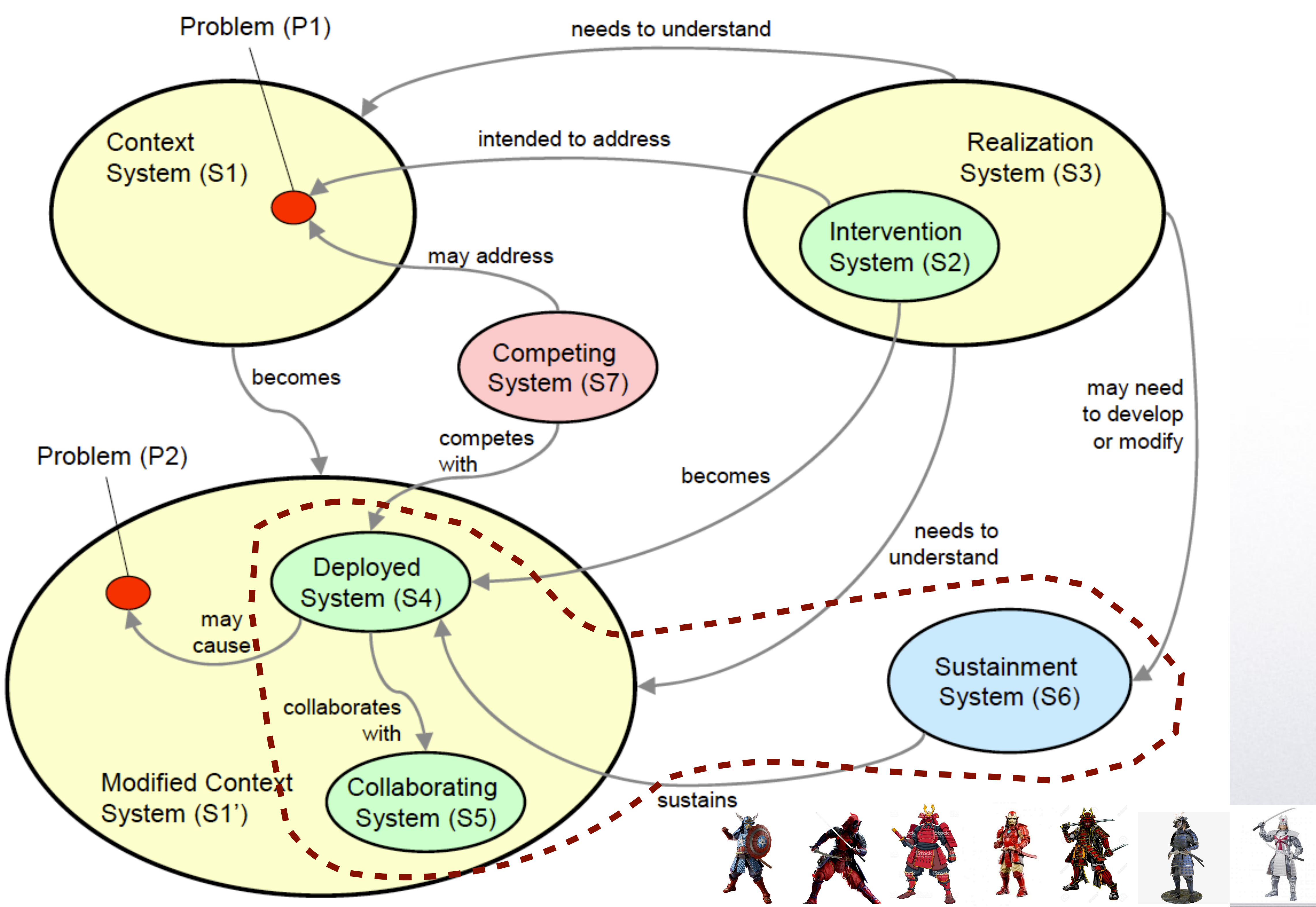


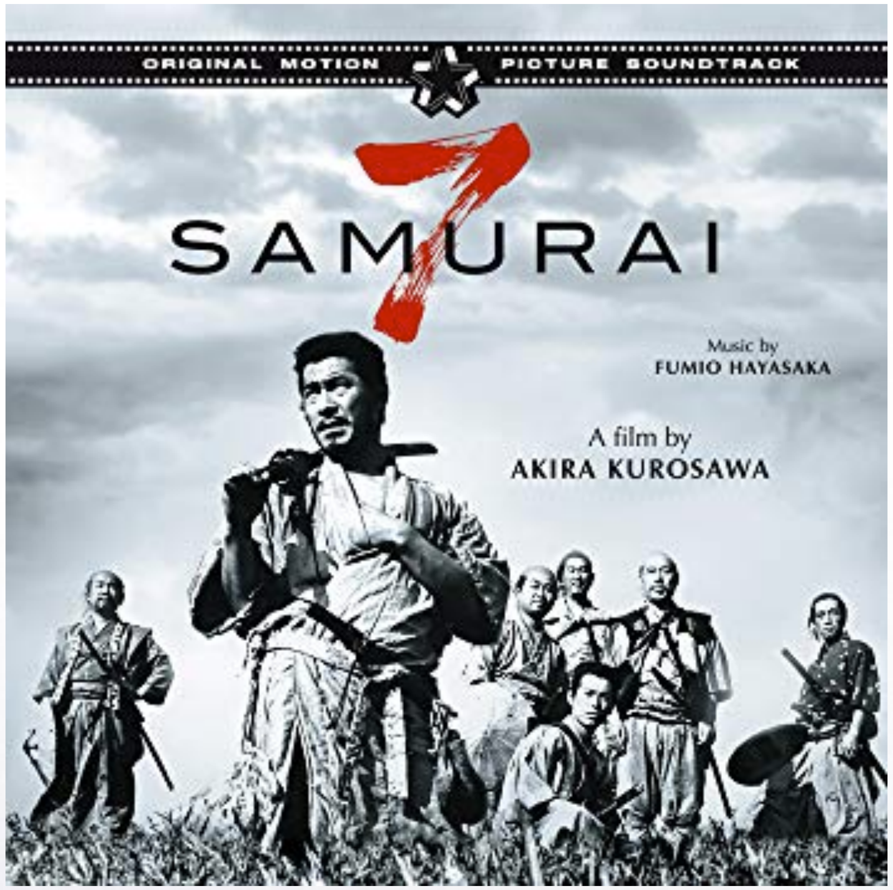
Abstraction focuses upon the essential characteristics of some object, relative to the perspective of the viewer.

Supermarket modeling



The Seven Samurais





The Seven Samurai of Systems Engineering: Dealing with the Complexity of 7 Interrelated Systems

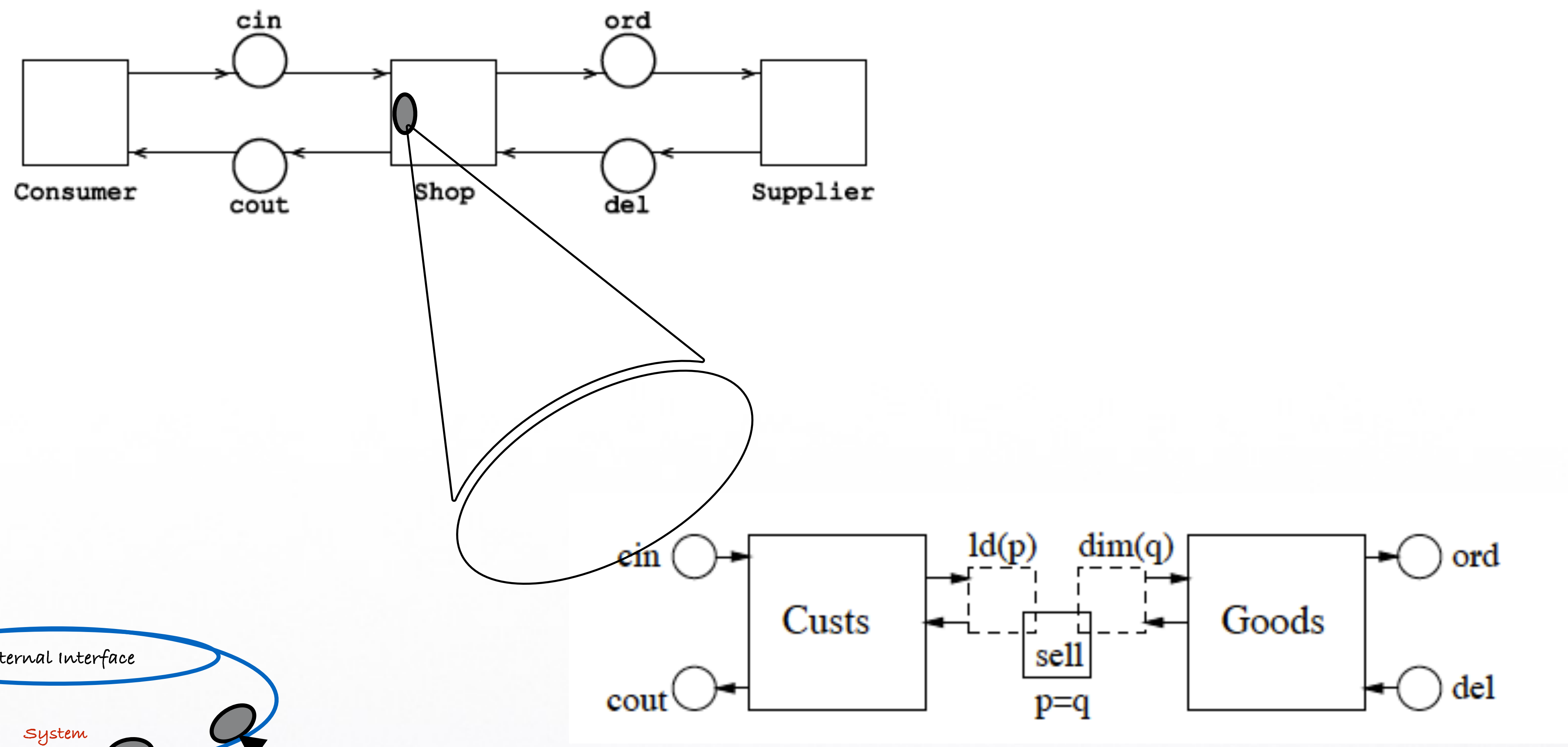
James N Martin
The Aerospace Corporation
M/S CH1-410, 15049 Conference Center Drive, Chantilly, VA 20151
James.Martin@incose.org

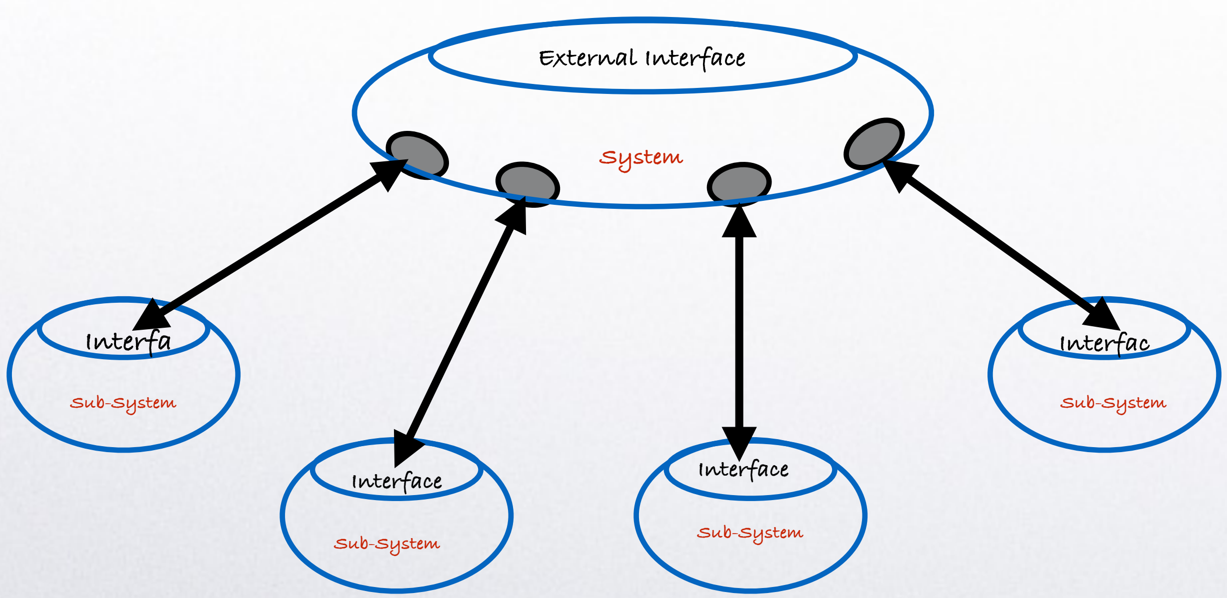
Abstract. There are seven different systems that must be acknowledged and understood by those who purport to do systems engineering. The main system to be engineered is the Intervention System that will be designed to solve a real or perceived problem. The Intervention System will be placed in a Context System and must be developed and deployed using a Realization System. The Intervention, when installed in the Context, becomes the Deployed System which is often different in substantial ways from the original intent of the Intervention. This Deployed System will interact with Collaborating Systems to accomplish its own functions. A Sustainment System provides services and materials to keep the Deployed System operational. Finally, there are one or more Competing Systems that may also solve the original problem and will compete for resources with your Deployed System. All seven systems must be properly reckoned with when engineering a system.

Introduction

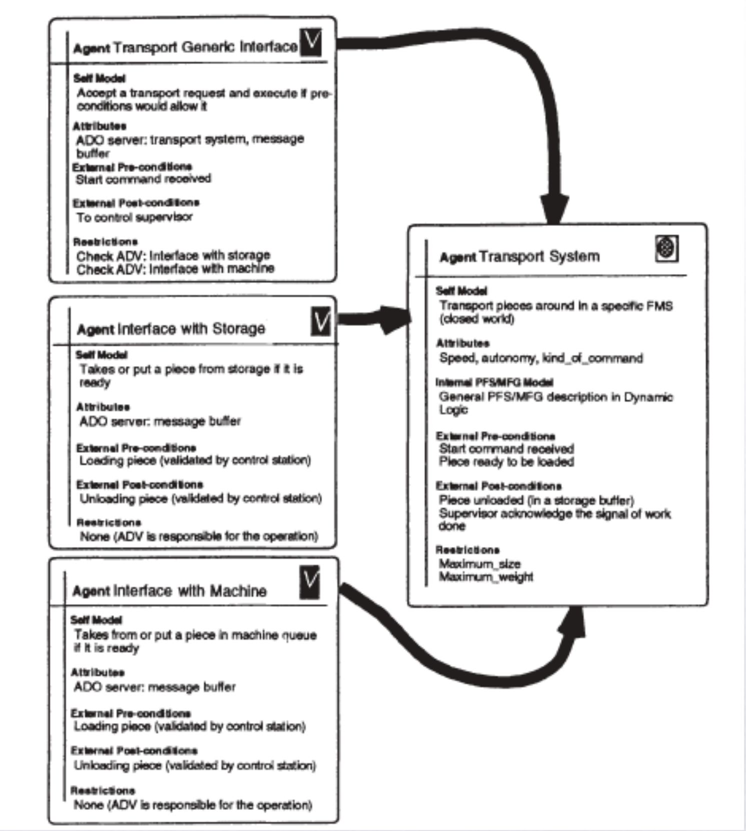
The Analogy. "*Shichinin No Samurai,*" the 1954 film classic directed by Akira Kursawa, is an apt illustration for the plight of the systems engineer. The Seven Samurai were the mighty warriors who became the seven national heroes of a small town. A poor village under attack by bandits recruits seven unemployed samurai to help them defend themselves. The notion of the "seven samurai" described in this paper illustrates the seven systems that are underemployed in the classical practice of systems engineering. When these 7 Samurai are employed with proper consideration and enthusiasm, they will become the seven national heroes of your small town

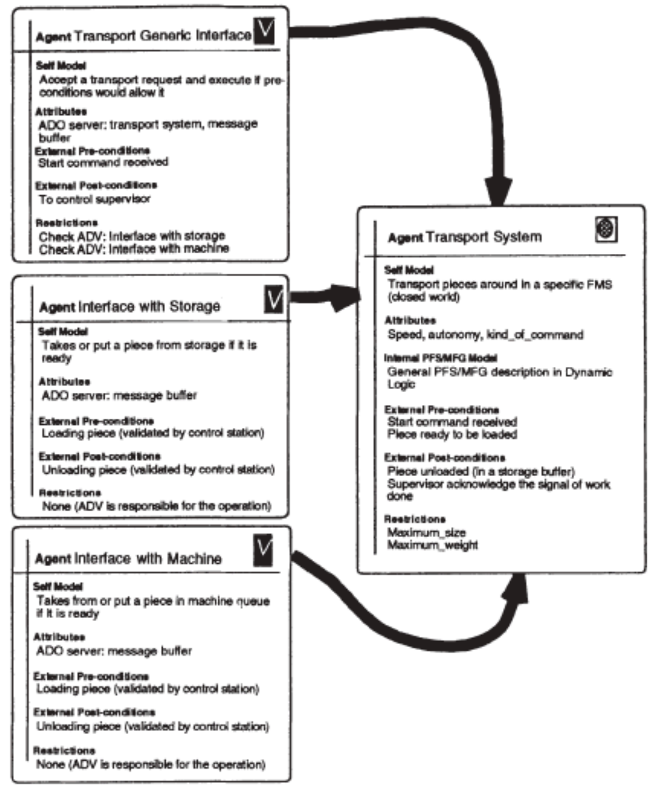
Presented at the 2004 Symposium of the International Council on Systems Engineering (INCOSE)
Received "Best Paper" Award

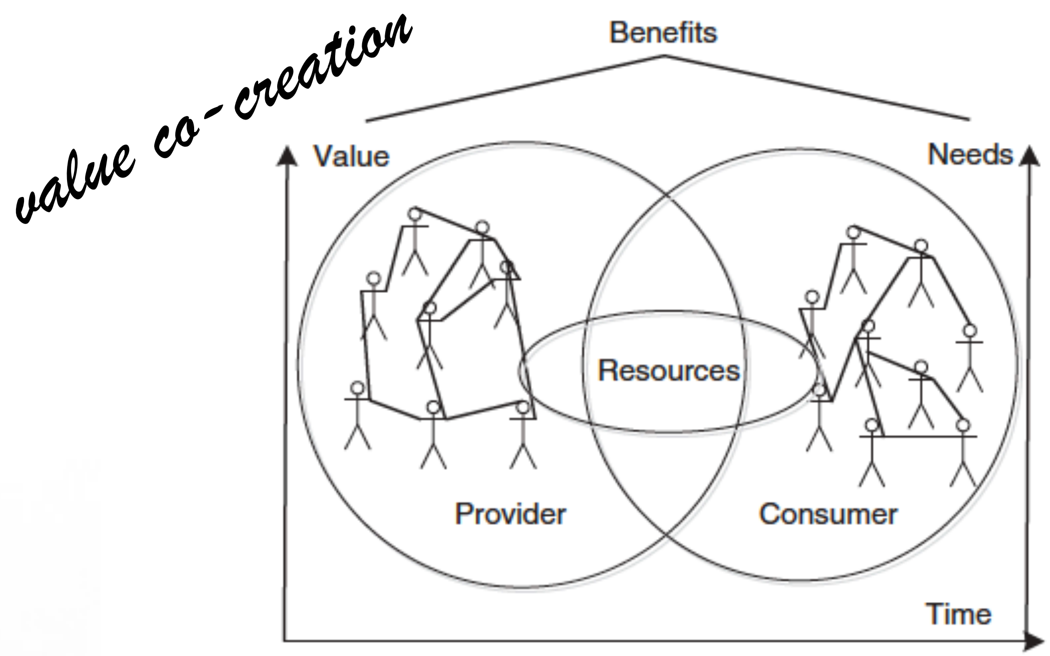

cin
ord
Consumer
cout
Shop
del
Supplier
cin
ld(p)
dim(q)
ord
Custs
Goods
sell
cout
p=q
del


External Interface
System
Interfa
Sub-System
Interface
Sub-System
Interface
Sub-System
Interfac
Sub-System

Sub-system (system) communication requires a specific “protocol" or transaction. Such transactions can be modeled as a net and denote a dynamic workflow.

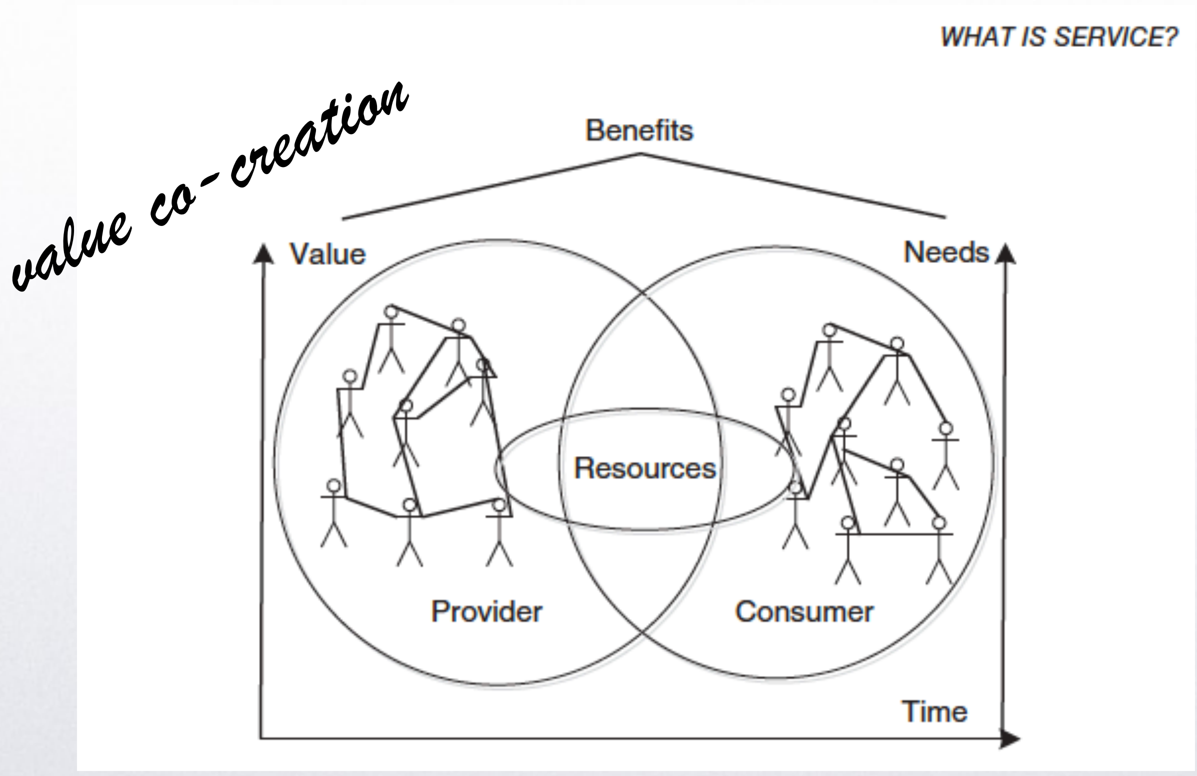




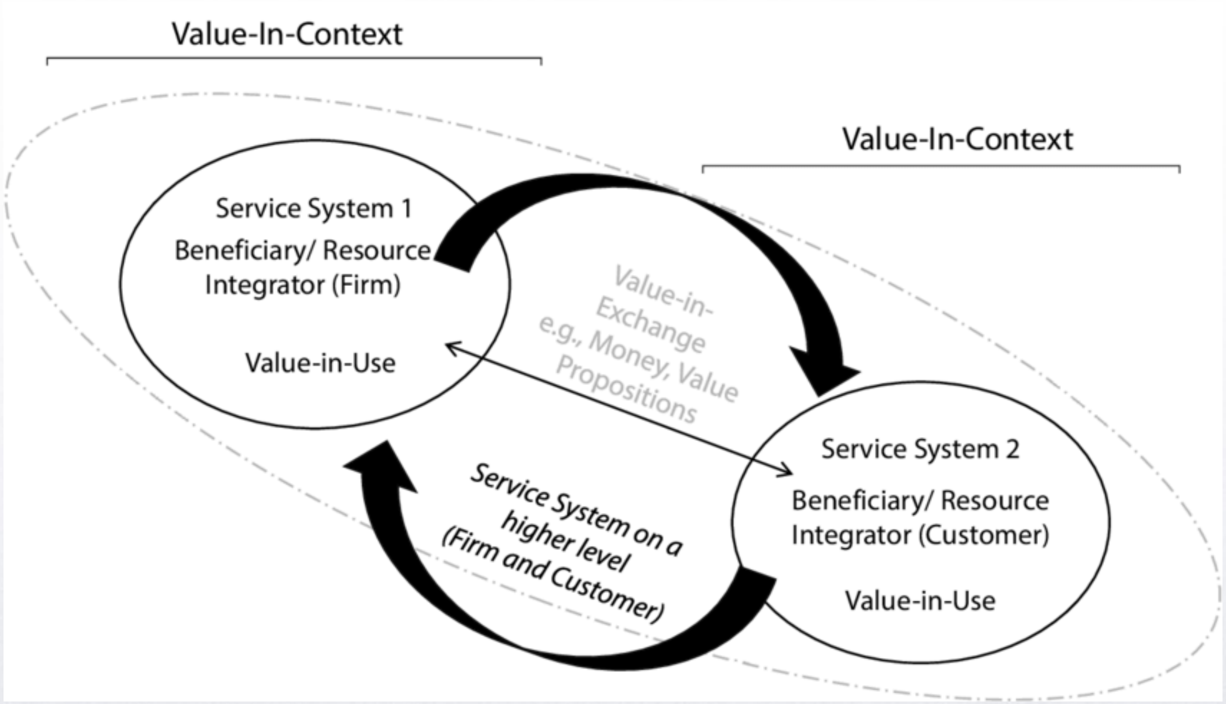


Modeling "services" is also a good example (like one of the supermarkets), where the "interaction protocol" is the target.

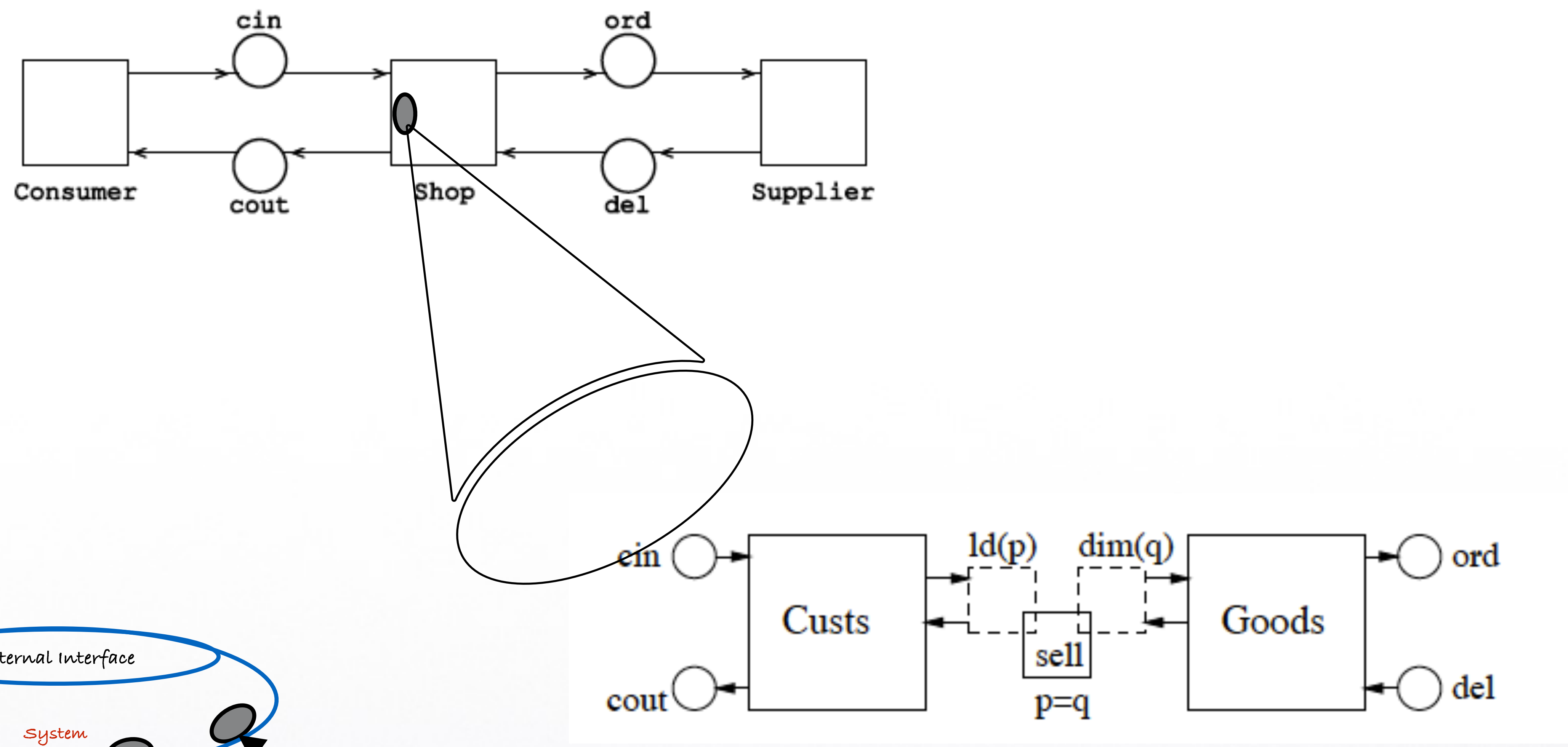
Modeling "services" implies a special protocol or transaction covering the interaction between the system and the user of the service (be it human or machine). It is called "value co-creation".

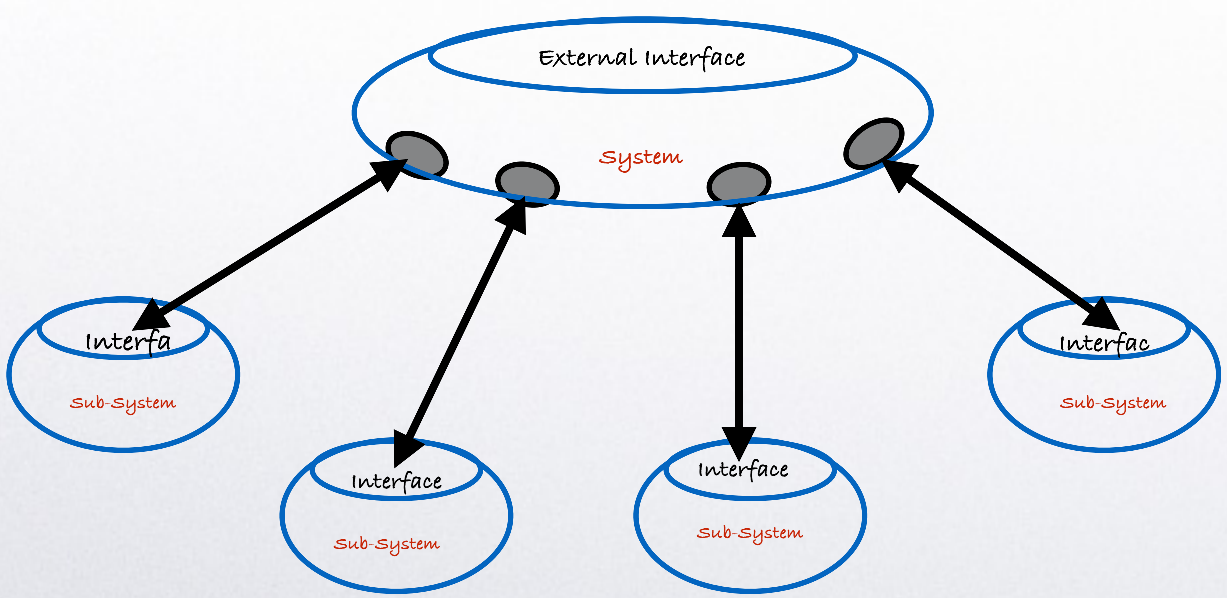


When modeling service, value co-creation is important because it is directly connected to the ROI (return of the investment) of Service Systems.



Protocols can be modeled by process algebra, which has matching with Petri nets. It is possible to describe a net locality (composed of pre and post conditions and a transition or step as a single protocol - see the textbook for further information).

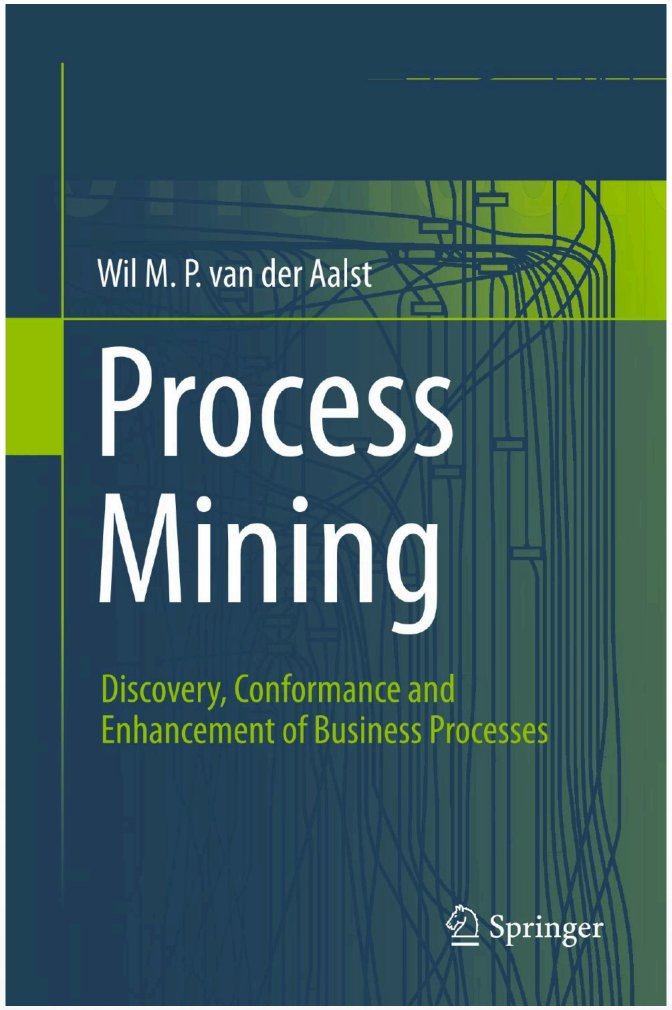


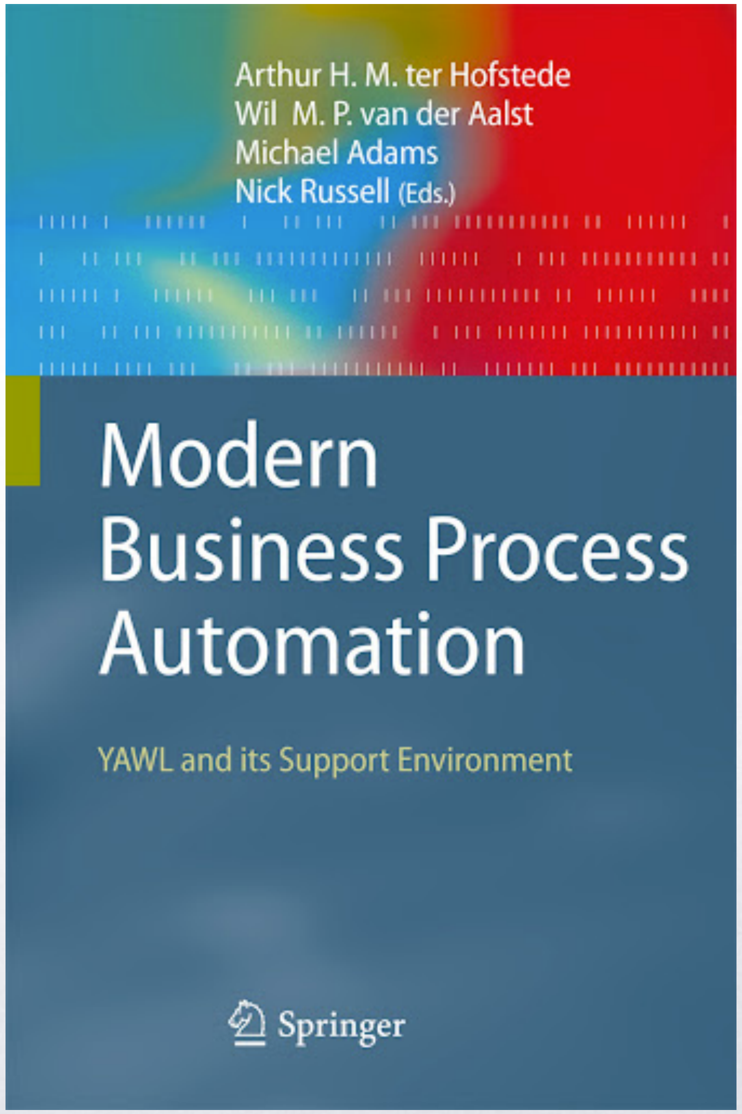


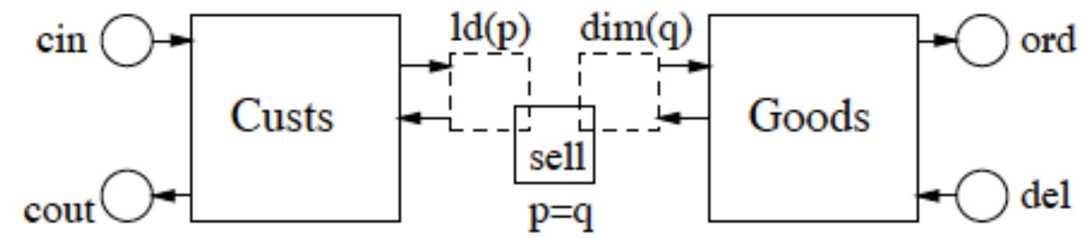
returning to the example...

The interaction modeled as Custs assumes that a customer comes to the supermarket and expend some money in exchange for some product. It is quite clear its importance to the business, even in this very simple example.

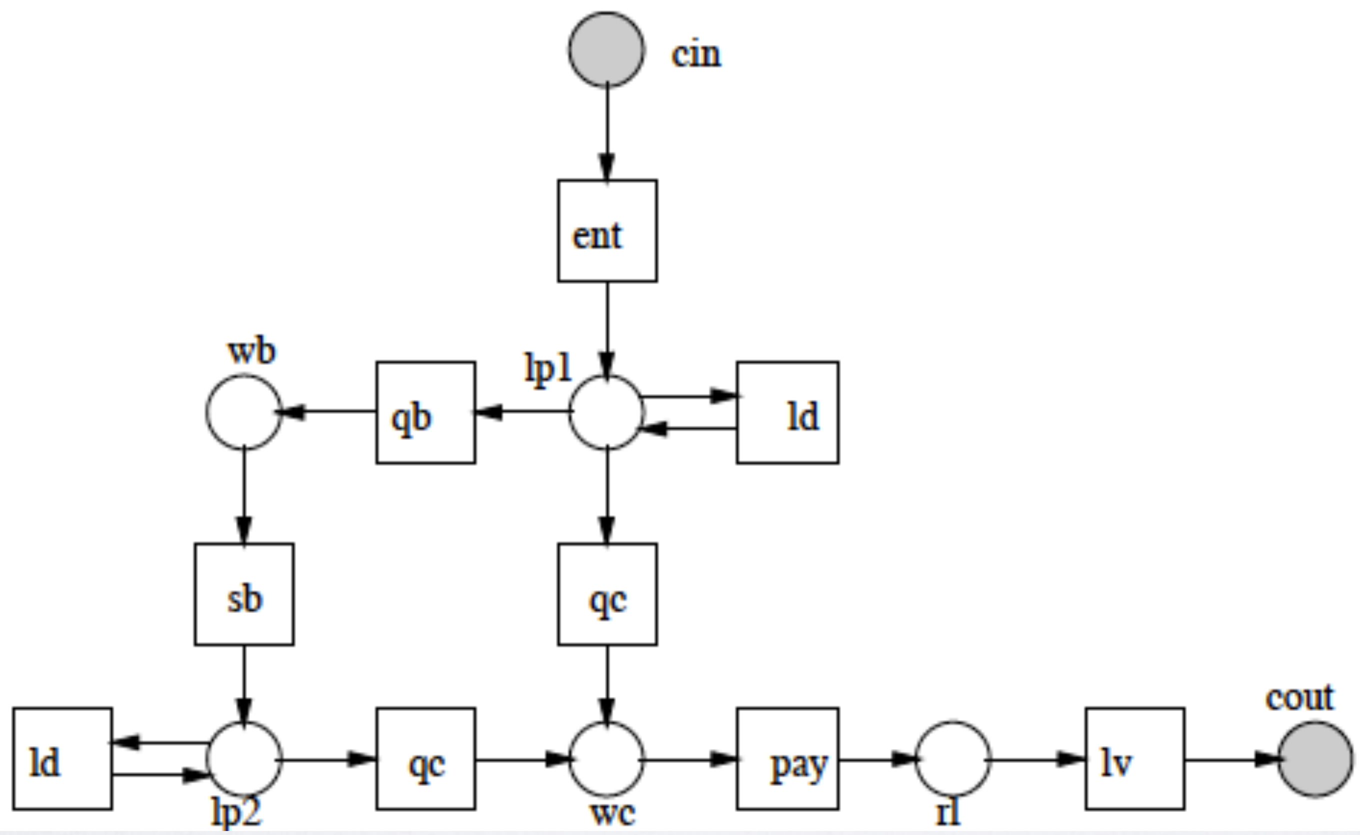
Wil van der Aalst







cin
cout
Custs
ld(p)
dim(q)
sell
p=q
Goods
ord
del



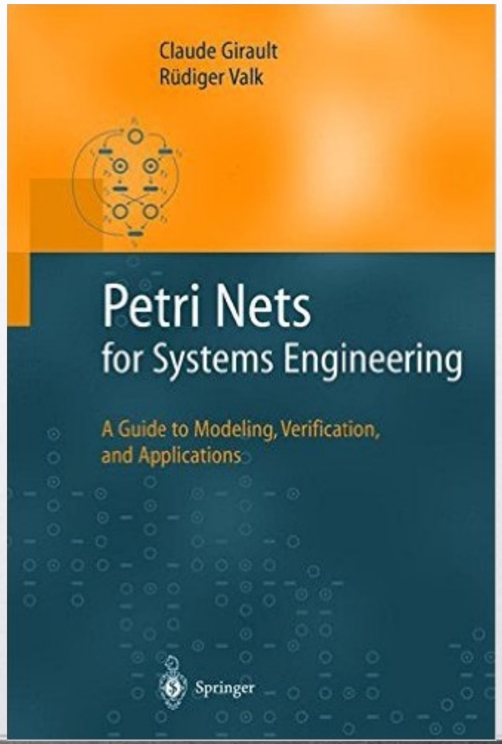
cin
ent
wb
qb
lp1
ld
sb
qc
ld
lp2
qc
wc
pay
rl
lv
cout

The construction method based o events explores the formalism of Petri nets but is also based on intuition and interpretation of the physics (or archetype) system. The necessary step after the modeling would be validation/verification.

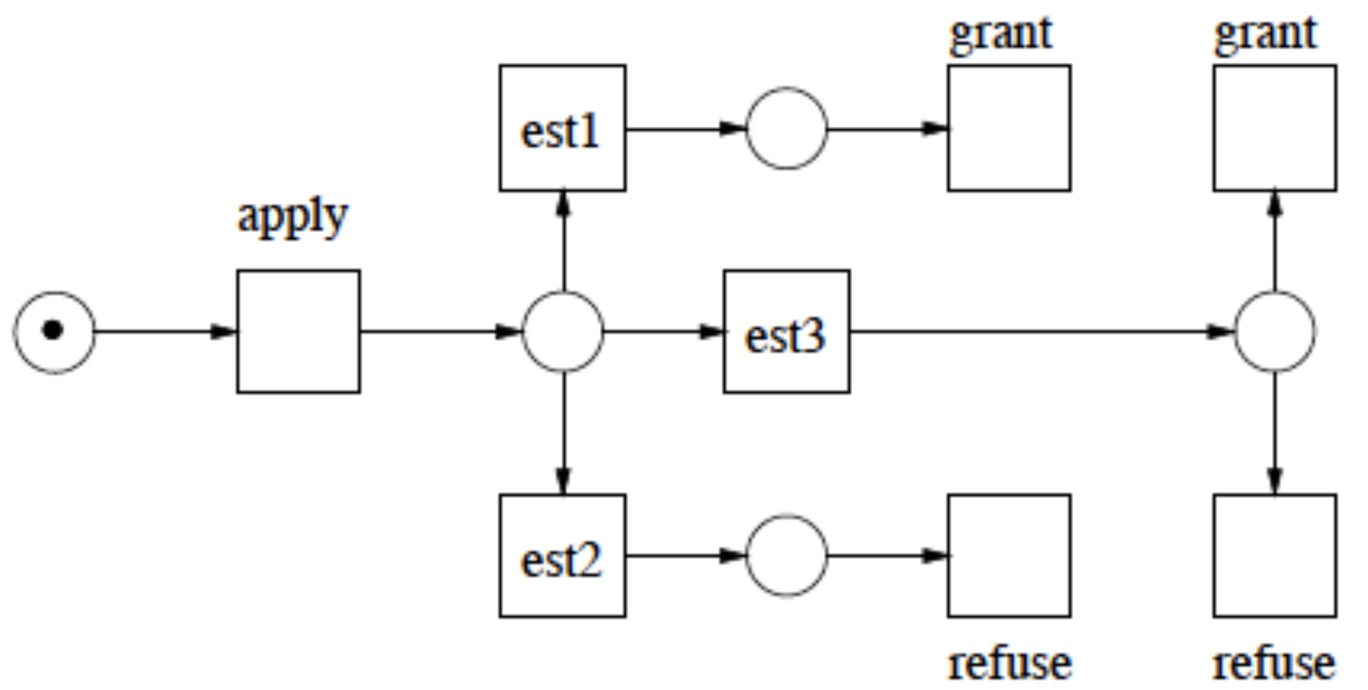
The first step to analysis and validation would be to look for possible reductions in the net, especially to detect problems with invisible transitions.

It is intuitive to use simulation to check for errors. But in systems modeling, it is also necessary to "prove" that errors are absent, especially in automated systems.

Abstraction consists of declaring the non-essential events to be *silent*, which is done notationally by labeling them with the label τ. The net thus modified is then reduced modulo an equivalence relation that disregards τ-labeled actions as much as possible. The reduced net is often a simple protocol net that embodies all desired behavior and nothing more.



Example: the bank loan process

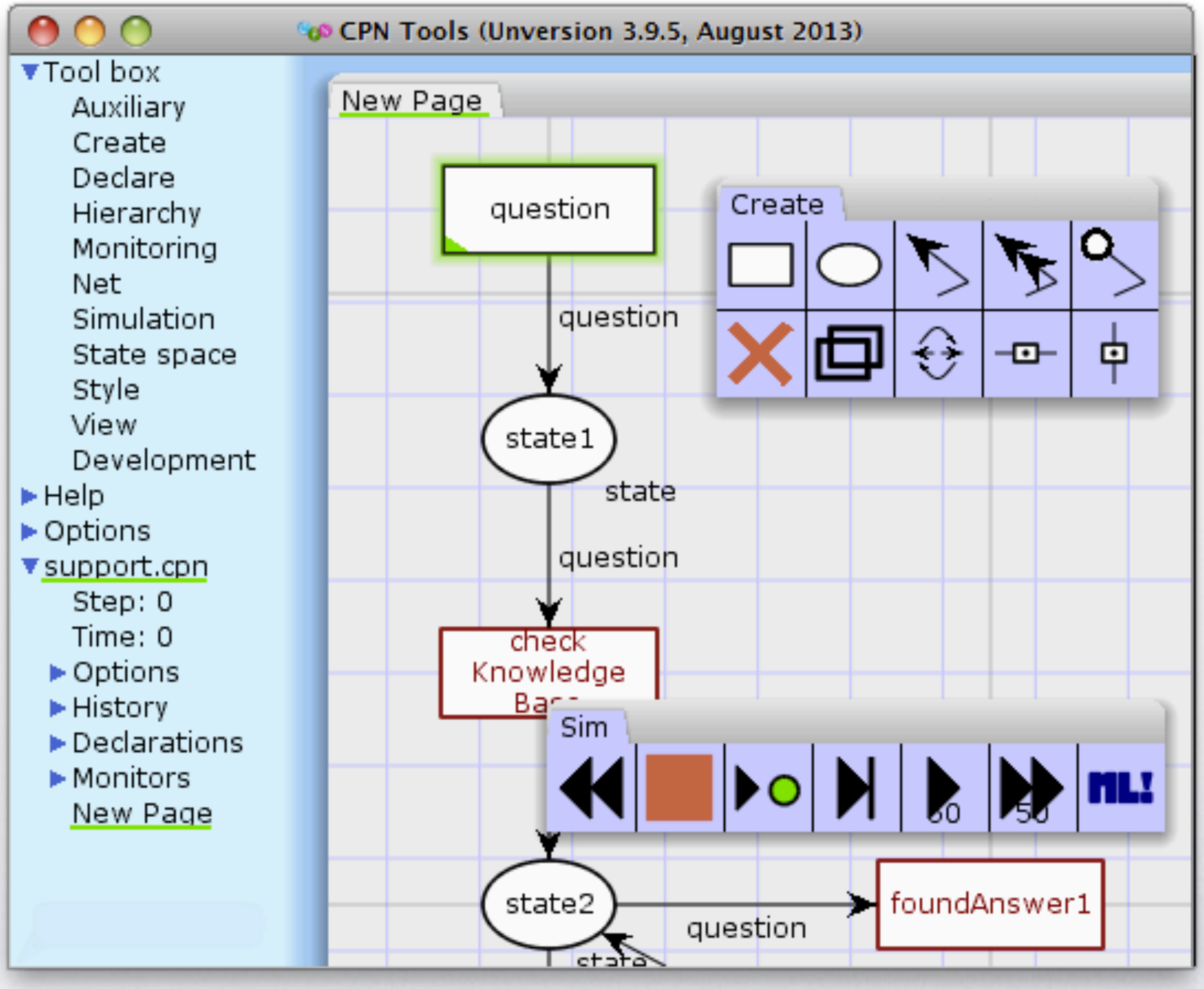


Bissimilarity

Bissimilarity and silent transitions can be faced with a process of abstraction, which concerns finding simpler models with clear attribution of transitions and respecting the occurrence of events that are observable but not controlled by the system.

Property analysis can also help, and address the dichotomy between validation and verification, besides being a way to combine methods.

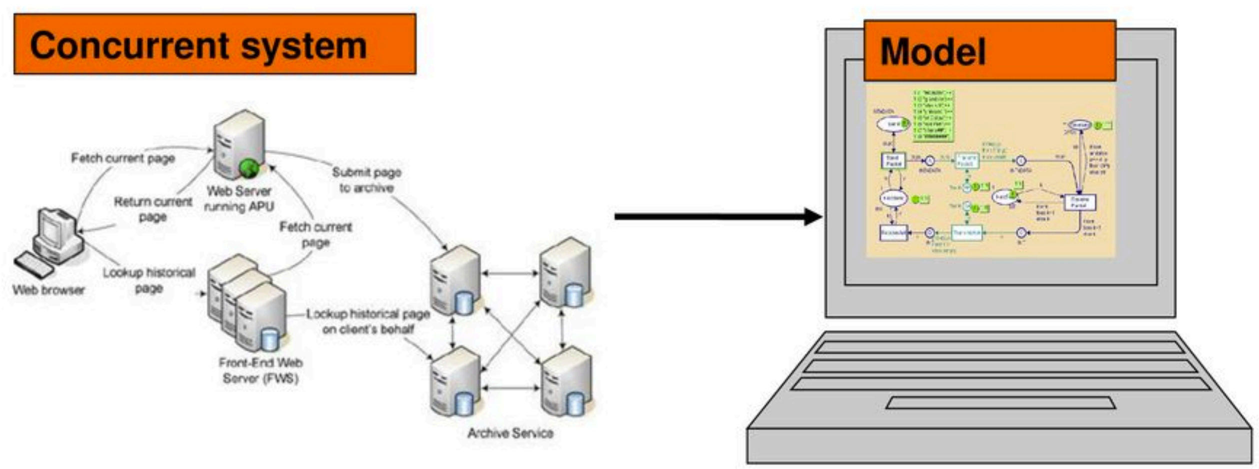
In the discussion so far there is no bias to classic or high-level nets, implying that we could use event-analysis with both net classes.



Old and new challenges to System Modeling (with Petri nets)

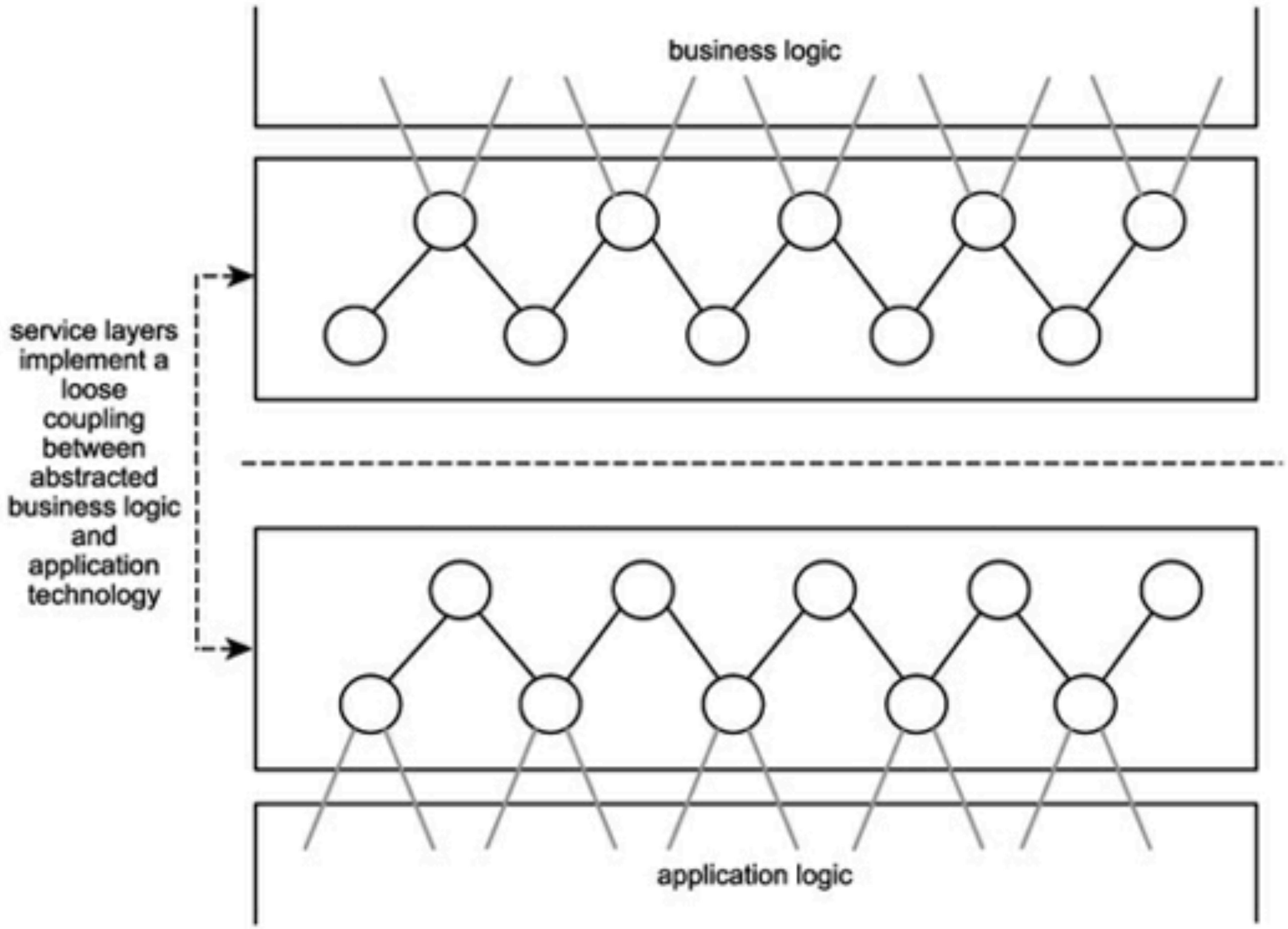
Model based system development

- One way to approach the challenges posed by concurrent systems is to build a model.
- A model is an abstract representation which can be manipulated by means of a computer tool.

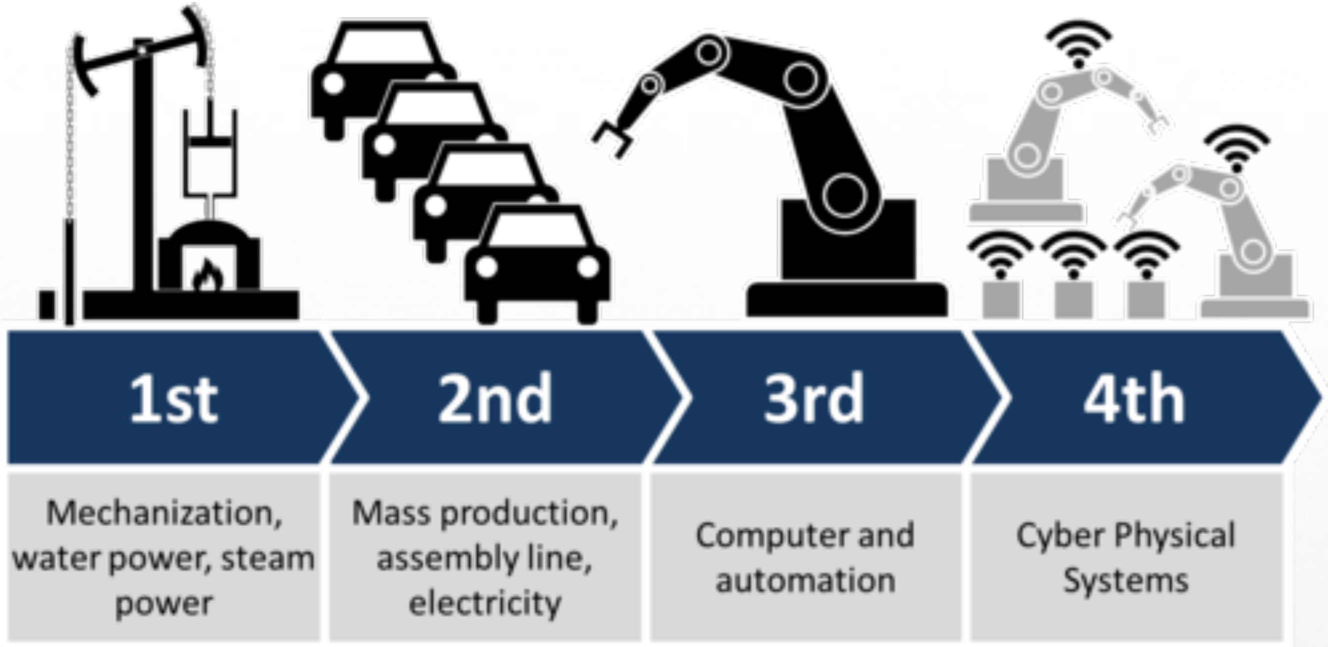


- Using a model it becomes possible to investigate how the system will behave and the properties it will possess.

Coloured Petri Nets
Department of Computer Science


business logic
service layers implement a loose coupling between abstracted business logic and application technology
application logic


INDUSTRY
4.0


1st
2nd
3rd
4th
Mechanization, water power, steam power
Mass production, assembly line, electricity
Computer and automation
Cyber Physical Systems

I4.0 brings new challenges for two main reasons:

i) it enhances concurrence in the implementation of production systems;
ii) it enhances the notion of manufacturing services or service-oriented manufacturing.

Industry new era

Service Science

Good-dominant
vs
Service-dominant

↓

"Servitization"

Industry 4.0

Embedded Systems

↓

Networked
Embedded Systems

↓

Cyber-Physical Systems

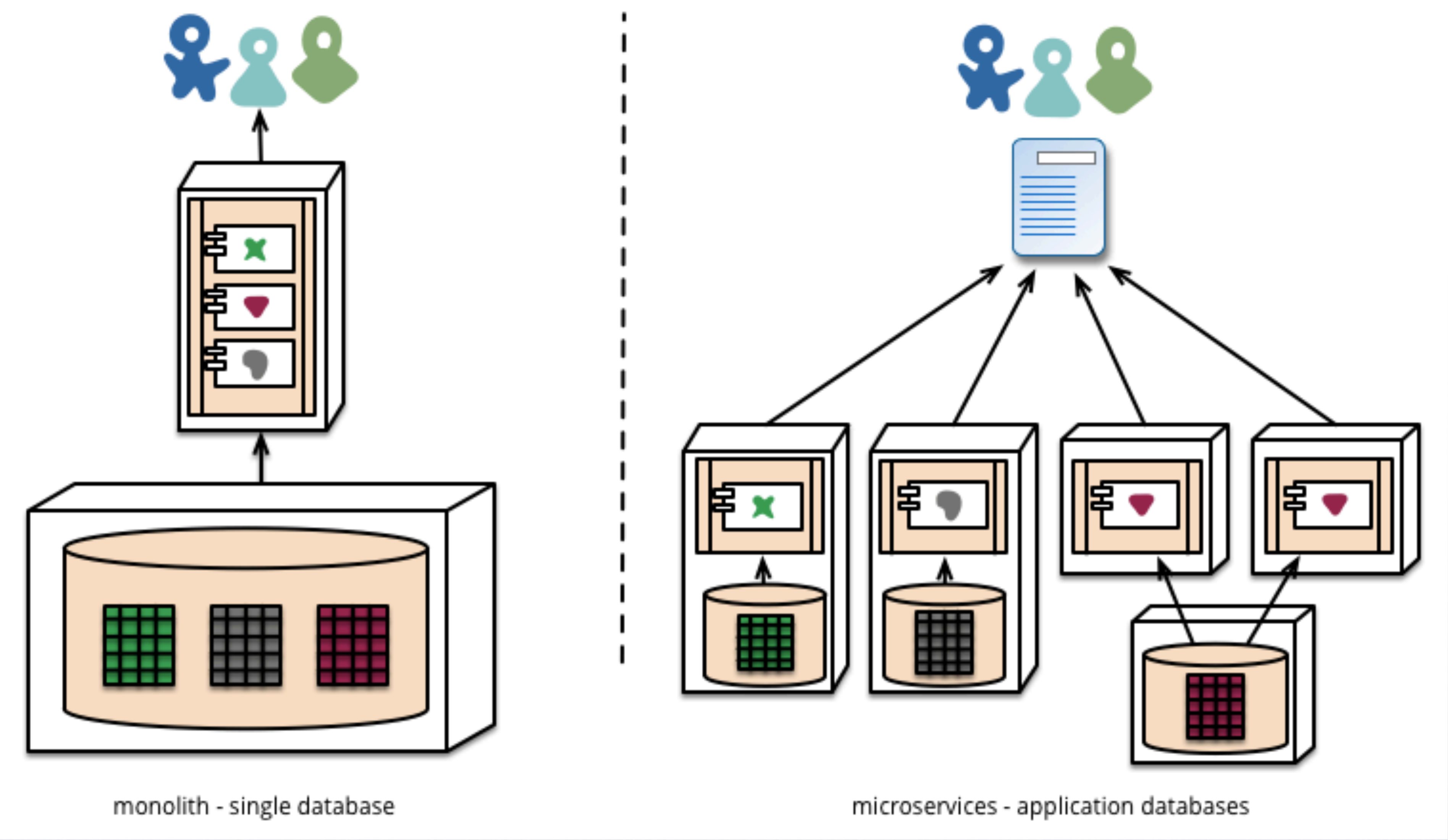
Industry new era

Service Science		Industry 4.0
Product-Service Systems	→	Smart products (Cyber-physical)
Co-production structure	PSA	Smart Factories

General requirements of new service systems

"… in the shoulders of thousands of pygmies".

Managing microservices



Next class we will reserve more time to discuss the paper and will summarize the discipline with an appreciation about Systems Modeling (with Petri nets) as a research area and its possibilities, as well as the possibilities in practice.