Environment Modeling based Requirements Engineering for Cyber-Physical Systems

Zhi Jin

Key Lab of High Confidence Software Technologies (MoE)
Peking University
zhijin@pku.edu.cn

高可信软件技术
教育部重点实验室

IEEE INMIC 2019，Islamabad, Pakistan, 2019.11.30

北京大学

Agenda

- Cyber Physical Systems bring in Challenges
- Environment Modeling based Requirements Engineering
- Some Non-functional Requirements
- More Efforts and Further Work

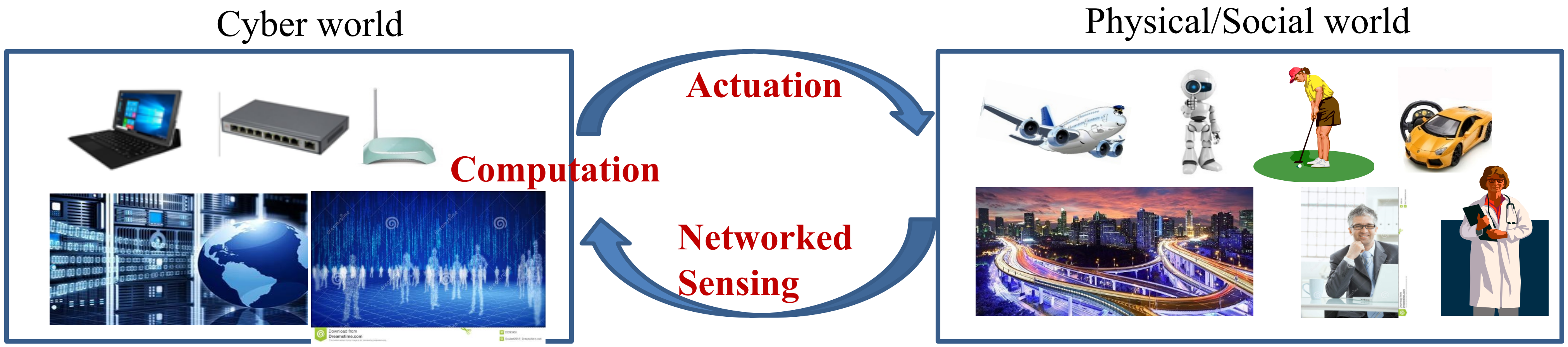
Cyber-Physical Systems

Cyber-Physical (-Social) Systems

Software systems are to be tightly *integrated with* the physical systems and the social systems

with

networked sensing, *computation*, *actuation*, etc.

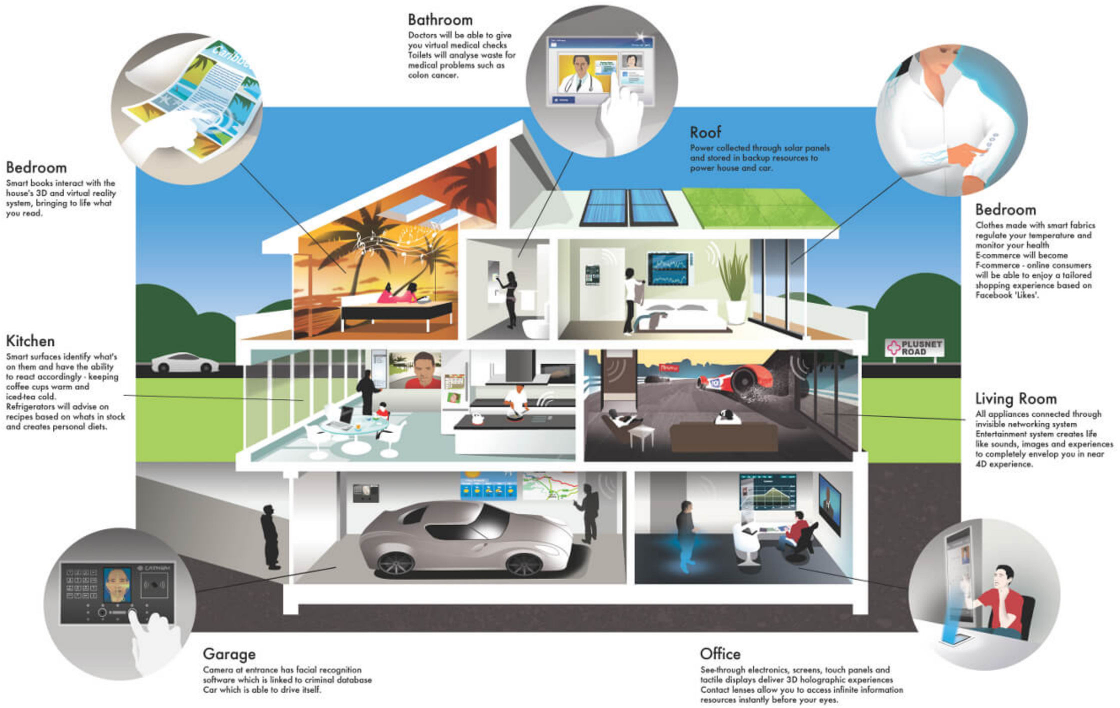


Cyber-Physical Systems

- Increasing number of critical applications in dependable sectors
  - Smart transportation
  - Large-scale critical infrastructures
  - Intelligent defense systems
  - Smart health care
  - aviation
  - ……
    - ……

Motivation: Smart Home

integrate the facilities of home life; Keep your home more comfortable, more convenient, and more secure via different kinds of sensors and actuators, wearable devices, ……



Service layer

Includes basic services provided by the sensor platform layer and can be combined into more complex applications for developers to invoke

↓

Sensor platform layer

Effectively translate the functions of various devices in the physical layer into services and describe them in a unified form.

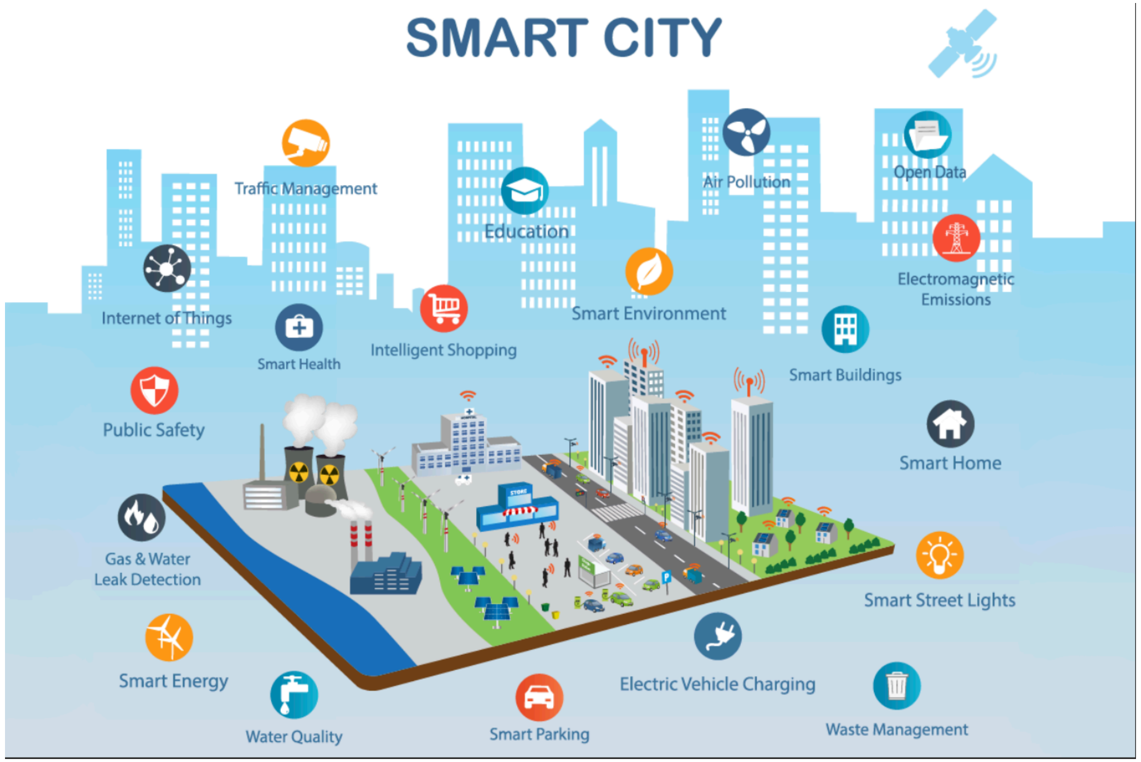
↓

Physical layer

Sensing, information acquisition and controlled equipment at the end of the home, such as air conditioning, TV, smoke sensor, etc.

Motivation: Smart Cities

City equipped with smart facilities

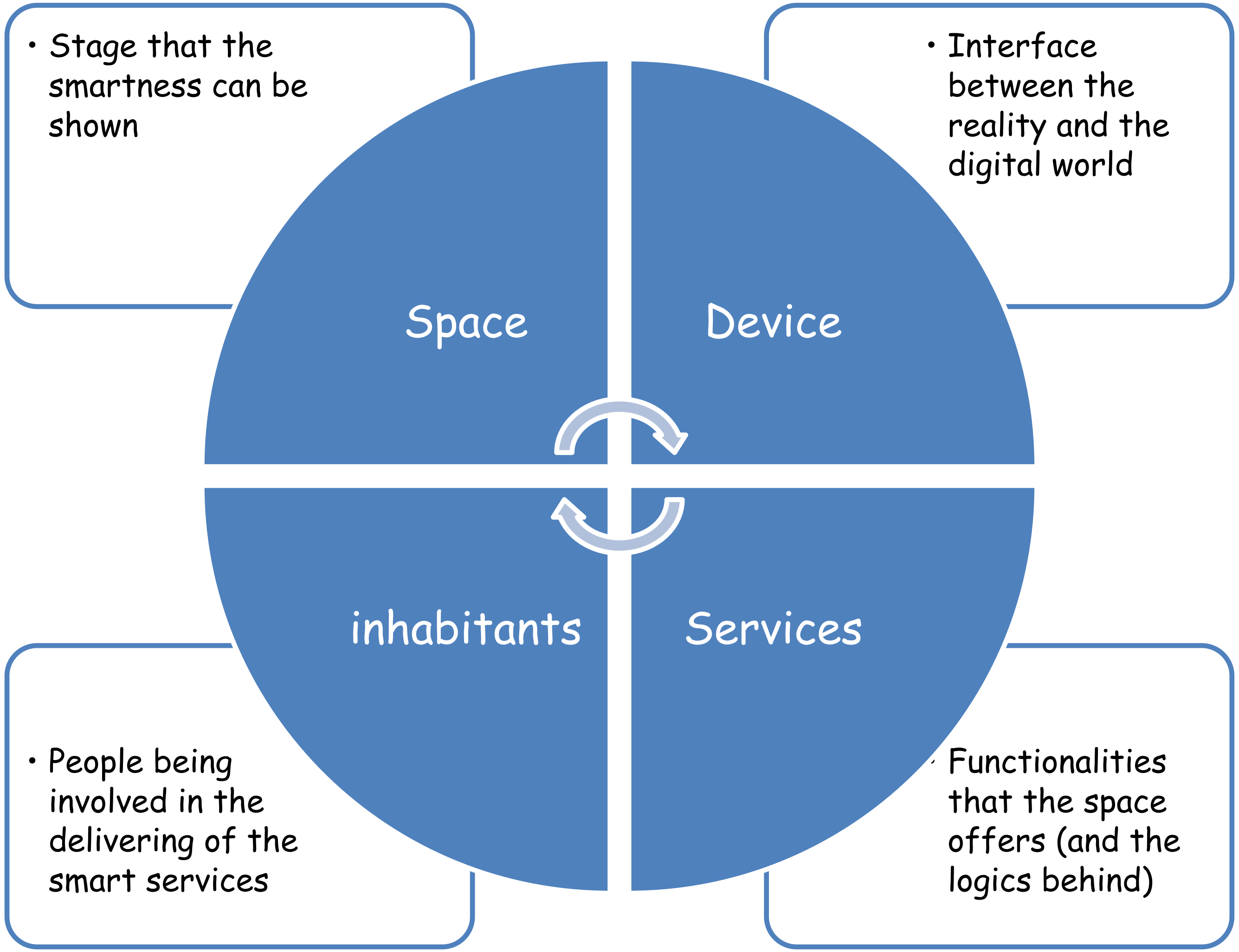


Not just about concrete and steel, but about how people want to live, e.g.

- define the most pressing transportation problems and envision bold new solutions
  - that could change the face of transportation in the cities
  - by meeting the needs of residents of all ages and abilities
- bridge the digital divide so that everyone, not just the tech-savvy, can be connected to everything their city offers

Four Fundamental Elements

- Smart spaces
  - combines technical intelligence and cognitive functions to provide an interface between the virtual and the physical world



Concerns for Cyber Physical Systems

Aerospace, automotive, chemical processes, energy, civil infrastructure, healthcare, manufacturing, consumer appliances, transportation, and entertainment

Need

- Security
- Privacy
- Time
- Physical Interaction modeling

Issues

CPS

Properties

- Environment
  - Openness
  - Dynamically changing
  - Perception
  - Actuation
  - Heterogeneity

Benefits

smart grid, autonomous automobile systems, medical monitoring, process control systems, robotics systems, and automatic pilot avionics

Cyber-Physical Systems

Devices: individual node of the system,
physical vs logic vs person
Content: data, information and knowledge
that need to be acquired, stored
and processed by the system

Operating environment
Openness and dynamic unpredictability

System Components
Diversity and Distribution Heterogeneity

Functionalities:
coming from the environmental entities
Adaptability:
adapting to new environment
new requirements
System assurance:
quality of service
security
confidentiality and reliability

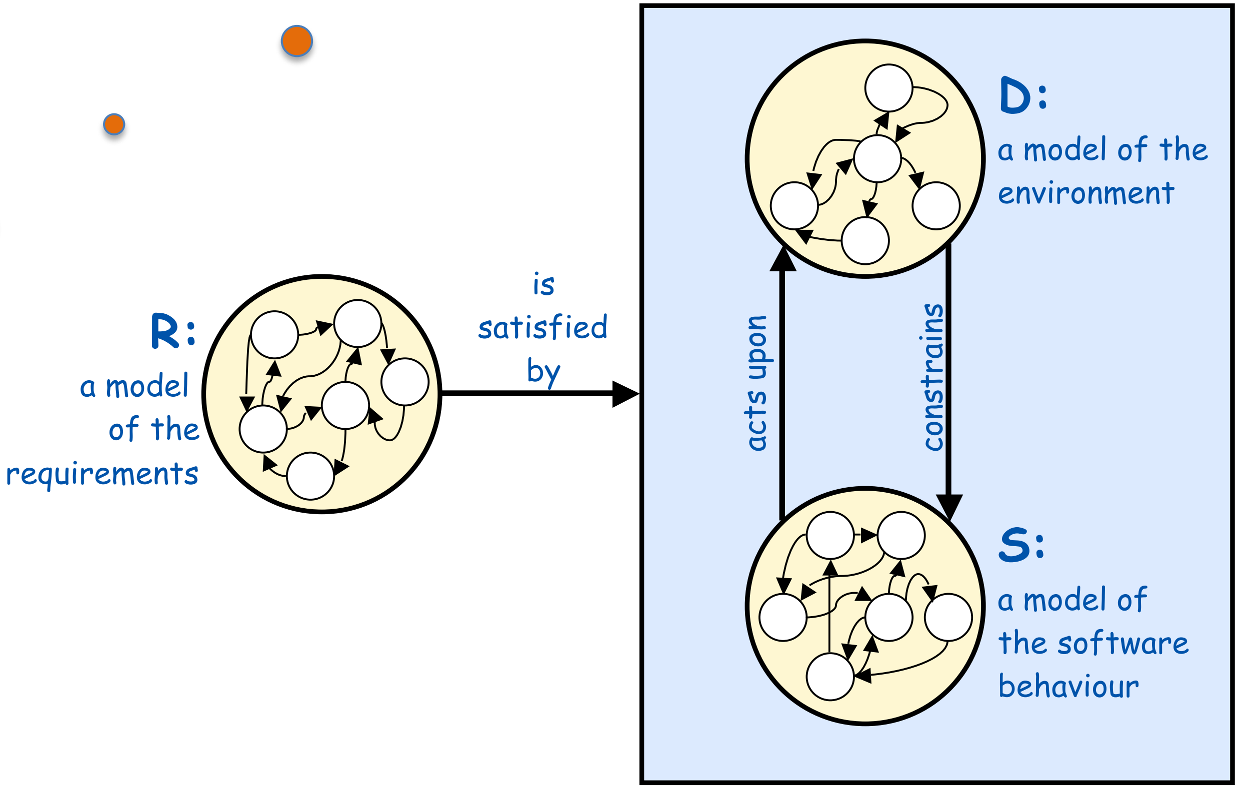
Challenges: Requirements Engineering

Environment Properties + Specification ⊢ Requirements

Environment is fixed

The task of RE

Given Environment Properties
And Users' Requirements
When Conduct RE process
Then System Specification
is decided



Methodologies

- Perspective of observing the problem
  - Goal-oriented approach
  - Intentional actors approach
  - Scenario-based approach

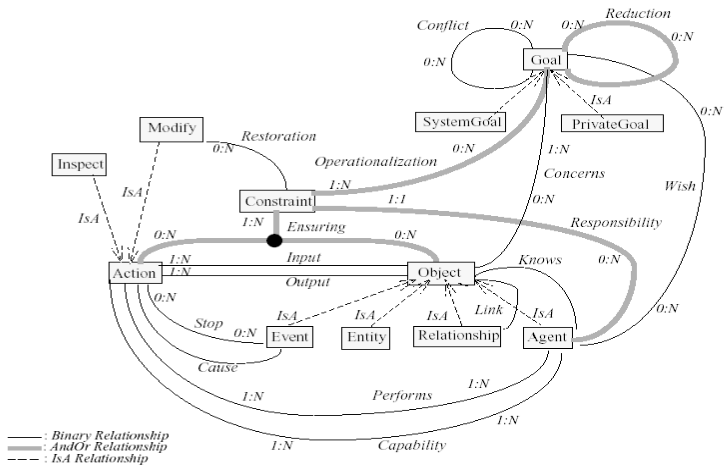
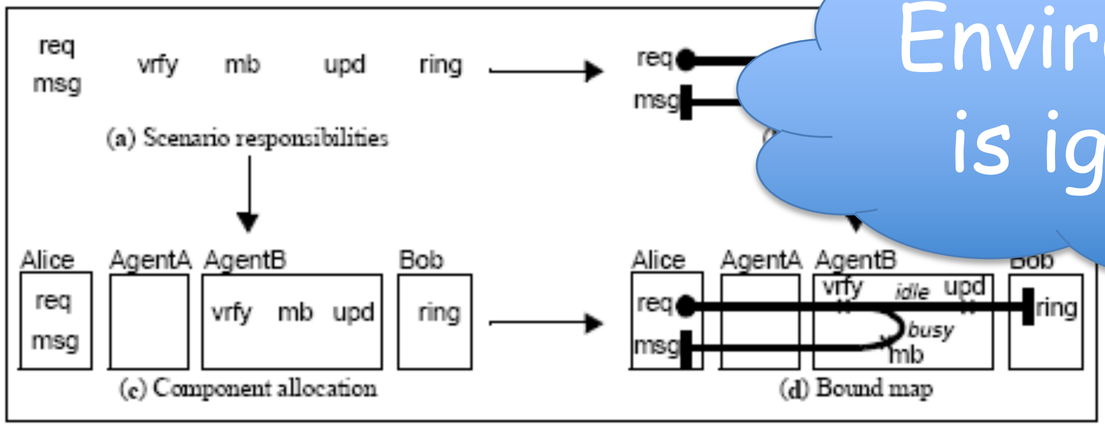
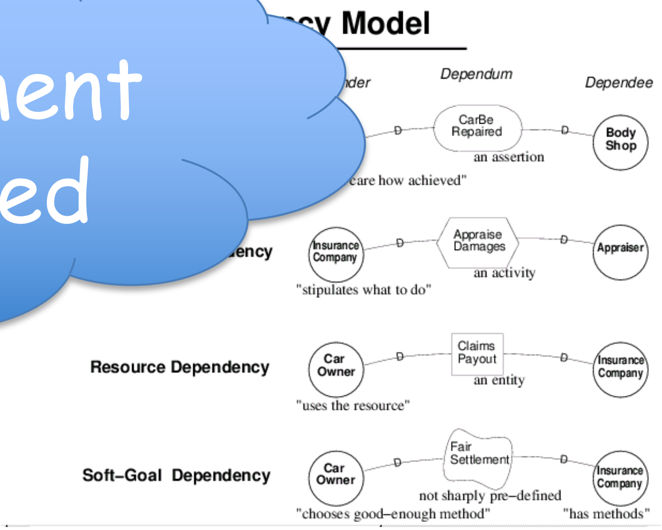


Figure 2: A Portion of the KAOS Conceptual Meta-model

Environment
is ignored

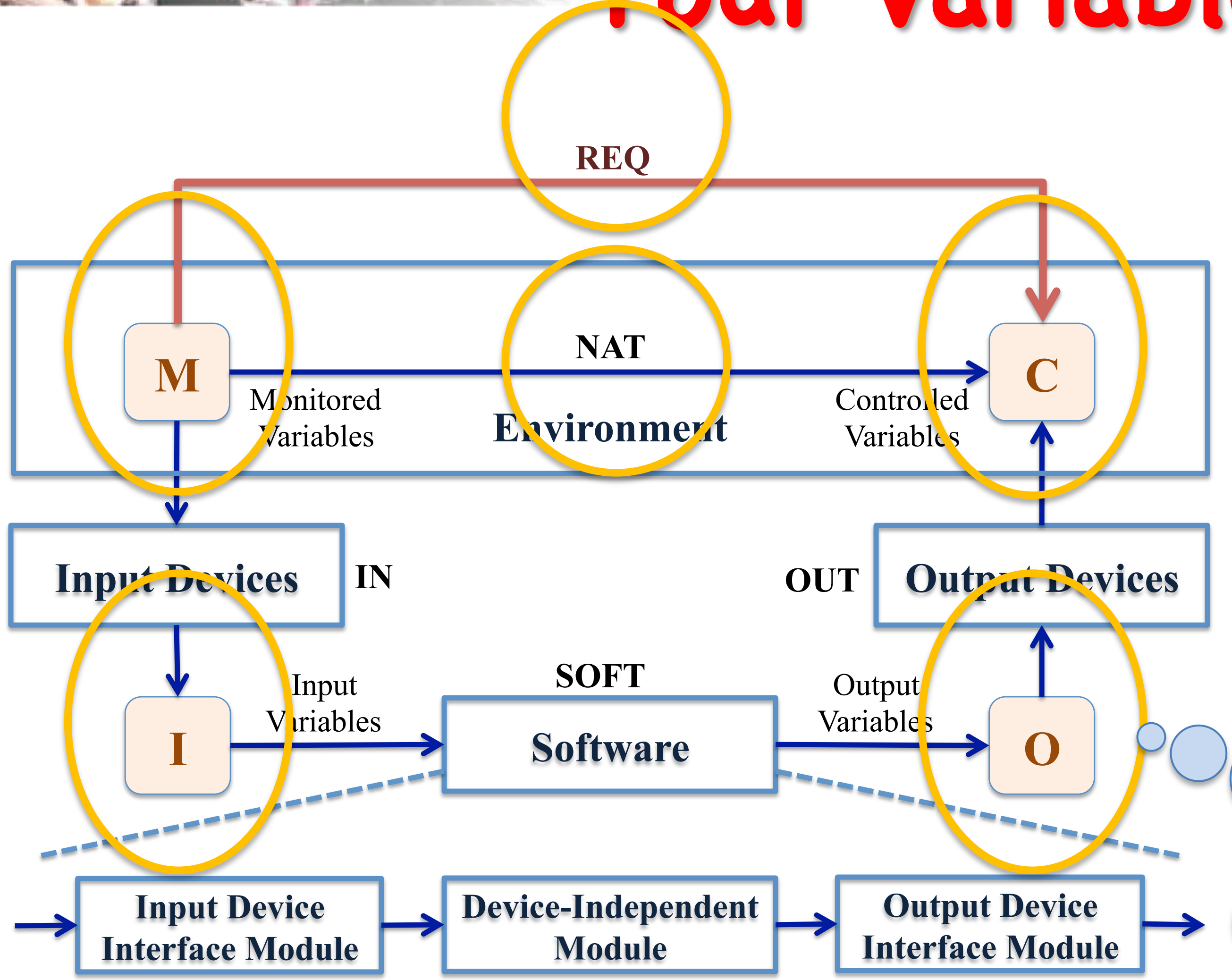




Agenda

- Cyber-Physical Systems bring Challenges
- Environment Modeling based Requirements Engineering
- Some Non-functional Requirements
- More Efforts and Further Work

Four Variable Model



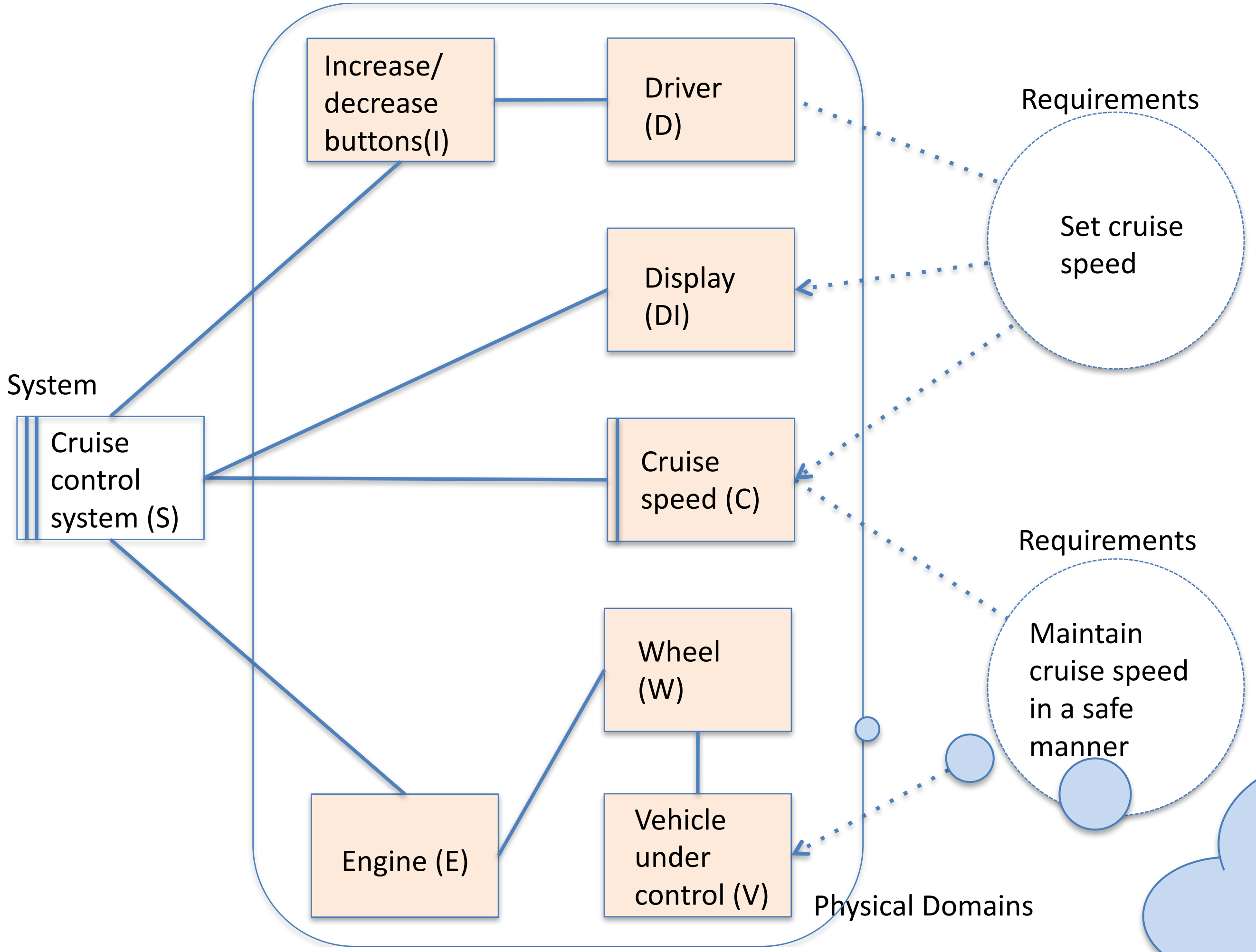
D. Parnas and J. Madey (1995), Functional Documents for Computer Systems, Science of Computer Programming 25: 41-61.

Metaphor:

To-be System is for Measuring and Controlling Reality Variables accordingly for enabling the expected relationships

Environment properties are first-class citizen and represented by sets of variables and the relationships between the variables

Problem Frames



Metaphor:

To-be System is for Establishing Relationships among Phenomena of Reality that are really expected

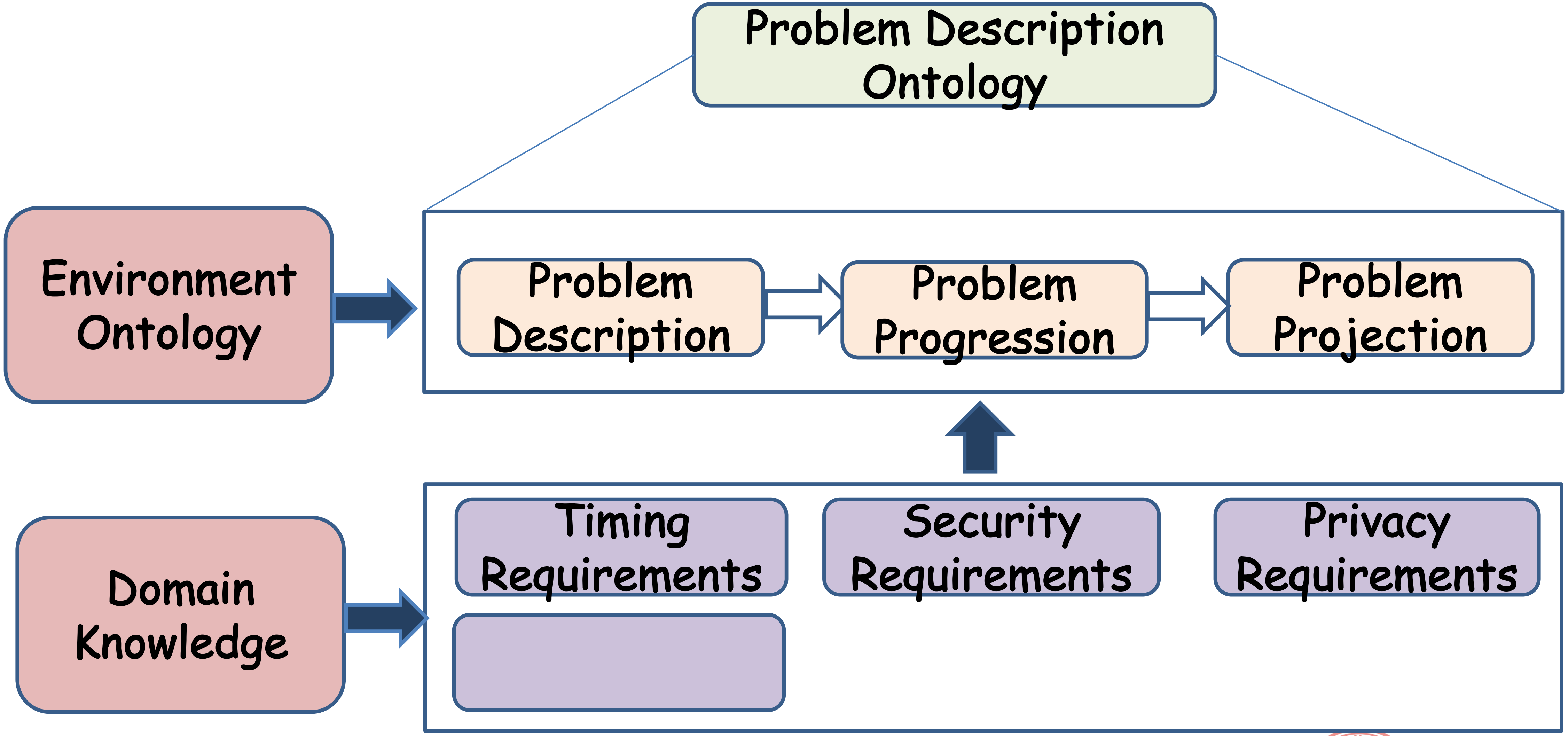
Environment entities and some of their dynamics are explicitly identified and represented

M. Jackson (2001), Problem Frames: Analyzing and Structuring Software Development Problems, Addison-Wesley

Still, The Difficulties

- Specify the environment entities with different characteristics
- Ensure the description completeness
- Locate the problem
- Decouple the problem
- Identify some related non-functionalities
- ……

The Solution: Framework



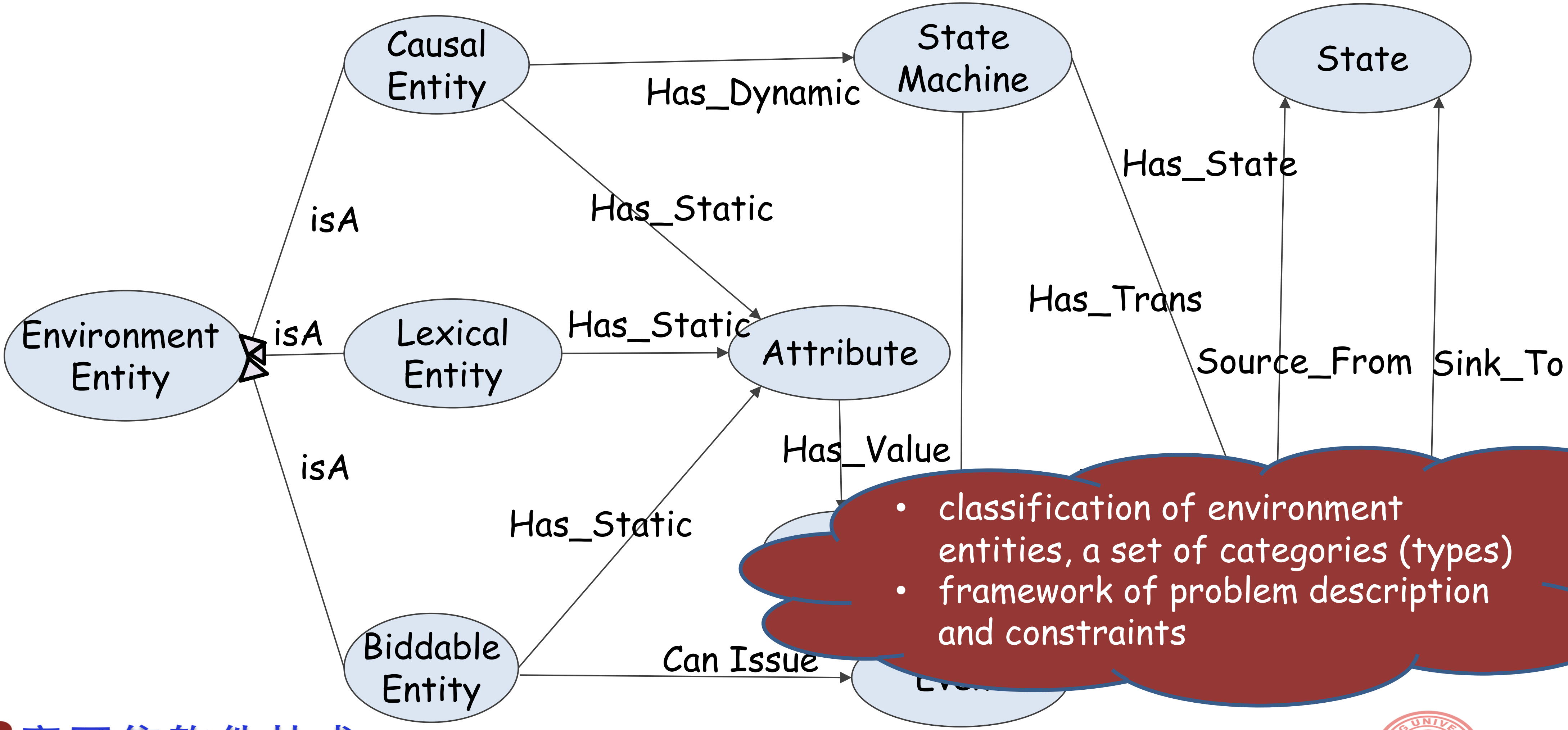
Main Principles

- Principe 1: (Classification) Environment is treated as the First-class Citizen, environment objects are classified
- Principle 2: (Statefulness) Environment objects are stateful, and then concerning environment is stateful. Environment state is the situation at a certain point
- Principle 3: (Dynamics) Environment Objects have Inner Behavior Regulations Apart from Attribute Aspect

Main Principles

- Biddable entity:
  - Typically consists of people
  - It is physical but lacks positive predictable internal causality
- Causal entity:
  - Their properties include predictable causal relationships among its causal phenomena
- Lexical entity:
  - A physical representation of data

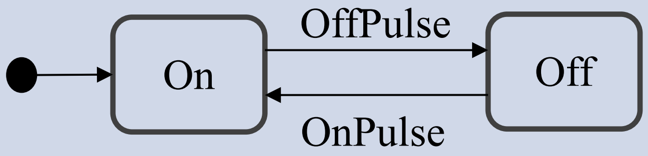
Environment Ontology: Classification and Description



- classification of environment entities, a set of categories (types)
- framework of problem description and constraints

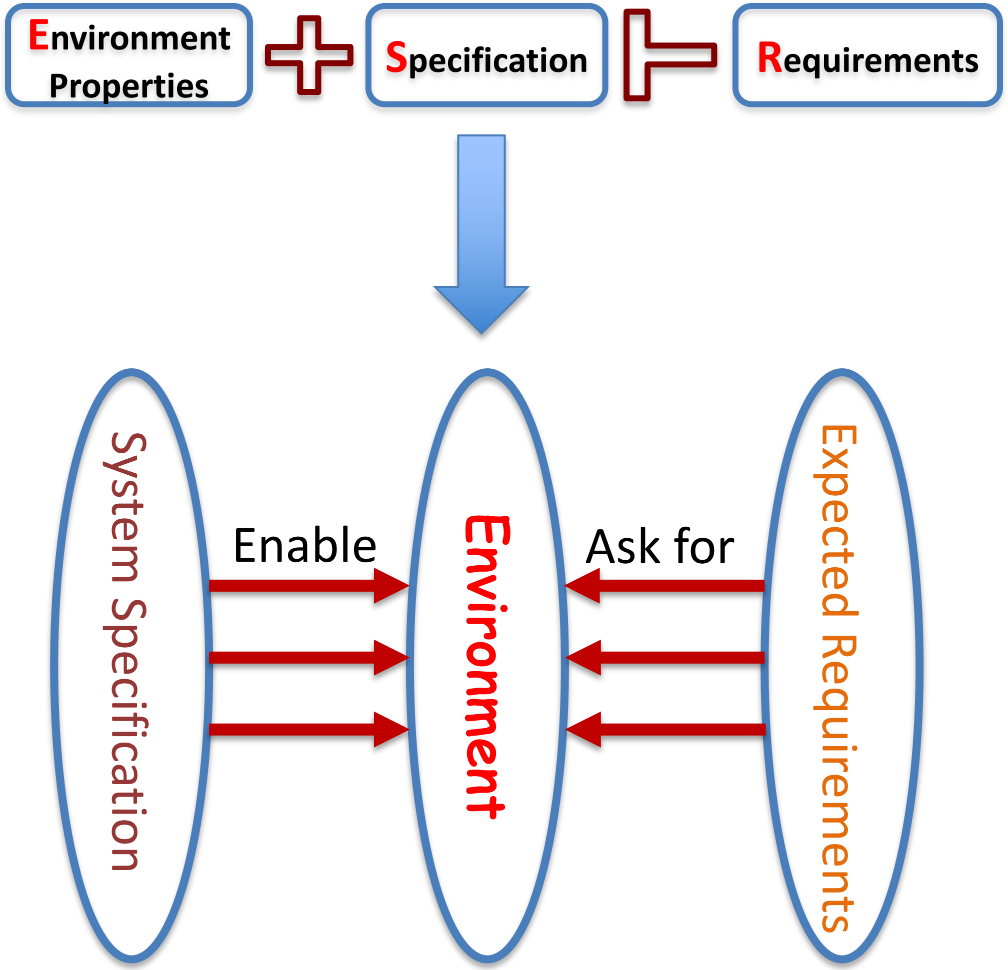
Environment Ontology: Dynamics

A set of light units are to be controlled by switch operators.

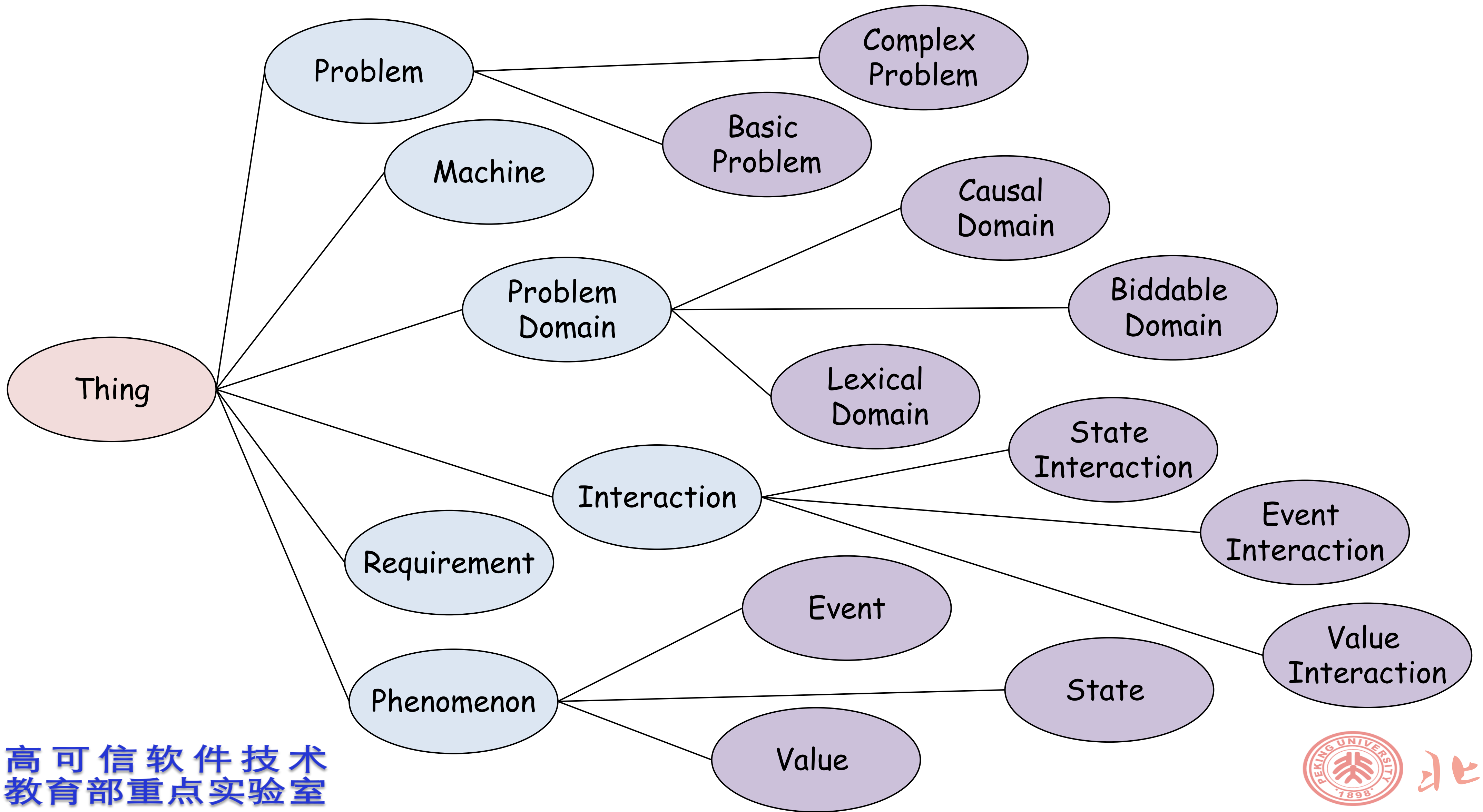
Environment entity	**Entity type**	**State machine**	**canIssue**
Light unit	Causal		
Switch operator	Biddable		OnButton, OffButton

Environment Properties are Matter

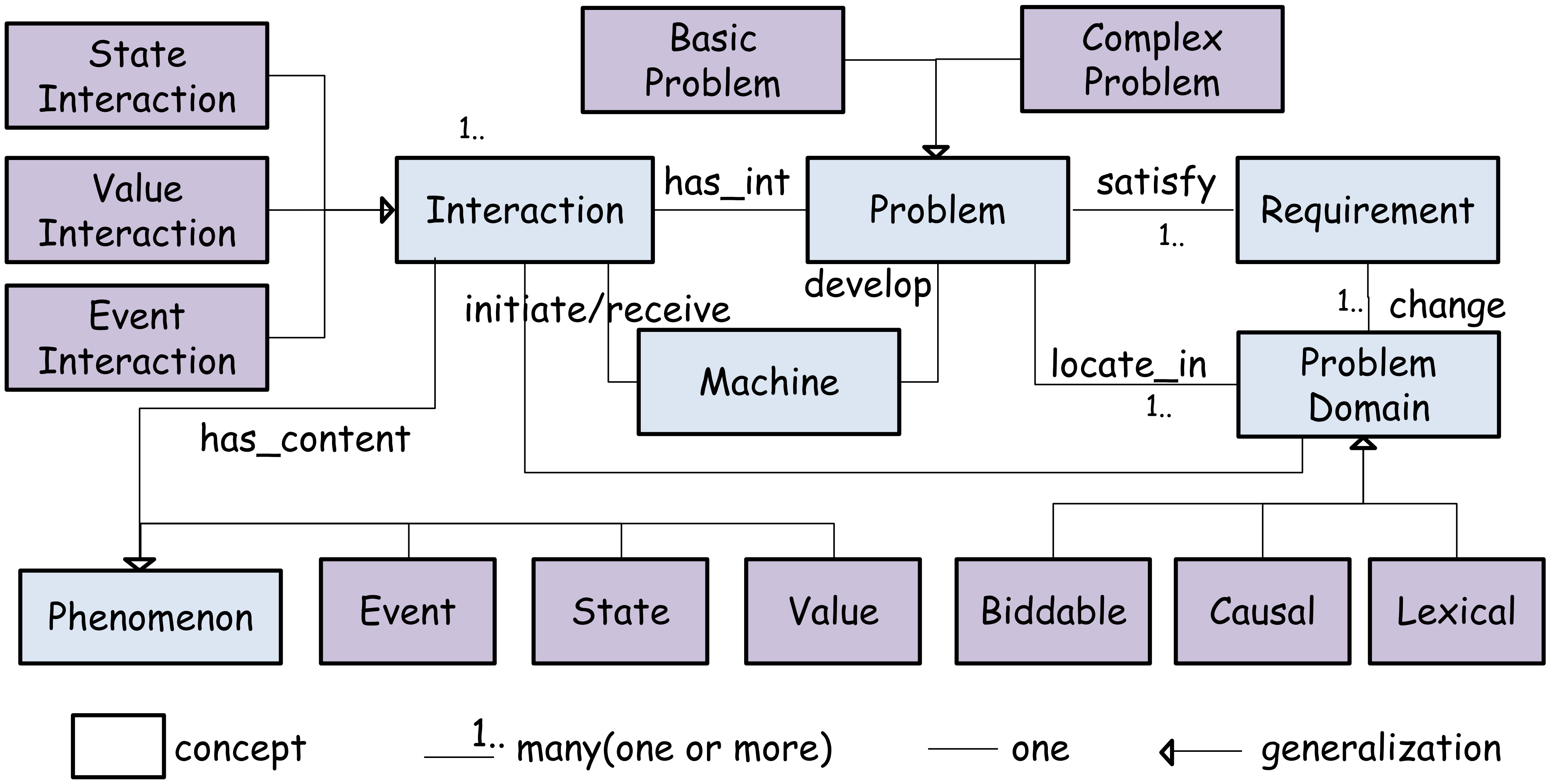
- Environment modeling based Approach
  - Extending the Problem Frames representations
    - structures the environment elements and then the requirements
    - provides analysis methods for deriving specification
  - Defining environment model
    - abstraction of the environment elements
    - capture the environment dynamics and openness



Problem Description Ontology



Problem Description Ontology



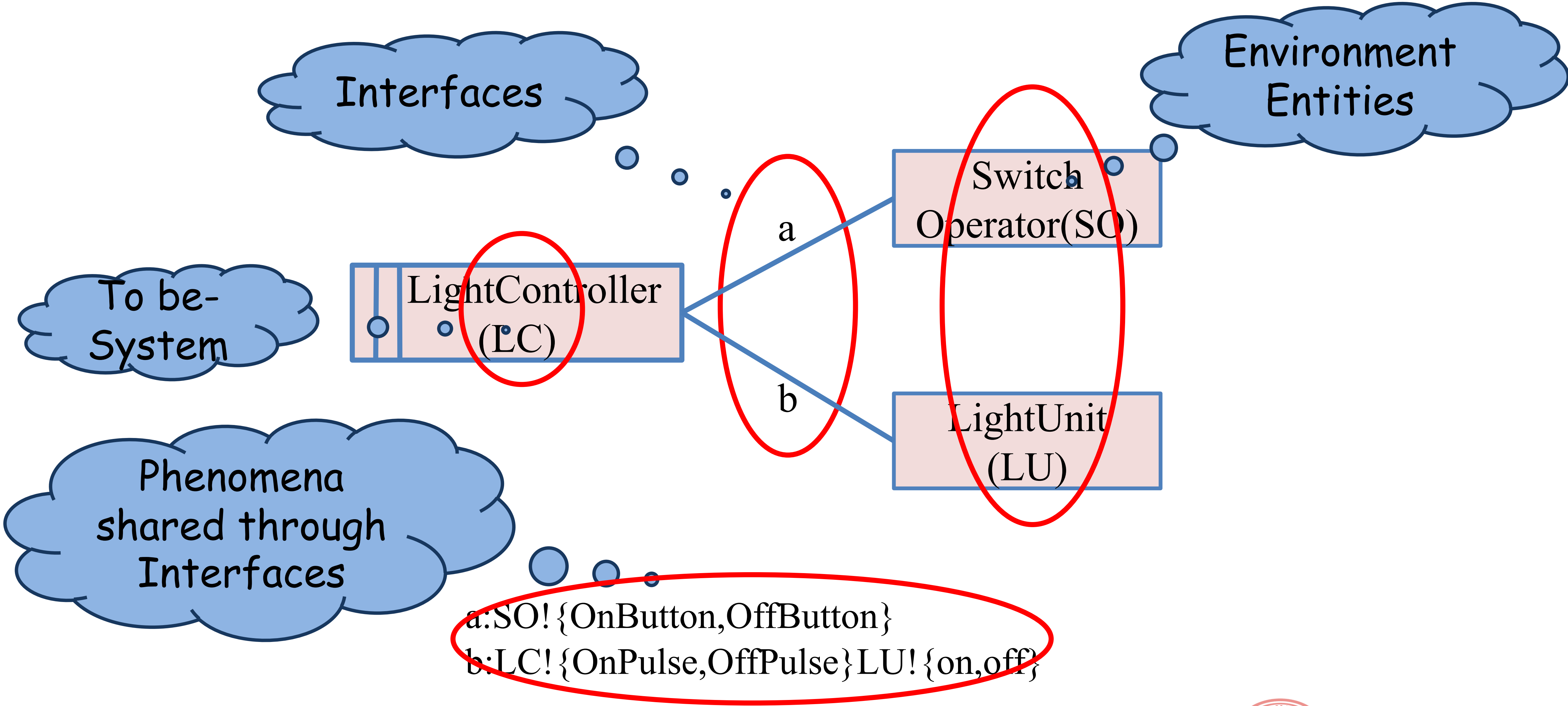
Problem Description Ontology: Constraints

$problem \sqsubseteq \geq 1\ develop \sqcap \leq 1\ develop$ Each problem only has a machine	(1)
$problem \sqsubseteq \geq 1\ locate\_in$ Each problem has a set of problem domains	(2)
$problem \sqsubseteq \geq 1\ has\_int$ Each problem has a set of interactions	(3)
$problem \sqsubseteq \geq 1\ satisfy$ Each problem meets a set of requirements	(4)
$scenario \sqsubseteq \geq 1\ realize \sqcap \leq 1\ realise$ Each scenario realize requirement	(5)
$Interaction(?x) \cap initiate(?x,?y) \cap receive(?x,?z) \cap Domain(?y) \rightarrow Machine(?z)$ Interaction connects software systems and problem domain	(6)
$Interaction(?x) \cap initiate(?x,?y) \cap receive(?x,?z) \cap Machine(?y) \rightarrow Domain(?z)$ Interaction connects software systems and problem domain	(7)
$Lexical\mathrm{Domain}(?x) \cap initiate(?x,?y) \rightarrow ValueInteraction(?y)$ Lexical domain can only initiate value phenomenon	(8)
$StateInteraction(?x) \cap initiate(?x,?y) \rightarrow CausalDomain\ (?y)$ Only the causal domain can initiate state phenomenon	(9)
$EventInteraction(?x) \cap has\_content(?x,?y) \rightarrow Event\ (?y)$ The content of the event interaction is the event	(10)
$StateInteraction(?x) \cap has\_content(?x,?y) \rightarrow State\ (?y)$ The content of the state interaction is the state	(11)
$ValueInteraction(?x) \cap has\_content(?x,?y) \rightarrow Value\ (?y)$ The content of the value interaction is the value	(12)

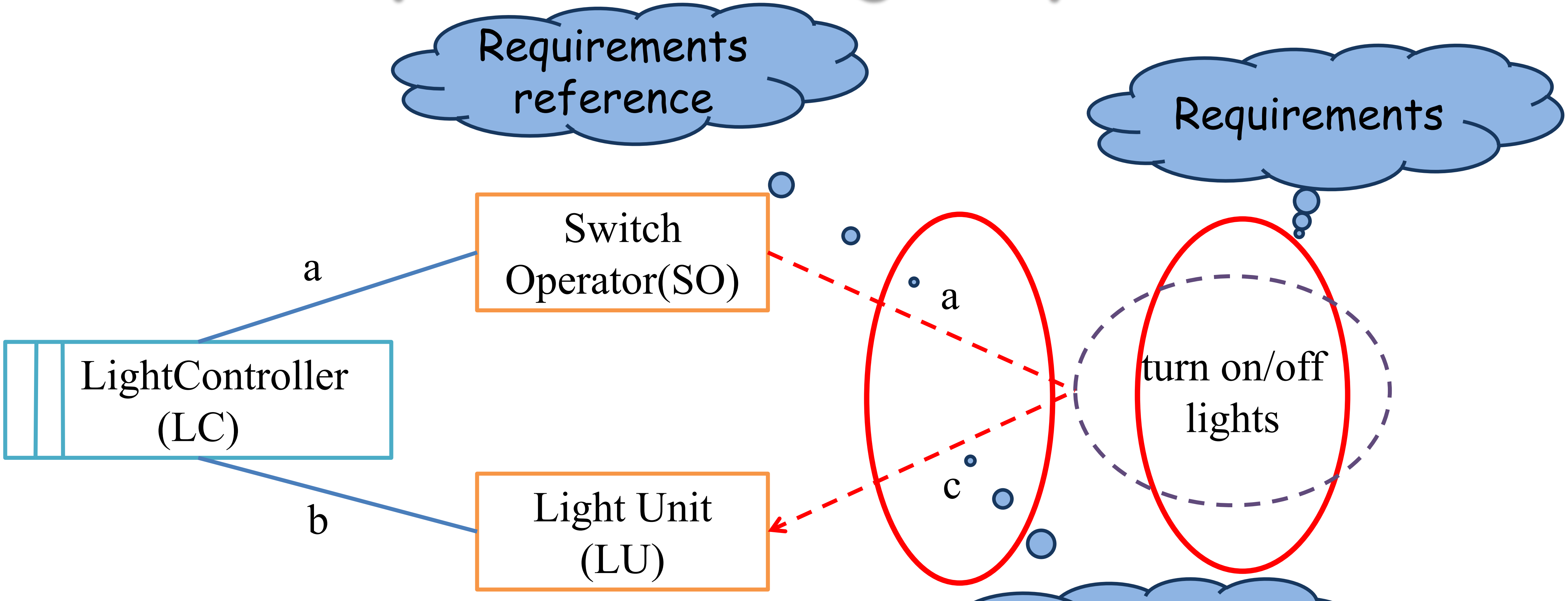
The number of concept instances

Relations between concepts

An Example: Problem Context



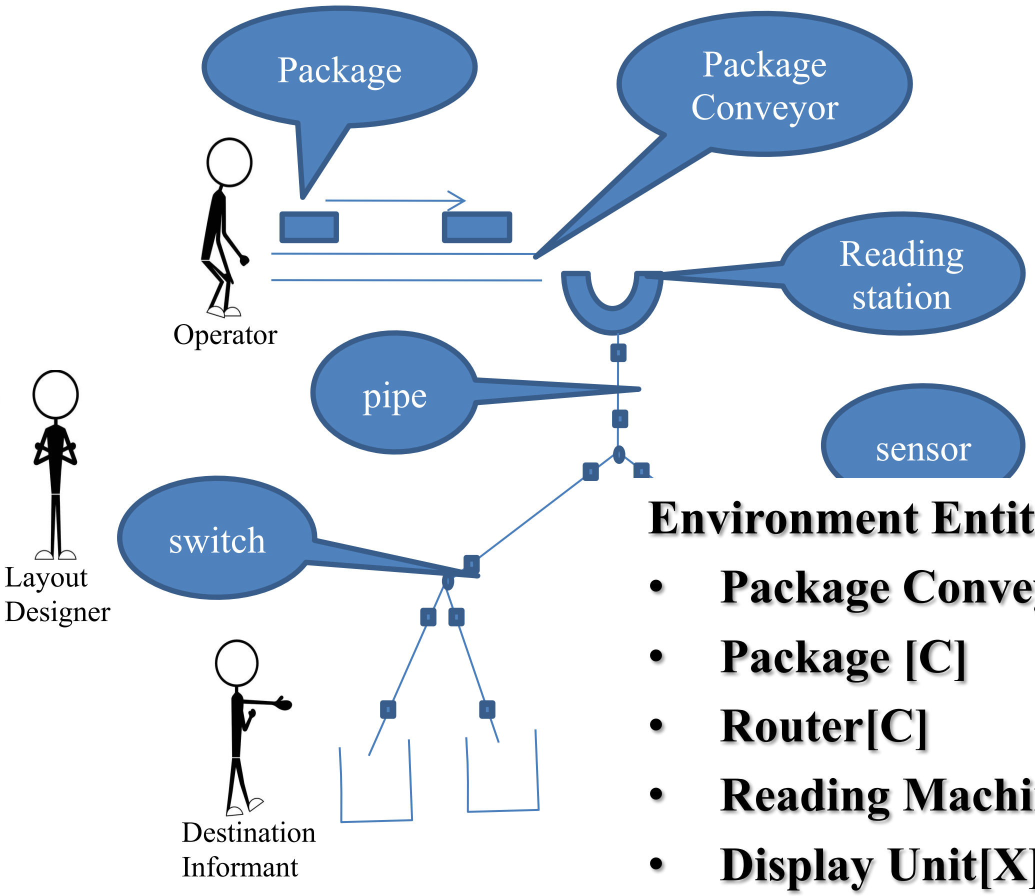
An Example: Including Requirements



a:SO!{OnButton,OffButton}
b:LC!{OnPulse,OffPulse}
c:LU!{On,Off}
Fig. Light control problem: problem diagram

An Real Example

- A package router is a large mechanical device
  - –used by postal and delivery organizations
  - –to sort packages into bins according to their destinations.
- Problem is to build the control machine
  1) to obey the operator's commands
  2) to route packages to their destination bins
  3) to report misrouted packages



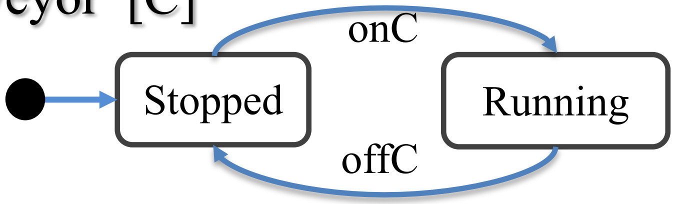
Environment Entities

- **Package Conveyor [C]**
- **Package [C]**
- **Router[C]**
- **Reading Machine[X]**
- **Display Unit[X]**
- **Operator[B]**
- **Layout Designer[B]**
- **Destination Informant[B]**

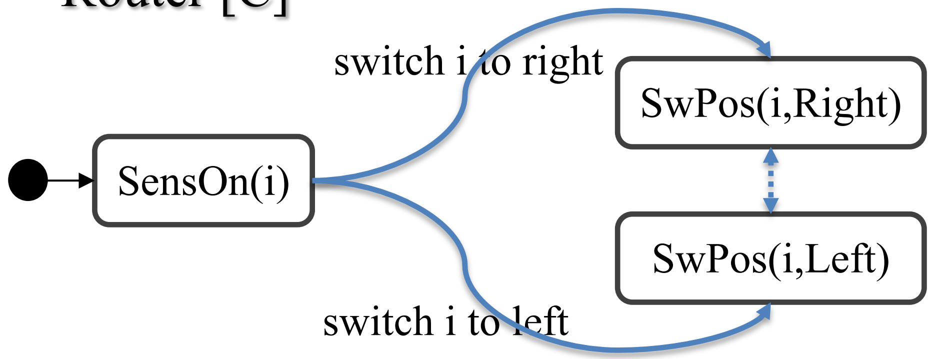
An Real Example: Entity Modeling

- Attributes of environment entities
  - Package: label
  - Router: pipe[l], sensor[m], switch[n], bin[o]
  - Reading Machine: label, packageID, destination
  - Display Unit: package ID, bin, destination
- Events that biddable entities issue
  - Operator[B]: OnBut, OffBut
  - Layout Designer[B]: Edit Layout Commands
  - Destination Informant[B]: Edit Destination-bin Commands

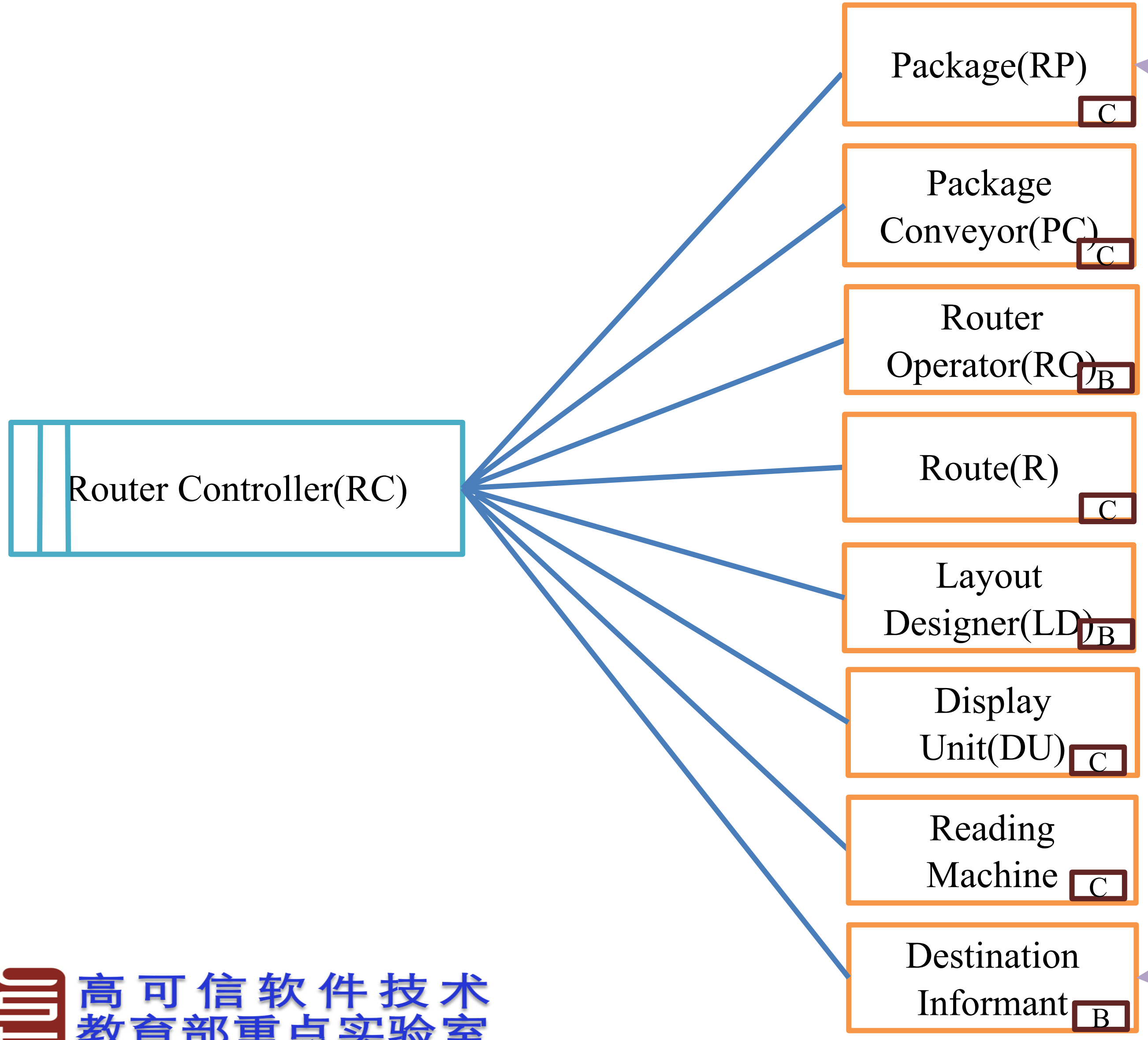
- State machine of causal entity
  - Package Conveyor [C]



  - Router [C]



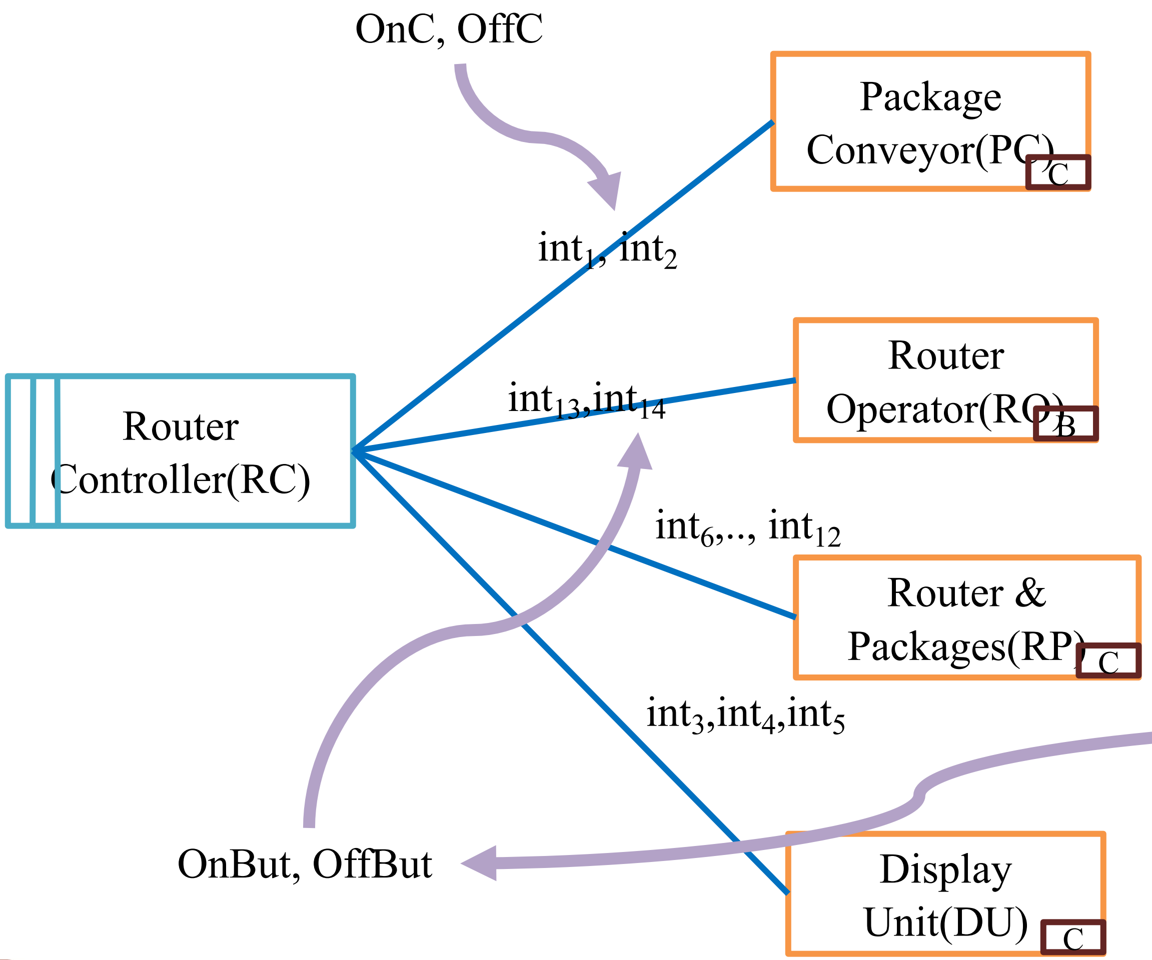
An Real Example: Locating Problam

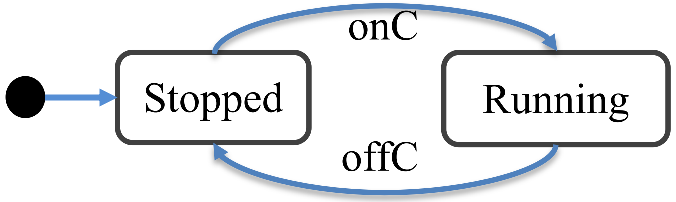


Environmental Entities

- **Router[C]**
- **Package Conveyor [C]**
- **Package [C]**
- **Operator[B]**
- **Reading Machine[X]**
- **Display Unit[X]**
- **Layout Designer[B]**
- **Destination Informant[B]**

An Real Example: Decide Interactions





State machine of Package Conveyor

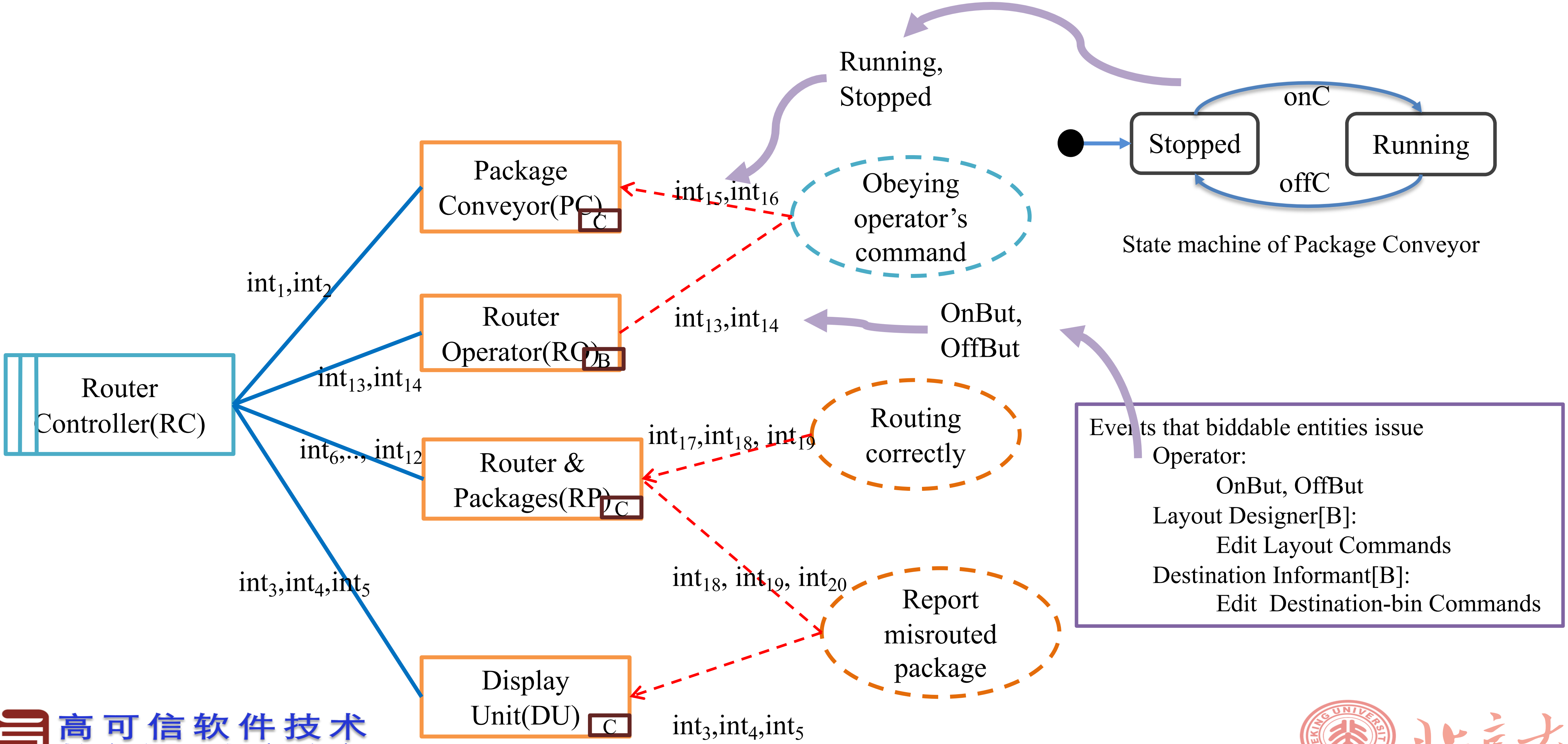
Checking list

1. One and only one machine
2. Multiple domains
3. Each domain has interface

Events that biddable entities issue

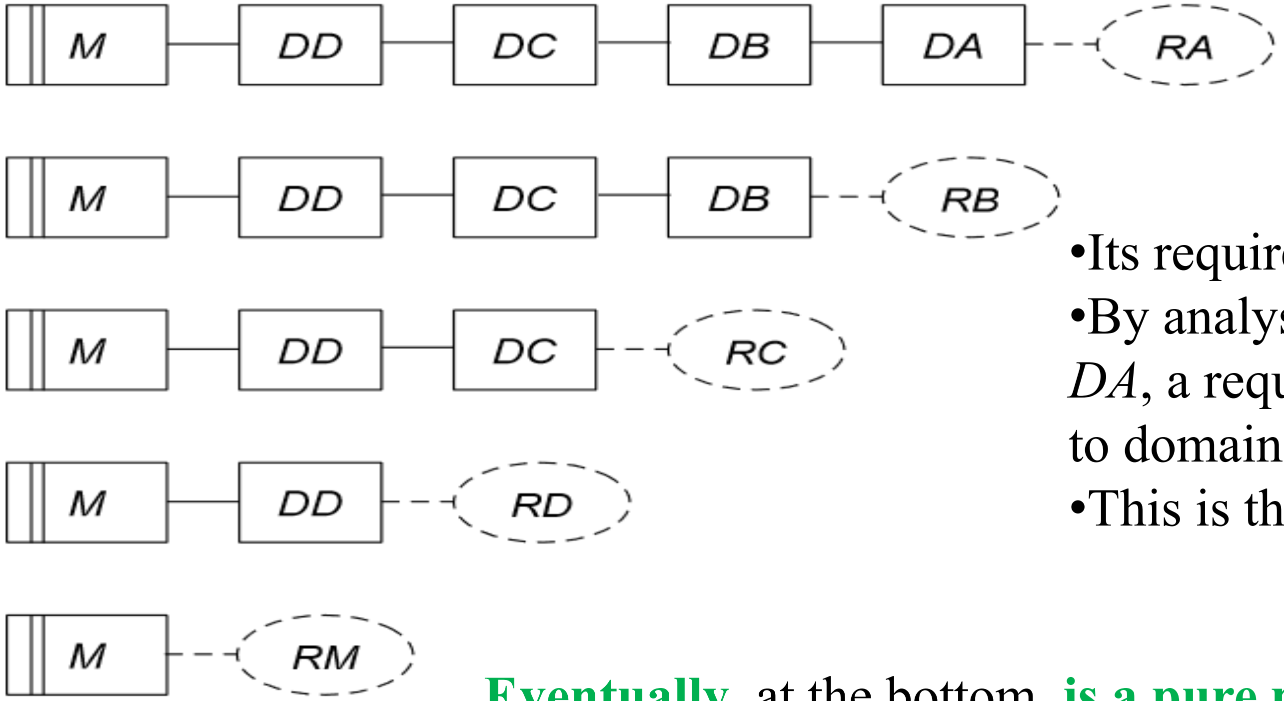
Operator[B]: OnBut, OffBut
Layout Designer[B]: Edit Layout Commands
Destination Informant[B]: Edit Destination-bin Commands

An Real Example: Decide Requirements Reference



Problem Propagation: When Entity is Too Far

- You can think of any problem [expressed in PF] as being somewhere on a progression towards the machine, like this:

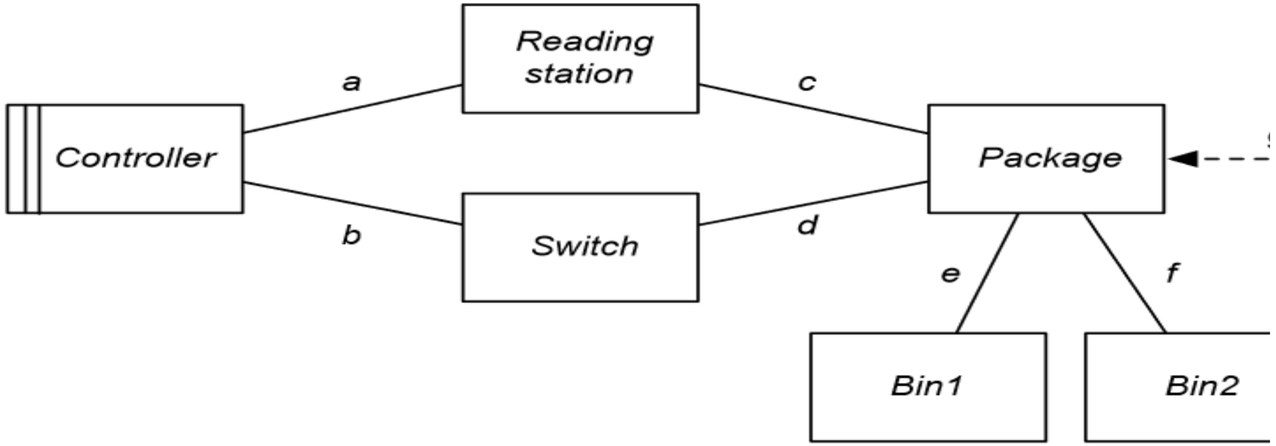


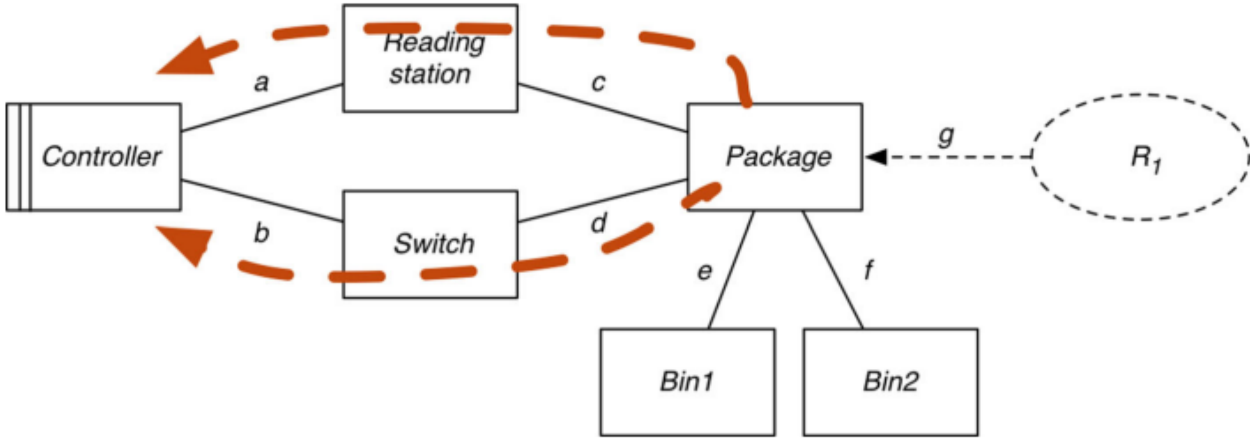
The top problem is deepest into the world [REQUIREMENTS].

•Its requirement *RA* refers to domain *DA*.
•By analysis of the requirement *RA* and the domain *DA*, a requirement *RB* can be found that refers only to domain *DB*, and guarantees satisfaction of *RA*.
•This is the requirement of the next problem down.

Eventually, at the bottom, **is a pure programming problem [SPECIFICATIONS]** whose requirement refers just to the machine and completely ignores all problem domains."

Causal Pairs Enable the Removal





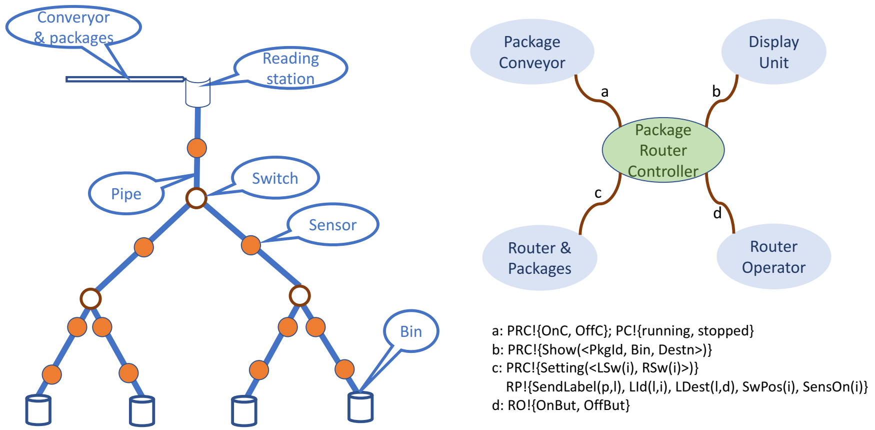
Name	Description
R_1	If the package's destination is *pkgDst*, with *pkgDst=left* or *pkgDst=right*, and the package enters the reading station (i.e. ↑ *inRS* occurs), then the package should enter the appropriate bin (i.e. either ↑ *inBin1* or ↑ *inBin2* occurs) after $x + y + z + w$ seconds
R_2 (by rule CET(6)r)	If the package's destination is *pkgDst*, with *pkgDst=left* or *pkgDst=right*, and the package shares *pkgDst* with the reading station (i.e. *share*(*pkgDst*) occurs), then the package should enter the appropriate bin (i.e. ↑ *inBin1* or ↑ *inBin2* occurs) after $x + y + z + w$ seconds
R_3 (by rule ITOr)	If the package's destination is *pkgDst*, with *pkgDst=left* or *pkgDst=right*, and the reading station reads *pkgDst* (i.e. *share*(*pkgDst*) occurs), then the package should enter the appropriate bin (i.e. either ↑ *inBin1* or ↑ *inBin2* occurs) after $x + y + z + w$ seconds
R_4 (by rule CTE(7)r)	If the package's destination is *pkgDst*, with *pkgDst=left* or *pkgDst=right*, and the reading station sends *pkgDst* to the controller (i.e. *send*(*pkgDst*) occurs), then the package should enter the appropriate bin (i.e. either ↑ *inBin1* or ↑ *inBin2* occurs) after $x + y + z + w$ seconds
R_5 (by rule ITOr)	If the package's destination is *pkgDst*, with *pkgDst=left* or *pkgDst=right*, and the controller receives *pkgDst* (i.e. *send*(*pkgDst*) occurs), then the package should enter the appropriate bin (i.e. either ↑ *inBin1* or ↑ *inBin2* occurs) after $x + y + z + w$ seconds
R_6 (by rule ETC(7)c)	If the package's destination is *pkgDst*, with *pkgDst=left* or *pkgDst=right*, and the controller receives *pkgDst* (i.e. *send*(*pkgDst*) occurs), then the package should enter the switch (i.e. ↑ *inSW* occurs) with the switch state appropriately set (i.e. either *swState*(left) or *swState*(right)), depending on the value of *pkgDst* after $x + y$ seconds
R_7 (by rule RD(3)c)	Assuming the behaviour of *Bin1* and *Bin2*, if the package's destination is *pkgDst*, with *pkgDst=left* or *pkgDst=right*, and the controller receives *pkgDst* (i.e. *send*(*pkgDst*) occurs), then the package should enter the switch (i.e. ↑ *inSW* occurs) with the switch state appropriately set (i.e. either *swState*(left) or *swState*(right)) depending on the value of *pkgDst* after $x + y$ seconds
R_8 (by rule ITOc)	Assuming the behaviour of *Bin1* and *Bin2*, if the package's destination is *pkgDst*, with *pkgDst=left* or *pkgDst=right*, and the controller receives *pkgDst* (i.e. *send*(*pkgDst*) occurs), then the switch observes the package entering (i.e. ↑ *inSW* occurs), with the switch state appropriately set (i.e. either *swState*(left) or *swState*(right)), depending on the value of *pkgDst* after $x + y$ seconds
R_9 (by rule ETC(7)c)	Assuming the behaviour of *Bin1* and *Bin2*, if the package's destination is *pkgDst*, with *pkgDst=left* or *pkgDst=right*, and the controller receives *pkgDst* (i.e. *send*(*pkgDst*) occurs), then the switch should receive commands from the controller to set its state appropriately set (i.e. either *set*(*left*) or *set*(*right*) occurs), depending on the value of *pkgDst* after $x + y$ seconds
R_{10} = (by rules RD(3)c)	Assuming the behaviour of *Bin1*, *Bin2* and *Package*, if the controller receives *pkgDst* (i.e. *send*(*pkgDst*) occurs), with *pkgDst=left* or *pkgDst=right*, then the switch should receive commands from the controller to set its state appropriately set (i.e. either *set*(*left*) or *set*(*right*) occurs), depending on the value of *pkgDst* after $x + y$ seconds
R_{11} = (by rules OTIc)	Assuming the behaviour of *Bin1*, *Bin2* and *Package*, if the controller receives *pkgDst* (i.e. *send*(*pkgDst*) occurs), with *pkgDst=left* or *pkgDst=right*, then the controller should issue commands to set the switch state appropriately set (i.e. either *set*(*left*) or *set*(*right*) occurs), depending on the value of *pkgDst* after $x + y$ seconds
R_2 = (by rules RD(1)c&(2)r)	Assuming the behaviour of *Bin1*, *Bin2*, *Package*, *Switch* and *Reading station*, if the controller receives *pkgDst* (i.e. *send*(*pkgDst*) occurs), with *pkgDst=left* or *pkgDst=right*, then the controller should issue appropriate commands (i.e. either *set*(*left*) or *set*(*right*) occurs), depending on the value of *pkgDst* after $x + y$ seconds

Projection: Including Complete Situation and Deriving the Separat Sub-problems

The idea:

- Requirements may interleaving. We need decouple them into smaller ones for identifying the function points.
- Projecting the desired environments changes upon the requirements can get a set smaller requirements

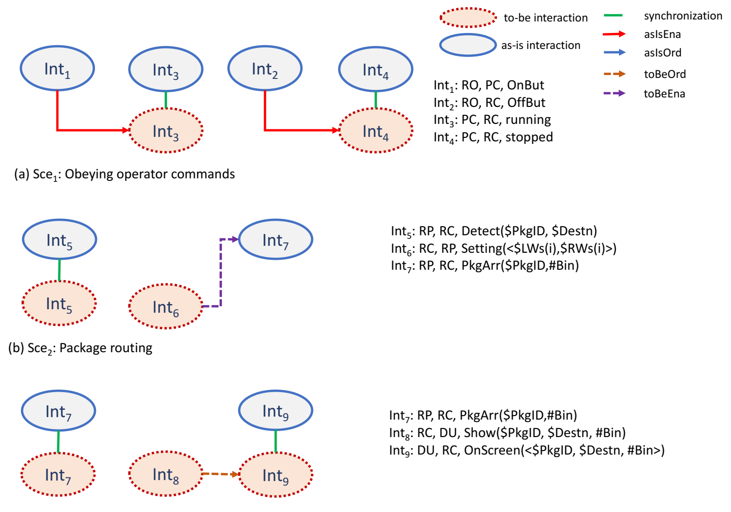
Three User Stories



(a) A Schematic Picture

(b) Initial Context Diagram

Example Context Diagram

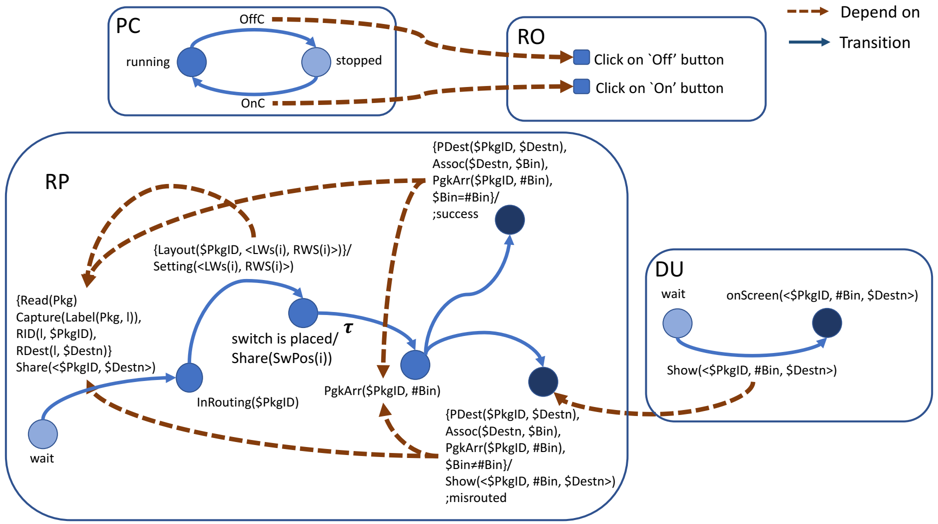


Initial Requirements:
Desired Relationships
among the Interactions

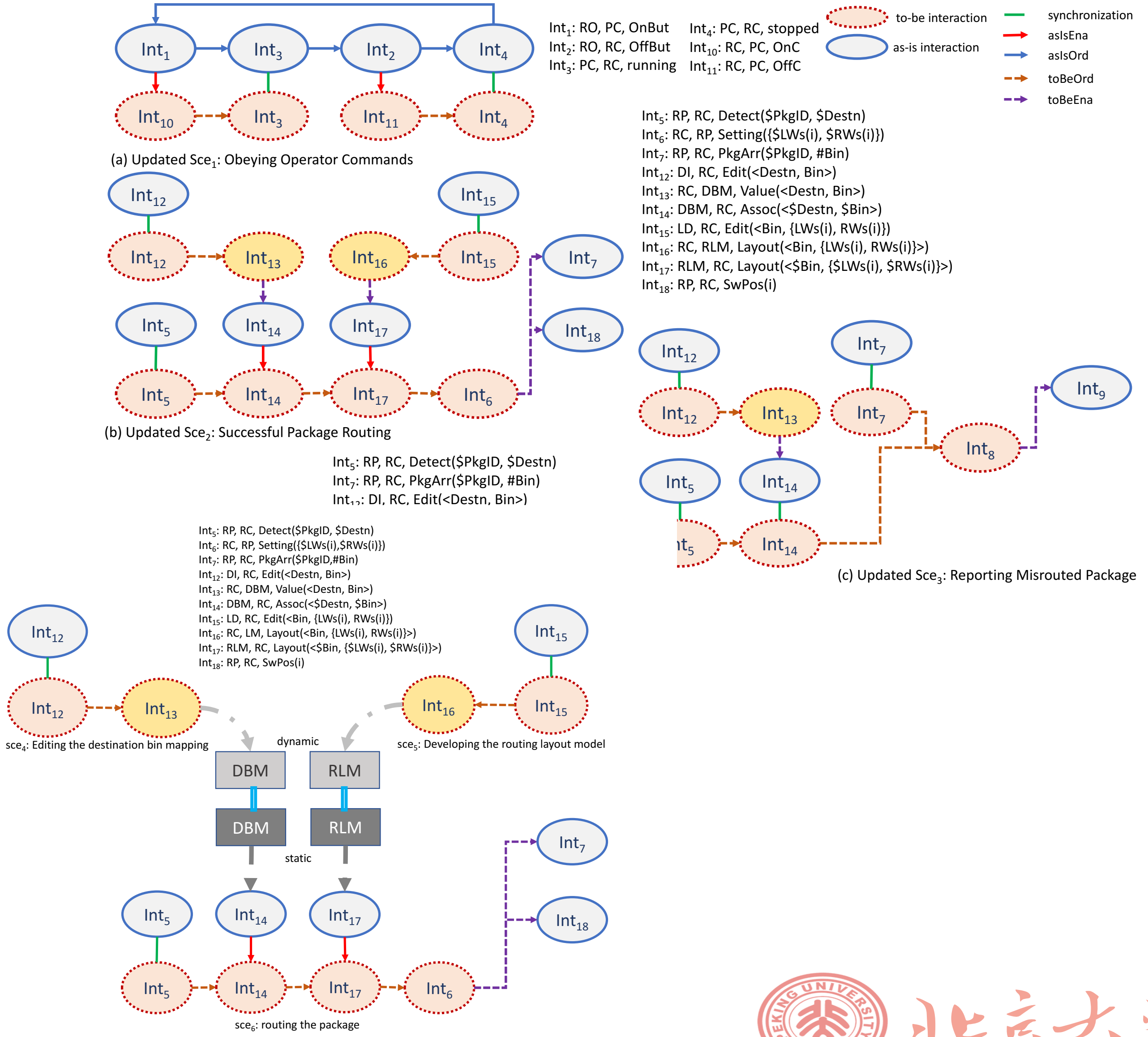
M. Jackson (2001), Problem Frames: Analyzing and Structuring Software Development Problems, Addison-Wesley

Projection based Refinement

Physical Device Behaviors



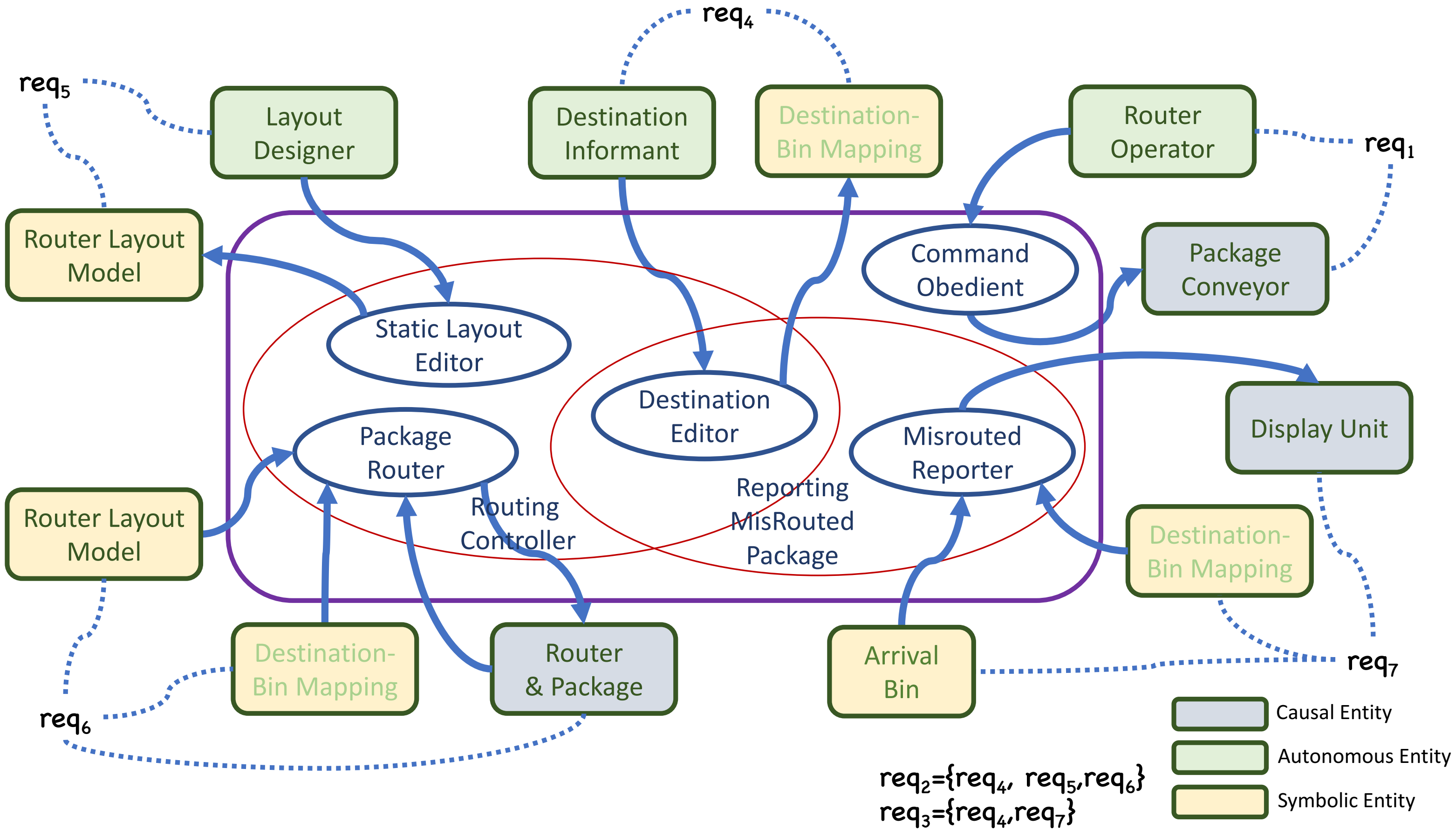
Domain Environment Ontology

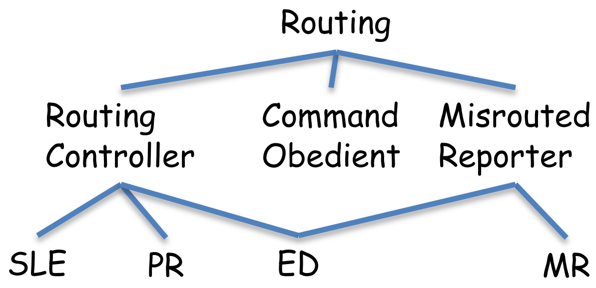


Two Strategies:

- Include missing interactions for making the interaction scenarios complete
- Separate multiple roles by including new entities

Reasoning: Projection and Refinement





Agenda

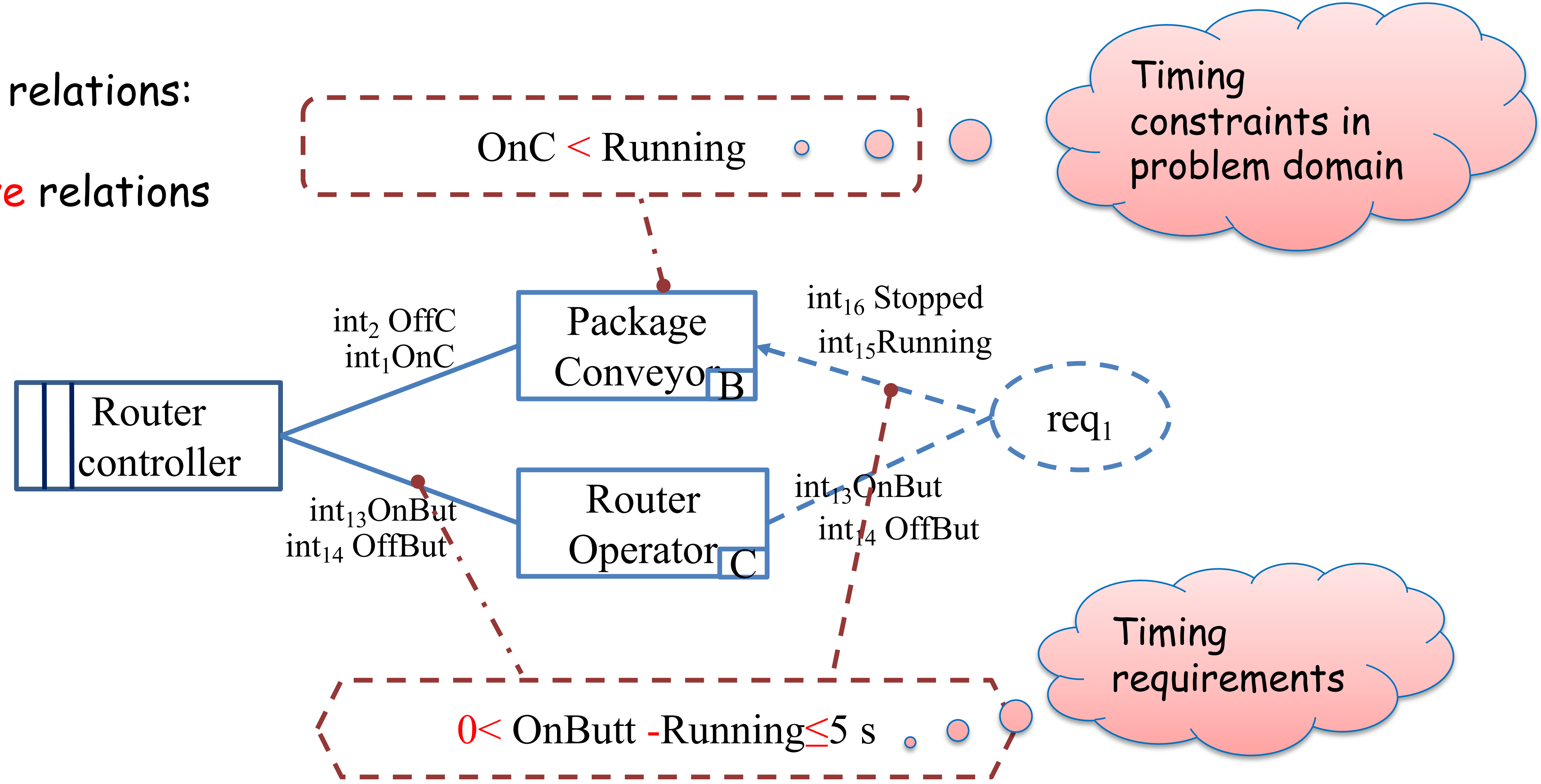
- Cyber-Physical Systems bring Challenges
- Environment Modeling based Requirements Engineering
- Some Non-functional Requirements
- More Efforts and Further Work

Timing Requirements

- especially important
  - for real-time safety-critical systems
- as prominent as functional requirements.

Attaching Timing Constraints

1. Qualitative relations: temporal
2. Quantitative relations



Extension and Benefits

Language extension

- express timing constraints with CCSL(Clock Constraint Specification Language)

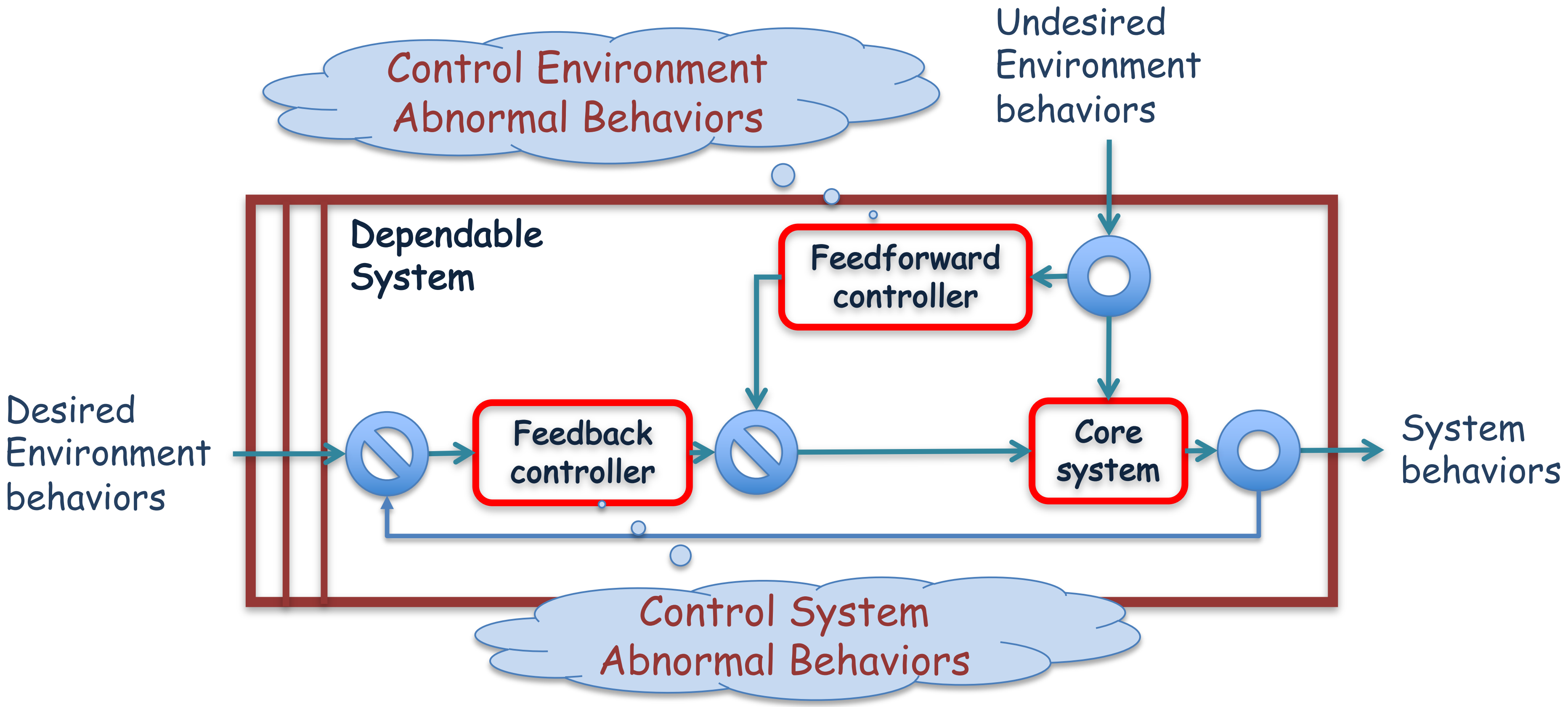
Timing requirements derivation

- the problem diagrams and scenario graphs
- from the state machines of causal domains

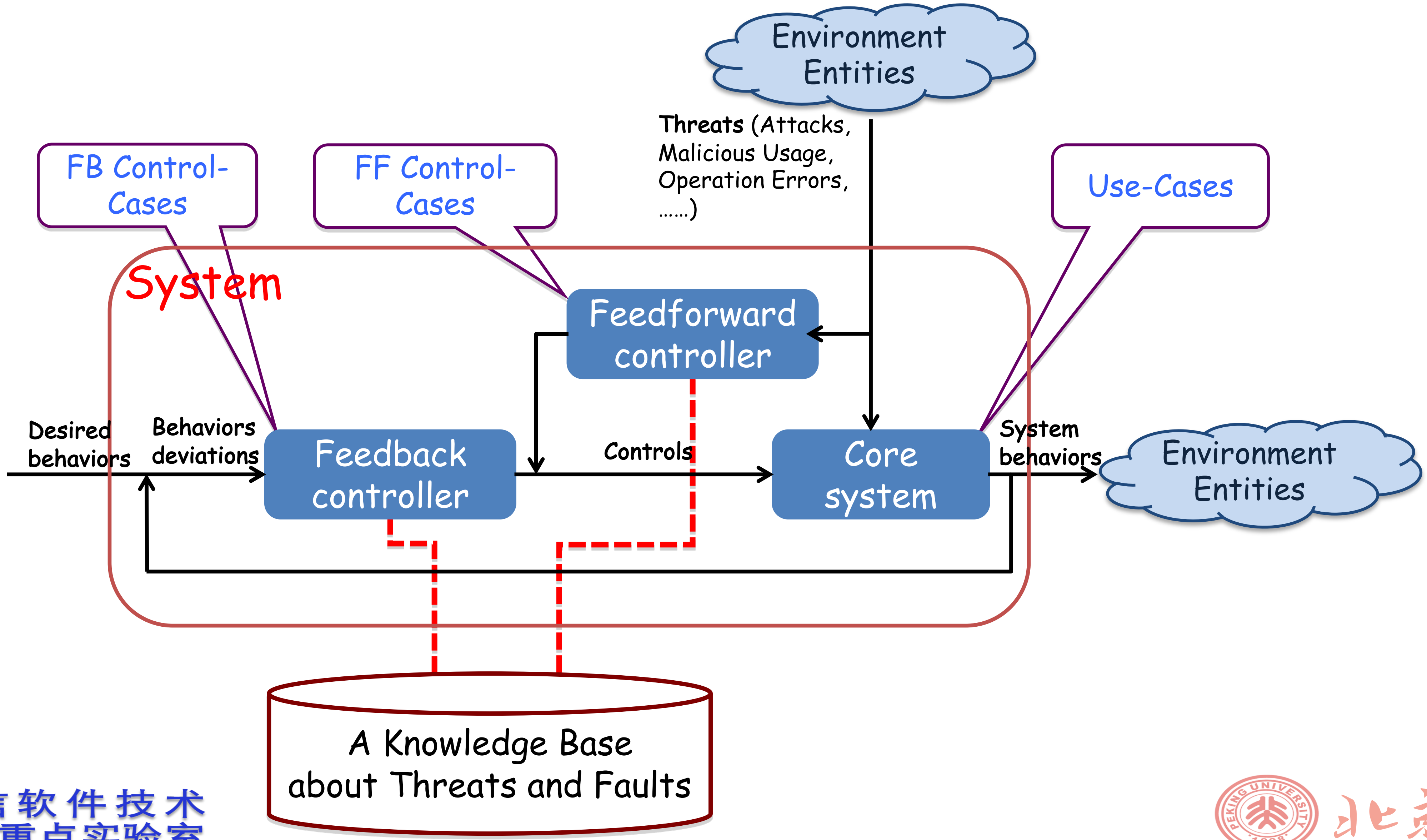
Timing requirements verification

•SMT model checker: MyCCSL （Z3, CVC4）

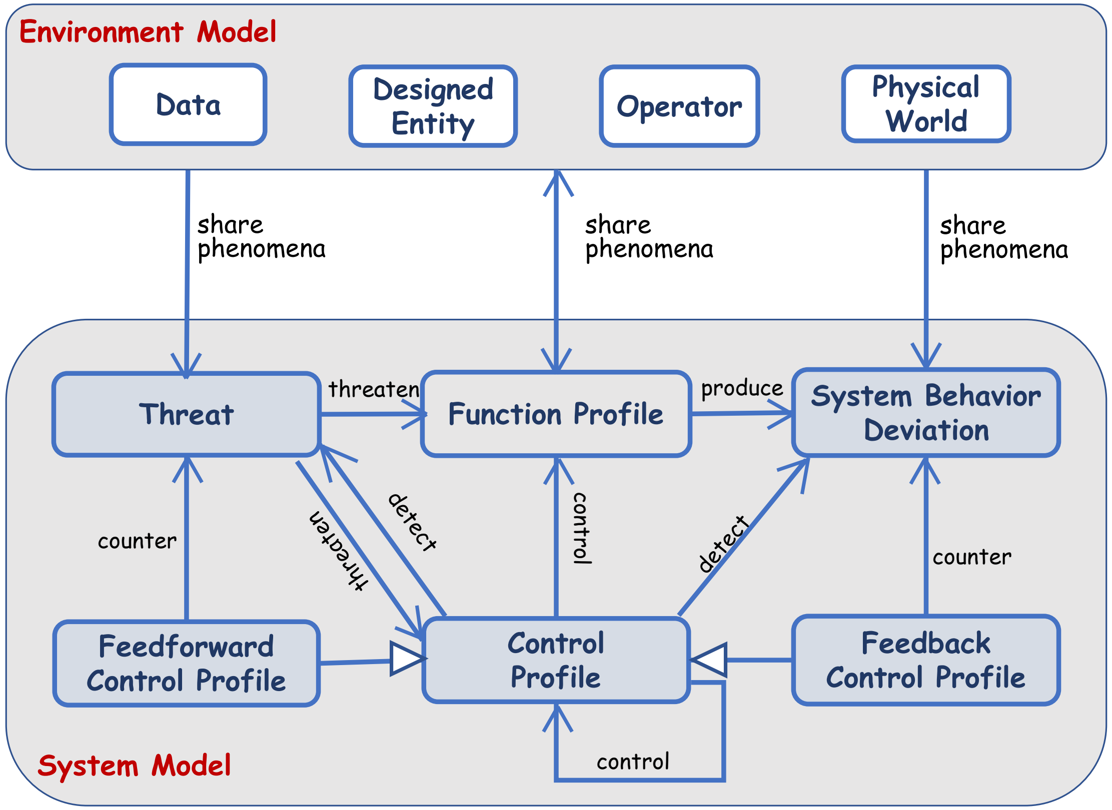
Security/Safety Requirements: Including Controllers



Security/Safety Requirements: Including Controllers



Conceptual Graph and Domain Knowledge



Environment Entity / System Asset / Interaction / Phenomenon	Undesired Feature	Implied Concern
External / Internal Autonomous Entity	Has malicious intent to access	Authorization Concern
System / System Component	Produce unexpected behavior / output; Failure	Fault tolerance Adaptation Concern
External Entity	Trigger known attack / virus	Security Concern
External Symbolic Entity	Has different levels of sensitiveness	Privacy Concern
External Physical Device	Produce unexpected input	Robustness Concern
External Entity	Valuable or Critical	Safety Concern
Connection	Be lost, Be tampered	Security Concern
Interactive Environment	Uncertain	Adaptation Concern

Entity / Threat / Countermeasure

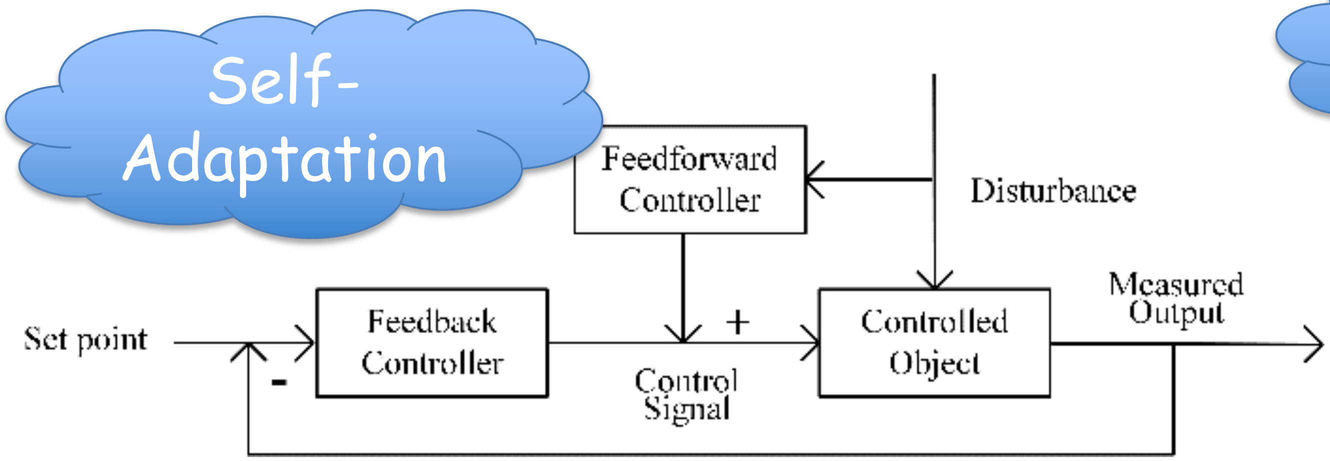
Featured Entity / Service / Interaction	Threat	Countermeasure
Private/sensitive data	Information disclosure in transmission or service delivering	Strong authorization to data accessing; Strong encryption to the data; Communication link securing with protocols that provides message confidentiality
High available system service	Denial of service by malicious user	Resource and bandwidth throttling; Input validation and filtering
Malicious operator	Spoofing for illegal usage	Strong authentication; Strong encryption to operators' login data; Authentication cookie protection with Secure Sockets Layer
Critical / Valuable data, Device, or Interactor that can result in big loss	Tampering with data in transmission or data storage and/or processing	Data hashing and signing; Digital signatures; Strong authorization; Tamper-resistant protocols across communication links; Communication link securing with protocols that provides message integrity
	System fault or behavior deviation	Oracle-based system behavior checking
Open system/service with highly-desired availability	Virus, e.g. Trojan horse, Worms, ……	Block all unnecessary ports at the firewall and host; Disable unused functionality; Harden weak, default configuration settings
	…… MicroSoft, Improve Web Application Security; Common Criteria for Information Technology Security Evaluation;	

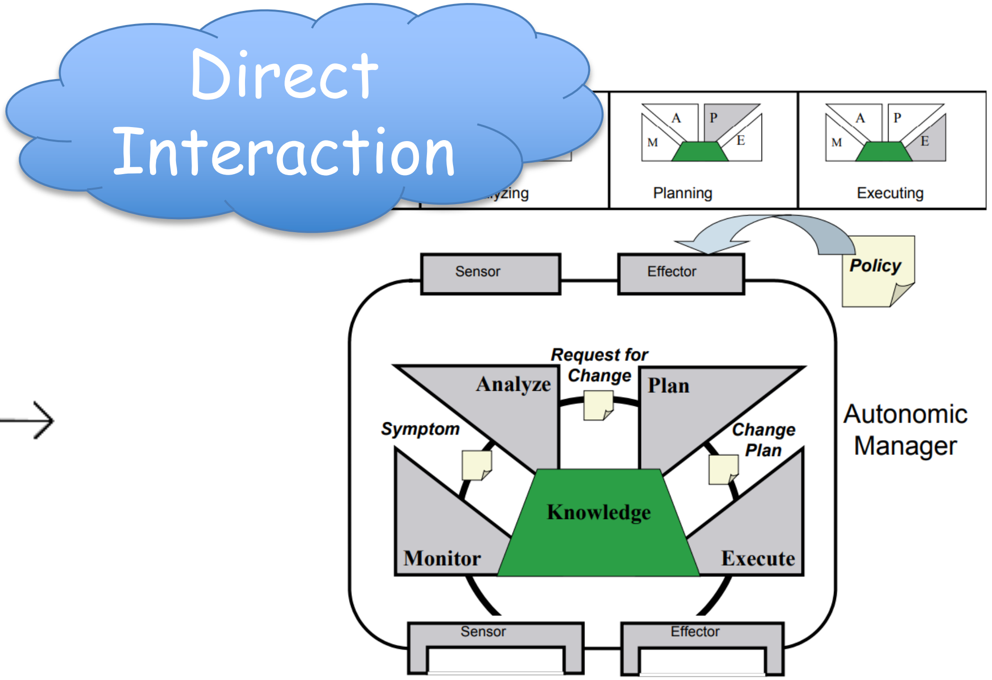
Reality is Uncertainty: A Necessary Concern

- Strategy is Autonomy or Situation-Aware Design
  - Be capable of robust, long-term autonomy requiring
    - minimal or no human operator intervention
    - in the face of
      - Uncertain, unanticipated, and dynamically changing situations
- Direct Interaction means
  - Combine
    - perception, cognition, communication, and actuation

    to sense and operate the reality
- This leads to three concerns:
  - architecture, environment modeling, situation awareness

Control Loop and MAPE-K





- Feedforward controller: takes into account the external *Disturbances* to produces a *Control Signal* that compensates for the *Disturbances*
- Feedback controller: computes *Control Signal* based on the deviation between desired goals and corresponding *Measured Output*

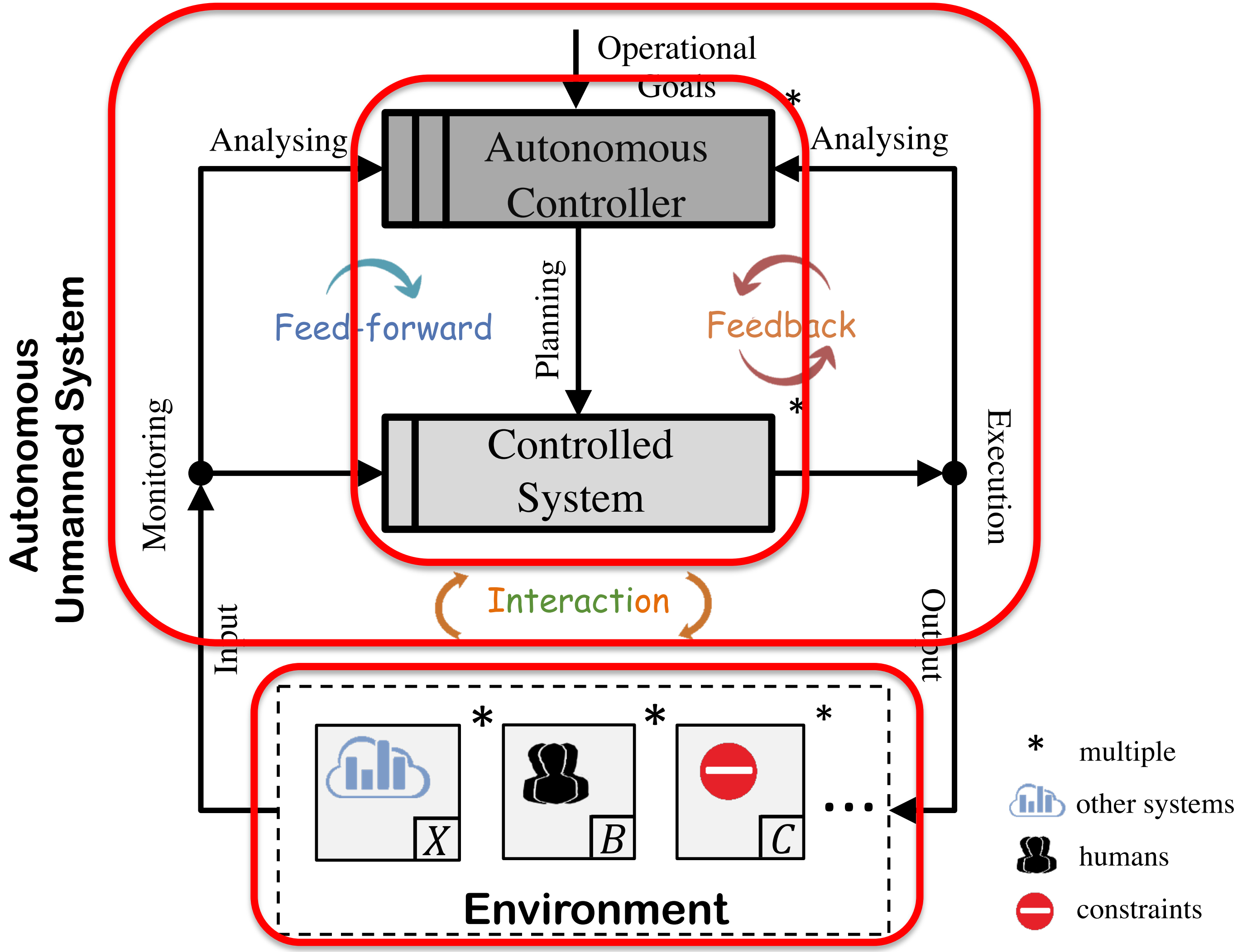
[Shevtsov 2017]

Self-adaption mechanism

- MAPE/MAPE-K control loop
- Components: sensor, controller, effector, communication link

[IBM 2006, Lemos et al. 2010]

Environment + Control Loops + MAPE-K



- Unified Control Loops
- Separation of Services and Controllers
- Explicit Environment Modeling

Yixing Luo, Yijun Yu, Zhi Jin and Haiyan Zhao, Environment-Centric Safety Requirements for Autonomous Unmanned Systems, Proceedings of the 27th IEEE International Requirements Engineering Conference (RE'19), 2019

Situation Awareness

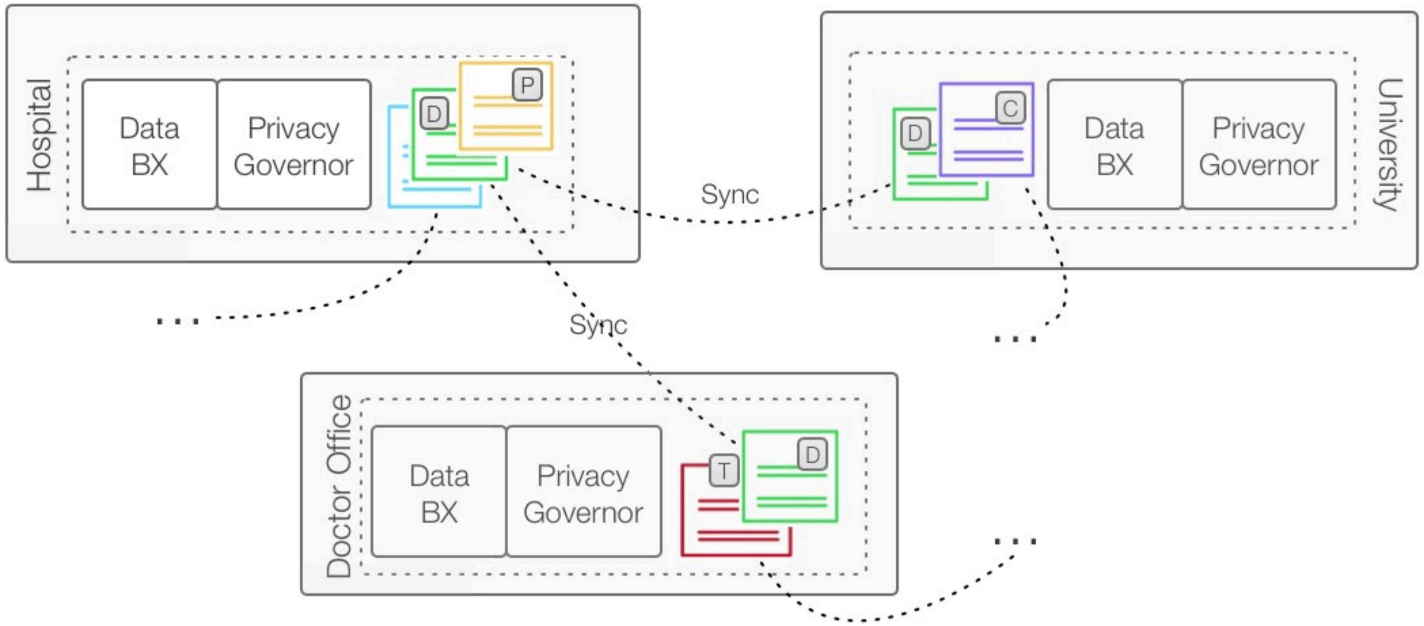
- the perception of environmental elements and events with respect to time or space
- the comprehension of their meaning, and
- the projection of their future status

- Not only aware, but also take actions

Distributed Data Privacy Protection

Data flows always respect privacy policies

- Each node sets its own data outflow or inflow privacy policies
- All nodes need to synchronize the shared data
- How to define and deploy?



Smart Hospital
Runtime Deployment

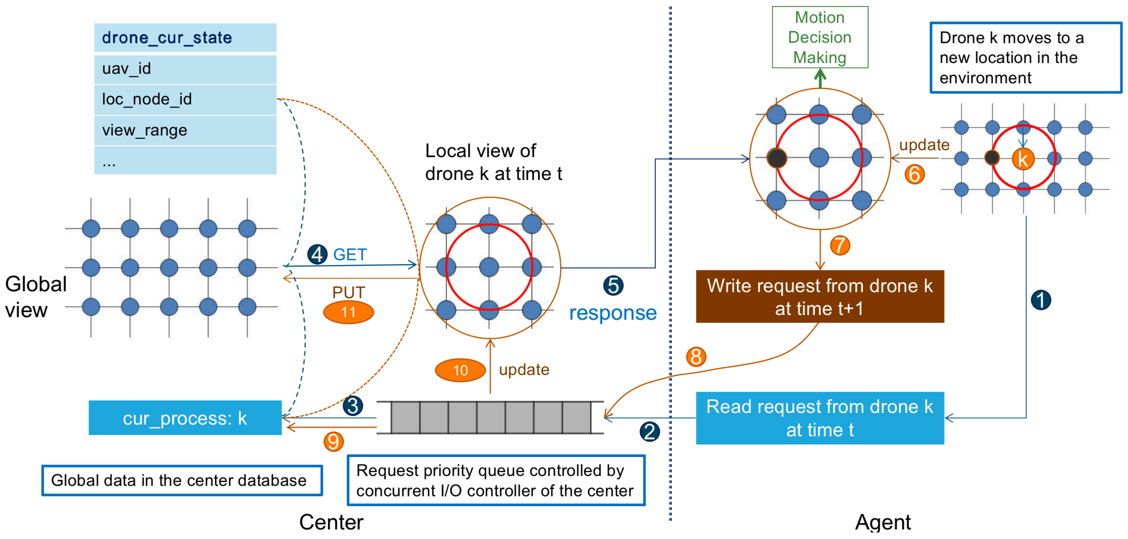
Nianyu Li, Christos Tsigkanos, Zhi Jin, Schahram Dustdar, Zhenjiang Hu and Carlo Ghezzi, POET: Privacy on the Edge with Bidirectional Data Transformations, 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom 2019): 1-10, 2019.

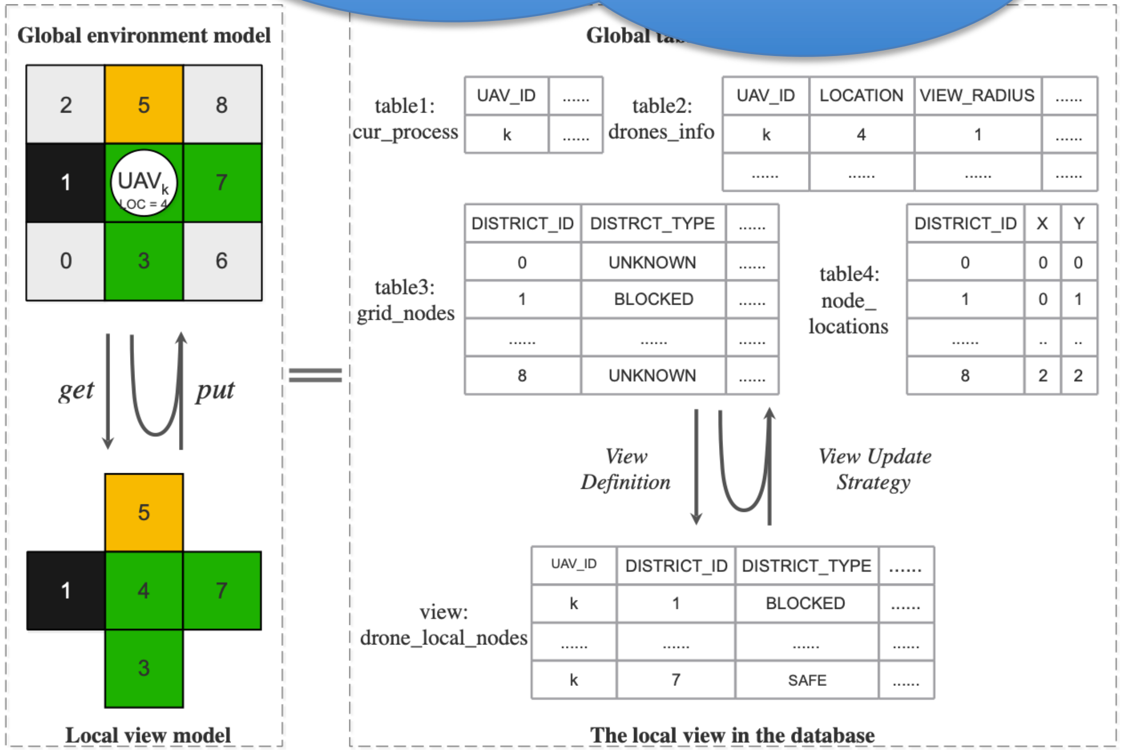
Location, Task, Resource Aware Data S

Different nodes needs different shared data depending on their location, their resource and their tasks

Victim Searching

Runtime Decision Making with





Scene and Risk aware Adaptive Configuration

Environment Friendly: Privacy Protection

Runtime Configuration with Scene Detection and Privacy Policy

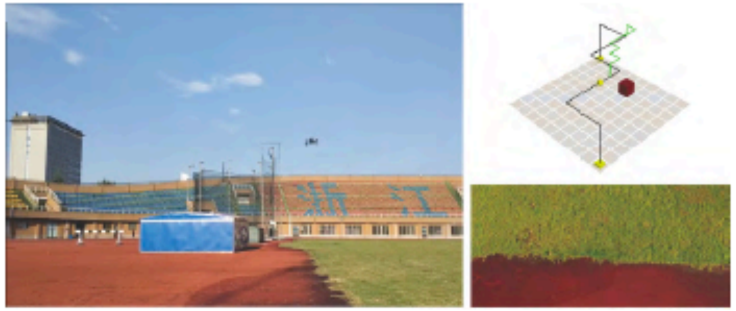
(a) The first time the private region is detected.

(b) The private region is always in sight.

(c) The UAV passes through the private region.

(a) The UAV detects the private region and re-plans its trajectory and camera orientation.

(b) The private region is avoided by changing trajectory and camera orientation is tuned to 30°.



(c) The UAV passes through the private region, and camera orientation is tuned back to −90°.

http://re4cps.org/mainPage/home

不安全 | re4cps.org/mainPage/home

RE4CPS: Requirements Engineering for Cyber-Physical Systems

Home | Team | Documentation | Introduction | Examples | Web Help | Bugs

Cyber-Physical Systems (CPSs) connect the cyber world with the physical world through a network of interrelated elements, such as sensors and actuators, robots, and other computing devices. There are inspiring increasing number of beneficial applications in dependable sectors such as aviation, transportation, aerospace, healthcare, etc..

The inherent characteristics of CPSs pose a number of challenges to requirements engineering. Unlike normal information systems, CPSs need to continuously detect and adapt to the environment changes. The interactive environment becomes the first-class citizen because the features and the changing patterns in environment are must-to-be considered. Moreover, in such systems, many non-functional requirements are environment related, like timing, safety, security, and privacy requirements.

We provide an environment modelling based approach to engineering the requirements of CPSs. Extending the Problem Frames representations, this approach structures the model of the environmental elements and provides analysis methods for deriving and specifying requirements.

This website aims to deliver a few supporting tools that assist the modelling and verification of the system specification. These tools includes:

Problem description

Problem progression

Problem projection

Timing requirements

Privacy requirement

Security requirements

They are developed in collaboration among the Peking University, China, East China Normal University, China, Guangxi Normal University, China and The Open University in UK.

The Open University 50 YEARS

Agenda

- Cyber-Physical Systems bring Challenges
- Environment Modeling based Requirements Engineering
- Some Non-functional Requirements
- More Efforts and Further Work

More Challenges

- How to identify and model the system bound ?
  - Human intention and domain knowledge
  - Static and dynamic
  - Normal use cases misuse cases, malicious use cases, ……
  - Contract-Based Model
  - ……
- How to deal with the time and space consistency among the environment entities in the real world
  - Space: The space and motion of communicating agents, Robin Milner
  - Time: Real time, time constraints, …….
  - Space and time ?
  - ……

Conclusions and Future Work

- Tighter interaction with the reality brings in the complexity of solution
  - Mixture of continuous and discrete components
  - Discretization and verification of the hybrid models
- Research topics in RE:
  - New patterns/frames for hybrid problem, serving for
    - Definition of the problem
    - Structure mechanism of problem analysis

Conclusions and Future Work

- Mobility of moving entity along the space topology means the location or context changes
  - Changes of the relationship
  - Location-aware smartness
  - Prediction of the changes
- Research topic in RE:
  - Model spaces and movement reactive rules
  - Time-dependent behavior and Time-Space Consistency

Conclusions and Future Work

- Environment-friendly, endogenous safety and security
  - Environment risk assessment before taking actions
- Research topic in RE
  - Environment modeling and simulation
    - Not only capture the dynamics but also the value- and risk-related properties
  - Compensating the lack of human intervention by adaptive control
    - System behaviors are controlled by: the action-taking, the decision-making and the policy learning for allowing coping with uncertainty

Acknowledgements

- National Grand Fundamental Research Program of China under Grant No. 2009CB320701, Ministry of Science and Technology
- Key Project of National Natural Science Foundation of China under Grant No. 90818026

- Thanks to the students and colleagues who contribute to these projects

Publications

- Zhi Jin, Environment Modeling-based Requirements Engineering for Software Intensive Systems, Elsevier, Morgan Kaufmann Publisher, 2018
- Yixing Luo, Yijun Yu, Zhi Jin and Haiyan Zhao, Environment-Centric Safety Requirements for Autonomous Unmanned Systems, Proceedings of the 27th IEEE International Requirements Engineering Conference (RE'19), 2019
- Xiaohong Chen, Zhiwei Zhong, Zhi Jin, Min Zhang, Tong Li, Xiang Chen and Tingliang Zhou, Automating Consistency Verification of Safety Requirements for Railway Interlocking Systems, Proceedings of the 27th IEEE International Requirements Engineering Conference (RE'19), 2019
- Lionel Montrieux, Naoyasu Ubayashi, Tianqi Zhao, Zhi Jin and Zhenjiang Hu, Bidirectional Transformations for Self-Adaptive Systems, Communications of NII Shonan Meetings, Engineering Adaptive Software System 2019: 95-114, Springer, 2019
- Nianyu Li, Christos Tsigkanos, Zhi Jin, Schahram Dustdar, Zhenjiang Hu and Carlo Ghezzi, POET: Privacy on the Edge with Bidirectional Data Transformations, 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom 2019):
- Christos Tsigkanos, Nianyu Li, Zhi Jin, Zhenjiang Hu and Carlo Ghezzi, On Early Statistical Requirements Validation of Cyber-Physical Space Systems, 2018 ACM/IEEE 4th International Workshop on Software Engineering for Smart Cyber-Physical Systems (ACM SEsCPS@ICSE 2018): 13-18
- Tianqi Zhao, Wei Zhang, Haiyan Zhao, Zhi Jin: A Reinforcement Learning-Based Framework for the Generation and Evolution of Adaptation Rules. ICAC 2017: 103-112
- Yixing Luo, Yijun Yu, Zhi Jin, Haiyan Zhao, Environment-Centric Safety Requirements for Autonomous Unmanned Systems, RE@NEXT 2019

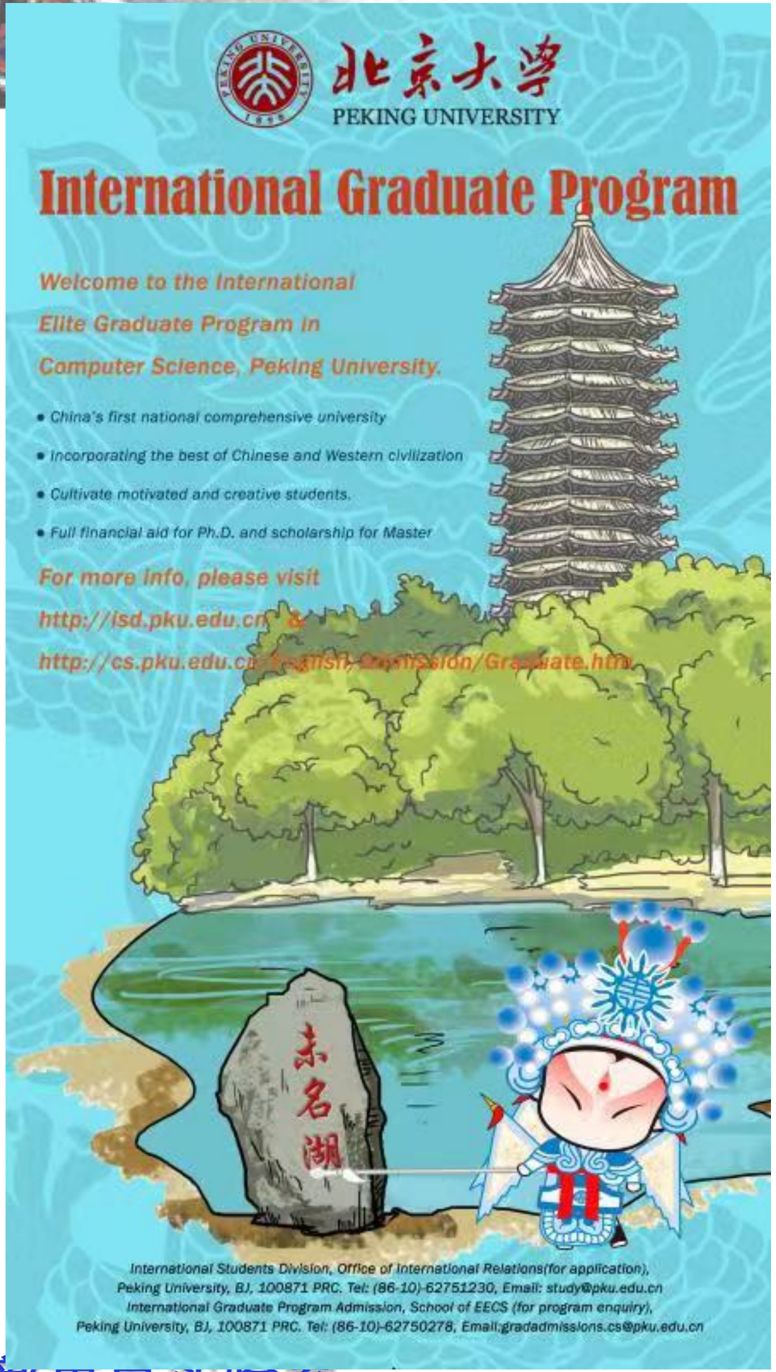
Acknowledgements

- Key Projects of National Natural Science Foundation of China under Grant Nos. 90818026, 61620106007, 61751210
- National Grand Fundamental Research Program of China under Grant No. 2009CB320701, Ministry of Science and Technology

国家自然科学基金委员会
National Natural Science Foundation of China

中华人民共和国科学技术部
Ministry of Science and Technology of the People's Republic of China

- Thanks are due to
  - Collaborators: Schahram Dustdar, Carlo Ghezzi, Zhenjiang Hu, Lionel Montrieux, Christos Tsigknos, Naoyasu Ubayashi, Yijun Yu, Haiyan Zhao, Wei Zhang
  - Students: Nianyu Li, Yixing Luo



Thanks
For Your Attentions

高可
教育部重点实验室

北京大學