

ER2005

24-28  
October  
2005



7<sup>th</sup> International Bi-Conference Workshop on  
**Agent-Oriented Information Systems @ ER'2005**

# Modeling **D**ynamic **E**ngineering **D**esign **P**rocesses in PSI

- ☑ **Vadim Ermolayev**      Zaporozhye National University, Ukraine
- Natalya Keberle**
- Oleg Karsayev**              Saint Petersburg Institute for Informatics and Automation, RAS, Russia
- Vladimir Samoylov**
- Eyck Jentsch**                  Cadence Design Systems GmbH, Germany
- Wolf-Ekkehard Matzke**

cadence®



Thursday, October 27, 2005


# The Outlook

---

- What is a **D**ynamic **E**ngineering **D**esign **P**rocess?
- What makes EDP Dynamic?
- The focus: How to assess (and increase) the **Productivity** of a ...?
- What do we need to **model** a DEDP and a Design System?
  - Actors and Teams
  - Tasks, Activities, and Dependencies
  - Goals, Design Artifacts
- Some results obtained so far in PSI
- Conclusions and future work

# What is a **Dynamic Engineering Design Process?**

---

- A **DEDP** is the process of aiming a **weakly defined engineering design workflow** to achieve its goal in an optimal way in the terms of:
  - Result **Quality** and
  - Gained **Productivity**
- A **DEDP** is dynamic because:
  - In PSI we consider that **workflow formation occurs at the run time**
  - Reasons/Factors: to be discussed 

# Factors Providing Dynamics

---

- Different **Actors** have **different knowledge** and **capabilities** wrt the parts of a **DEDP**
  - Requires **distributed planning** at run time
- **Task decomposition** is performed **subjectively** and **partially**
  - **Implies** Resulting Activities may be sequenced and conveyed differently - **distributed scheduling** at run time
- No of Activity **Iterations** is **not pre-defined** (quality checks, bad results at prior or intermediate steps)
  - **Implies**: run-time **re-planning** and **re-scheduling**
- **Activity duration** depends on the available **Capacity** of the Actor (different)
  - **Implies** run-time **re-scheduling**
- Actors are **not assigned in advance** - **Contracted** when needed (runtime)
  - Requires **Negotiation** Mechanisms

# DEDP Productivity Assessment

- **Definition:** Productivity is the amount of **output** created (in terms of goods produced or services rendered) per unit **input** used\* (by a **system** in a process)

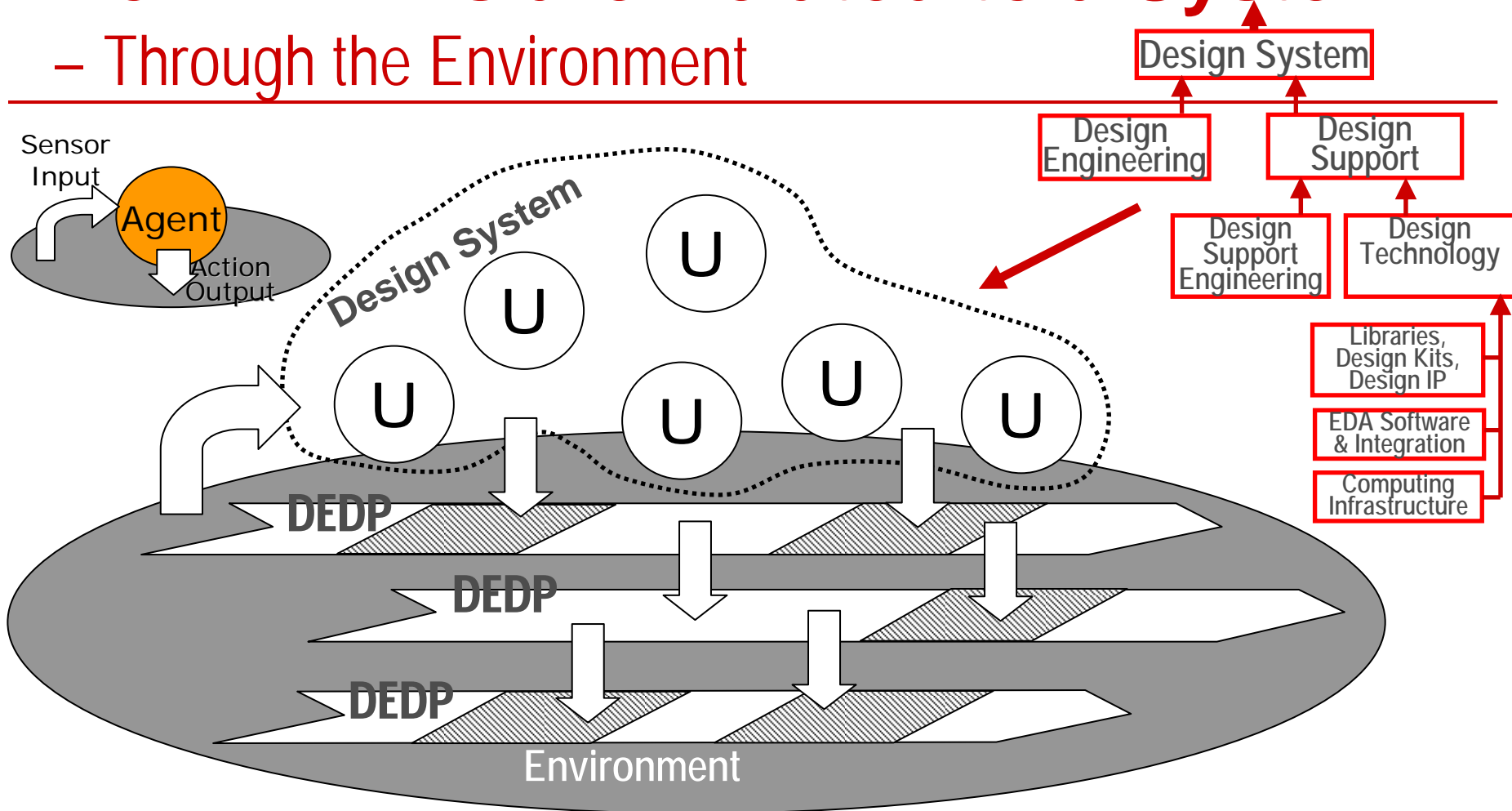


- Productivity of? A System? A Unit? An Organization?  
A Process?
- Who does the work? How DEDPs are related to a System?
- How to measure (& compare) inputs (often money) and outputs (sometimes the knowledge which is negative)
  - E.g.: Is it productive to spend 20MY for getting clear understanding that the approach was fake?

\*Wikipedia, <http://www.wikipedia.org/>

# How DEDPs are Related to a System?


- Through the Environment



Action Output is NOT the **OUTPUT** in the Productivity model  
The **OUTPUT** is the Design Value Assessment of the Action Output

# Productivity Questions: the Answers

---

- **Productivity** of the **System** AND the **Units** within the System (white box)
  - An **Organization** is the subclass of a System
  - A **Unit** is the subclass of a System ← 
  - A **System** is the COLLECTION of Units
- The **Unit** (and, sometimes, the whole **System**) does the work
- Use the Utilitarian approach: measure in **UTILITY**
- E.g. A: **YES** – productive if having this knowledge saves 25MY for the System
  - I.e.: the UTILITY gained by the System is more than the UTILITY spent by the System

# DEDP Productivity Assessment

---

- Use the **Utilitarian** approach: measure in **UTILITY**
- The main point in **Utilities** is that they are **RELATIVE**
- **Corollary:**
  - **Productivities** are **RELATIVE** and
  - System Laws (social aspect) should be accounted in the Assessment

# Welfare-Based Productivity Measure

## Utilitarian Approach

---

- **Productivity** of a DEDP:
  - Assessed as the **accumulated productivity of the participants**
  - Measured by the number of the accumulated **Units of Welfare (UoW)** – abstract **UTILITY** units
- In these settings:
  - An **economically rational actor** (a Unit or a System modeled by an agent or a MAS) is **the locus of Utility accumulation**
  - An **actor receives** the **UoW** for:
    - Performing DEDP **(sub-)tasks**
    - Providing his **Design Solutions (DS)**
  - **Otherwise**, an actor may outsource a **(sub-)task**, or require a **DS** and **spend** his **UoW** for that

# Actors and Teams Compared by their Level of Welfare

---

- An **Actor** may be considered **more Productive** if he **receives more** and **spends less UoW**
- **In a long run** (dozens of different **DEDPs**) the **relative Productivity** of an actor may be reliably measured by the **Level of his Welfare**
- The **Productivity** of an **Organization** or a **Team** may also be **assessed** as the **sum of the Welfare** of its members
- **Important:**
  - This **productivity measure** is **invariant** to the **DEDPs** which were actually used to collect the Utility

# UoW may be Gained, Spent, or Lost through Collaboration

---

- **Collaboration** occurs when:
  - An **Actor** assigns a **(sub-)task** to its sub-ordinate by directive
  - An **Actor** contracts another actor for a **(sub-)task**
  - A **DS** of the **Actor** is re-used in different **DEDPs**
- Types of encounters:
  - Directive **assignments**
  - Contracting **negotiations**
- **Mechanisms** comprise the **protocol**, the **strategy**, and the **social norms**
  - Should be **Utilitarian** (decisions based on the **UoW**)

# DEDP Model:

## the Building Blocks

---

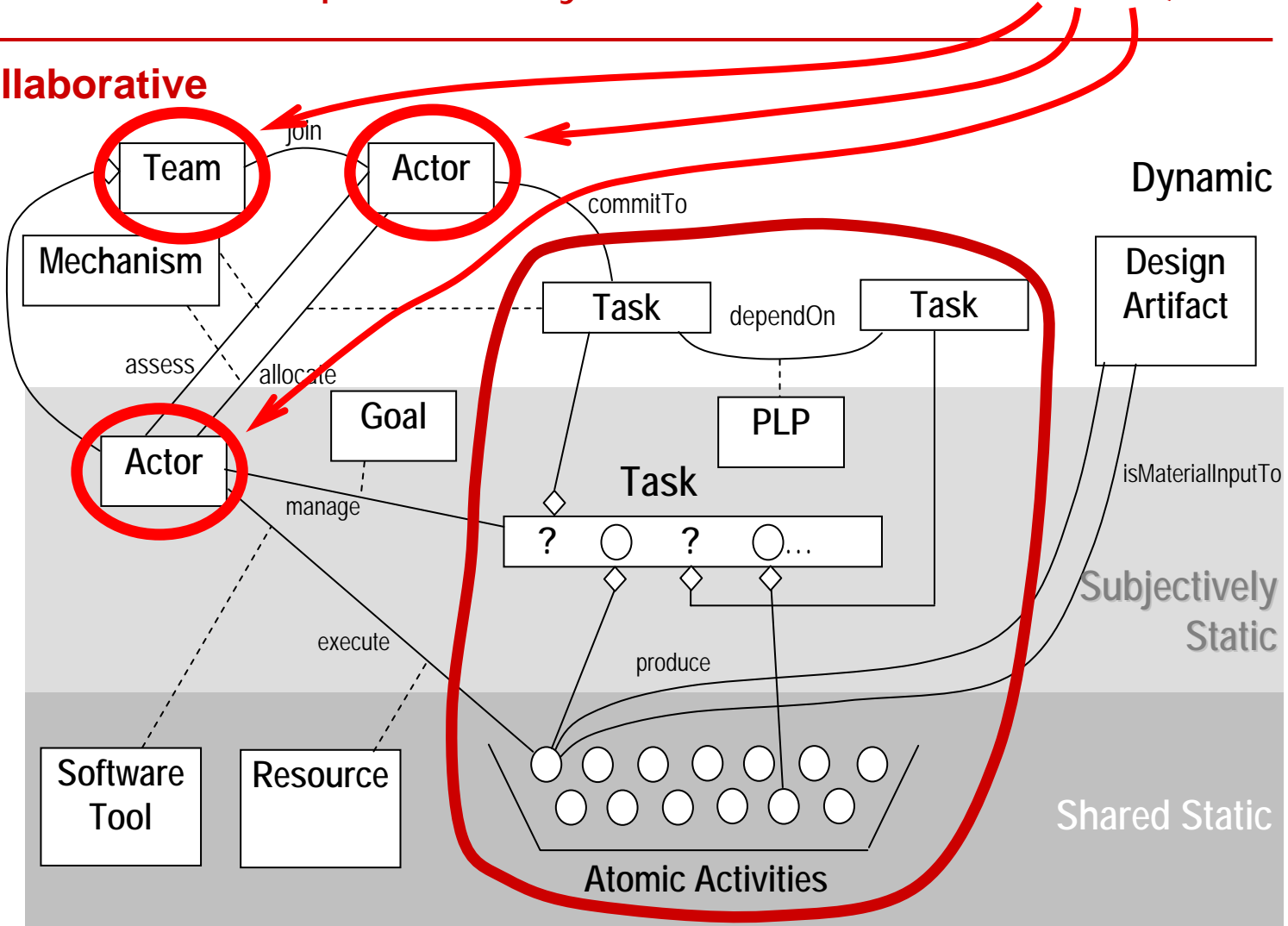
- Descriptive models (**Ontologies**) for:
  - An **Actor** (Unit)
  - A **Team** (Set of Collaborative Units + Constraints + Binding Conventions)
  - A **Process** (Tasks, Activities, Dependencies)
  - DEDP **objectives** (comprising **Design Artifacts**)
- **Software Models** (agent-based) of the same
- **Mechanisms** to arrange Actors' Collaboration:
  - **Protocols** for different encounters
  - Behavior **Strategies**

# What do we Need to Model a DEDP?

(Mind dynamics factors, productivity measure and the "Units")

A **DEDP** is a **collaborative** problem solving process

A **DEDP** is a **dynamically** and **subjectively** formed, planned and scheduled **hierarchy** of **tasks**, **subtasks** and **atomic activities** which may have **dependencies**



# What do we Need to Model a DEDP?

(Mind dynamics factors, productivity measure and the "Units")

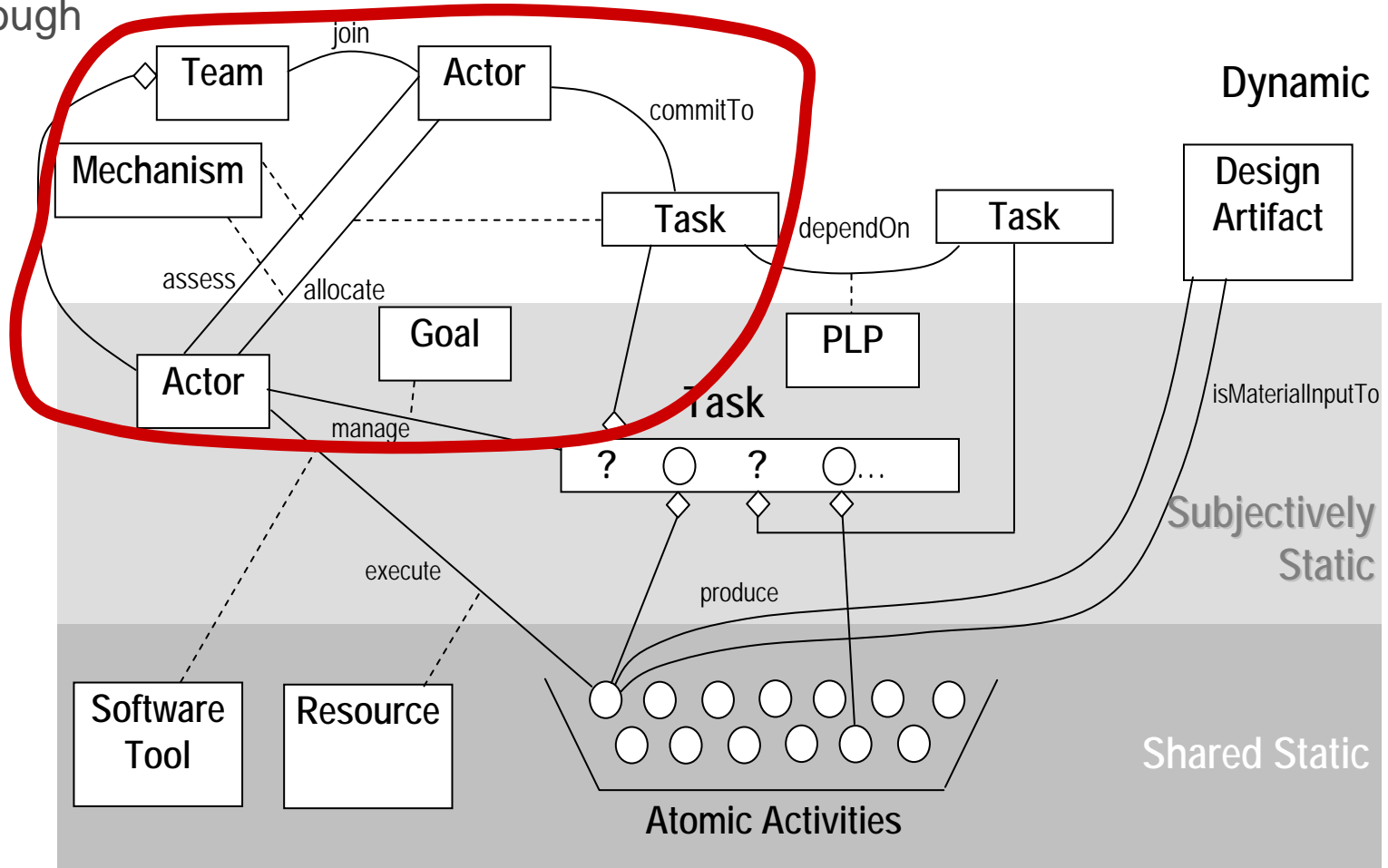
A DEDP is performed by **Actors** which collaborate in **Teams** – earn and spend their **UoW** through

- **Managing Tasks**

and

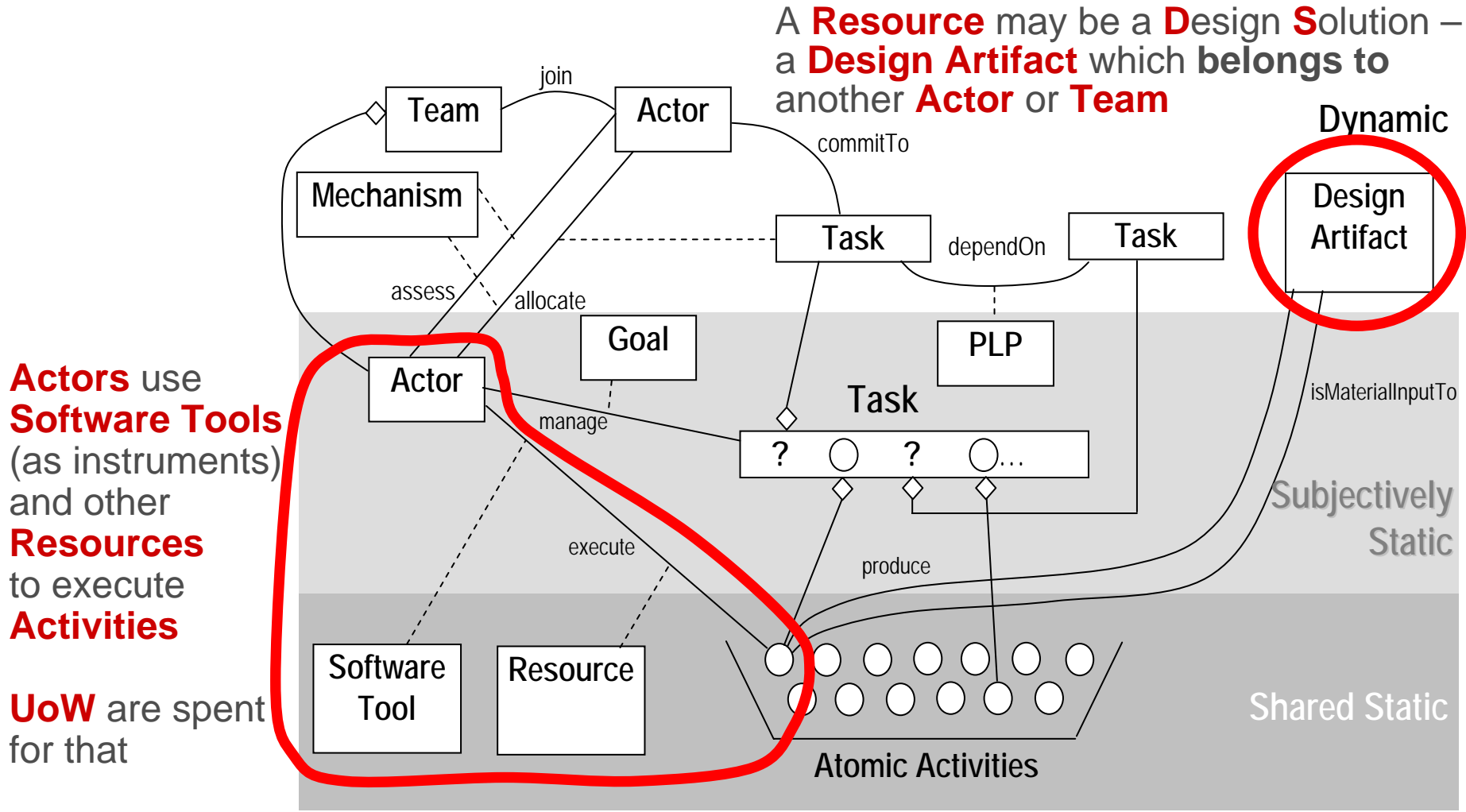
- **Executing Activities**

The **Teams** are formed by using Contracting **Mechanisms** through negotiations



# What do we Need to Model a DEDP?

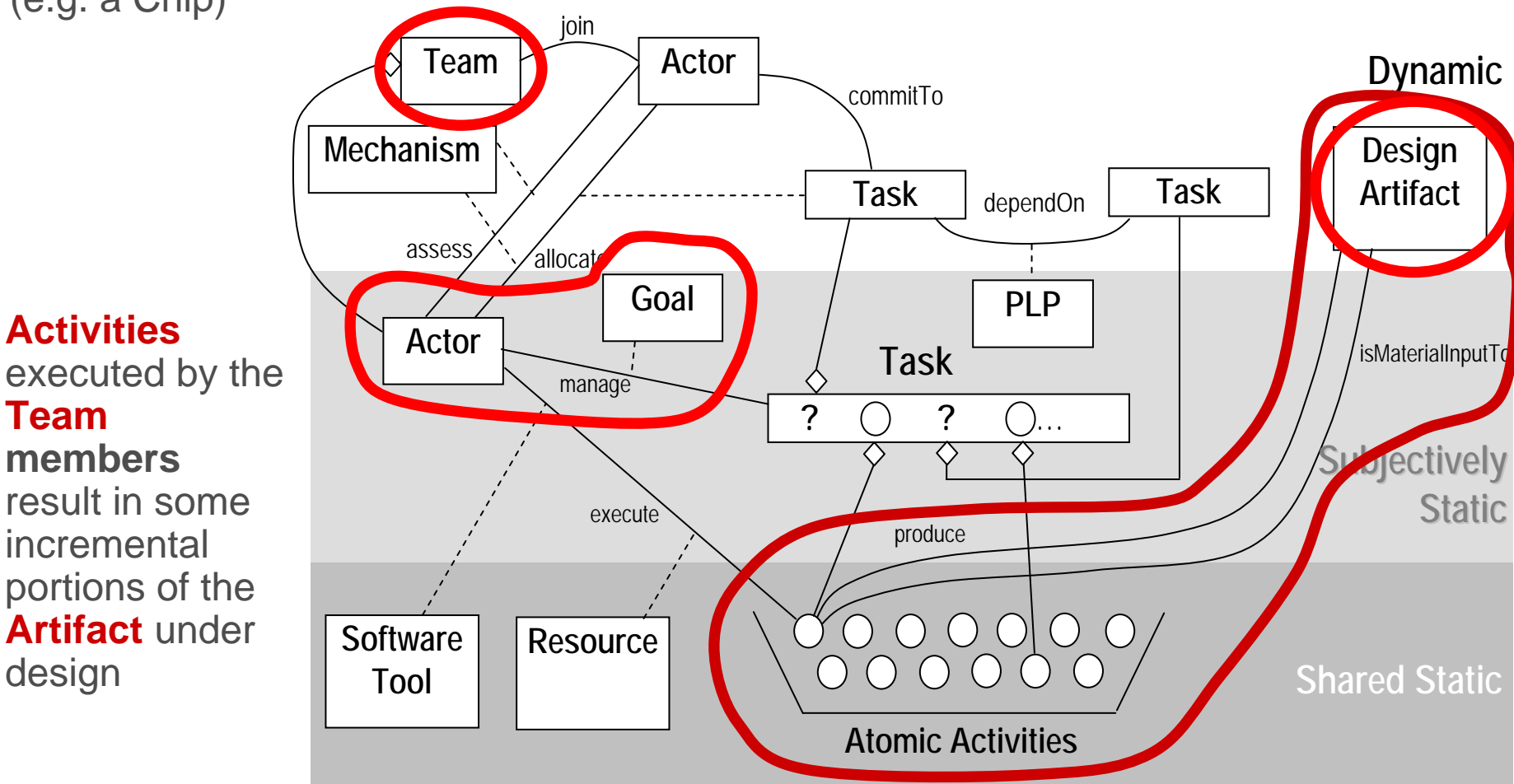
(Mind dynamics factors, productivity measure and the "Units")



# What do we Need to Model a DEDP?

(Mind dynamics factors, productivity measure and the "Units")

The overall **Goal** of a(n **Actor** managing the) **DEDP** is to **design** the **Artifact** (e.g. a Chip)



**Activities** executed by the **Team** members result in some incremental portions of the **Artifact** under design

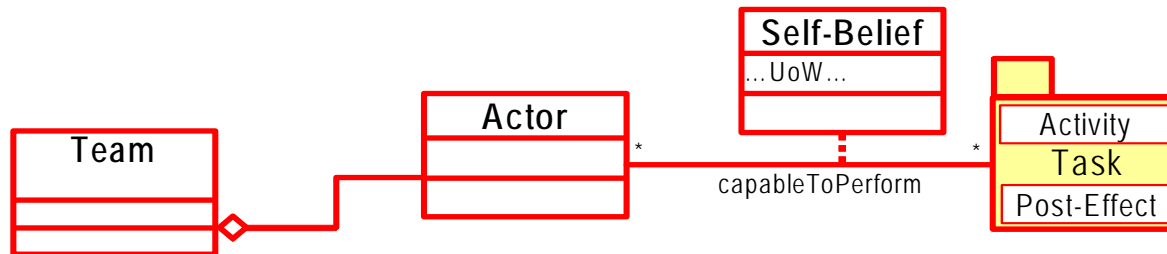
# Actors:

Self-Beliefs (Capabilities, Capacities), Team Members

---

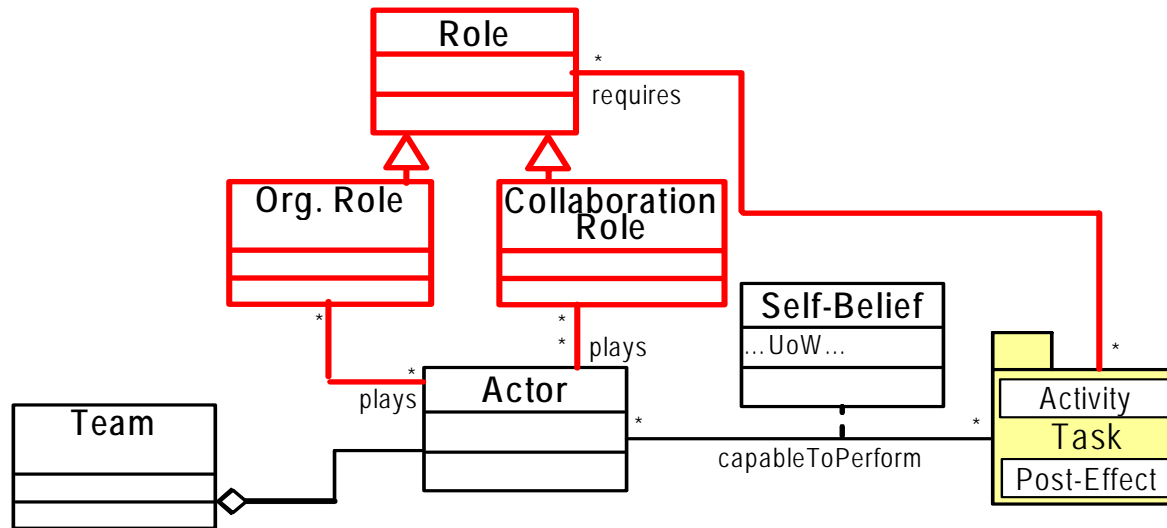
**Actors** have various capabilities and capacities (**Self-Beliefs**) wrt **Tasks** and **Activities**

**Actors** may form groups (**Teams**) to perform a **DEDP**



# Actors:

## Roles in Teams, DEDPs, Encounters



**Actors** play different **Roles** in these **Teams**

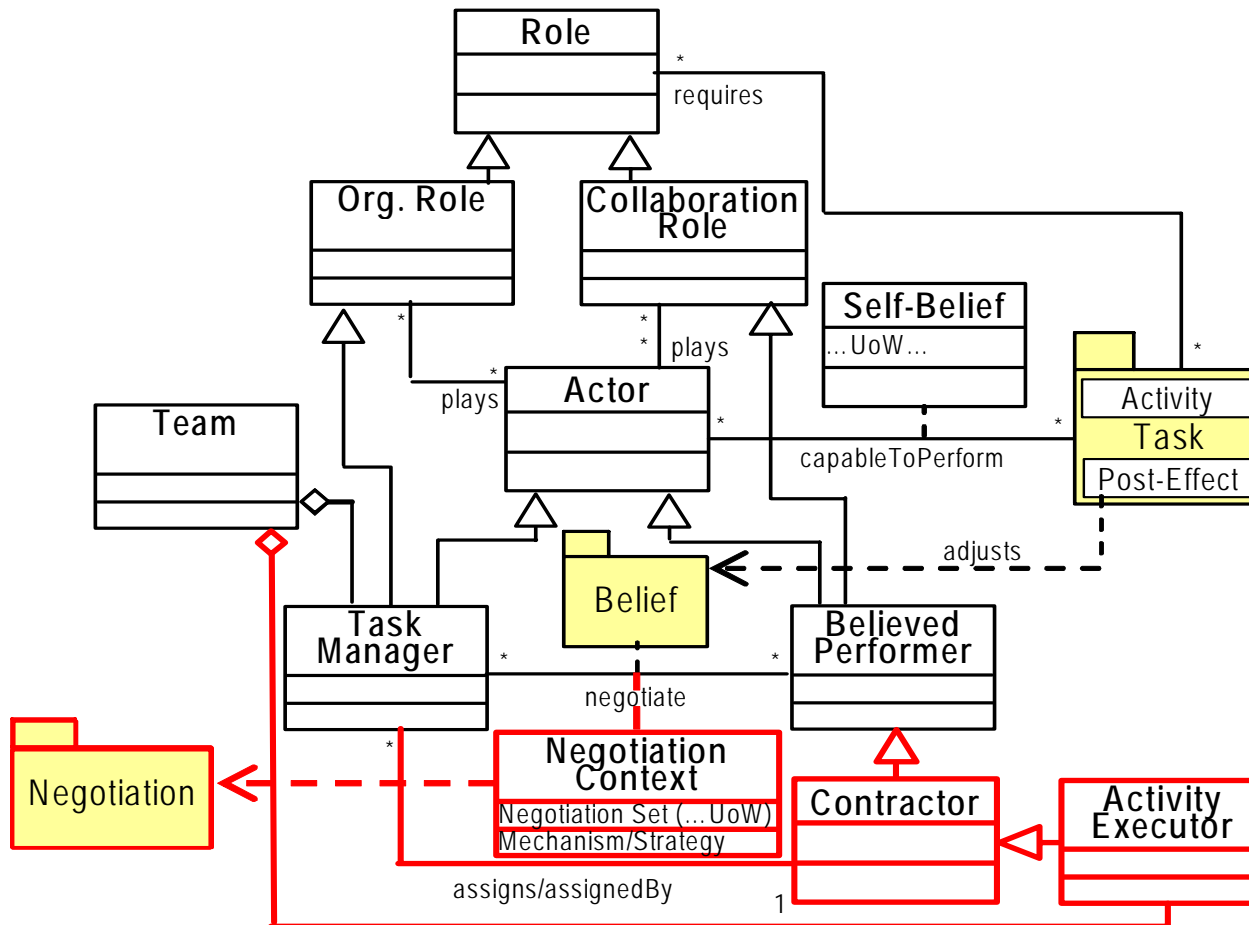
The **Role** may be:

- **Organizational** (e.g. Project Manager)
- **Collaboration** (i.e., the role in the team-forming encounter, e.g., the Initiator of the Contracting Negotiation)



# Actors:

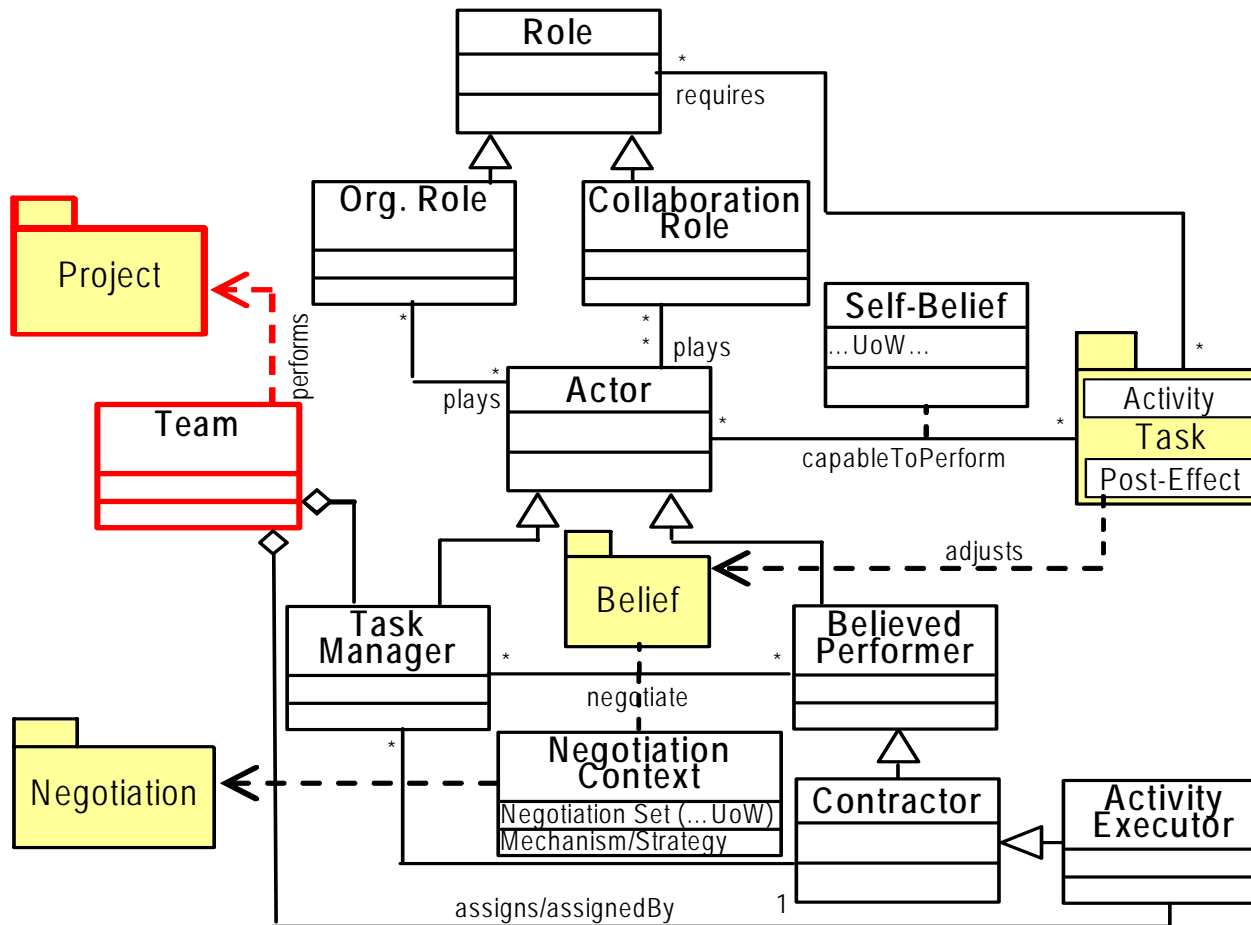
## Collaboration Mechanisms



**Actors** should have the mechanisms for communication and **collaboration** (Utilitarian)

# Actors:

## A Team per Project (DEDP): arbitrary Actor Combinations



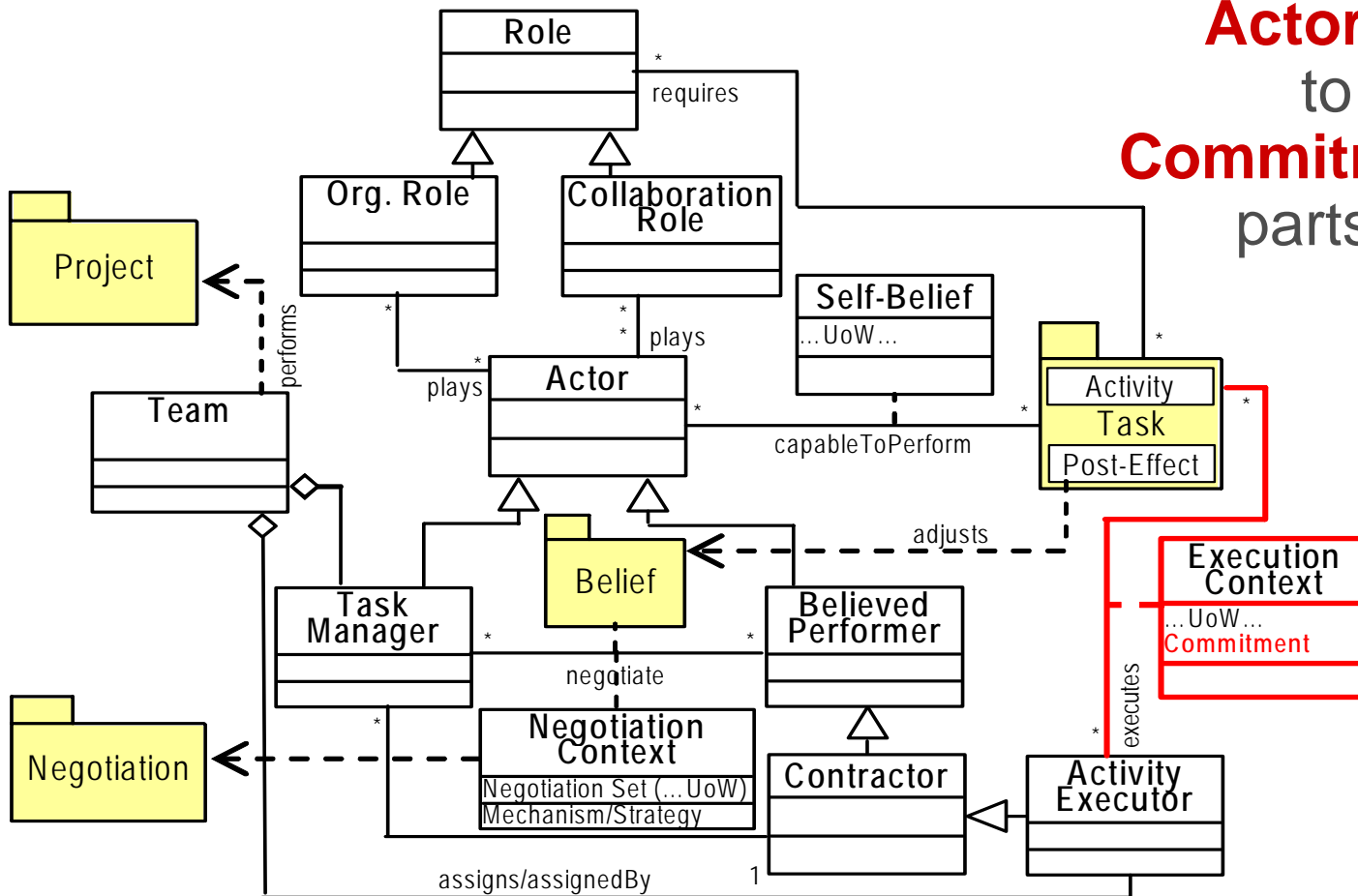
**Actors** may take part in different DEDPs (Projects) at a time

A **Team** is bijectionally related to a **Project**

An **Actor** may belong to different **Teams** at a time

# Actors:

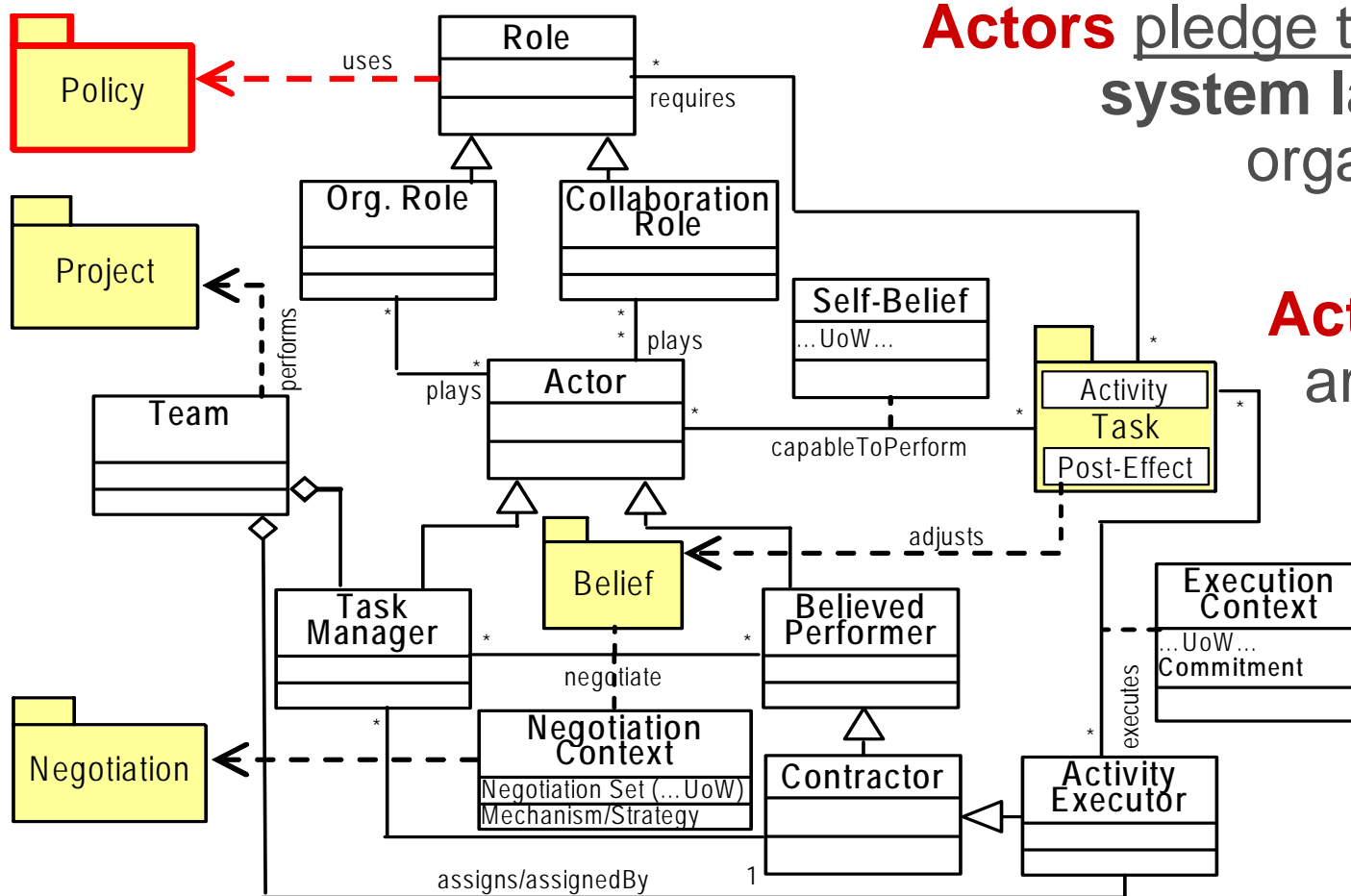
## Commitments



**Actors** who belong to a **Team** have **Commitments** wrt the parts of the DEDP

# Actors:

## System Laws as Policies

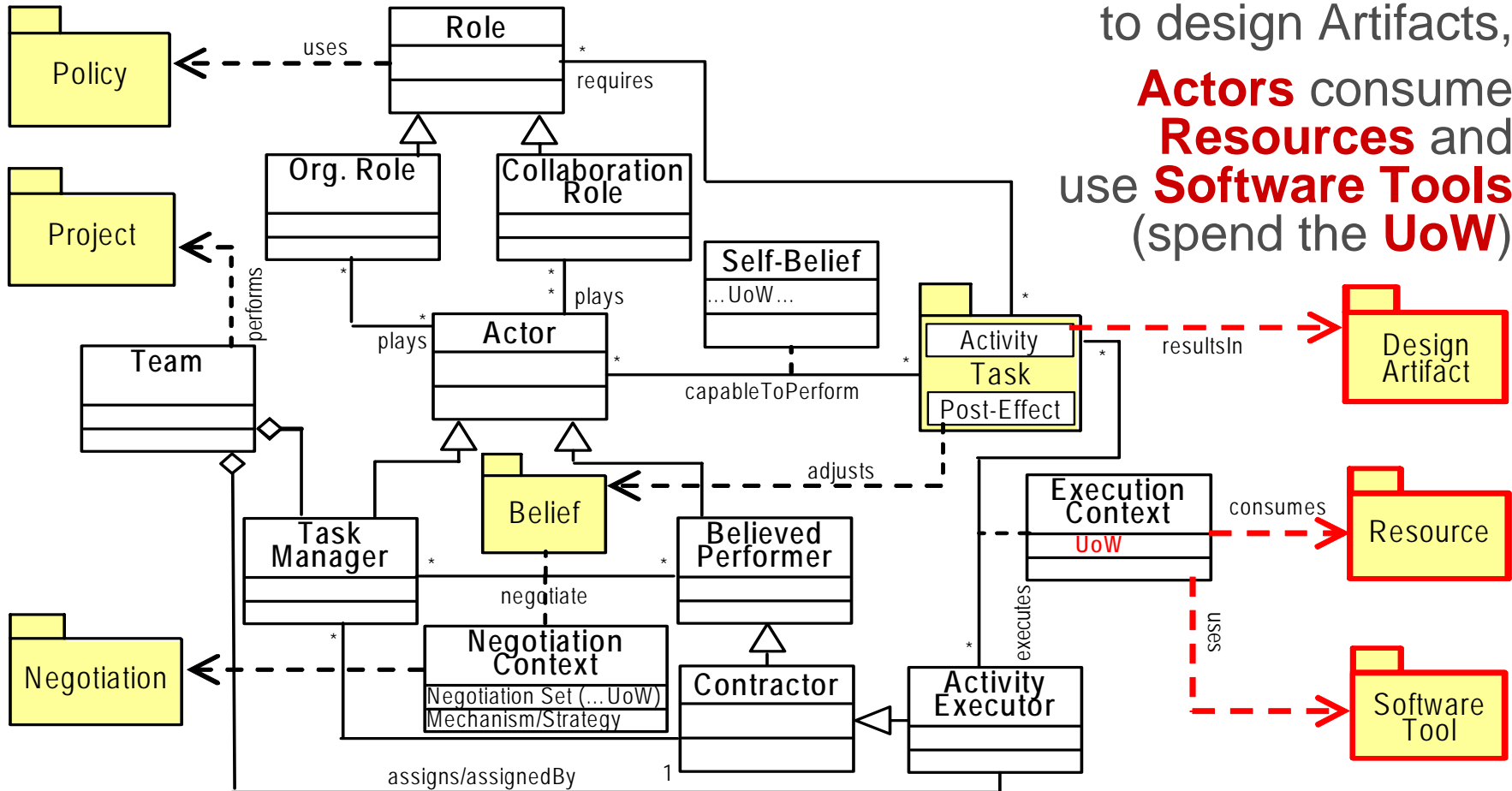


**Actors** pledge to follow some **system laws** (team- or organization-level conventions)

**Actors'** activities are constrained by **Policies**

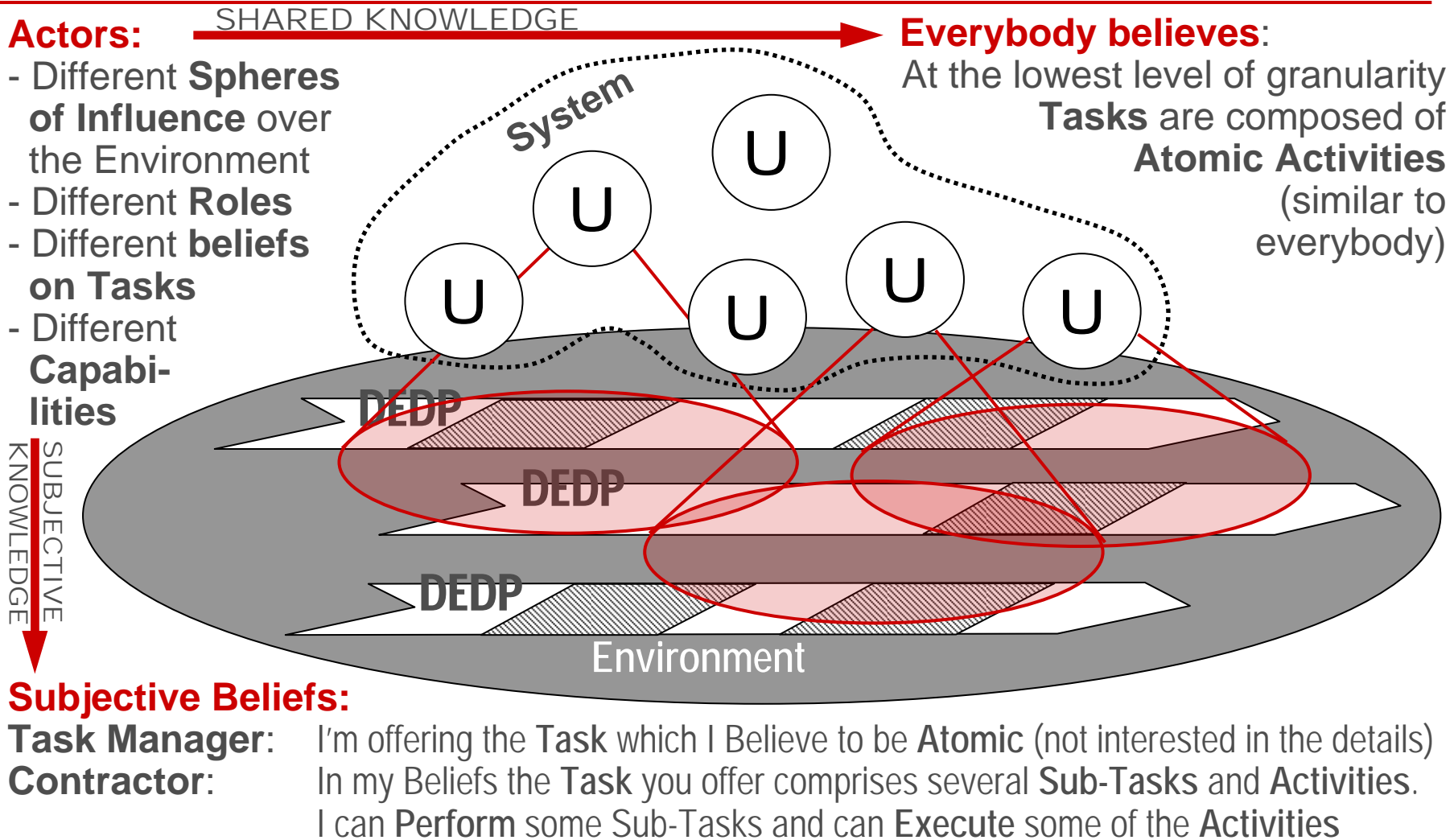
# Actors:

## The Goal and the Price to Pay (UoW)



**Actors** execute **Activities** to design **Artifacts**,  
**Actors** consume **Resources** and use **Software Tools** (spend the **UoW**)

# DEDP: Tasks and Activities

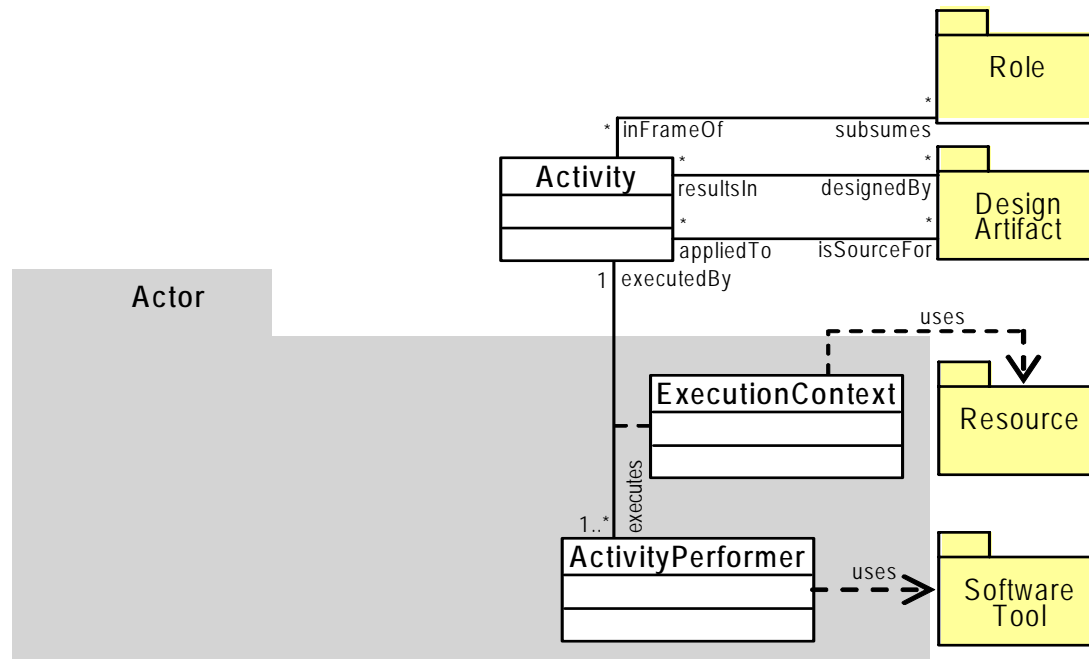


# Tasks-n-Activities

## Basic Building Blocks. Material In-Out-s

An **Activity** – the basic building block (for everybody), defined by the **Design Technology (SHARED and STATIC)**

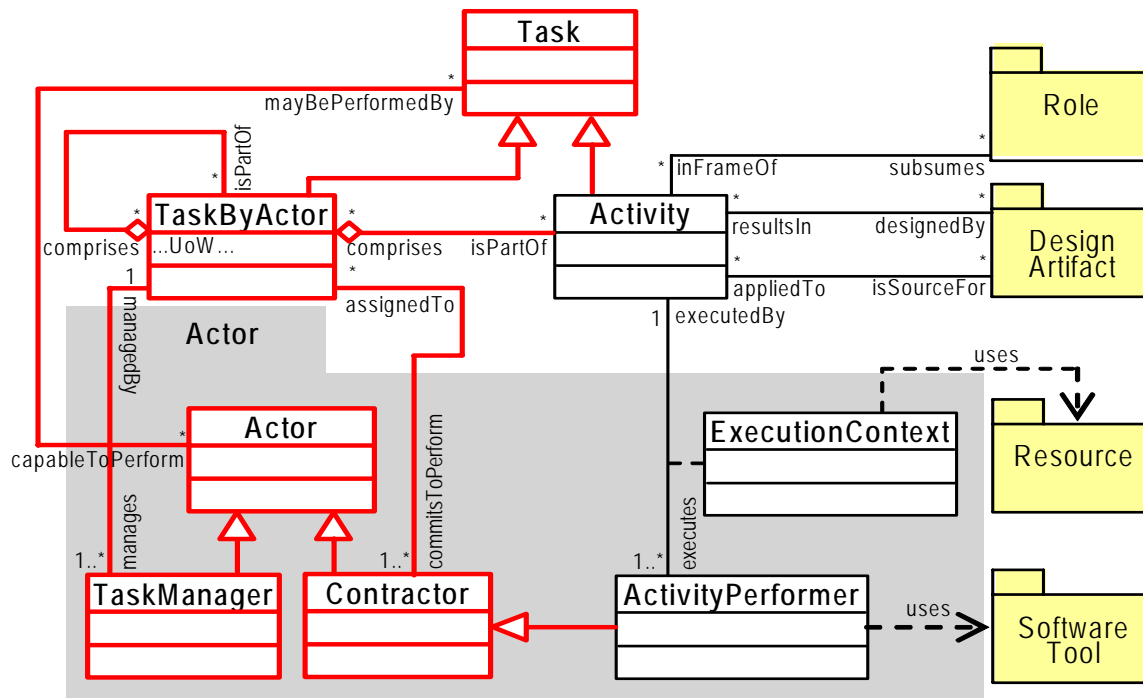
An **Activity** is **Executed** on its **Material Inputs** (Design Artifacts) and **Produces Material Outputs** (Design Artifacts)



# Tasks-n-Activities:

## A Task – a Hierarchical Combination of Activities

- A **Task** is the hierarchical (Sub-Tasks) **combination** of **Activities**
- This combination may be **believed different** by different **Actors**
  - In the simplest case a **Task** comprises the only **Activity**
- A **Task** comprising more than the only **Activity** is not Executed but **Managed** and has NO Material Inputs and **Material Outputs**



A **DEDP** is the Design Artifact transformation process modeled as the **Task** managed By the certain Actor (the **Task Manager**)

# Task Dependencies

## Strong and Weak Dependencies

---

- $t_1$  is **strongly dependent** of  $t_2$ 
  - $t_1$  can't be started before the **Results** of  $t_2$  become available
  - The **Results** of a **Task** are the **Material Outputs** of all **Activities** executed in a **Task**
- $t_1$  is **weakly dependent** of  $t_2$ 
  - If the results of  $t_2$  are available  $t_1$  may be performed for less **UoW** (means quicker, with better quality, fewer iterations, ...)
- $t_1$  is **independent** of  $t_2$ 
  - In all other cases

# Task Dependencies are Subjective

## Partial Local Plans (PLP)

---

- **Actors** have different **Beliefs of Task Dependencies**
- **Actors Plan and Schedule** managed **Tasks** autonomously
  - Do not use the knowledge of other Actors
- $t_1$  is **strongly dependent** of  $t_2$  implies:
  - All the **Material Outputs** of  $t_2$  **Activities** are available and will be used as the **Material Inputs** by the **Activities** of  $t_1$
  - The **Pre-condition** of  $t_1$  is the event of the appearance of the **Material Inputs** produced in  $t_2$  (**Eventual Output**)
  - **Eventual Input** of  $t_1$  is the **Eventual Output** of  $t_2$
- Similarly **for weak dependencies**

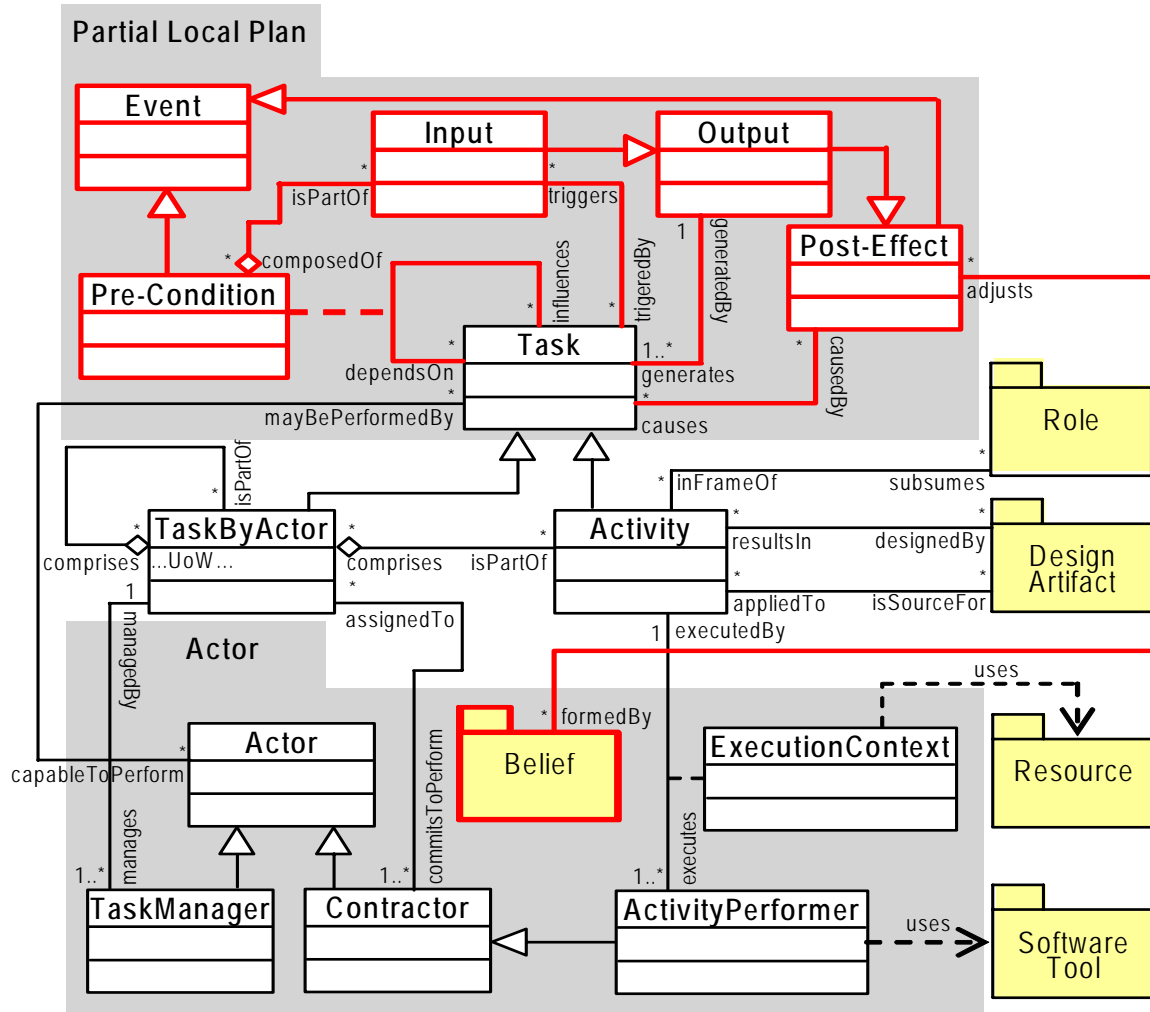
# Task Post-Effects

---

- Only some **Eventual Outputs** become **Eventual Inputs**
- An **Eventual Output** is the sub-class of a **Post-Effect**
- A **Post-Effect** is the abstraction of the changes implied by the performance of a **Task** onto the **Environment**:
  - E.g., **deadline violation** causes re-scheduling, penalties, the changes in the **Beliefs** of an **Actor** on the other Actors

# Tasks-n-Activities:

## Dependencies and Partial Local Plans



# A Design Artifact

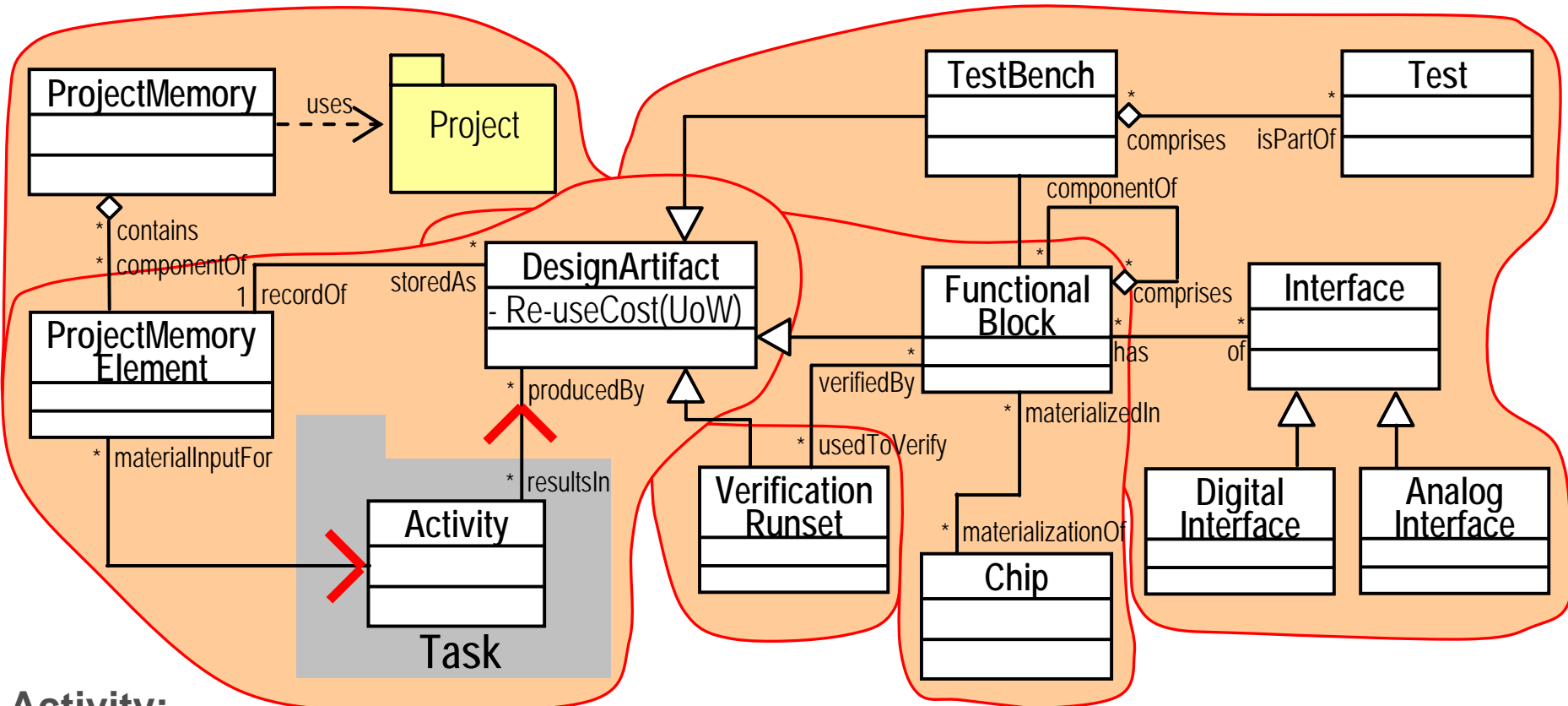
---

- Describes the **Material Output(s)** of an **Activity**, the **Activities** of a **Task**, ..., of a **Task**, ... of a **DEDP**
- Grounds it to **SES** Design Domain
  - E.g., by structuring a **Design Artifact** as appropriate for **SES**
  - E.g., by stating that a **Design Artifact** in this Domain is further on materialized in a **Chip**
- Reflects the project-oriented nature of a DEDP:
  - States that a **Design Artifact** is stored as the **Project Memory Element**
  - A **Project Memory Element** (but not a **Design Artifact**) is used as the **Material Input** for an **Activity**

# A Design Artifact

Project-oriented nature of Design

Structure appropriately for SES Design



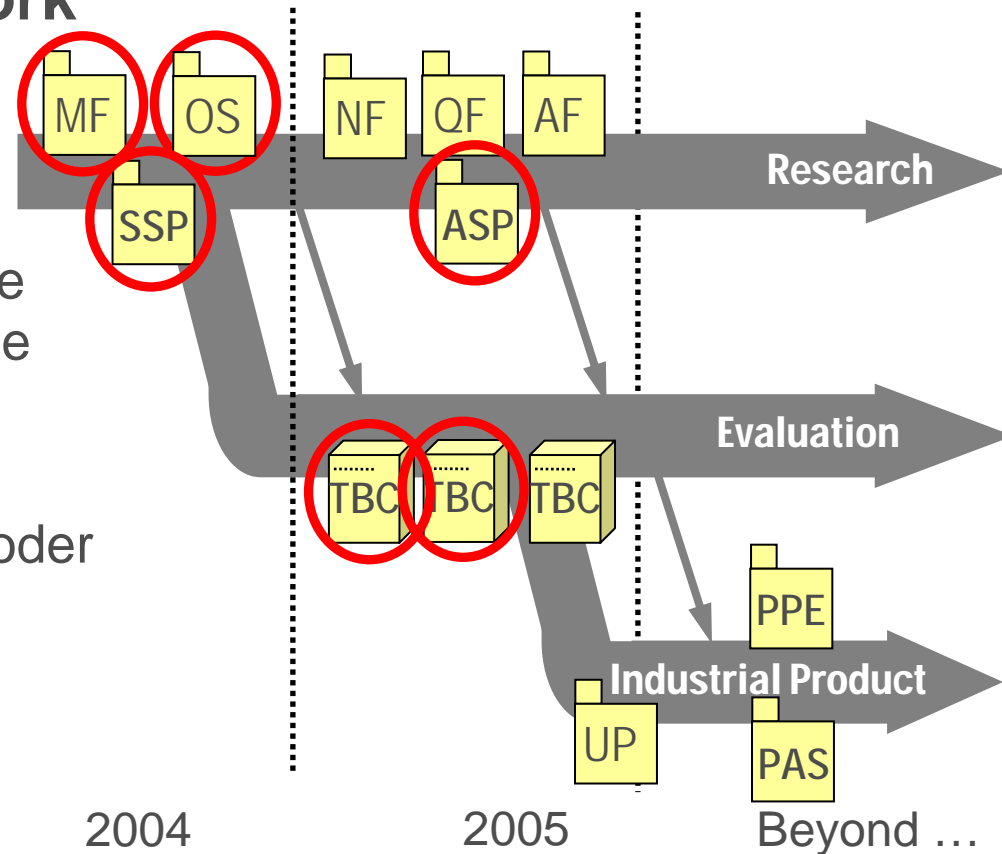
Activity:

Material Inputs and Outputs

Materialization in a Chip

# PSI Results

- **DEDP Modeling Framework**
- **DEDP Ontologies** in OWL
- **USED in 2 Research Prototypes**
  - Simplified Simulator Prototype
  - Advanced Simulator Prototype
- **2 Test Cases** (simplified) stored to the Test-Bed
  - Configurable multimedia encoder (digital)
  - Controlled amplifier (analog)
- **Evaluation** experiments on the initial test-bed performed



# Conclusions

---

- **Done:**

- Descriptive framework for modeling DEDPs
- The family of DEDP ontologies
  - Used in the design of the research prototypes of DEDP Simulator
  - Used in framing the data and the knowledge on PSI Test-bed – 2 cases

- **Future work:**

- Ongoing: Evaluation by a real-life design project of Cadence
- Harmonization (e.g., by checking consistency with DOLCE)

**Shall be Happy  
to Answer  
your Questions**

**cadence**



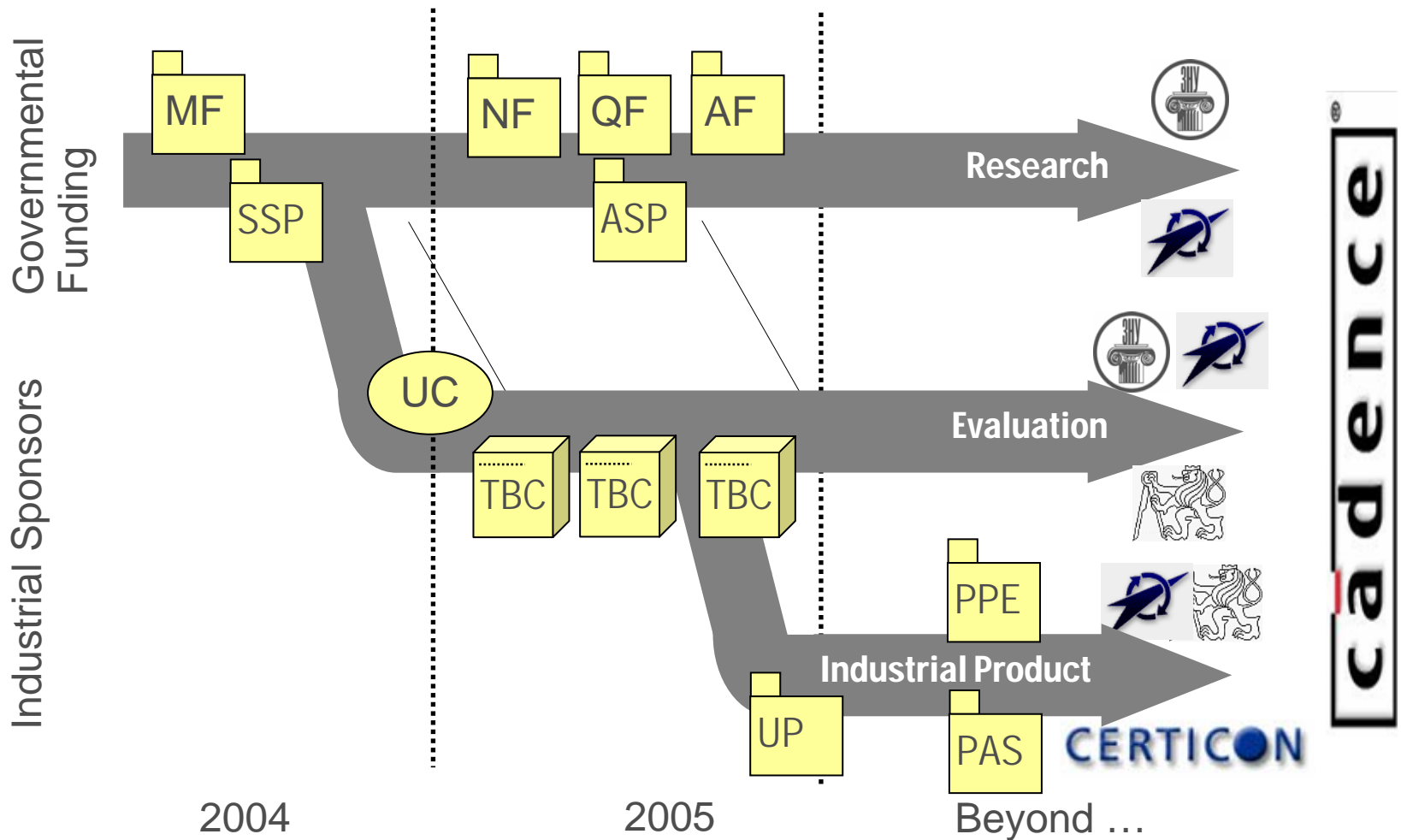
Thursday, October 27, 2005

# BACK-UP SLIDES



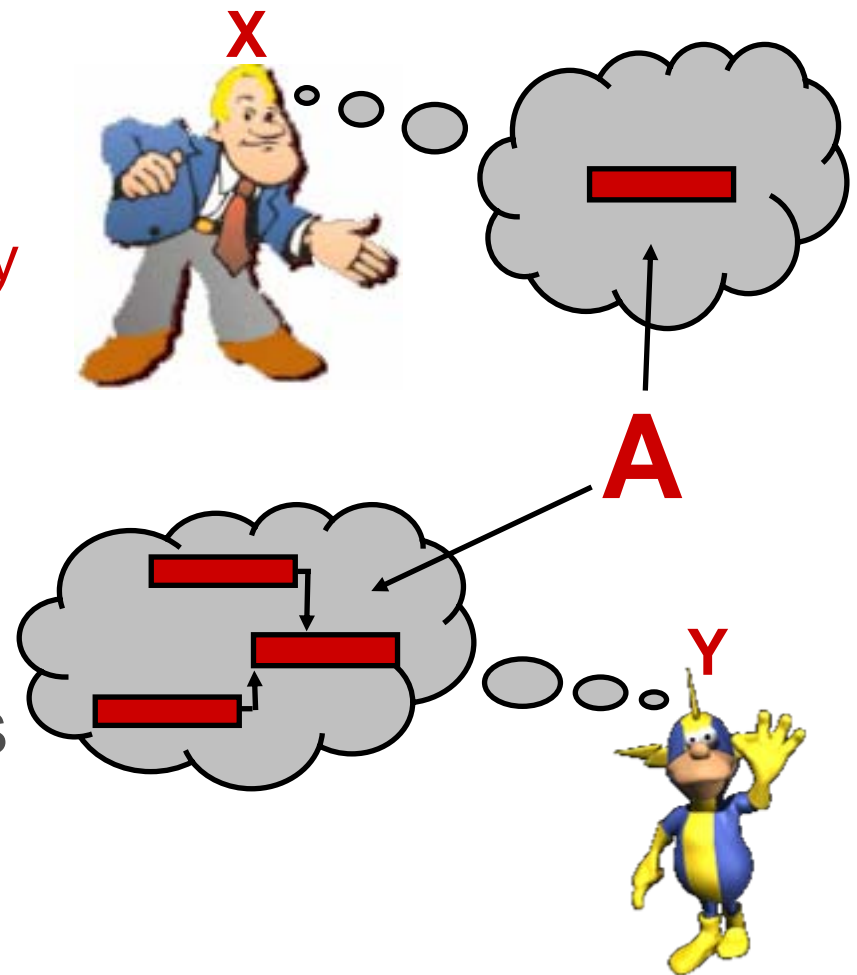
# Productivity Simulation Initiative

## Project Lines and Partners



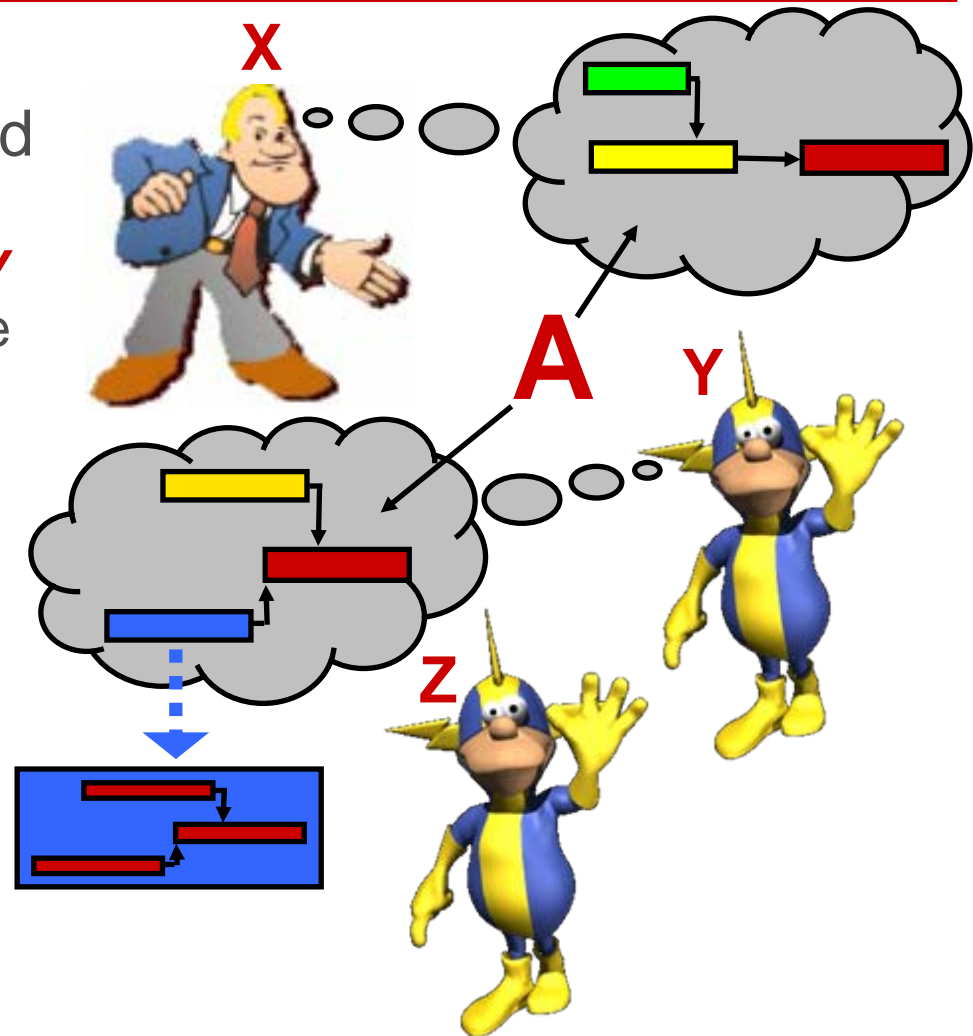
# Factors affecting DEDP Dynamics: Subjective Knowledge on Activities

- Different **Agents** have different knowledge and capabilities wrt a **DEDP**
  - **Agent X** may treat an **Activity A** as atomic – i.e. non decomposable
  - **Agent Y** may treat **A** as composite – i.e. a **Task**
- **X** and **Y** (if assigned) will perform **A** in different ways (with different levels of distress)
- Requires **distributed planning**



# Factors affecting DEDP Dynamics: Composition is Subjective and Partial

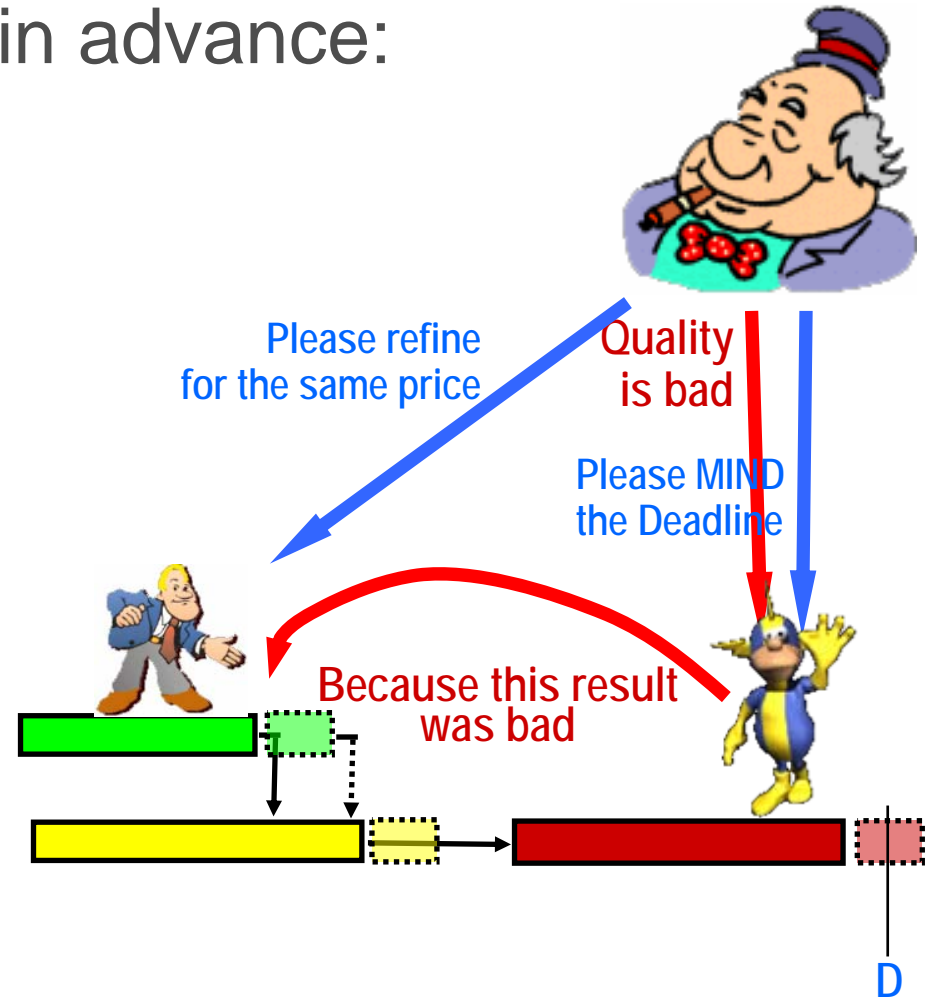
- Activity **composition** is performed **subjectively** and **partially**:
  - **Subjectively**: Agents **X** and **Y** may have different knowledge on how to compose a Task of Activities
  - **Partially**: Activities may also (further, e.g., by Actor **Z**) appear to be Tasks
- **Implication**: Activities may be sequenced and conveyed differently
- Requires **distributed scheduling** at run time



# Factors affecting DEDP Dynamics:

No of Activity Loops is not Predefined -

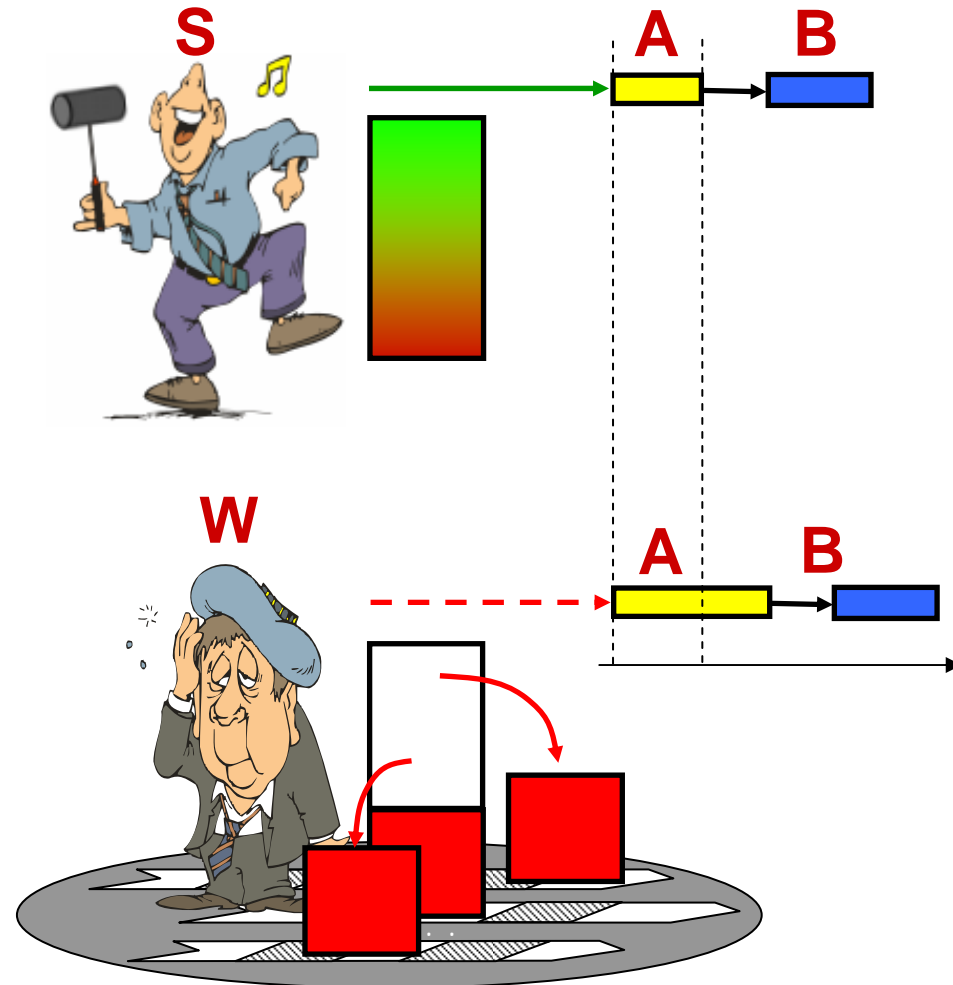
- Can not be determined in advance:
  - **Quality checks**
  - **Poor results** at prior or intermediate steps
- Increasing No of Loops implies increased duration (same price)
- Associated penalties may be triggered
- Requires run-time **re-planning** and **re-scheduling**



# Factors affecting DEDP Dynamics:

Activity duration depends on the available capacity

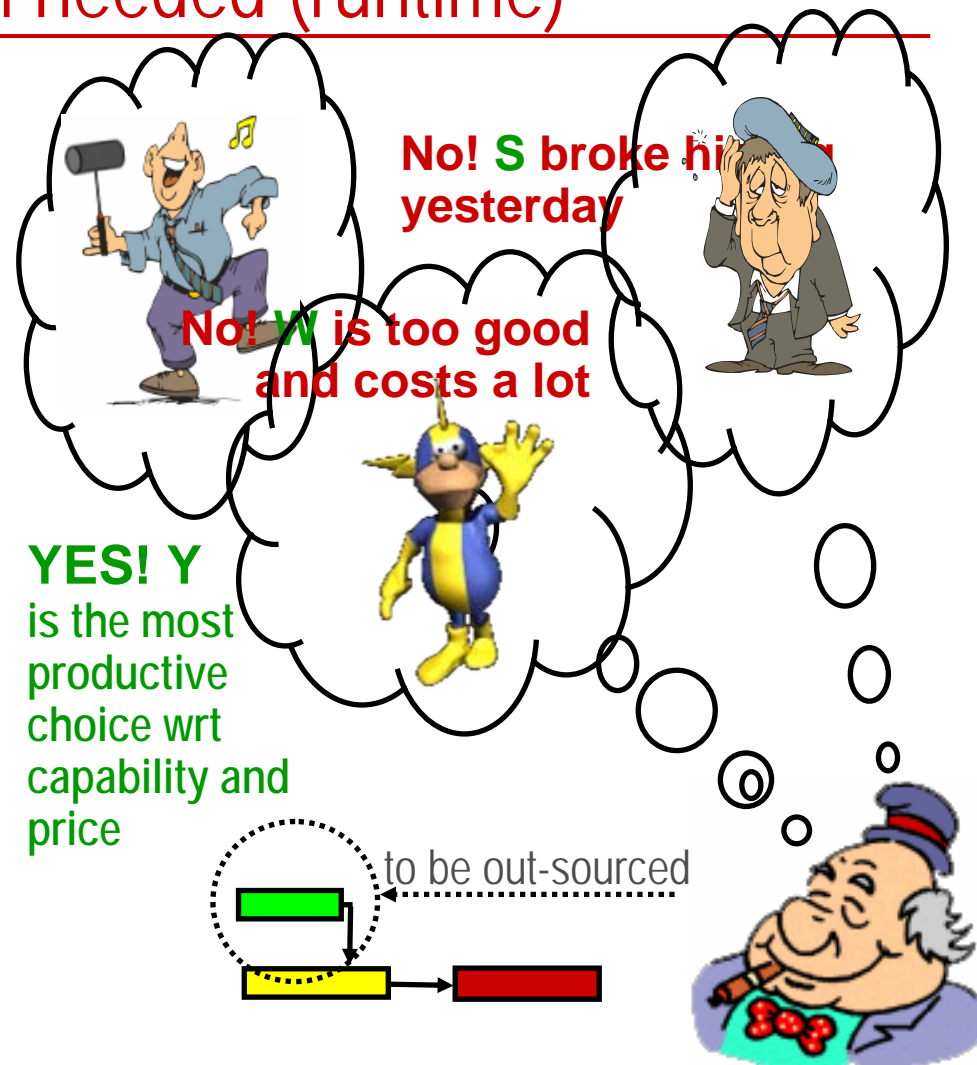
- Mr. **S** is highly productive wrt **A**
- Mr. **W**:
  - Can also be highly productive wrt **A**
  - But spends his capacity to several other **DEDPs**
- **B**, though allocated, remains idle for different time (cant be pre-determined)
- Requires run-time **re-scheduling**



# Factors affecting DEDP Dynamics:

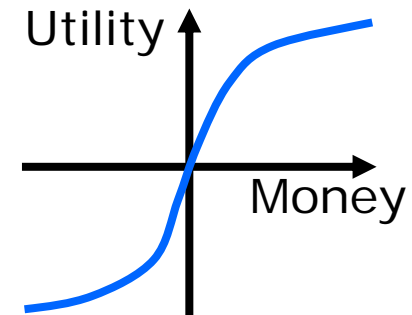
Actors are contracted when needed (runtime)

- Actors are often not assigned in advance to perform certain activities
- An actor is contracted by the Task Manager when s/he decides to assign or to out-source the activity
- Contracting decision is done and taken through **negotiations**



# Utilities are Relative ...

- **Utility** is not money (but it is a useful analogy)
- **Utility functions** are just a way of representing an agent's preferences
- They do not simply equate to money
- Suppose “You have all and I have nothing” (recall “The Bodyguard”) – say, more rationally, € 1 000 compared to €5 000 000:
  - A generous donator coming with 1 000 000
  - For me the utility will be enormous – a raise in 1 000 times
  - And for you – just something more
- Typical relationship between utility & money – on the chart



# More Information

---

- **ER'2005 tutorial** “Modeling and Simulation of Dynamic Engineering Design Processes”
  - Abstract: <http://eva.zsu.zp.ua/psi-public/psi-tutorial-abstract.pdf>
  - Presentation slides: <http://eva.zsu.zp.ua/psi-public/psi-tutorial.pdf>
- The **Overview of the SOTA** in Agent-Based Design Modeling ...
  - **Ermolayev, V. et al:** Agent-Based Dynamic Engineering Design Process Modeling Framework. Technical Report. Cadence Design Systems, GmbH, 29 p., 2004,
    - <http://eva.zsu.zp.ua/psi-public/SOTA-TR-PSI-2-2004.pdf>
- **PSI DEDP Modeling Framework**
  - **Ermolayev, V. et al:** Agent-Based Dynamic Engineering Design Process Modeling Framework. Technical Report. Cadence Design Systems, GmbH, 29 p., 2004,
    - [http://eva.zsu.zp.ua/eva\\_personal/PS/PSI-DEDP-MF-v10-Feb-2004.pdf](http://eva.zsu.zp.ua/eva_personal/PS/PSI-DEDP-MF-v10-Feb-2004.pdf)

# To Read More

---

- **PSI Papers**

- **Matzke, W.-E.:** Engineering Design Performance Management – from Alchemy to Science through ISTa. In: R. Kashek, H. C. Mayr, S. Liddle (Eds.): Information Systems Technology and its Applications (ISTA'05) 4th Int. Conf. 23-25 May 2005, Palmerston North, New Zealand GI LNI vol P-63, pp. 154-179, 2005
- **Gorodetsky, V., Ermolayev, V., Matzke, W.-E., Jentzsch, E., Karsayev, O., Keberle, N., Samoylov, V.:** Agent-Based Framework for Simulation and Support of Dynamic Engineering Design Processes in PSI. In: Pechouchek, M., Petta, P., Varga, L. Z. (Eds.) Proc. 4<sup>th</sup> Int. Central and Eastern European Conf. on Multi-Agent Systems (CEEMAS'05), 15-17 September 2005, Budapest, Hungary, LNAI 3690, pp. 511-520, 2005
- **Ermolayev, V., Keberle, N., Matzke, W.-E., Vladimirov, V.:** A Strategy for Automated Meaning Negotiation in Distributed Information Retrieval. In: Y. Gil et al. (Eds.): ISWC 2005, LNCS 3729, pp. 201 – 215, 2005