ER2005

24-28 October 2005

cadence

Modeling and Simulation of Dynamic Engineering Design Processes

(Agent-Based Approaches in Design Process Modeling)

☑ **Vadim Ermolayev** — Zaporozhye National University, Ukraine

Vladimir Gorodetski — Saint Petersburg Institute for Informatics and Automation, RAS, Russia

☑ **Eyck Jentzsch** — Cadence Design Systems GmbH, Germany

Wolf-Ekkehard Matzke — Cadence Design Systems GmbH, Germany

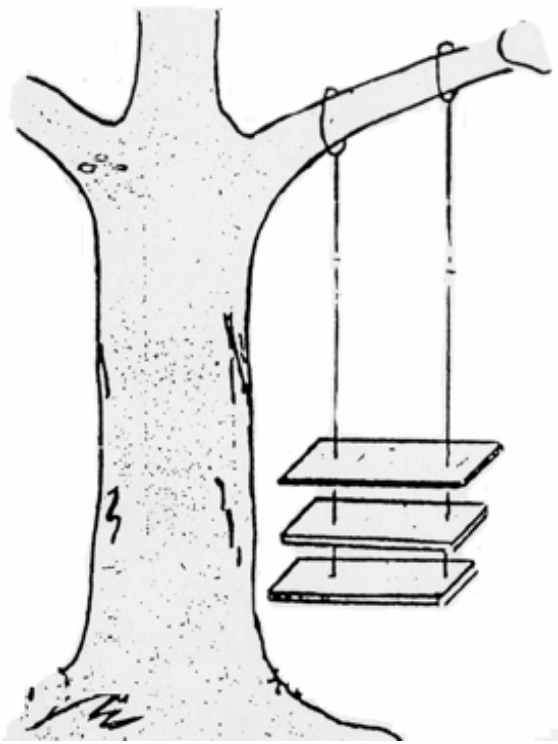
CERTICON

The Outlook

- **Motivation**:
  - Why to model Design?
  - Are there difficulties?
  - How to? Agents?
- Agent-based approaches and models in IPD 1993-2005+
- Modeling and Simulation of DEDP-s in PSI
- Conclusions and Discussion

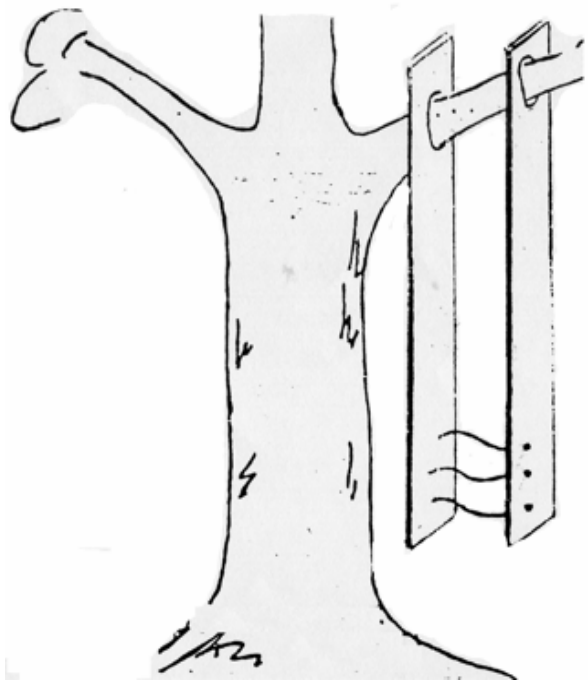
A Swing in Design …

MOTIVATION



required by the contract

… produced by
Design Engineer…

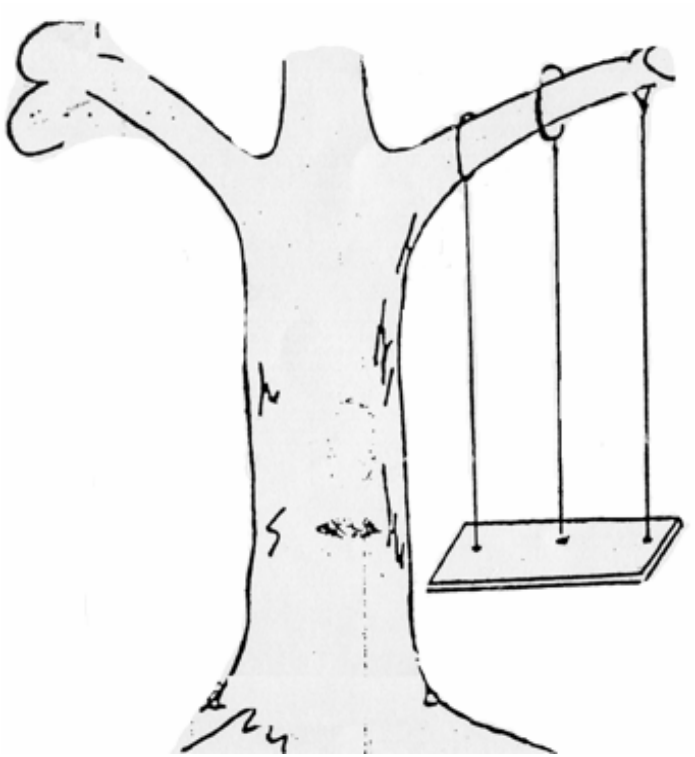


CERTICON

cadence

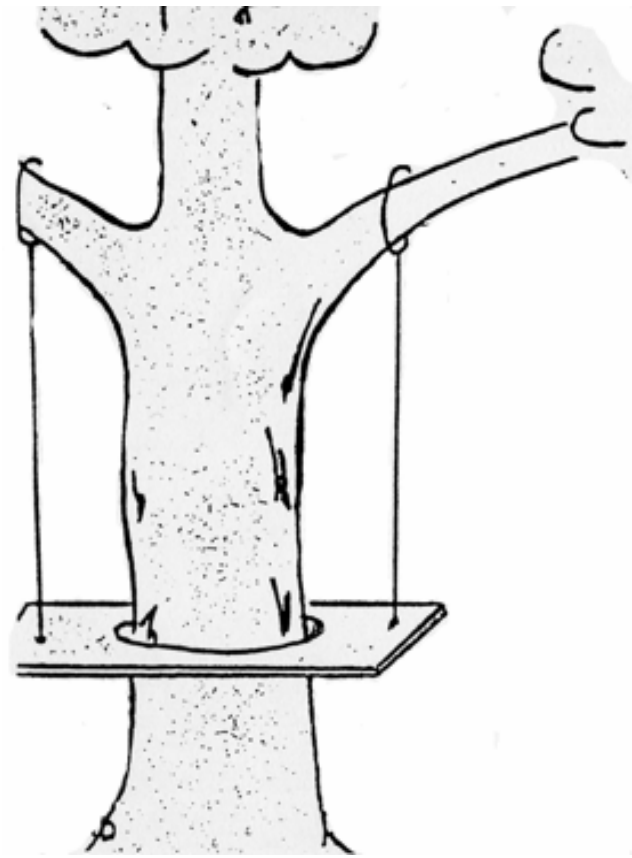
A Swing in Design …

MOTIVATION



written in technical documentation

the research prototype

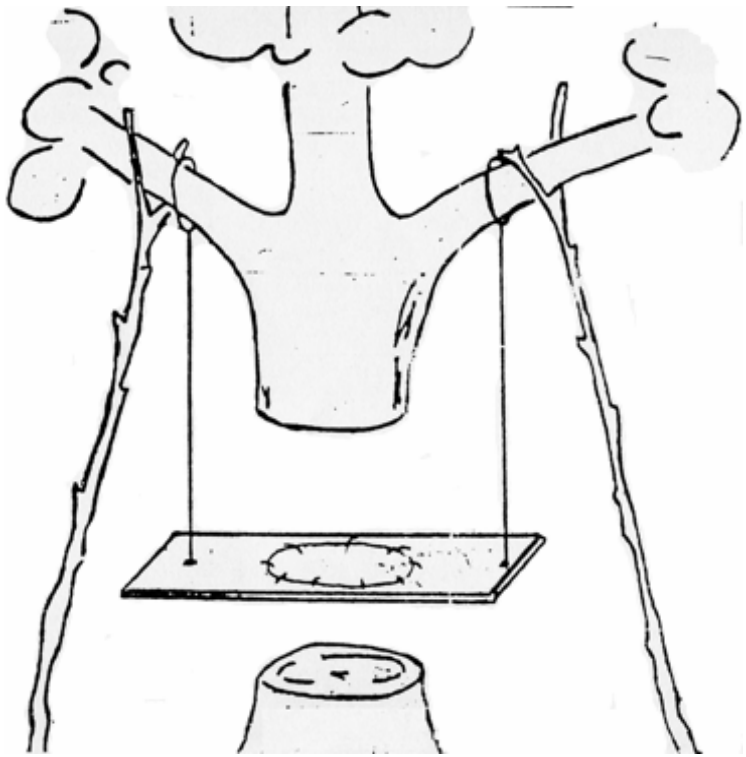


CERTICON

cadence

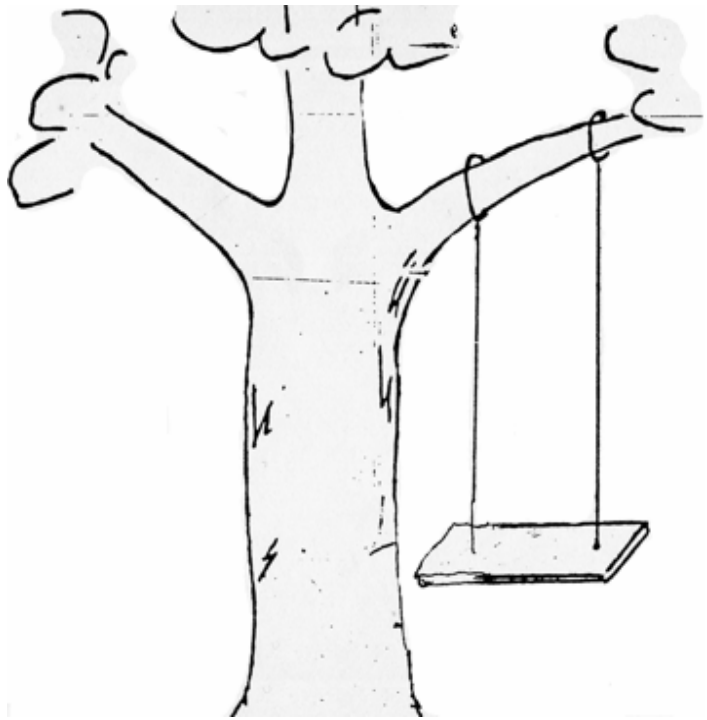
A Swing in Design …

MOTIVATION



… has been manufactured

…expected by a consumer!!!



Design – a Challenge for AI:

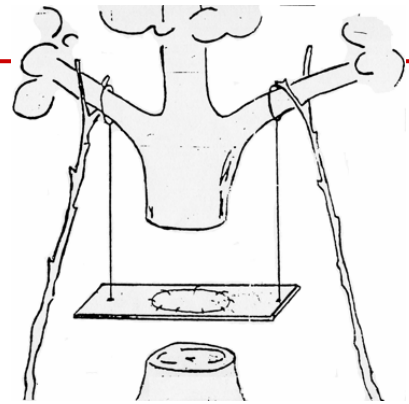
MOTIVATION

Design – a signature
of **human** intelligence (HI)…

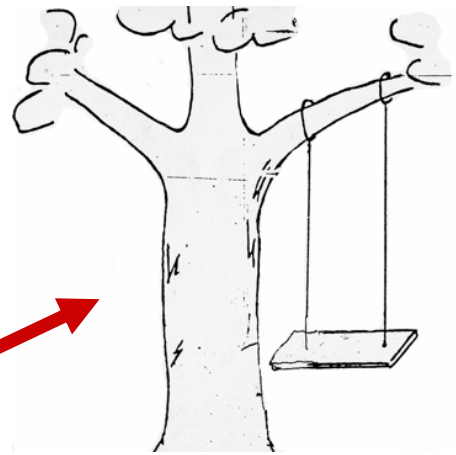
…– was always a great
challenge …

Esp., for AI research

E.g., **that's why**



… has been manufactured



**…expected
by a consumer!!!**

The Outlook

MOTIVATION

- **Motivation**:
  - Design – a signature of HI and a challenge for AI ☑
  - The reasons for such a “Swing” to appear
  - The dimensions of complexity in design modeling
  - The proper focus
  - The solution strategies
  - Are Agents the suitable modeling metaphor?

A Swing in Design! - Why?

MOTIVATION

- Design of a Complex Integrated System (Integrated Product) is often:
  - **Multidisciplinary**
  - Performed by **distributed teams**
  - Reuses the **components** which **do not always suit**
  - Needs (often **underestimated**) effort for proper **organization** and **coordination**
  - **More??? - Lots of more reasons!!!**
    - **E.g., designers are humans – nobody is perfect ☺**
    - **...**

Why Death March Projects Exist?*

MOTIVATION

- Industry (e.g. Automotive or Semiconductor) System Laws
- Intense competition due to globalization
- Intense competition due to new technology

Glocalization, boundaries between disciplines

- Marine Corps mentality - real engineers don't sleep
- Politics in a large, complex project
- Naive promises - unrealistic commitments
- Naive optimism of youth

Poor management

- Startup mentality of fledgling entrepreneurs
- Intense pressure of unexpected customer requirements
- Unexpected/unplanned crises of all kinds

Non-determinism, dynamics

- … and even more …

*partly based on Edward Yourdon: Death March, Prentice Hall, pp. 253-257

The Concept of a System Law
(please note the key words)

MOTIVATION

- A **System Law**
  - is a **rule** (or a set of rules) within a concrete system
  - tells an **agent** of the system about the expected **behavior** pattern
  - can enforce **change** or represent a barrier to change
  - can be used to predict certain aspects of the system behavior, which are based on the force, or influence it exerts on the internal **environment** of the system

System Laws vs Natural Laws

MOTIVATION

- **System laws** (in contrast to natural laws) are neither universal nor does they need to be true, correct, etc
- It doesn't matter where they came from
- **They are**, for whatever reasons, **IN FORCE**
- A **System Law** (in contrast to a natural law) may be very well **changed by human powers**
  - E.g., by **parliamentary voting**

System Laws: Semiconductor Industry

MOTIVATION

- The two most important are:
  - The Moore's Law
  - The Noyce's Minimum Information Principle
- Note: **Gordon Moore** and **Robert Noyce** founded Intel in 1968
- The **Moore's Law** states:
  - Every 18 months, processing power (of the product) doubles while cost holds constant (Exponential Growth)

System Laws: Semiconductor Industry

MOTIVATION

- The two most important are:
  - The Moore's Law
  - The Noyce's Minimum Information Principle
- According to the **Noyce's Minimum Information Principle**:
  - A researcher **guesses** what **the answer** to a problem is and goes as far as s/he can in a heuristic way
  - If that doesn't solve the problem, he backs up and learns enough to try something else (try to get by with as little information as possible to make something that work)
  - Developing a deliverable product is the only goal
  - …
  - The clock is ticking (see before)

System Laws: Semiconductor Industry

MOTIVATION

- The two most important are:
  - The Moore's Law
  - The Noyce's Minimum Information Principle
- Implications on design process performance, management, …:
  - **<u>The Moore's Law</u>** imposes **a clock rate** on the technological progress to be achieved
  - **<u>The Noyce's Minimum Information Principle</u>** enforces the **concept of how to act**

The Outlook

MOTIVATION

- **Motivation**:
  - Design – a signature of HI and a challenge for AI ☑
  - The reasons for such a “Swing” to appear ☑
  - The dimensions of complexity in design (modeling)
  - The proper focus
  - The solution strategies
  - Are Agents the suitable modeling metaphor?

Dimensions of Complexity in IPD*

MOTIVATION

- There are **Boundaries between Disciplines**
- Representatives of different disciplines have **different Built-in Goals**
- **Disciplinary Design** is performed **in Big Chunks**
- Complex Systems, like IPD teams, possess **Counter-Intuitive Behavior**

* Shakeri, C., Brown, D.C., Noori, N.: Discovering Methodologies for Integrated Product Design. Proc. Artificial Intelligence and Manufacturing: Second Bi-annual AI & Mfg Workshop, Albuquerque, New Mexico, 1998.

Dimensions of Complexity:

MOTIVATION

- **Boundaries between Disciplines**:
  - Knowledge is conceptualized and represented differently
  - Different special internal languages and ontologies
  - No means to communicate with the outside world

- **Consequence** - Difficult to:
  - Communicate, collaborate, resolve conflicts between disciplines

Dimensions of Complexity:

MOTIVATION

- **Different Built-in Goals**:
  - Knowledge is accumulated and used **independently**
  - **Decisions** are done and taken to **pursue the local goals**
  - The **local goals** of autonomous participants are often **in conflict** with the **global goals** of the design
- **Ignoring the conflicts** between **local** and **global goals drives** the design process to possible **lifelocks** or even **deadlocks**

Dimensions of Complexity:

MOTIVATION

- **Disciplinary Design in Big Chunks:**
  - Disciplinary designs are processed in **large segments**
    - **E.g.:** an engine, a chassis, a body of a car, an engine controller
  - Big chunks make **integration difficult** because **valuable information is hidden** from the rest of participants
    - **E.g.:** decisions that may lead to conflicts
  - There is also a **large overhead** in repeating large, discipline-based segments
    - **E.g.,** because of:
      - Possible failures
      - The iterative nature of design (Noyce's Minimal Information Principle)

Dimensions of Complexity:

MOTIVATION

- **Counter-Intuitive Behavior**:

  "It has become clear that complex systems are counter-intuitive, that is they give indications that suggest corrective action which will often be ineffective or even adverse in its results"*

- **Reason**: impossible to oversee all the details and the consequences
  - Systems (e.g., **design teams** for complex products like **B747**, **Pentium P4** processor) are the examples of complex systems with counter-intuitive behavior

- **Recipe**: Proper corrective actions should **emerge within the system**

*Forrester, J. W.: Urban Dynamics, MIT Press, 1969

What is the Proper Focus for Improvement?

MOTIVATION

- What to concentrate on?
- Which complexity brings most disasters?

- …Not yet clear

The Outlook

MOTIVATION

- **Motivation**:
  - Design – a signature of HI and a challenge for AI ☑
  - The reasons for such a "Swing" to appear ☑
  - The dimensions of complexity in design (modeling) ☑
  - The proper focus – postponed until more evidence received
  - The solution strategies
  - Are Agents the suitable modeling metaphor?

Solution Strategies for IPD*

…Bringing Complexities Upfront – to Analyse

MOTIVATION

- **Small Design Methods**
- **Opportunistic Contribution**
- **Cooperation**
- **Least Commitment**
- **Concurrency**

- **Design Models and Methodologies**

* Shakeri, C., Brown, D.C., Noori, N.: Discovering Methodologies for Integrated Product Design. Proc. Artificial Intelligence and Manufacturing: Second Bi-annual AI & Mfg Workshop, Albuquerque, New Mexico, 1998.

Small Design Methods

MOTIVATION

- To be smoothly integrated the **big chunks of design** should be **broken** into (atomic) **pieces** – e.g., **design activities**
- A **design activity** is a **procedure** or a **body of orderly procedures** for accomplishing a design task (**e.g.**, design synthesis, design selection, and design evaluation)
  - Breaking up the design into pieces corresponds to breaking design activities into smaller activities
  - **Smaller design activities** means:
    - **Fewer decisions** are made in each activity
    - **Shorter time** is spent in an activity
    - **Less information** is **produced** as a result of executing that activity
  - Smaller design activities are **simpler** and **consume less resources**
- **However**, **various degrees of granularity** for different executives at different organizational levels

Opportunistic Contribution

MOTIVATION

- An **opportunistic problem solving strategy** facilitates integration of the contributions of different parties in the design process
  - An **opportunistic approach** in contrast to a **predetermined order of contribution** allows to **take advantage** of the **diversity** of different opinions and **candidate solutions**
  - Every participant should get a **fair chance** to contribute to the goals of the design process so that **all points-of-view are explored**

- **However**, **less order**, more **difficult to find out the preferred outcome**

Cooperation

MOTIVATION

- A **Cooperative Strategy** provides **mechanisms**
  - By using which **different participants adopt common goals**
  - While **rationally trying to reach** their **local goals**
- Implementation of the cooperative strategy in a distributed design process results in **favoring** the **common goals** of the design **over local goals**
  - As a result of such strategy different parties spend their diverse resources in the same direction, **coherently**
  - Cooperation also means that the parties are **aware of the other parties** when posing their **design constraints** and **proposing their solutions**
- Cooperation doesn't mean that the **conflicts do** not **arise**
  - It provides **mechanisms for conflict resolution**
- **However**, **computationally complex**

Least Commitment

MOTIVATION

- **Deferring** the **decisions** that **constrain** future choices for **as long as possible**
- A least commitment strategy **reduces the number of conflicts**
  - **E.g.**, because it avoids committing to decisions that are made based on incomplete information
  - Otherwise, decisions may be made as soon as they can be, even if incomplete, arbitrary, or less trusted information is used
  - As a consequence, there is more chance for conflicts to occur in the future, because such information may turn out to be invalid
- **However**, **how to measure the commitment if the information is incomplete?**

Concurrency

MOTIVATION

- A **Concurrent Strategy**, in contrast to a **Sequential Strategy**
  - Carries out **some of the problem solving activities in parallel** to each other
- Concurrent design is the **main theme** of the well-established **Concurrent Engineering** field
- Concurrency in design gives freedom to **all participants** to **contribute** to the current state of the design **in parallel**
- As a result, the **design process speeds up**, because the participants in the design do not have to wait in a line if they can make a contribution

- **However**, **coordination** and **planning** become **more difficult**

Design Models and Methodologies

MOTIVATION

- **Axiomatic** Design Model
  - **Independence** axiom (suggests maintaining independence between functional requirements)
  - **Information** axiom (suggests minimizing the information content) – Noyce's Principle?
- **Systematic** Design Model
  - Engineering design must be **carefully planned** and **systematically executed**
  - A **design method** (activity) must **integrate** many **different aspects of engineering**
- **Decision-Based** Design Model
  - Considers design process as the **Cooperative Problem Solving Activity**
- **Good in theory**, but **don't facilitate enough** to the implementation in the **real world settings (e.g., resource constrained)**

Complexities Encountered

MOTIVATION

- **Boundaries between Disciplines** cause problems in:
  - Communication, collaboration, conflict resolution
- **Differences in Local Goals cause conflicts:**
  - **If ignored, drive** the design process to possible **lifelocks** or even **deadlocks**
- **Big Chunks in Design**
  - make **integration difficult** because **valuable information is hidden** from the rest of participants
  - Cause **large overhead** in repeating large, discipline-based segments
- **Counter-intuitive behaviour** in complex design systems
  - A system is inefficient until corrective actions do not **emerge within the system**

Solution Strategies Proposed

MOTIVATION

- **Small Design Methods**
  - Provide capabilities to decompose the Big Chunks into the Smaller Ones. In a Distributed manner. Arrange and Coordinate
- **Opportunistic Contribution**
  - Forget about the pre-determined order of activities – arrange them opportunistically, re-plan, re-schedule
- **Cooperation**
  - Use mechanisms providing to the growth of the group welfare, but not for the sake of the individual utilities

Humanize a Design System

MOTIVATION

- **In a word**:
  - A (model of a) design system should possess the substantial set of human-like capabilities
- **From the other hand:**
  - **Intelligent software agents** are used when the software
    - Does not execute blindly
    - But needs to possess some '**human**' features
    - Needs to demonstrate human-like behaviour

The Outlook

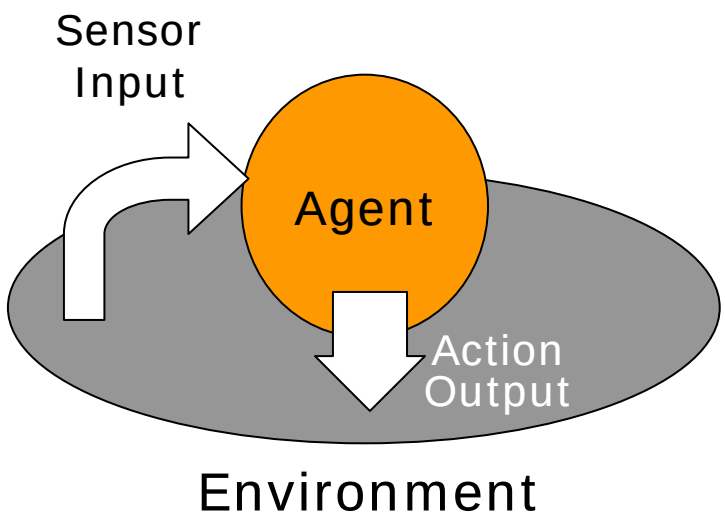
MOTIVATION

- **Motivation**:
  - Design – a signature of HI and a challenge for AI ☑
  - The reasons for such a "Swing" to appear ☑
  - The dimensions of complexity in design modeling ☑
  - The proper focus – postponed until more evidence received
  - The solution strategies ☑
  - Are Agents the suitable modeling metaphor?

What is an Agent?

MOTIVATION

- **The main point** about **agents** is that they are **autonomous** – i.e. capable of:
  - Acting independently
  - Exhibiting control over their internal state
- **Thus**: an **agent** is a **computer system** capable of **autonomous action** in some **environment**



An Agent: Intelligence

MOTIVATION

- **Trivial** (non-interesting) agents:
  - E.g., thermostat
  - E.g., e-mail filtering robot
- An **intelligent** agent is a computer system capable of **flexible** autonomous action in some environment
- By flexible, we mean:
  - **Reactive**
  - **Pro-active**
  - **Social**

Reactivity

MOTIVATION

- If a program's **environment** is guaranteed to be **fixed**, the program
  - Need never worry about its own success or failure
  - Just executes blindly
  - E.g., a compiler
- Many (**most!**) interesting environments are **dynamic** (**e.g., Engineering Design Processes**)
  - A program must take into account the possibility of failure – ask itself whether it is worth executing!
- A **reactive system** is one that:
  - Maintains an ongoing interaction with its environment
  - And responds to the changes that occur in it (in time for the response to be useful)

Pro-active Behavior

MOTIVATION

- Simple reacting to an environment is easy
  - **E.g.:** stimulus → response rules:
  - **E.g.:** temperature is more than 20°C → turn off heating
- But we generally want agents to **do things for us, on our behalf**
- Hence, **goal directed behaviour**
- **Pro-activeness** =
  - Generating and attempting to achieve goals
  - Not driven solely by events
  - Taking the initiative
  - Recognising opportunities

Social Ability

MOTIVATION

- The real world is a **multi-agent environment**:
  - We cannot go around attempting to achieve goals without taking others into account
    - E.g.: We care about the people in the audience …
  - Some goals can only be achieved with the cooperation of others
    - E.g.: you'd go to the bakery instead of attempting to make bread for your breakfast yourselves – you can't bake
- Similarly for many computer environments: witness the INTERNET

Social Ability

MOTIVATION

- **Social ability** in **Agents** is the ability to:
  - **Interact** with other agents (and possibly humans) via some kind of **agent-communication language**
  - **Adjust its behavior** according to **social norms** (e.g. obey the Moore's Law and the Noyce's MIP)
  - and perhaps **Cooperate with others**
- **Implies:** the ability to **Negotiate** and to **Understand the Meaning** of what is communicated

Ontologies: the Means to Understand the Meaning

MOTIVATION

- **E.g.:** A **Hot Dog** refused in New-York City …
  - Conceptual misunderstanding between the two agents
  - Needs proper resolution
- Imagine an intelligent information system confused by the query:
  - …
  - "**Show all the places selling hot dogs in X metropolitan area. Order per price**"
  - …
  - needs the **alignment of the semantics** …
- The concept of a **Hot Dog** should be denoted in the terms familiar to the both parties

Ontologies can help …

MOTIVATION

- **An ontology defines the terms used to describe and represent an area of knowledge**

  Ontologies are used by people, databases, and applications that need to share domain information (a domain is just a specific subject area or area of knowledge like **DEDP**, medicine, tool manufacturing, real estate, automobile repair, financial management, etc.). Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them ...

- **Ontologies encode knowledge in a domain and also knowledge that spans domains**
  - In this way, they make that knowledge reusable

Working Draft, W3C Web Ontology Working Group

Ontologies can help …

MOTIVATION

- **Ontologies** are developed to provide a **machine-processable semantics** of **information** that can be **communicated** between **different agents** (software and humans)
- **Domain Ontologies** are:
  - Descriptive models of the Domain
  - Domain Theories

Going Back to the Reality: Industry
(at least the Semiconductor)

MOTIVATION

MR. DEED

- Look on the current design management practices in the semiconductor industry
- Benefits for executives, shareholders and other stakeholders alike are discussed to death all over the place, and there is nothing new to add here
- Here we shall take the viewpoint of a managed victim
- Introduction of a generic character – a model prototype:
  - Mr. Deed (a design engineer)

The World
as Experienced by Mr. Deed

MOTIVATION

- Mr. Deed: "I want to change the world!" (**PRO-ACTIVE** behaviour)
- "Secret" motivations to join Death March projects [see Yourdon pp. 258-262]:
  - Become a superhero
  - Risks are high, but so are the (potential) rewards
  - Challenge of "Mt Everest" or "Marathon" syndrome
  - Naivety or optimism of youth (and inexperience)
  - Avoiding unemployment (scarce job opportunities)
  - Required for future job advancement
  - Avoid (or delay) bankruptcy or other calamity
  - Opportunity to escape "normal" bureaucracy
  - Revenge (sometimes sweet?)

The World
as Experienced by Mr. Deed

MOTIVATION

All the above is somehow sweet, somehow encouraging, but very often conflict-prone

Design
Engineer
Experiencing
Distress

The World
as Experienced by Mr. Deed

MOTIVATION

Preamble: It took mankind 2M years to proceed from the hunting and gathering societies, to the horticultural and pastoral societies, further to the agricultural societies and finally to the industrial societies. That's four types of societies in 2M years.

- Encounter #1:
  "What's on offer for today?"
  - Communication Society
  - Digital Economy
  - Dream Society
  - Experience Economy
  - Era of Glocalization (Globalization & Localization)
  - Information Society
  - Knowledge Society
  - Learning Society
  - Network Society
  - Post-Industrial Society
  - Ubiquitous Network Society
  - ...

The World
as Experienced by Mr. Deed

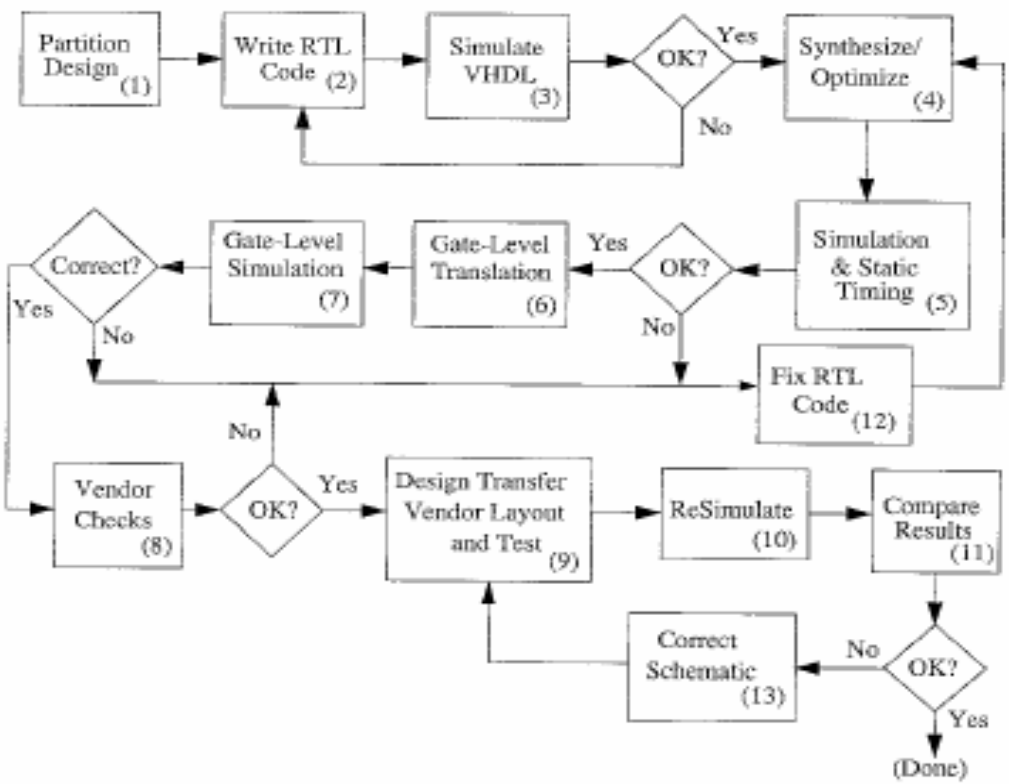
MOTIVATION

- Encounter #2: Use the Automated Travel Expense System
- Encounter #3: Use the Automated Purchase System
- Encounter #4: Use the Automated Human Resource Development System.
- Encounter #5: Use the Automated Project Management System
- Encounter #6: Use the Critical Incidents Reporting System
- Encounter #7: Use the Automated Time Tracking System
- Encounter #8: Use the Customer Relationship Management System
- Encounter #9: …

More Important for Modeling:
as Experienced by Mr. Deed

MOTIVATION

Front-end ASIC Design Process Model



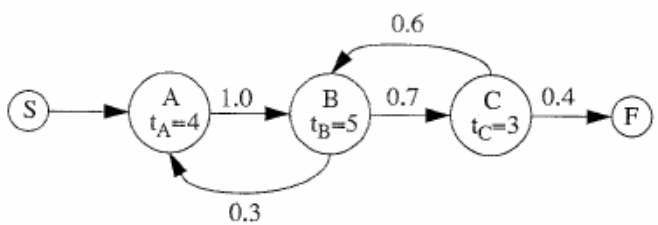
Where is Mr. Deed in this model?

Picture taken from: Eric W. Johnson and jay B. Brockman, “Measurement and Analysis of Sequential Design Processes”, *ACM Transactions on Design Automation of Electronic Systems*, Vol. 3, No. 1, January 1998, Pages 1–20.

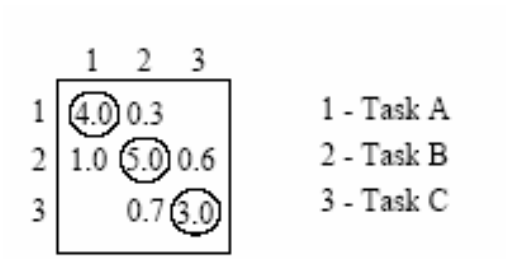
More Important for Modeling:
as Experienced by Mr. Deed

MOTIVATION

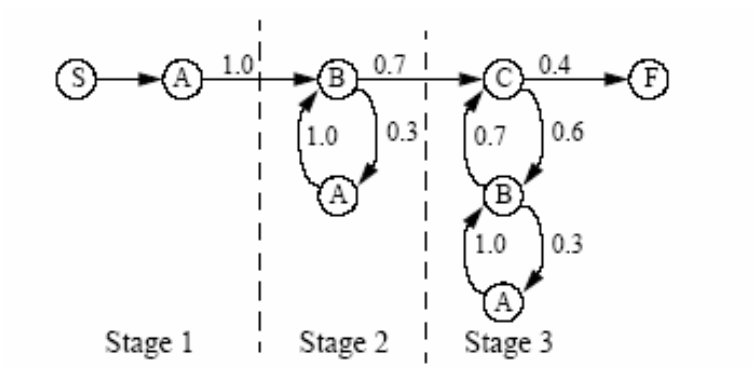
The Underlying Design Process Modes



Example Design Process



Design Structure Matrix



Markov Chain for Updated Design Structure Matrix

Where is Mr. Deed in these models?

An Agent as Mr. Deed

MOTIVATION

- Is the proper metaphor
- However often ignored by Industry
- Has been extensively used in many projects in the past – the **Antique**, the **Middle Ages**
- Is being productively used to model and manage design today – the **Renaissance**
- … Not only because we use it in PSI

The Outlook

- **Motivation**: ☑
  - Why to model Design?
  - Are there difficulties?
  - How to? Agents?
- Agent-based approaches and models in IPD 1993-2005+
- Modeling and Simulation of DEDP-s in PSI
- Conclusions and Discussion

The Outlook

AGENTS IN IPD

- Agent-based approaches and models in IPD 1993-2005+
  - The **Antique** (<1996): Agent-based models in IPD
  - The **Middle Ages** (1996-2002): Refining the basics - models, methodologies, technologies
  - The **Renaissance** (2003+): Refined models and Enhanced opportunities
  - Choosing the **proper focus**
  - One of 2003+: Modeling and Simulation of **DEDP-s in PSI**

Antique (<=1996): Agent-based models in IPD
(just some of the topical examples)

AGENTS IN IPD

- **PACT:** Palo Alto Cooperative Testbed
  +**SHADE**: Shared Dependency Engineering
  - (Stanford U., Lockheed, HP, Enterprise Integration Technologies, 1993)
- **ACDS:** Automated Configuration-Design Service
  - (U Michigan, 1994)
- **DIDE**: Distributed Intelligent Design Environment
  - (TU Compiegne, 1996)
- **ABCDE:** Agent-Based Concurrent Design Environment
  - (U. Calgary, 1996)
- **SHARE:** A Methodology and Environment for Collaborative Product Development
  expanding further into: **FirstLink**, **NextLink**, **ProcessLink**
  - (Stanford Centre for Design Research, EIT Inc., 1996)
- **SiFA**: Single Function Agents
  - (AI in Design Group at Wochester Politech Inst, 1996)

PACT: the Approach*

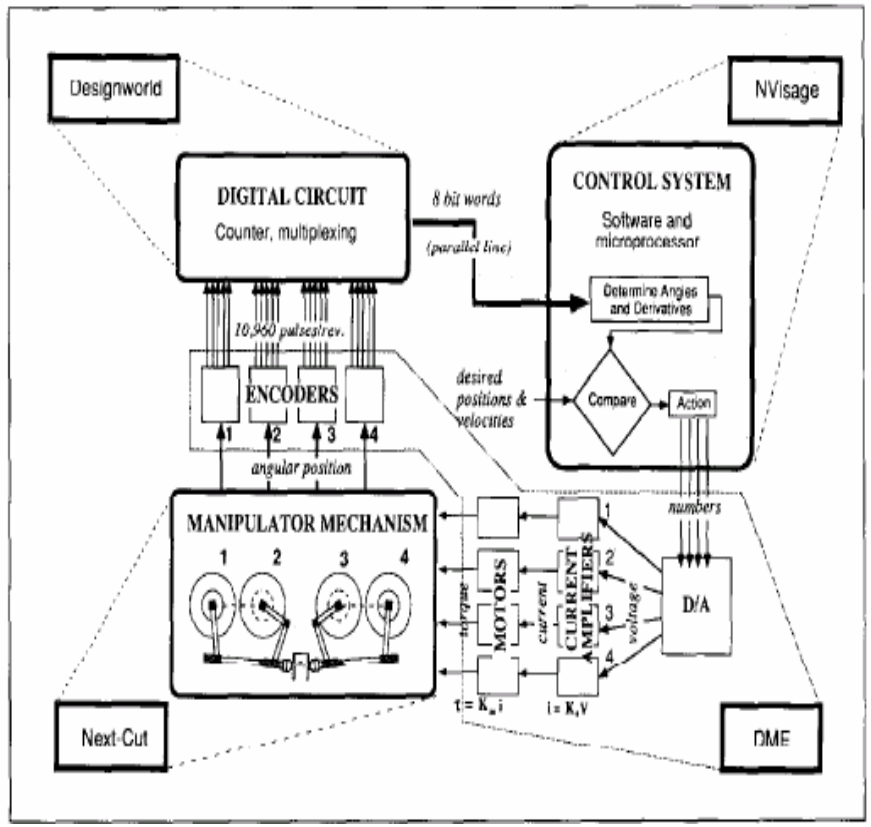
- **The Goal**: to integrate existing multi-tool systems that are themselves frameworks
- **The Constraint**:
  - Individual engineering groups prefer to use their own tool suites and integration environments – there is significant investment in these self-contained systems
- **The Niche**: the projects that involve large segments of an enterprise or multiply enterprises (multi-disciplinary)
- **The task**: to provide the framework for coordination and integration of such activities
- **Dimensions (focus)**:
  - Cooperative development of **interfaces**
  - **Knowledge sharing** among systems
  - Computer-aided support for (human) **negotiation** and **decision making**

* Cutkosky, M.R., Engelmore, R.S., Fikes, R.E., Genereseth, M.R., Gruber, T.R., Mark, W.S., Tenenbaum, J.M. and Weber, J.C.: PACT: An Experiment in Integrating Concurrent Engineering Systems. IEEE Computer 26(1), p. 28-38, 1993

PACT: the Approach

AGENTS IN IPD

- Different agents model different design groups using their own tool suites
- Uses agent communication and **common ontology** (**Shared Design Model**):
  - A unified model is not needed. Instead **tool models are encapsulated**
  - **Shared engineering language** is needed for communication. It only has to cover the shared design model
- Running example of a robotic manipulator design



DIDE: the Approach*

AGENTS IN IPD

- **D**istributed **I**ntelligent **D**esign **E**nvironment is based on an architecture called **OSACA**
  - **O**pen **S**ystem for **A**synchronous **C**ognitive **A**gents
- Very close to **PACT**
- Distributed Design Model
- Agents are "First Class":
  - Truly autonomous
  - Facilitators are not used

* Shen, W. & Barthes, J.-P.: An Experimental Multi-Agent Environment for Engineering Design, International Journal of Cooperative Information Systems, 5(2-3), pp 131-151, 1996

ACDS: the Approach*

AGENTS IN IPD

- Solution of **configuration-design problem** that achieves the benefits of the concurrent engineering (CE) design paradigm
- Design concerns (manufacturability, testability, etc.) are applied to an **evolving design** throughout the **design cycle**
- Attempt to **identify conflicts early on**, which avoids costly redesign and can lead to better products
- Framework is based on a **distributed, dynamic, interval constraint satisfaction problem** (**DDICSP**) **model**
  - Persistent **catalog agents** map onto **DDICSP** variables and **constraint agents** map onto **DDICSP** constraints
  - These agents:
    - Use a set of operations and heuristics to **navigate** through the **space of possible designs**
    - Rapidly **eliminate sets of designs** until a solution is found

*Timothy P. Darr and William P. Birmingham: An Attribute-Space Representation and Algorithm for Concurrent Engineering. CSE-TR-221-94, The University of Michigan, Department of Electrical Engineering and Computer Science, Ann Arbor, Michigan 48109-2122, 1994

ACDS: the Approach

- **ACDS** is a collection of loosely-coupled, autonomous agents that organize communication among themselves based on design constraints
- These agents represent part catalogs and design constraints, and consist of **catalog agents**, **system agents**, **bid agents** and **constraint agents**
  - **ACDS** agent is a computational process:
    - with expertise about a limited portion of a design problem
    - capable of achieving specific goals
  - Agents have the capability to direct other agents to perform operations within the context of the design representation and algorithm
- **ACDS** agents
  - Are distributed functionally and geographically
  - Communicate by passing messages
- To use **ACDS**, a designer provides a **high-level specification of the desired design** and uses this to configure the **ACDS** network

SiFA: The Approach*

AGENTS IN IPD

- **Design** is modeled as a **cooperative multi-agent problem solving task**
  - where different agents possess different knowledge and evaluation criteria (single functions)
- The multi-agent paradigm intuitively captures the concept of deep, modular expertise that is at the heart of knowledge-based design
- By implementing the **opportunistic strategy** in the multi-agent design system, methods are dynamically selected based on:
  - The **individual agents' view** of the problem-solving situation
  - **Shared information** about the capabilities of agents in the system
- Therefore, the **design methodology emerges** at run time

* Shakeri, C., Brown, D.C., Noori, N.: Discovering Methodologies for Integrated Product Design. Proc. Artificial Intelligence and Manufacturing: Second Bi-annual AI & Mfg Workshop, Albuquerque, New Mexico, 1998.

Antique: Lessons Learned

- Many **great insights** for further materialization!
- But, did these activities produce any **industrial-strength solutions**?
  - No evidence in open sources …
- If not, **the reasons**:
  - Design activities are **very difficult to formalize**:
    - Intuitive, creative, highly non-deterministic, …
  - Agent technology was **too immature** by that time:
    - Powerful yet computationally efficient (in **RBS**) formal **frameworks for essential features** were absent
    - **Agent-Oriented Software Engineering Methodologies** were too immature
    - Means for **consensual knowledge representation** were not sufficient

The Outlook

AGENTS IN IP

- Agent-based approaches and models in IPD 1993-2005+
  - The **Antique** (<1996): Agent-based models in IPD ☑
  - The **Middle Ages** (1996-2002): Refining the basics - models, methodologies, technologies
  - The **Renaissance** (2003+): Refined models and Enhanced opportunities
  - Choosing the **proper focus**
  - One of 2003+: Modeling and Simulation of **DEDP-s in PSI**

CERTICON cadence

Middle Ages (>1996): Enabling Solutions

AGENTS IN IPD

- **RAPPID** - Dynamic Design Process Models
- **ADN -** Dynamic Distributed Planning and Coordination

Not only in Design …

- **B-MAN** - Business Mobile Agent Network
- **COMMA** - Corporate Memory Management through Agents
- **E-COLLEG** - Advanced Infrastructure for Pan-European Collaborative Engineering
- **MACRO** – A tool to support Distributed Multi-disciplinary Design and Optimization

- Lots of more examples …

Enabling Solutions (>1996): RAPPID*

AGENTS IN IPD

- **Focus: Dynamic Design Process Models**
- **RAPPID** - **R**esponsible **A**gents for **P**roduct-**P**rocess **I**ntegrated **D**esign
  - (Van Parunak, Altarum, ARPA MADE funded project, 1999)
- **RAPPID** uses a **marketplace** to establish a price-per-unit for each characteristic of a design
- **Agents**, representing design **stakeholders** for each **component**, buy and sell units of these **characteristic**s on a network-based market server
  - A **Component** – Represents a part of the design; buys and sells characteristics in the market; might be organized in a hierarchy; Might be controlled by a human user
  - A **Characteristic** – Definable attribute like weight or power; maintains a marketplace for that item

*H.V.D. Parunak, A. Ward, M. Fleischer, and J. Sauter: The RAPPID Project: Symbiosis between Industrial Requirements and MAS Research. Autonomous Agents and Multi-Agent Systems 2:2 (June 1999), 111-140

CERTICON cadence

Enabling Solutions (>1996): RAPID

- A **Component** that needs more latitude in a given **Characteristic** (say, more weight)
  - Can purchase increments of that characteristic from another Component
  - But may need to sell another characteristic to raise resources for this purchase
- The market-based prices:
  - Reflect the relative scarcity of the various characteristics (that is, which ones constrain the design more closely)
  - Rationalize communication among designers
- In some cases, analytical models of the dependencies between characteristics may help designers estimate their relative costs
- But even where such models are clumsy or nonexistent, prices set in the marketplace define the coupling among characteristics

RAPPID: Set Based Reasoning

AGENTS IN IPD

- A specific **Design Space** is the **Cartesian Product** in the Cartesian **Space of Design Characteristics**
- Low prices = slack characteristics
- High prices = constrained characteristics
- Space can be collapsed by buying up allocations of characteristics…
- Which gives other agents more funds to purchase other characteristics instead…
- Which causes the amount of certain characteristics to get fewer and converge on a price

- **However**, **Cartesian metaphor works well only if the characteristics are orthogonal**
- Difficult to determine the basic set

Enabling Solutions (>1996): ADN*

AGENTS IN IPD

- **Dynamic Distributed Planning** and **Coordination**
- **ADN** – Agent-Based Decision Network (U of Southern Calif, **IMPACT** Lab)
- Focuses on making designers consider other team members' decisions when making their own
- Attempts to achieve coherent design decisions among designers by explicitly representing and enhancing individual design decision making and negotiation processes
- **ADN** is composed of:
  - A decision-based **design process model** (**DDPM**) - captures individual designers' design processes
  - An objective-based **negotiation model** (**OBNM**) - facilitates objective-based negotiation and tracks both dependencies generated and decisions made at each design stage for downstream negotiation support
  - A number of **intelligent agents**, each associated with a **human designer** - generate and utilize the DDPM and OBNM information to support their designers

*Danesh, M. R. and Jin, Y.: An Agent-Based Decision Network for Concurrent Engineering Design. CERA 9(1), 2001, pp 37-47

Enabling Solutions: More examples

AGENTS IN IPD

- **Not only in Design** …
  - **B-MAN** - Business Mobile Agent Network
  - **COMMA** - Corporate Memory Management through Agents
  - **E-COLLEG** - Advanced Infrastructure for Pan-European Collaborative Engineering
  - **MACRO** – A tool to support Distributed Multi-disciplinary Design and Optimization
  - …
  - Lots of more examples …

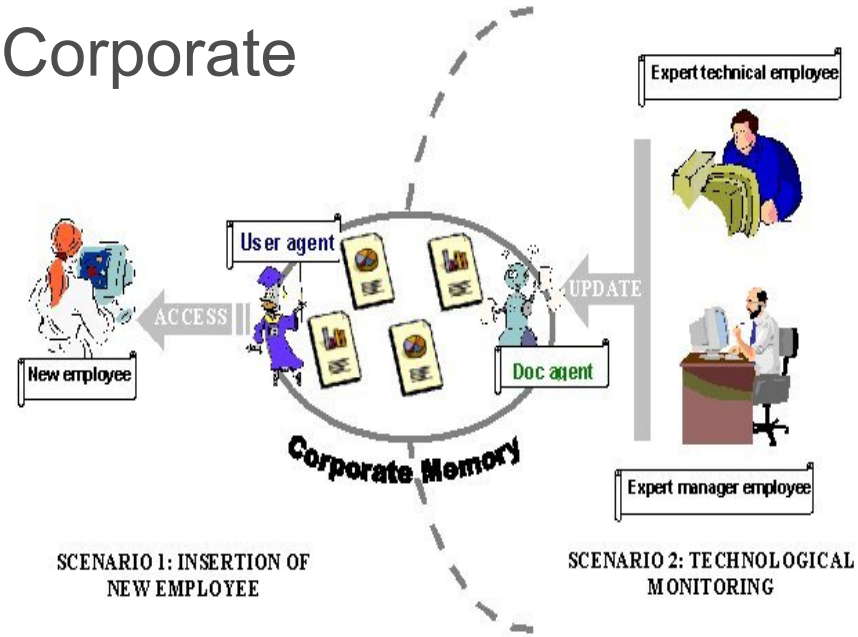
Decentralized Workflow Engine

- **B-MAN - Business Mobile Agent Network**,
  - (IST-2001-32285, http://www.b-man.org/)
- A software platform that aims at enabling the **definition**, **enactment** and **management** of **cross-organizational business processes** on top of the Internet, combining:
  - A decentralized agent-based workflow engine
  - Secure and trusted **contract-based business interactions**
- The **features** that make **B-MAN** different from the previous workflow solutions are:
  - A fully **decentralized** workflow process control engine,
  - Explicitly support **mobile computing**,
  - Provide **trustworthy** (as opposed to merely secure) cross-organizational **process enactment**

Interoperability, CKM, CSCW

AGENTS IN IPD

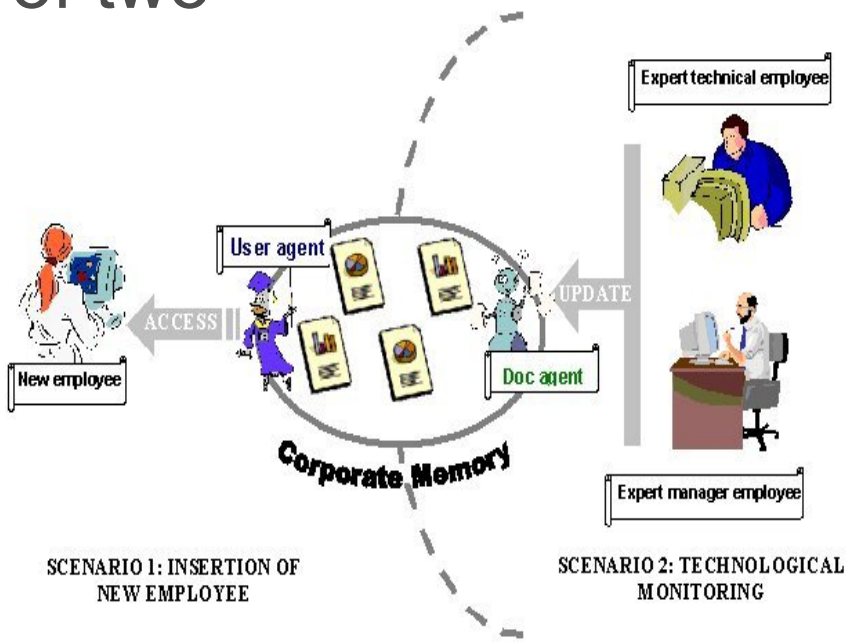
- **COMMA** - **Corporate Memory Management through Agents**
- **Goal**: Implement and test a Corporate Memory management framework integrating several emerging technologies:
  - Agent technology
  - Knowledge modeling
  - XML technology
  - Information retrieval
  - Machine learning techniques
- (IST,2000, http://www.si.fr.atosorigin.com/sophia/comma/)



COMMA - Corporate Memory Management

AGENTS IN IPD

- The project intends to implement the system in the context of two scenarios:
  - Enhance the insertion of new employees in the company,
  - Perform processes that detect, identify and interpret technology movements and interactions for matching technology evolutions with market opportunities to diffuse among employees innovative ideas related to technology monitoring activities



Interoperability: Advanced Infrastructures

AGENTS IN IPD

- **E-COLLEG** - Advanced Infrastructure for Pan-European Collaborative Engineering
  - (IST-1999-11746, Jan. 2000 - Dec. 2003, http://www.ecolleg.org)
- **Key target - ACI:** Advanced Collaborative Infrastructure that will enable seamless Internet-based (multi-site and multi-platform) integration and management of **tools** and **data**
- **ACI** components:
  - Basic Collaborative Services
  - Advanced Tool Registration and Management Services (TRMS)
  - XML-based Integration Technologies
  - Collaborative Extensions to Design Tools
- **Mr DEED ???**

Networked CSCW Methodologies

- **MACRO** – A tool to support Distributed Multi-disciplinary Design and Optimization (DMDO)
  - (EPSRC Project (GR/L91245) June, 1998 – June, 2001, http://www.cranfield.ac.uk/coa/macro/)

- **The Goal**: to create **a system of working** in a **distributed environment** with **non-collocated design teams** such that
  - the resulting process is as effective and efficient as that achieved with **collocated Integrated Product Teams**

Middle Ages (>1996): The Key Assets

AGENTS IN IPD

- Dynamic Design Process Models
  - Like Design MarketPlace in Rappid on which Design Characteristics are tradable by agents
- Set-based reasoning to combine design characteristics on in the Design Space
- Agent-based decision framework for dynamic distributed planning and coordination
- Several activities aimed to develop the basics – different aspects of agent technology

Middle Ages (>1996): the Impression is:

AGENTS IN IPD

- The **Antique** and the **Middle Ages** in **Agent-Enabled Engineering Design** have ended up with **NO SILVER BULLET**
- The **successors** of the mentioned projects:
  - though **been declared** (e.g., Design Space Colonization (DSC), Engineering Design Modeling (EDM)
  - have either **gone to industry** (classified)
  - or provide **no evidence of substantial progress** (open sources)
- The (academic) research has concentrated on the development of the enabling **solutions**, **methodologies** and **infrastructures**

The Outlook

AGENTS IN IPD

- Agent-based approaches and models in IPD 1993-2005+
  - The **Antique** (<1996): Agent-based models in IPD ☑
  - The **Middle Ages** (1996-2002): Refining the basics - models, methodologies, technologies ☑
  - The **Renaissance** (2003+): Refined models and extended opportunities
  - Choosing the **proper focus**
  - One of 2003+: Modeling and Simulation of **DEDP-s in PSI**

CERTICON cadence

Renaissance (2003+): Refined Models
Just few (3) topical examples

AGENTS IN IPD

- **Addressing Design Process Dynamics**
  - Catholijn Jonker and Jan Treur
    - Vrije Universiteit Amsterdam, Universiteit Utrecht
  - a declarative, logical approach for specification of dynamic properties of design processes, supported by a formal temporal language
- **Creativity** in Design is not an **Individual** but a **Social Property**
  - John S. Gero's group
    - Key Centre of Design Computing and Cognition, University of Sydney
- **I-DIMS** – a Holistic approach to describe Design Processes with Ontologies
  - CODE Group at NUI Galway

Design Process Dynamics*:
Temporal Logic

- Component-based design processes (highly dynamic)
- Re-use of the components in design
- Important concepts:
  - A **design problem statement**:
    - A set of requirements in the form of dynamic properties on the overall system behaviour that have to be fulfilled
    - A partial description of (prescribed) system architecture that has to be incorporated
    - A partial description of (prescribed) dynamic properties of elements of the system that have to be incorporated; e.g., for components, for transfers, for parts, for interactions between parts
  - A **solution specification** for a design problem
    - A specification of a design object (both structure and behaviour)
      - that fulfils the imposed requirements on overall behaviour
      - and includes the given (prescribed) descriptions of structure and behaviour
- Formalisation of design process dynamics is discussed in terms of **design states**, **design transitions** and design **traces**

*Tibor Bosse, Catholijn Jonker, and Jan Treur: Analysis of Design Process Dynamics. In: Proc. of the 16th European Conference on Artificial Intelligence, ECAI'04. IOS Press, 2004.

Design Process: States and Transitions

- The state of a **DP** at a certain time point – a combined state comprising the two states, ***S*** = <*S1*, *S2*> :
  - *S1* **requirements manipulation state** (RM-state), including the current requirements set
  - *S2* **design object description manipulation state** (DM-state), including the current design object description state
- A particular design process shows the sequence of transitions from one state ***S*** to another (next) state ***S'***

Design Process: Transitions and Traces

- Design **traces** are time-indexed sequences of design states
- In a design trace each subsequent pair of states is a design **transition**
- A fixed **time frame T** is assumed which is **linearly ordered**
- Depending on the application, the **time frame T** may be:
  - **Dense**
    - e.g., the real numbers
  - **Discrete**
    - e.g., the set of integers or natural numbers
    - a finite initial segment of the natural numbers
  - Any **other form**

Linearly Ordered

Design Process: Traces

- A **trace** γ over:
  - A state ontology **Ont** (including ontology for design objects and requirements)
  - And time frame T

  is a <u>mapping</u> γ: T $\rightarrow$ **STATES(Ont)**, i.e., a sequence of states $\gamma_t (t \in \mathrm{T})$ in **STATES(Ont)**

- The **<u>set of all traces</u>** over state ontology **Ont** is denoted by **TRACES(Ont)**

Design Process: Dynamic Properties, TTL*

- **DYNPROP(*Ont*)** is the set of temporal statements that can be formulated wrt traces based on the state ontology ***Ont***
- Given a trace γ over the state ontology ***Ont***, a certain state of a design process at time point **t** is denoted by **state**(γ ,**t**), which as a TTL-expression refers to γ_t
- These **states** can be related to **state properties** via the formally defined **satisfaction relation |=**, comparable to the **Holds** predicate in the **Situation Calculus**:
  - **state**(γ, **t**) |= **p** denotes that state property **p** holds in trace γ at time **t**
- Based on these, dynamic properties can be formally presented in a sorted first-order predicate logic with:
  - sorts **T** for time points, **Traces** for traces, and **F** for state formulae

using quantifiers over time and first-order logical connectives $\neg, \wedge, \vee, \Rightarrow, \forall, \exists$

*TTL – Temporal Trace Language

Creativity in Social Structure* (Agents):
To Adopt or to Abstain a Design Solution?

AGENTS IN IPD

- Creativity is the property which value is determined:
  - Not wrt an individual "generator"
  - But by the group of "evaluators"
- The approach focuses on the relation between individual-generative and group-evaluative processes
- Creativity is seen as a social construct or a communal judgment (following Feldman et al. 1994)
- A creative designer is considered not in isolation but in interaction with an environment of social and epistemological dimensions
- Creativity is determined by the diffusion of a design solution across a social group

*Sosa, R and Gero, JS: A computational framework to investigate creativity and innovation in design, *AIEDAM* (to appear) 2005

Experimental Test-Bed Based on DIFI

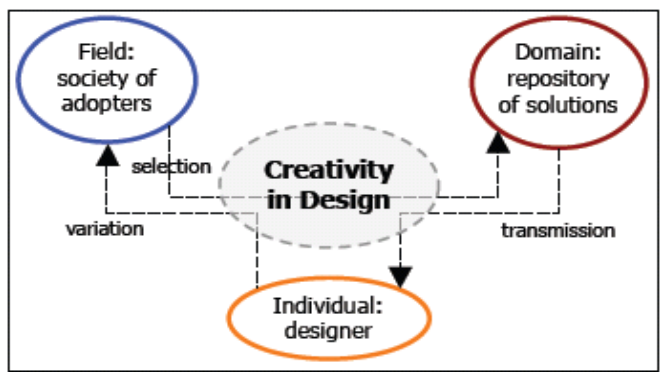
- To explore qualitative generalisations about the nature of creative behaviour in design
- The framework is based on the **D**omain-**I**ndividual-**F**ield **I**nteraction model (**DIFI**) (Feldman et al. 1994):
  - Locates **creativity** outside the **individual creator**
  - Places it in the **interrelations of** three main parts of a system: **domain**, **field** and **individual**
    - The **Domain** consists of the set of solutions, knowledge, techniques, and evaluation criteria **shared by** the members of a given **community**
    - **Fields** include groups of individuals who share a common domain
- The key implication of the **DIFI** model:
  - **Situated in a dynamic environment, creative designers are those who:**
    - **generate the right product**
    - **at the right place**
    - **and at the right time**

    where 'rightness' is largely defined by evolving social standards

Method of Study: Agent-Based Simulation

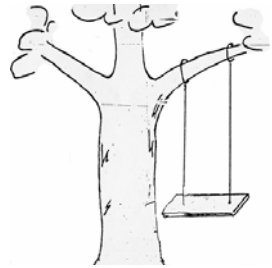
AGENTS IN IPD

- Multi-agent based simulation of social phenomena is the primary method of inquiry (following Gilbert and Troitzsch, 1999)
- To define and implement in computer simulations:
  - Different actors and components of a system
  - The rules that may determine their behaviour and interaction.
- Results in the systematic study of their likely characteristics and effects when the system is run over simulated time
- Manipulate the experimental variables at initial time – to extract patterns from the observed results over time and build hypotheses in relation to the target system
- A framework of social agency includes:
  - a small number of competing designer agents
  - large social groups of clients or adopter agents
  - and a cumulative repository of design artefacts that represent the design domain



The Adoption Framework

… has been designed



…adopter's pattern

AGENTS IN IPD

- Adopter behaviour consists of evaluating artifacts generated by designers and deciding to adopt or abstain
- Artifacts are simplified shapes (linear representation)
  - Illustrates an evaluation function based on intuitive visual geometric features
  - Supports multiple interpretations by adopters and shape emergence
- The objective is to support some of the key aspects of design problems in multi-objective decision making
- Clients or adopters in this system evaluate artefacts according to individual thresholds of perception and preference
- Variation of perception across a population enables different interpretations of artifacts

I-DIMS*: Ontologies to Define Collaborative Design Processes

AGENTS IN IPD

- Addresses the problem of **collaborative design**
- Investigates and develops a **holistic approach** to knowledge management for the design process
  - **Ontologies**
  - Intelligent **agent based systems**
  - A distributed design environment (DDE).
- Ontologies are used as t**he kernel** for the future **development** of a **multi-agent system**
- the users and information are structured in clusters (Social Groups) defined by specific perceptual, effectual, and intellectual capabilities

*Camelia Chira, Ovidiu Chira, Thomas Roche: Multi-agent Support for Distributed Engineering Design. IEA/AIE 2005: 155-164

CERTICON cadence

I-DIMS: Engineering Design Process

Emphasis on Designers and Teams

AGENTS IN IPD

- Engineering design is a (dynamic) process
- Designers and design teams make decisions (at run-time) to solve specific problems, based on their own:
  - implicit knowledge, (e.g. creativity, knowledge, experience, imagination, originality etc.),
  - explicit knowledge, (e.g. case bases, design catalogues, design reuse etc.).
- Rich representation of a Designer (a Team) facilitates the information transformation
  - from customer requirements
  - to the detailed design of an artifact (inc. function, use, behaviour, performance, manufacture, operation, maintenance, and disposal)
- **The scope of the engineering design ontology is to create a common shared understanding of the engineering design domain so that information and knowledge can be shared among the actors of the distributed design environment (DDE)**
- These **Actors** can be **humans** (e.g. designers) or **software agents**

Renaissance (2003+): The Summary

AGENTS IN IPD

- The **features** which received special attention:
  - Design Process Dynamics
  - Heterogeneity and Openness of Design Teams
  - Designers (individuals) and Teams (Social Structures)
  - Interoperability, collaboration, design ontologies
- …
- **PSI** has tried to carefully combine these aspects in its modeling framework
  - Process Dynamics
  - Performed by teams with social laws (obeying the System Laws)
  - Mr Deed is the 1-st class citizen
  - Assessing and increasing the Productivity of a DP
  - Aiming at the Industrial strength solution

The Outlook

AGENTS IN IPD

- Agent-based approaches and models in IPD 1993-2005+
  - The **Antique** (<1996): Agent-based models in IPD ☑
  - The **Middle Ages** (1996-2002): Refining the basics - models, methodologies, technologies ☑
  - The **Renaissance** (2003+): Refined models and Enhanced opportunities ☑
  - Choosing the **proper focus – ready now**
  - One of 2003+: Modeling and Simulation of **DEDP-s in PSI**

CERTICON cadence

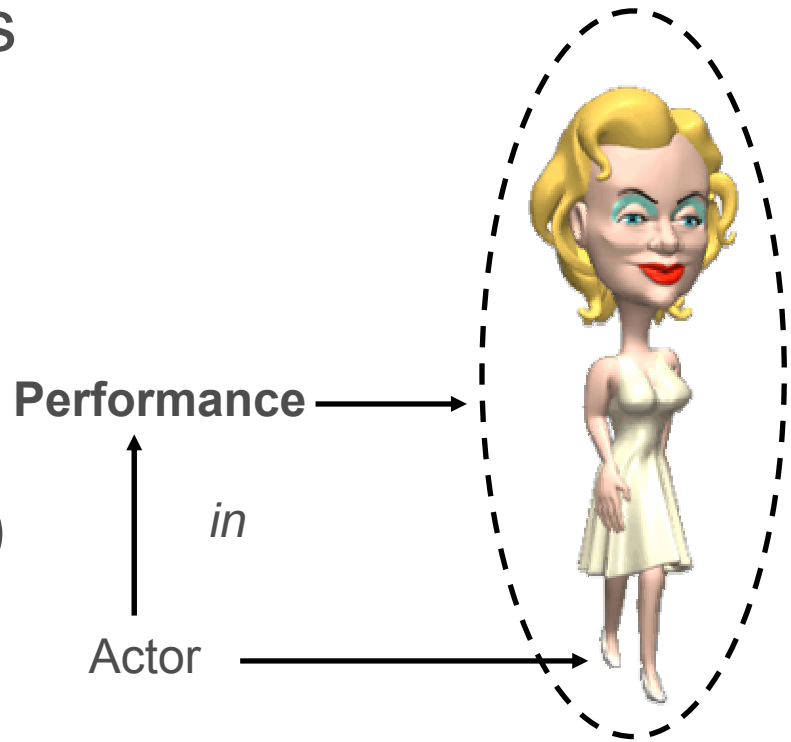
What should be Left to a Human?

In an Engineering Design Process

AGENTS IN IPD

- Being an Individual: inspired, intuitive, creative
- Performing design
- Generating Design Artifacts

- Be rational – don't expect agents pretend to be humans (Turing Test)



The Outlook

AGENTS IN IPD

- Agent-based approaches and models in IPD 1993-2005+
  - The **Antique** (<1996): Agent-based models in IPD ☑
  - The **Middle Ages** (1996-2002): Refining the basics - models, methodologies, technologies ☑
  - The **Renaissance** (2003+): Refined models and Enhanced opportunities ☑
  - Choosing the **proper focus** ☑
  - One of 2003+: Modeling and Simulation of **DEDP-s in PSI**

ER2005

24-28 October 2005

cadence

Modeling and Simulation
of Dynamic Engineering Design Processes

PART 2: Productivity Simulation Initiative

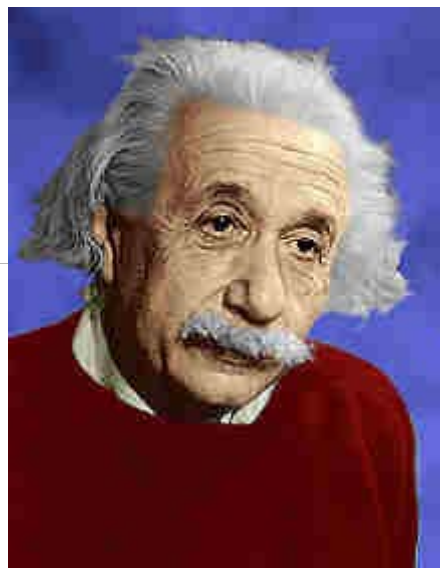
CERTICON

The Outlook

- **Motivation**: ☑
  - Why to model Design?
  - Are there difficulties?
  - How to? Agents?
- Agent-based approaches and models in IPD 1993-2005+ ☑
- Modeling and Simulation of DEDP-s in PSI
- Conclusions and Discussion

There is Knowledge …

… that shapes out
the world we live in



$$E = m c^2$$

There are Rumors …

Source, e.g.:

Matzke, W.-E.: **Engineering Design Performance Management – from Alchemy to Science through ISTa**. Invited talk at the 4th International Conference on Information Systems Technology and Applications, 23-25 May 2005, Palmerston North, NZ

$$E = mc^2 + \Delta$$

at least in SES design

FOR EYES ONLY

CERTICON cadence

That's what the Story is About!

PSI: how to gain this Δ in Engineering Design Processes

The Recipe: "Intrapreneural" Spirit

To read more, e.g.: **FT, Monday July 25 2005, p. 9**

"**Intrapreneural**":

- Invented by Norman Macrae (Economist, 1976)

  "… **dynamic** corporations of the future should simultaneously be trying alternative ways of doing things in **COMPETITION** WITH **THEMSELVES**."

- **The keywords: Dynamic, Competition, Themselves**
- **Engineering Design Processes are right there:**
  - **Dynamic Engineering Design Processes (DEDPs)**

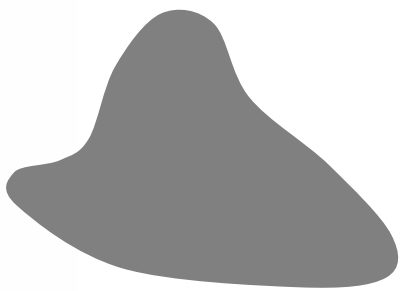
Suppose you Want to Build
a House of your Dream

- So, you have (at least) to:
  - dig the foundation pit
  - take care of the dump

Looking for Δ

- If you do it yourself:
  - forget about your senior designer's salary for at least a month
  - spend more for food
  - still pay a dump-truck to take the dump off

Let expected expenses be **E**



Striving for Δ

- If you hire a mate:
  - need to pay him for the job
  - may have your salary
  - may have your beer
  - with your family

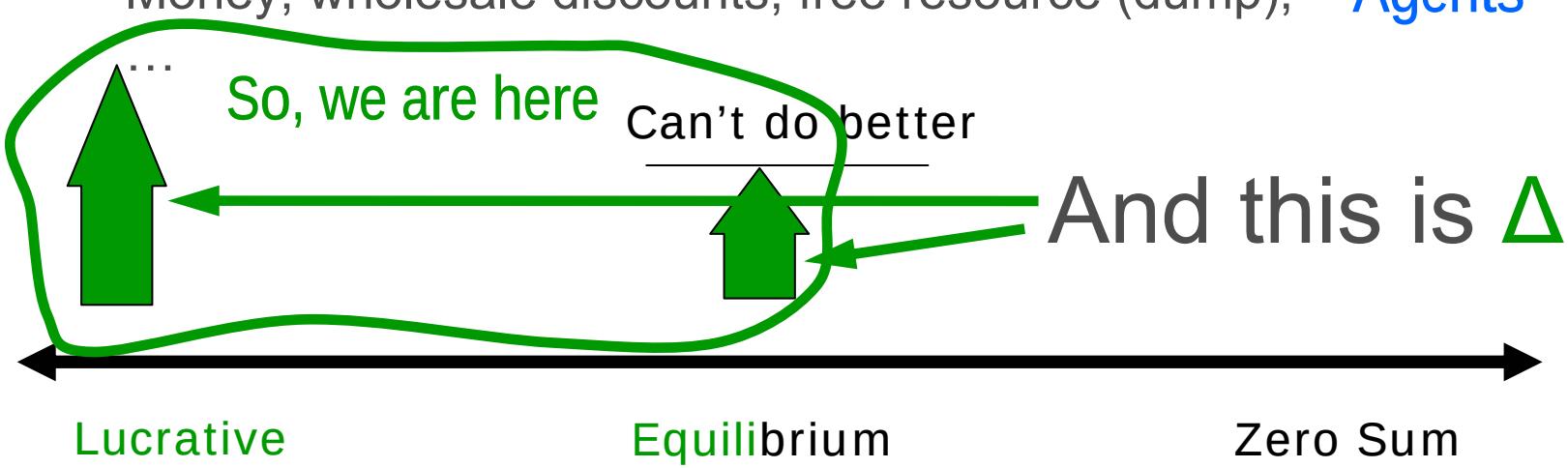
- The mate:
  - digs the soil at your neighbor's
  - has tools and skills
  - gets discounts for fuel (wholesale)
  - will be happy to take your dump for free – smb asked him to cover up a hole

Suppose he asks for **P** <**E**, =**E**, …

Hiring a Mate is SUPER ADDITIVE (Δ)
(in the terms of the Game Theory)

- In this **ENCOUNTER**
  - You are doing better as if you are on your own
    - Income, dump-for-free, time, fun, family, …, beer
  - The mate is doing better as if he is on his own
    - Money, wholesale discounts, free resource (dump), …

A Mechanism to find out where we are is Negotiation among Software Agents

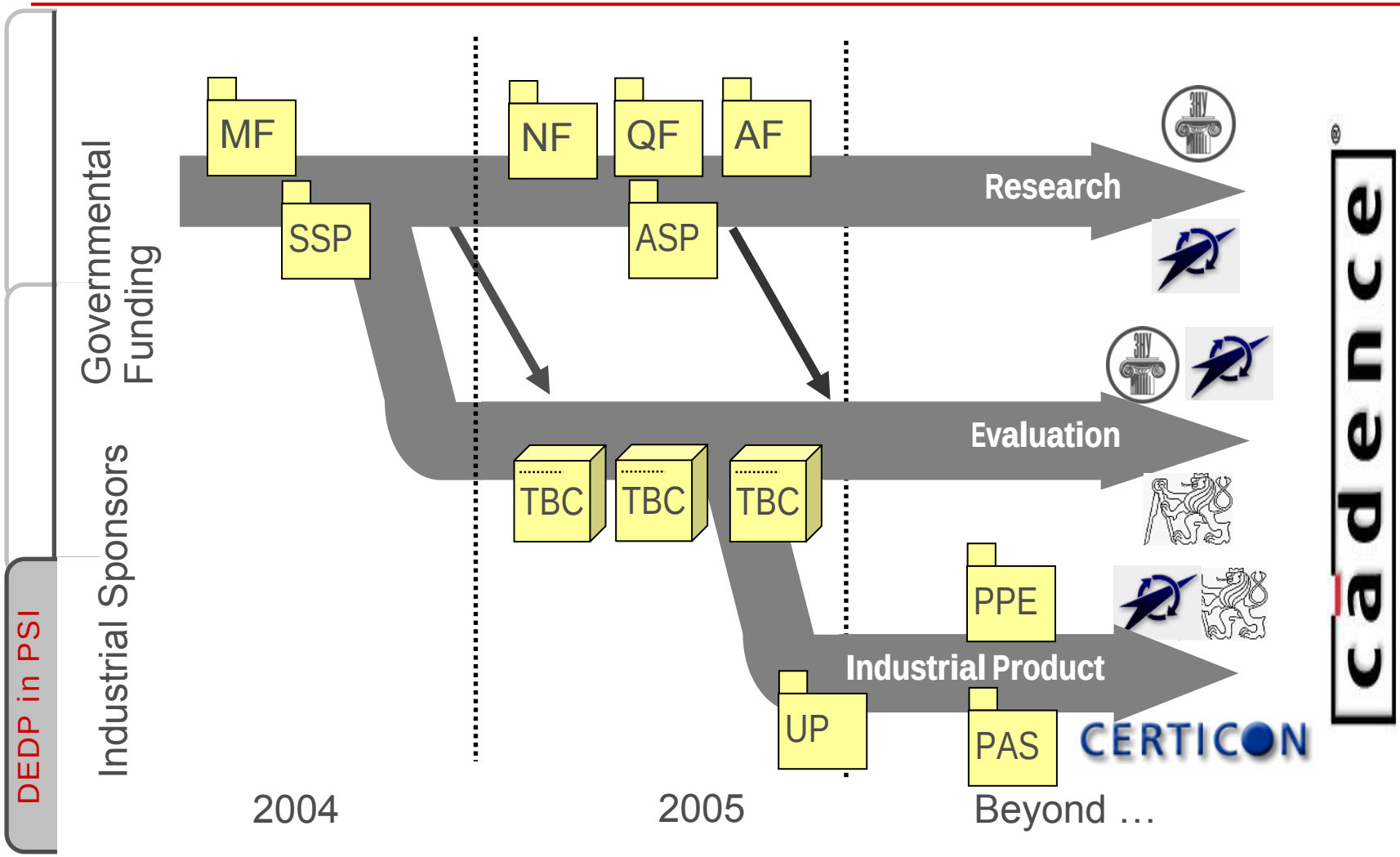


The Outlook

- Modeling and Simulation of DEDP-s in PSI
  - PSI Motivation – emerging collaboration in a DEDP is super-additive ☑
  - PSI project lines and partners: research, prototyping, industrial implementation
  - The factors providing D in EDPs
  - Productivity assessment and process run-time optimization (PSI DEDP modeling framework)
  - PSI simulation modes (playback, predictive)
  - MASDK as the rapid prototyping toolkit

DEDP in PSI

PSI Project Lines and Partners



PSI Project Lines: Research

- **Frameworks**
  - Modeling Framework: Conceptual, Architectural
  - Negotiation Framework
  - Productivity Assessment, Quality, Ability (Mr. Deed)
- DEDP **Ontologies** Family:
  - Actor Ontology
  - Task-Activity Ontology
  - Design Artifact Ontology
  - Design Tool (usage) Ontology
- **Collaboration Mechanisms**

Research Prototypes

The Outlook

- Modeling and Simulation of DEDP-s in PSI
  - PSI Motivation – emerging collaboration in a DEDP is super-additive ☑
  - PSI project lines and partners: research, prototyping, industrial implementation ☑
  - The factors providing D in EDPs
  - Productivity assessment and process run-time optimization (PSI DEDP modeling framework)
  - PSI simulation modes (playback, predictive)
  - MASDK as the rapid prototyping toolkit

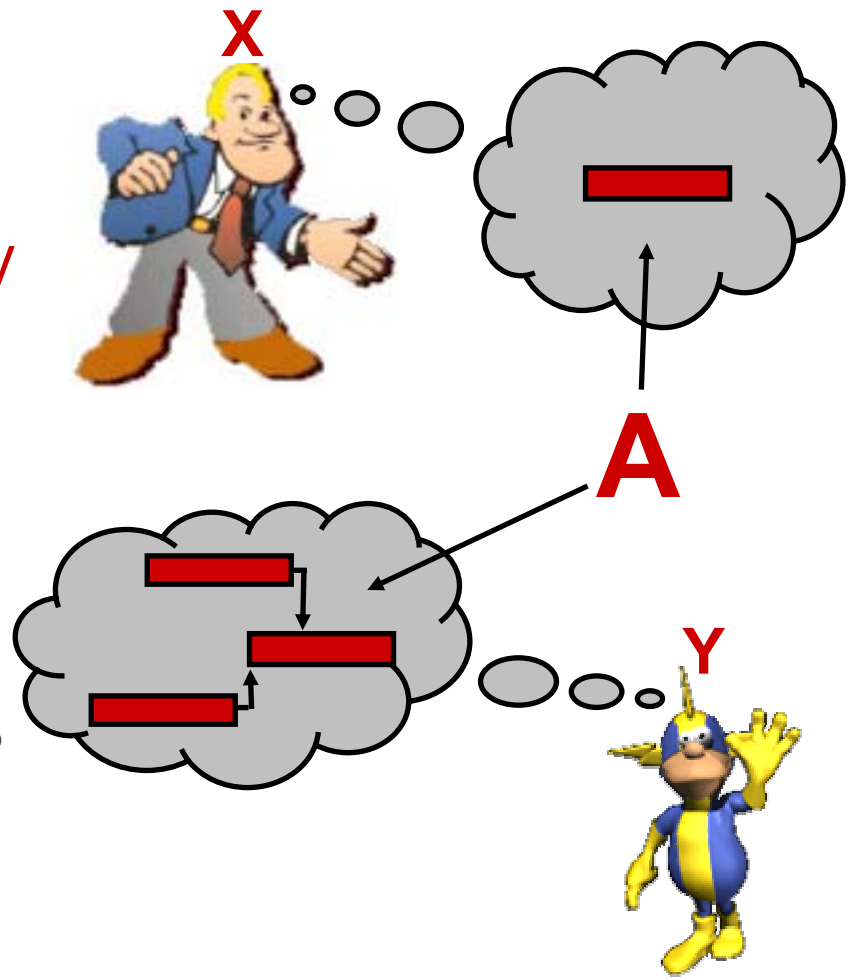
DEDP in PSI

What is a **D**ynamic **E**ngineering **D**esign **P**rocess?

- A **DEDP** is the process of aiming a weakly defined **engineering design workflow** to achieve its goal in an optimal way (**Δ**) 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 affecting DEDP Dynamics:
Subjective knowledge on Activities

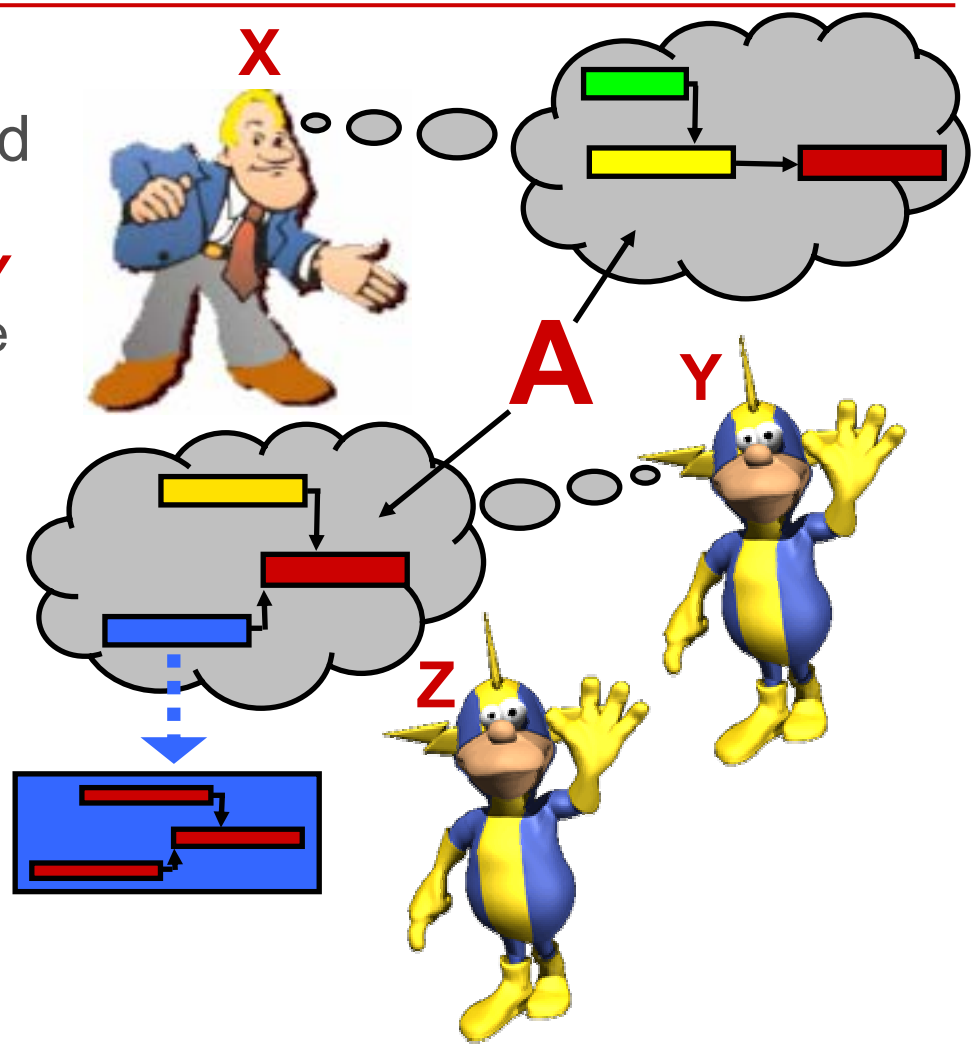
- Different **Agents** have different knowledge and capabilities wrt a **DEDP**
  - Agent **X** may treat an Activity **A** as atomic – i.e. non decomposable
  - Agent **Y** may treat **A** as composite – i.e. a **Task**
- **X** and **Y** (if assigned) will perform **A** in different ways (with different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



Factors affecting DEDP Dynamics:

Composition is subjective and partial

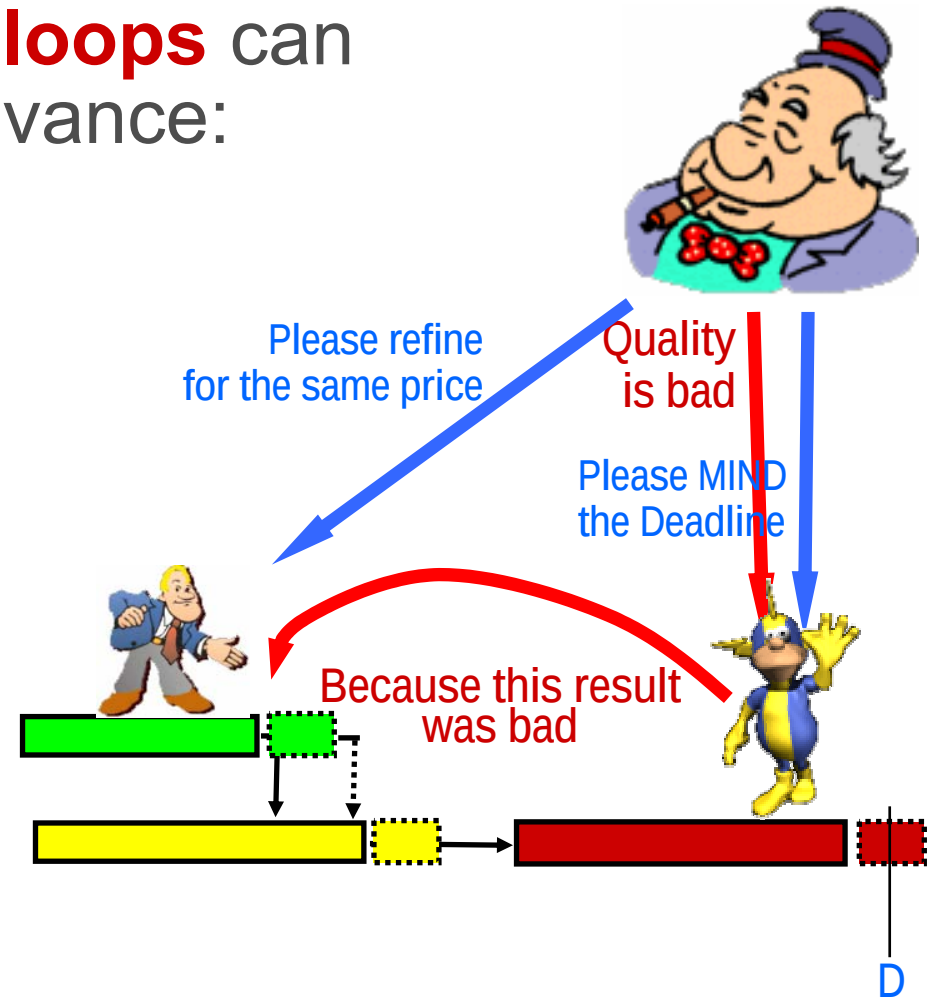
- Activity **composition** is performed **subjectively** and **partially:**
  - **Subjectively**: Agents **X** and **Y** may have different knowledge on how to compose a Task of Activities
  - **Partially**: Activities may also (further, e.g., by Actor **Z**) appear to be Tasks
- **Implication**: Activities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



Factors affecting DEDP Dynamics:

No of activity loops is not predefined

- The number of **activity loops** 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**

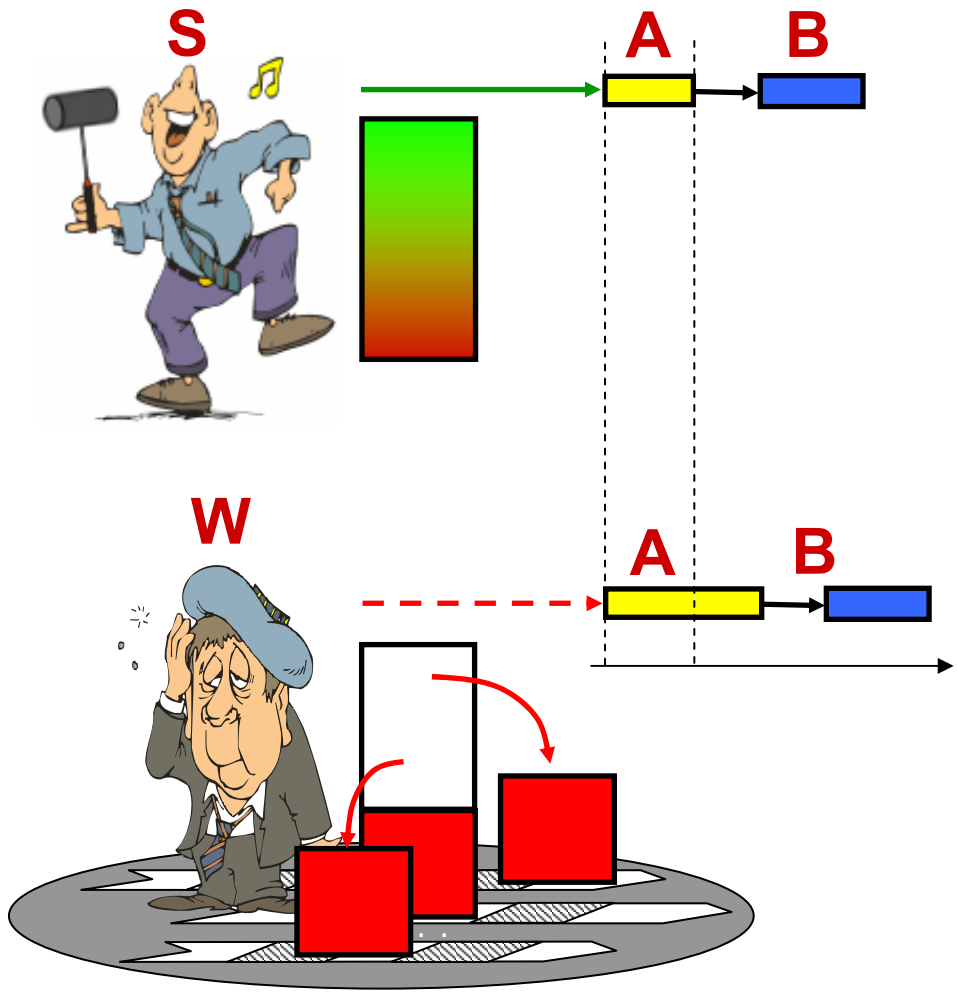


DEDP in PSI

Factors affecting DEDP Dynamics:

Activity duration depends on the available capacity

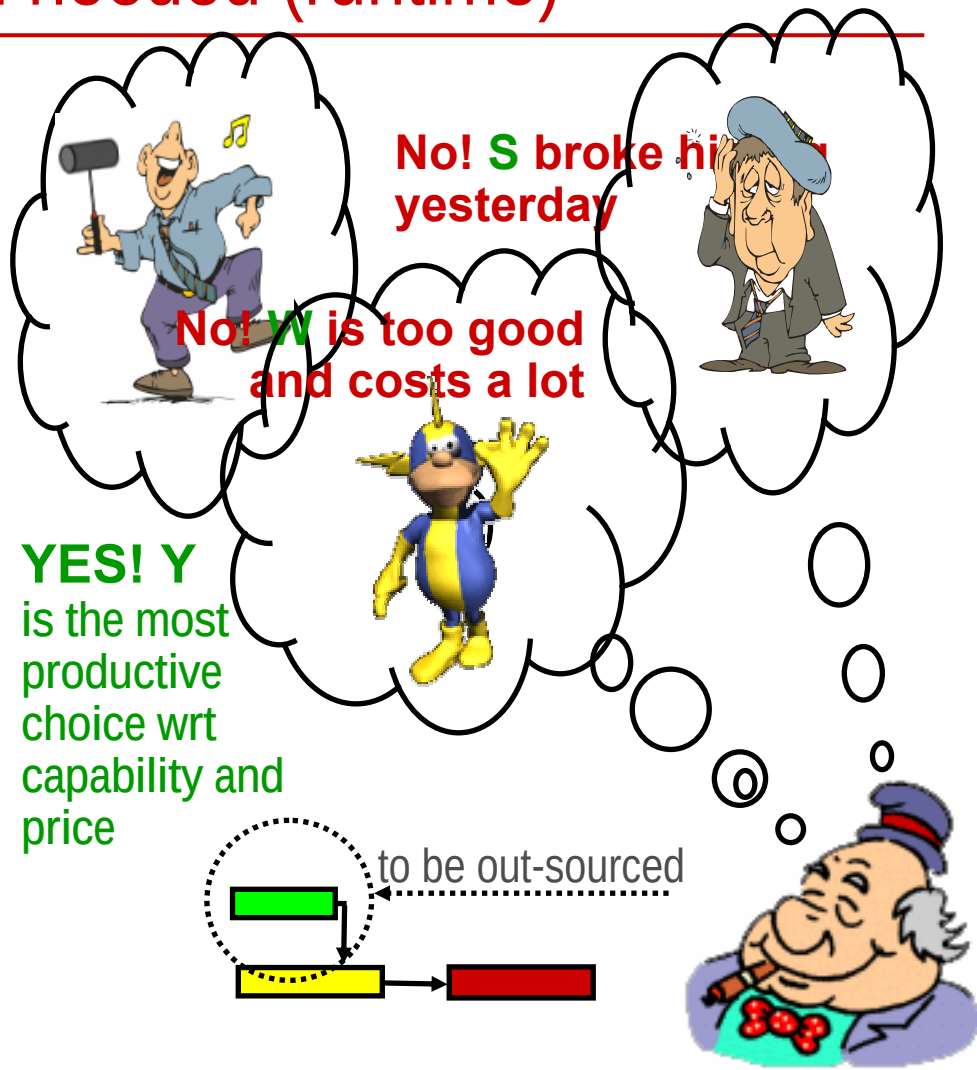
- Mr. **S** is highly productive wrt **A**
- Mr. **W**:
  - Can also be highly productive wrt **A**
  - But spends his capacity to several other **DEDP**s
- **B**, though allocated, remains idle for different time (cant be pre-determined)
- Requires run-time **re-scheduling**



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**



The Outlook

- Modeling and Simulation of DEDP-s in PSI
  - PSI Motivation – emerging collaboration in a DEDP is super-additive ☑
  - PSI project lines and partners: research, prototyping, industrial implementation ☑
  - The factors providing D in EDPs ☑
  - Productivity assessment and process run-time optimization (PSI DEDP modeling framework)
  - PSI simulation modes (playback, predictive)
  - MASDK as the rapid prototyping toolkit

DEDP in PSI

CERTICON cadence

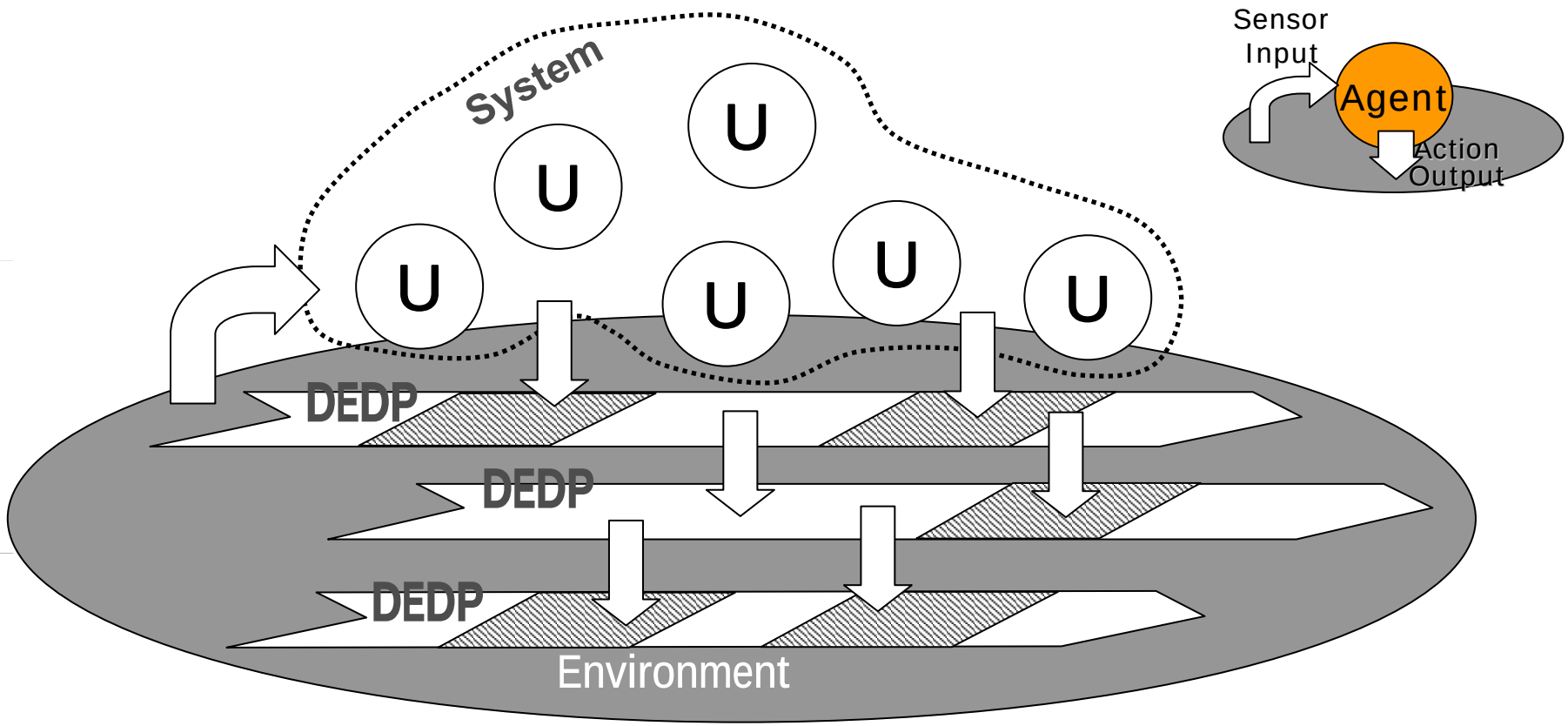
DEDP Productivity Assessment

- **Definition**: Productivity is the amount of **output** produced per unit **input** used by a **system** (in a process)

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

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**

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 **DEDP**s) 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 **DEDP**s which were actually used to collect the Utility

UoW may be Gained, Spent, or Lost through Collaboration

- **Collaboration** occurs when:
  - An **Actor out-sources** 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 **DEDP**s
- 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**)

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** (agents) of the same
- **Mechanisms** to arrange their Collaboration:
  - **Protocols** for different encounters
  - Behaviour **Strategies**

What do we Need to Model a DEDP?

(Mind dynamics factors, productivity measure and Mr. Deed)

- 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
- A DEDP is performed by actors which collaborate in teams – earn and spend their UoW
- Actors use software tools and other resources to execute activities
- The overall goal of a DEDP is to design the Artifact (e.g. a Chip)
- Activities result in some incremental portions of the artifact under design

DEDP: Actors – 1-st Class Citizens
(Economically rational, balanced rationality)

- Actors have various capabilities and capacities (Self-Beliefs)
- Actors may form groups (teams) to perform a DEDP
- Actors play different roles in these teams
- Actors should have beliefs on the others in their team or around
  - To have an idea on what they are capable to do
  - To have an assessment of how much the collegues are credible
- Actors should have the mechanisms for communication and collaboration (Utilitarian)
- Actors may take part in different DEDP-s at a time
- Actors have commitments wrt the parts of the DEDP
- Actors pledge to follow some system laws (team- or organization-level conventions)
- Actors' activities are constrained by policies
- Actors execute activities to design artifacts, consume resources and use software tools for that

DEDP: Tasks and Activities

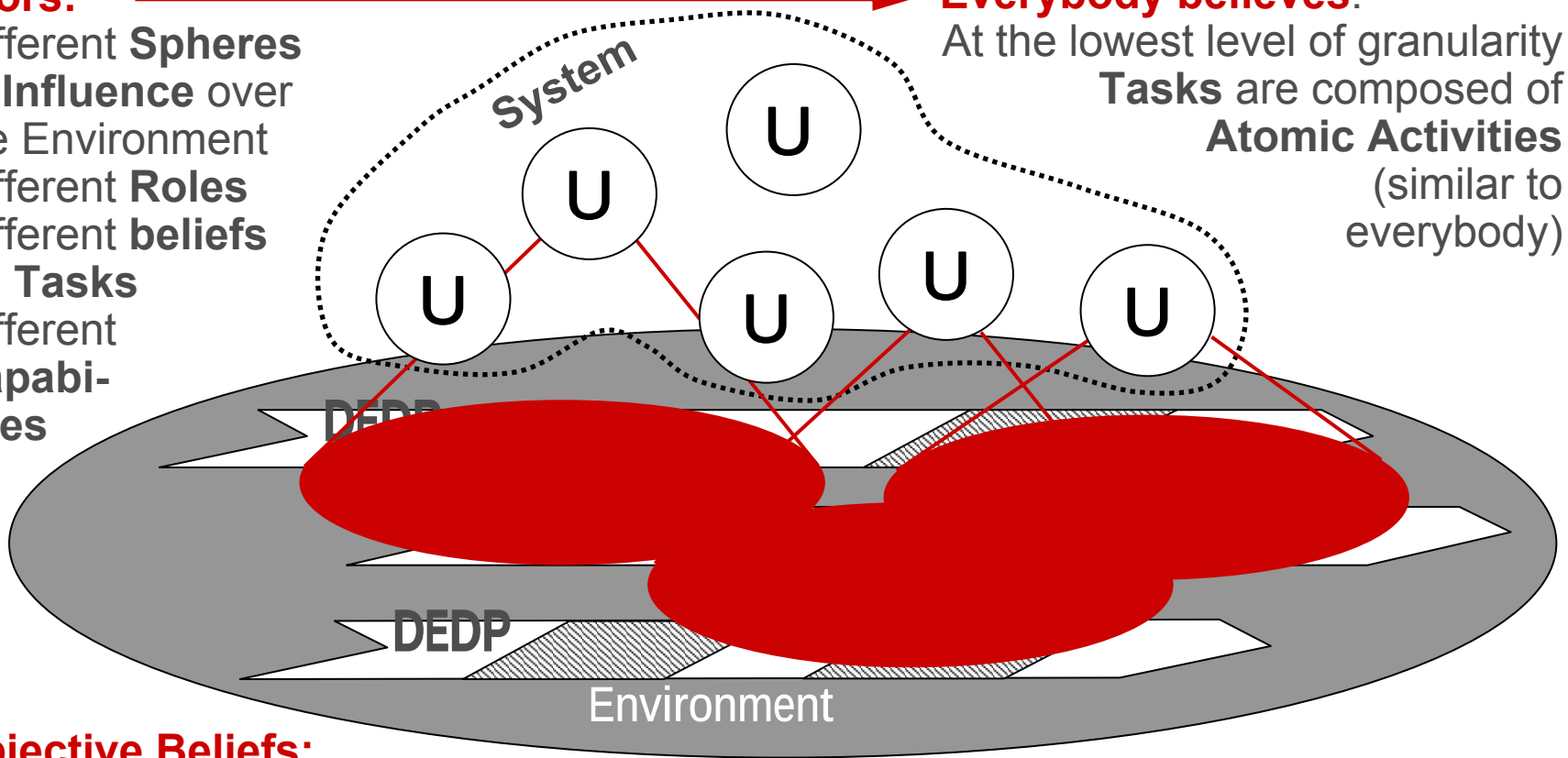
DEDP in PSI

Actors:

- Different **Spheres of Influence** over the Environment
- Different **Roles**
- Different **beliefs on Tasks**
- Different **Capabi-lities**

SHARED KNOWLEDGE

Everybody believes:
At the lowest level of granularity **Tasks** are composed of **Atomic Activities** (similar to everybody)



SUBJECTIVE KNOWLEDGE

Subjective Beliefs:

Task Manager: I'm offering the Task which I Believe to be Atomic (not interested in the details)

Contractor: In my Beliefs the Task you offer comprises several Sub-Tasks and Activities. I can Perform some Sub-Tasks and can Execute some of the Activities

DEDP: Tasks and Activities

- An **Activity** – the basic building block (for everybody), defined by the **Design Technology**
- An **Activity** is **Executed** on its **Material Inputs** (Design Artifacts) and **Produces Material Outputs** (Design Artifacts)
- 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 (knowledge - Design Artifact) transformation process modeled as the **Task managed By** the certain **Actor** (the Task Manager)
- A **Contractor** Commits himself to a **Task**(managed)**ByActor** by **striking the deal** in Contracting negotiations
- A **Contractor** may become either the **Task Manager** or the **Activity Performer**
- The **UoW** are associated with the **Task**(managed)**ByActor**

DEDP: 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

DEDP: 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
- $\mathbf{t_1}$ is **strongly dependent** of $\mathbf{t_2}$ implies:
  - All the **Material Outputs** of $\mathbf{t_2}$ **Activities** are **available** and **will be used** as the **Material Inputs** by the **Activities** of $\mathbf{t_1}$
  - The **Pre-condition** of $\mathbf{t_1}$ is the **event** of the appearance of the **Material Inputs** produced in $\mathbf{t_2}$ (**Eventual Output**)
  - **Eventual Input** of $\mathbf{t_1}$ is the **Eventual Output** of $\mathbf{t_2}$
- Similarly **for weak dependencies**

DEDP: Task Post-Effects

- Only <u>some</u> **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, <u>the **changes** in the **Beliefs**</u> of an **Actor** on the other Actors

DEDP: Design Artifact

- 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**
- Describes the **Material Output** of an **Activity**, the **Activities** of a **Task**, of a **DEDP**

DEDP: Collaboration Mechanisms

- Based on the **Beliefs** on the other Actors and **Self-Beliefs**
- E.g., the others:
  - Are they **Capable** to provide this or that Design Solution (DS)?
  - Are they **Capable** to perform this or that Task?
  - Are they **Credible** partners?
- At least 2 types of **Contracting** Encounters resolved through **Negotiations**:
  - Contracting an Actor to **Perform the Task**
  - Striking the deal on **Design Artifact re-use** (as the DS)

Fellows' Capability Assessment

Adjusting the Beliefs by the Collaboration Experience

- When it is necessary to assess the capabilities of the others?
  - Actor **I** believes (<u>self</u>) that s/he is not capable to perform a^j
  - Actor **I** believes (<u>self</u>) that s/he does not have the available **DS**
- **Allows:** Actor **I** believes (<u>others</u>) that some of A_i are:
  - Capable to perform a^j
  - Capable to provide a **DS**
- Source: **Fellows' Capabilities Estimation Matrix C**
- **Capability Estimations** c_i are adjusted by the results of the previous negotiations on the activity/**DS** provision
  - **IF** ($p_i > threshold^j$)A_i is **capable** to perform a^j (provide the **DS**)

DEDP in PSI

CERTICON cadence

Fellows' Credibility Assessment

Adjusting the Beliefs by the Collaboration Experience

- A **Task Manager** expects the results to be delivered by the **Contractor** at the **agreed time**:
  - **E.g.**, air ticket at the gate counter in 30 min before the check-in
    - What if the ticket appears to arrive in 5 min before the check-in?
    - What if there is still no ticket and the plane is taking off?
- We may associate a kind of a **results desirability value** (des) with each of these outcomes
  - Indicating, e.g., the **part of the agreed incentive** for the service provision we are ready to pay

Fellows' Credibility Assessment

Adjusting the Beliefs by the Collaboration Experience

- The **Credibility Value** (Cr_i^j) of the certain **fellow** A_i wrt the certain **activity** a^j may be reduced according to the **lost desirability** in the case the contract is not fulfilled
- An **Actor** maintains its **Fellows' Credibility Matrix** likewise the **Capability Estimations**

$$\mathbf{C}r = \begin{array}{c} \\ \mathrm{A}_1 \\ \\ \ldots \\ \\ \mathrm{A}_n \end{array} \begin{array}{c} \begin{array}{ccccc} a^1 & \ldots & a^j & \ldots & a^m \end{array} \\ \begin{bmatrix} Cr_1^1 & & Cr_1^j & & Cr_1^k \\ & & \ldots & & \\ & \ldots & Cr_i^j & \ldots & \\ & & \ldots & & \\ Cr_n^1 & & Cr_n^j & & Cr_n^m \end{bmatrix} \end{array}$$

Negotiation on DS Re-use
The Protocol

- Extended Iterated CNP (FIPA)
- **Initial CfP** (I→):
  - **DS type only** (e.g., 'Input Signal Amplifier')
  - In case no proposals were received back from **P**s at this round **I** will proceed with the design from scratch
- **CfP**: **DS desirability** (I→):
  - **P**s: Desirability feedbacks
  - The Deal

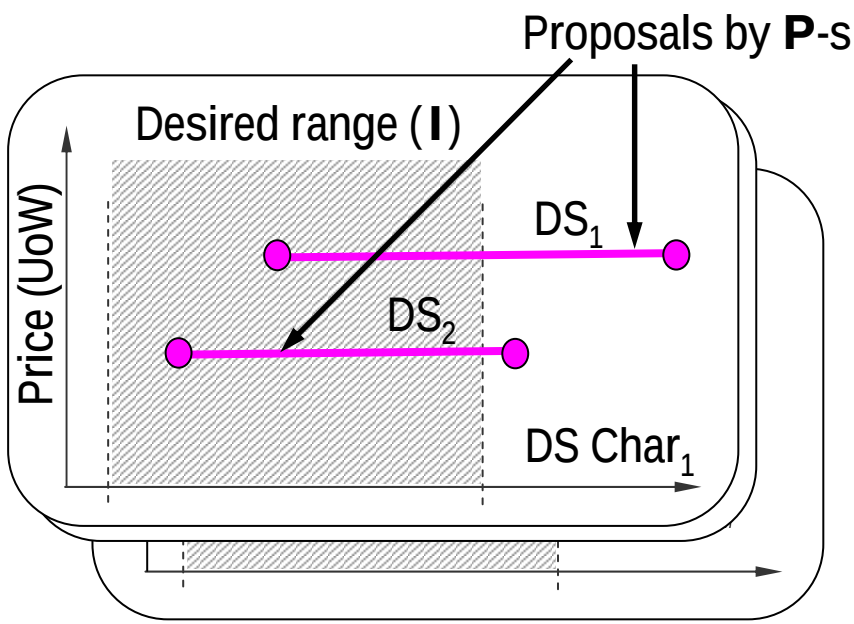
Negotiation on DS Re-use
The Protocol

- Negotiation on the **1-st round**: **availability of the DS**
- Negotiation on the **second** and the **subsequent rounds**: **terms of the possible contract**
  - **I** advertises:
    - The Space of the desired ranges of **DS** characteristics
    - Discrete **DS** desirability function (price over **DS** match ratio)
  - **P-**s reply with:
    - Their **DS** characteristics ranges
    - The price of the **DS**
  - **I** than computes the match ratio ***M*** for each received feedback

Negotiation on DS Re-Use

Desirability Function

- **Desirability CfP** (I→): **desired ranges** of DS characteristics, e.g.:
  - input voltage, V (2.5 – 5.4)
  - output frequency, MHz (1.80 – 1.85)
  - linear dimensions: length, mm (20-30), width, mm (10-15)
  - etc.
- Formal description: **Design Artifact Ontology**

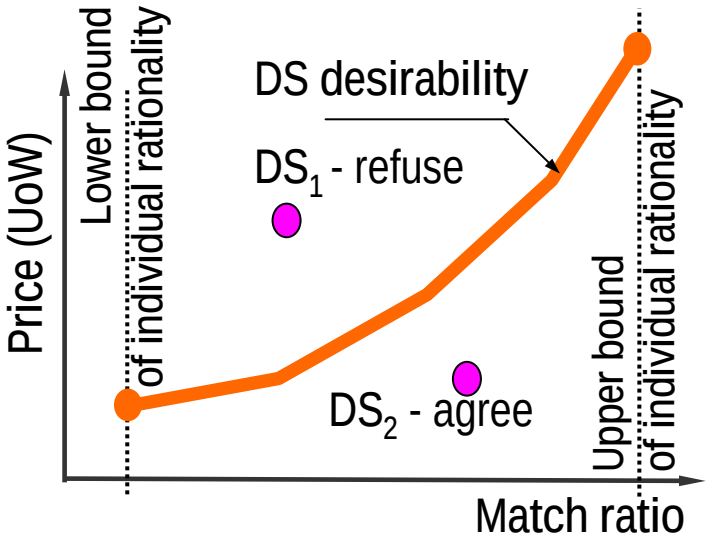


DS price (UoW) as the scalar function of DS characteristics (1-*n*)

DEDP in PSI

Negotiation on DS Re-Use
The Negotiation Strategy

- If 1 or more proposals by **P**-s indicate **agreement**:
  - **I** chooses the **best proposal** weighted by the respective **credibility value**
- If **no agreements**, Subsequent rounds are used to concede:
  - Attempt to find the agreement on the DS desirability – DS price in the case if no one of the **P**s has agreed on the previous round



DS desirability by **I** as the function of match ratio

DS proposals by **P**articipants

DEDP in PSI

Negotiation on DS Re-Use
The Negotiation Strategy

- **I** may continue to concede in the series of rounds if:
  - 1-st, the **P**s concede accordingly in a monotonic way
  - 2-nd, the concession still makes the possible deal individual rational for **I** with respect to the upper bound of the price value
- **I** considers the negotiation round as final if it can accept one of the **P**s' proposition and strike the contract deal
- The chosen **P** becomes the Contractor and commits to provide the DS to **I**
- **I** may declare the negotiation round as final by repeating the desirability function without concession
- Hence, if **P**s do not concede enough to make agreement in the last round, negotiation ends without reaching the agreement

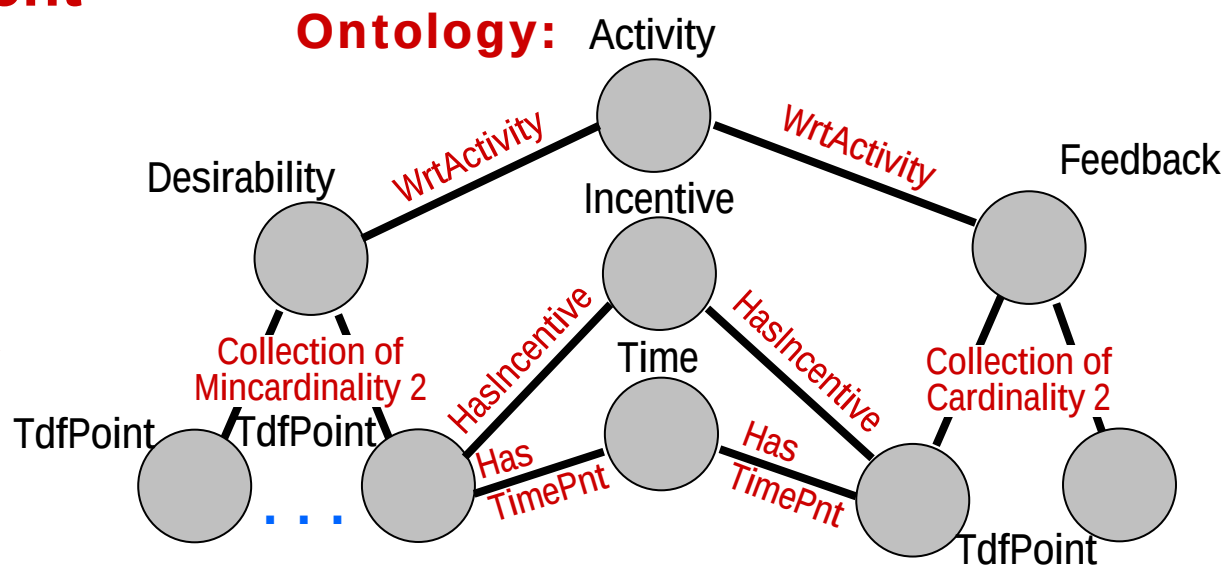
Contracting an Actor for the Sub-Task
The Protocol

- Extended **FIPA Iterated Contract Net protocol**:
- **Initiator** (**I**) – Task Requestor
- **Participants** (**P**) – Anticipated Task Performers (**Capable**)
  - **1-st round** – get initial proposals from **P**-s
  - **2-nd round** – **negotiate**: CfP – **task inputs + desirability**;
  - If several proposals result in **agreement** choose the best weighted by **Credibility**
  - **Subsequent rounds** – **adjust task inputs** in CfP if the proposals on the previous round **do not agree** with CfP
    - E.g.: the deadline, the incentive (**UoW**)

Contracting an Actor for the Sub-Task
Agreement and Disagreement

- **I**'s **Desirability** and the **Feedbacks** of **P**-s are specified in the terms of **Negotiation Ontology** as discrete functions of **Incentive** per **Time**
- The **Feedbacks** which lie above the **Desirability function** area indicate **Disagreement**
- Possible **Agreement** points belong to the intersection of the Desirability function area and the Feedback functions

Fragment of Negotiation Ontology:



DEDP in PSI

The Outlook

- Modeling and Simulation of DEDP-s in PSI
  - PSI Motivation – emerging collaboration in a DEDP is super-additive ☑
  - PSI project lines and partners: research, prototyping, industrial implementation ☑
  - The factors providing D in EDPs ☑
  - Productivity assessment and process run-time optimization (PSI DEDP modeling framework) ☑
  - PSI simulation modes (playback, predictive)
  - MASDK as the rapid prototyping toolkit

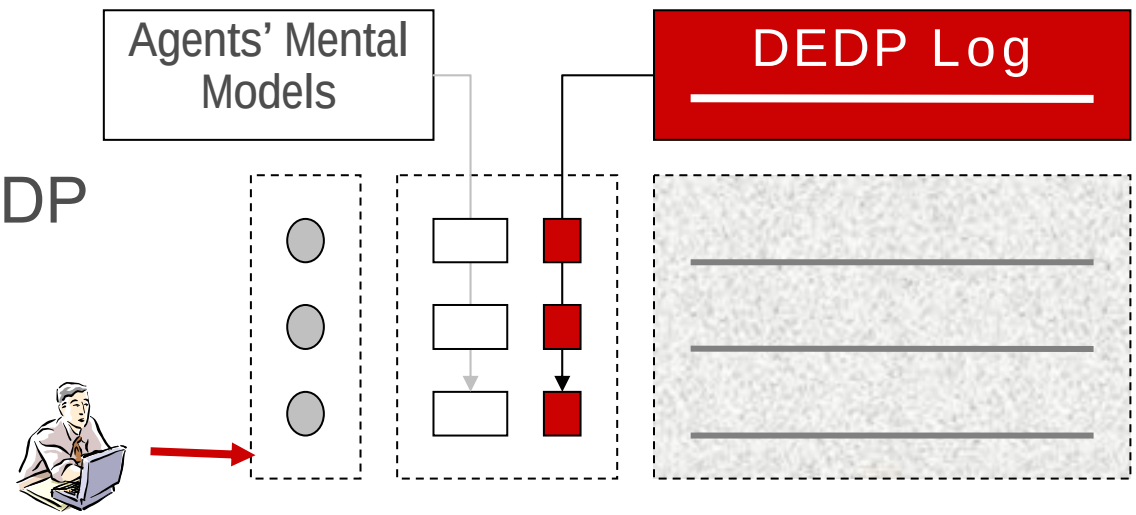
DEDP in PS

CERTICON cadence

DEDP Simulation Modes

The Boundary Cases

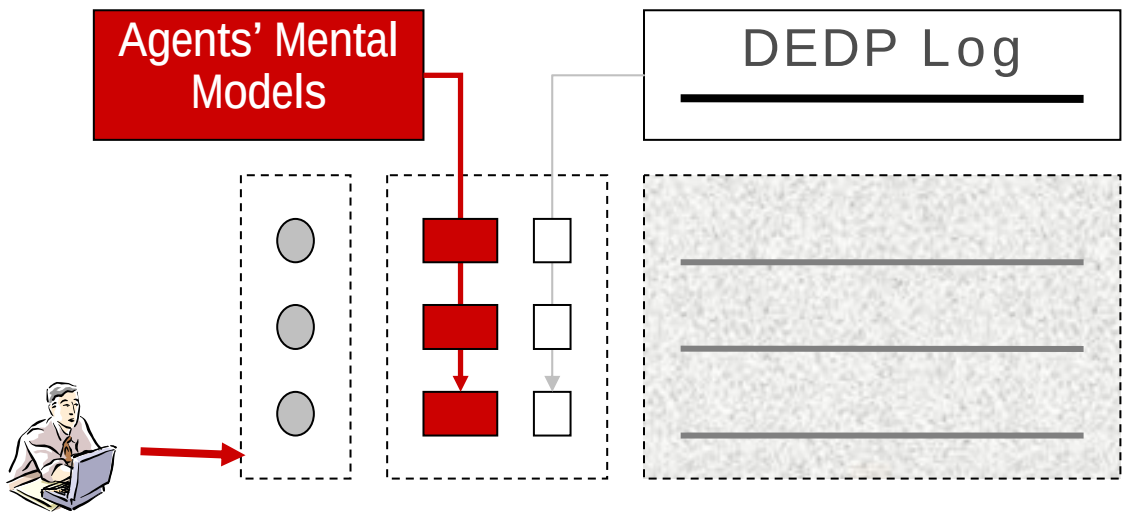
- **Playback**:
  - The process has been accomplished and recorded to PSI Test-bed
  - Simulations are playing back the record
  - Some Parameters are “screwed”, e.g., actors’ availability
  - The results are used to find out the sensitivity of the recorded DEDP to the screwed parameters



DEDP Simulation Modes

The Boundary Cases

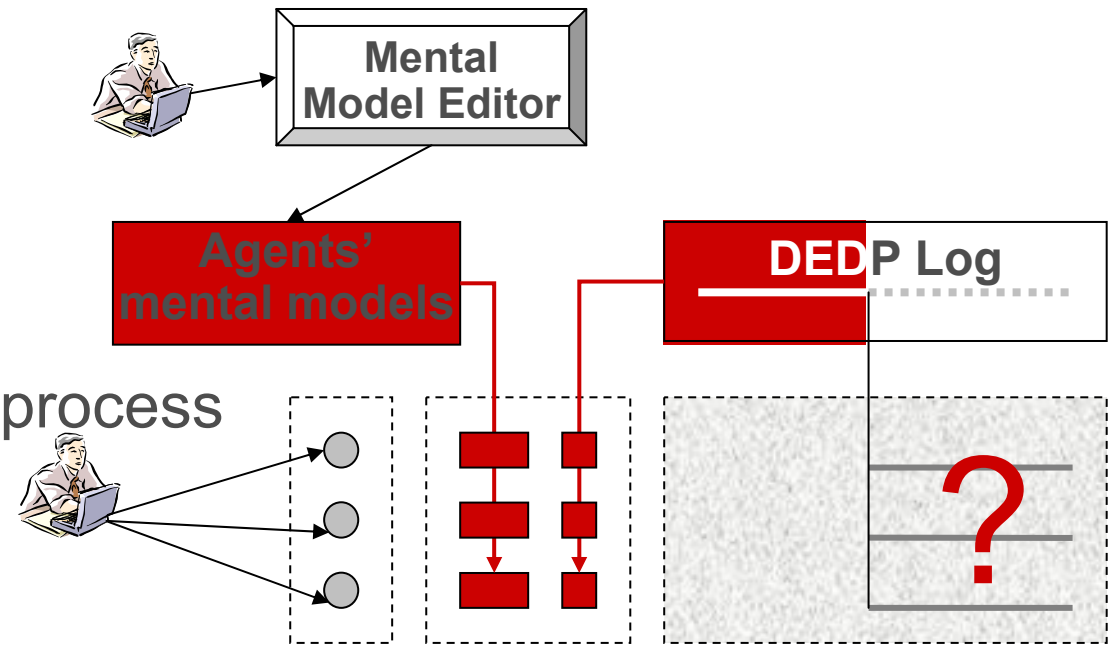
- **Predictive**:
  - The process has not started yet
  - Actors (agents) use their Mental Models and Collaboration Mechanisms to simulate the performance
  - Some Parameters are “screwed” by the user (as before)
  - The results are used for project planning
  - The Prototype for PPE



DEDP Simulation

Prediction-Correction Methodology

- **Playback – Predictive** mix:
  - Predict to the certain point in time
  - Perform the process and record
    - Playback
    - Compare the prediction and the playback
    - Screw parameters and find the optimal combination of the screw values
  - Provide corrective actions for the next process stage – if needed
  - Repeat until the process ends



Incremental Test-bed Development

Actors' Mental Models are Enhanced as well

- Recorded DEDPs are stored to the PSI Test-bed
  - Incrementally, process by process
- Mental Models of the Actors are stored to and updated at the PSI Test-bed
  - Designers are interviewed
  - Questionnaires are filled-in
  - Capability and Credibility Assessments are updated (simulation by simulation)
- Enhanced Mental Models are used for more precise simulations

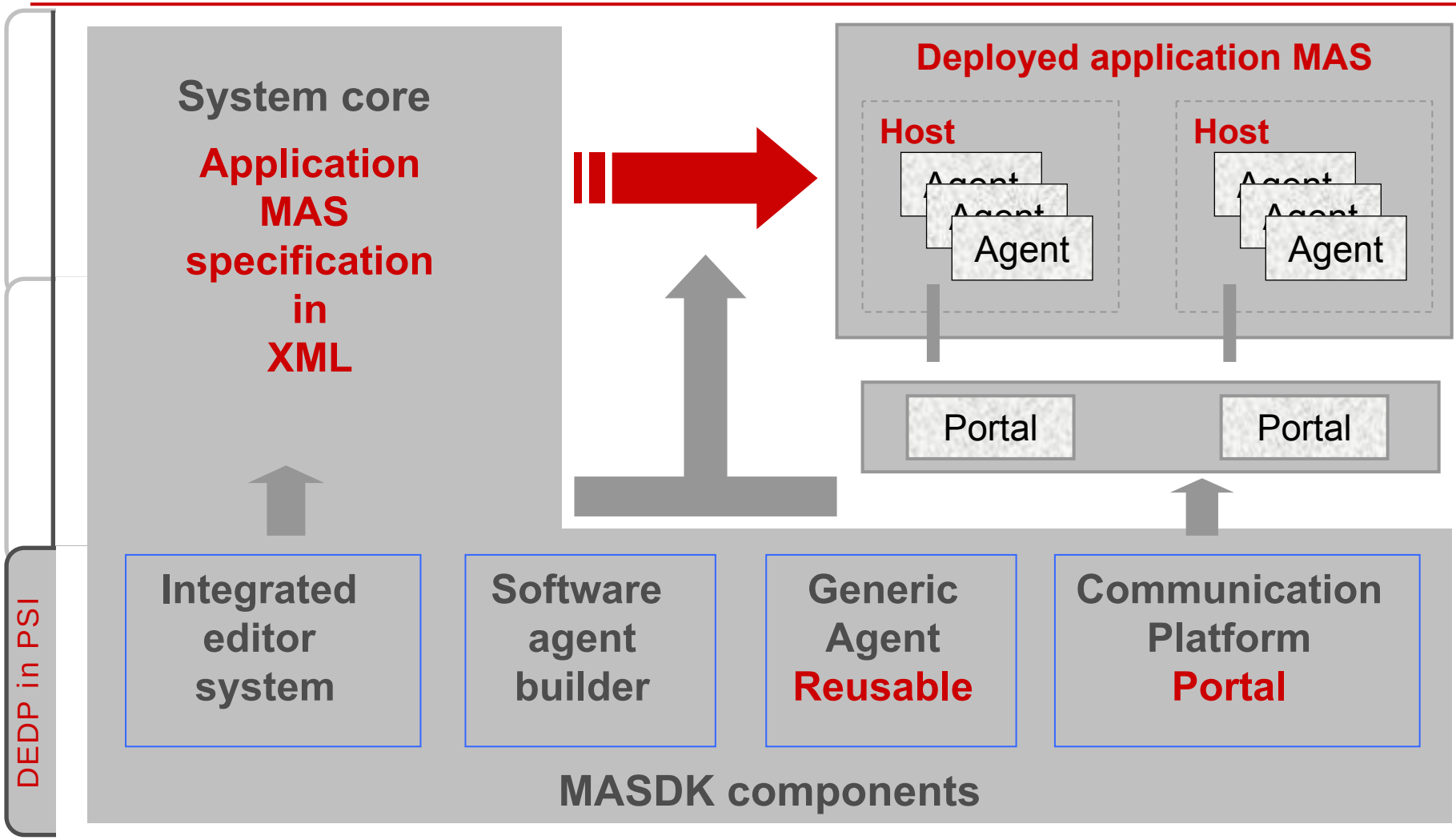
The Outlook

- Modeling and Simulation of DEDP-s in PSI
  - PSI Motivation – emerging collaboration in a DEDP is super-additive ☑
  - PSI project lines and partners: research, prototyping, industrial implementation ☑
  - The factors providing D in EDPs ☑
  - Productivity assessment and process run-time optimization (PSI DEDP modeling framework) ☑
  - PSI simulation modes (playback, predictive) ☑
  - MASDK as the rapid prototyping toolkit

DEDP in PSI

General description of MASDK

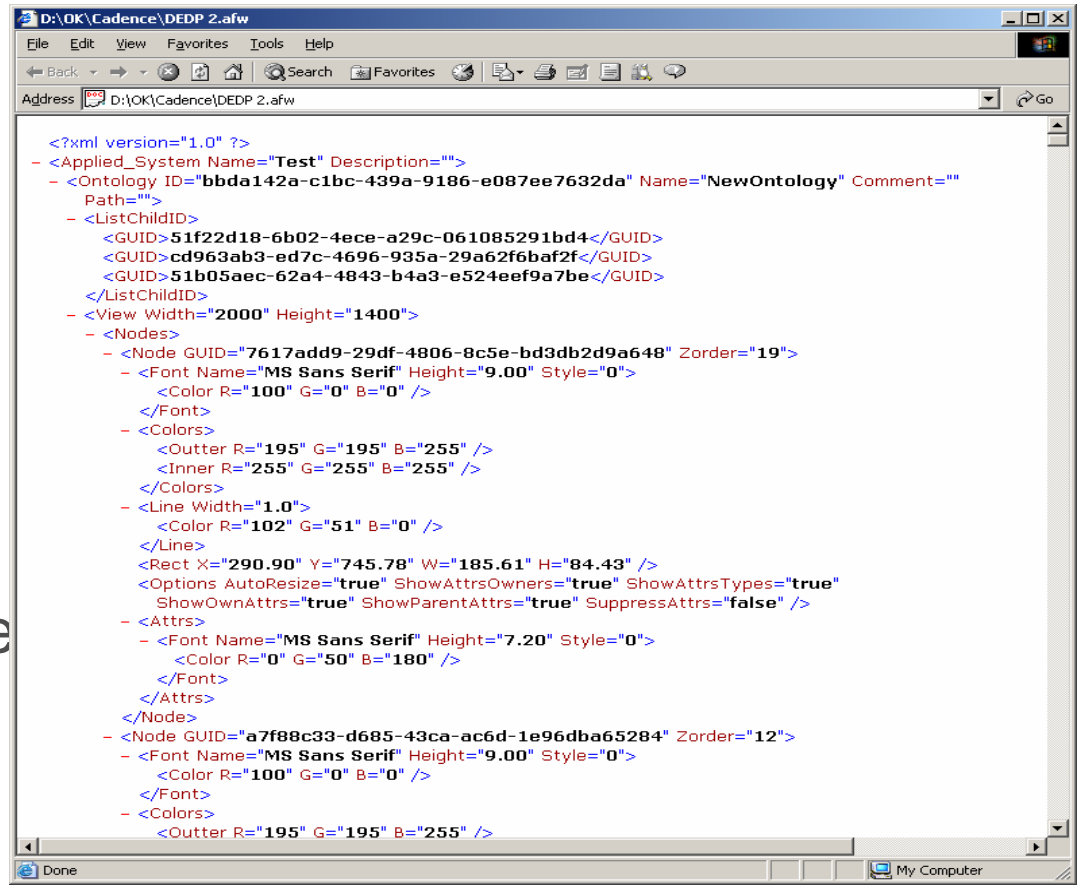
Components



MASDK Components:

System core

- Communication Role
- Protocol
- Message template
- (Shared) Ontology
- Agent class
- (Private) Ontology
- Behavior model
- Pro-active model
- User interface (ref to e
- State machine
- State
- Script

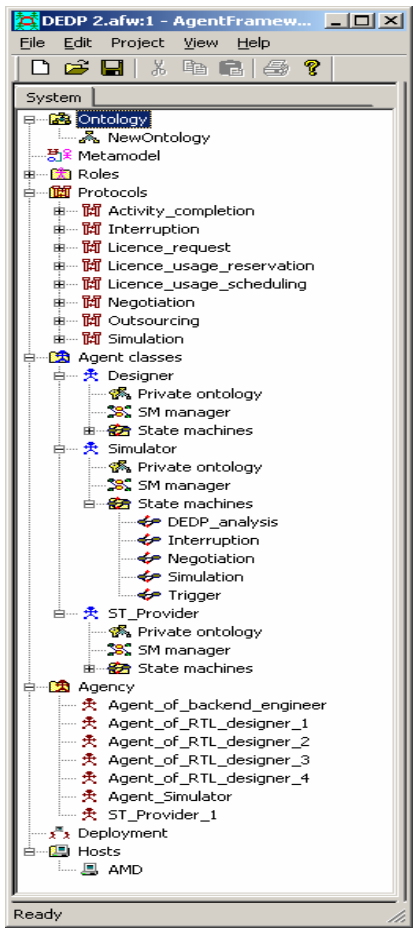


Application description is elaborated through **MASDK Editors**

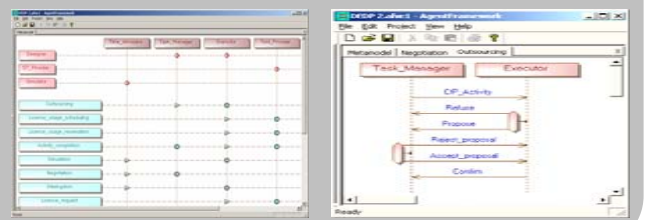
DEDP in PSI

MASDK Editors

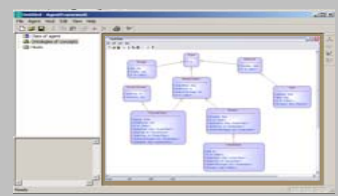
Browser



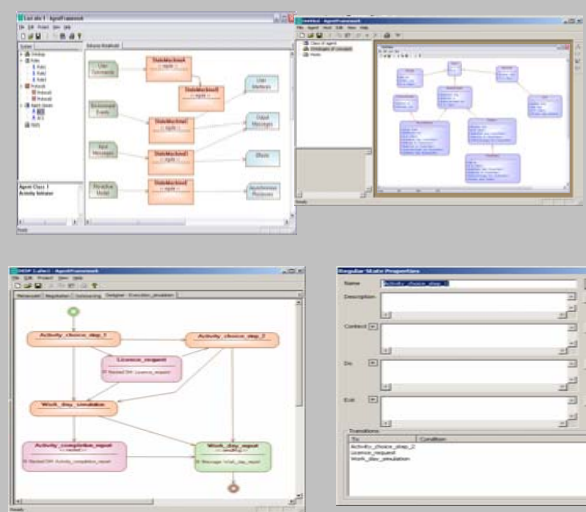
MAS organization



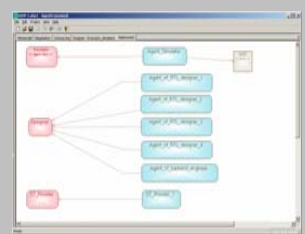
Ontology



Agent class



MAS configuration



DEDP in PSI

CERTICON cadence

MASDK Components

Generic agent

- **GA** – the generic component that includes implementation of the common meta–behavior rules of agent. The design of an application agent builds upon GA and is reduced to the specification of the problem-oriented aspects

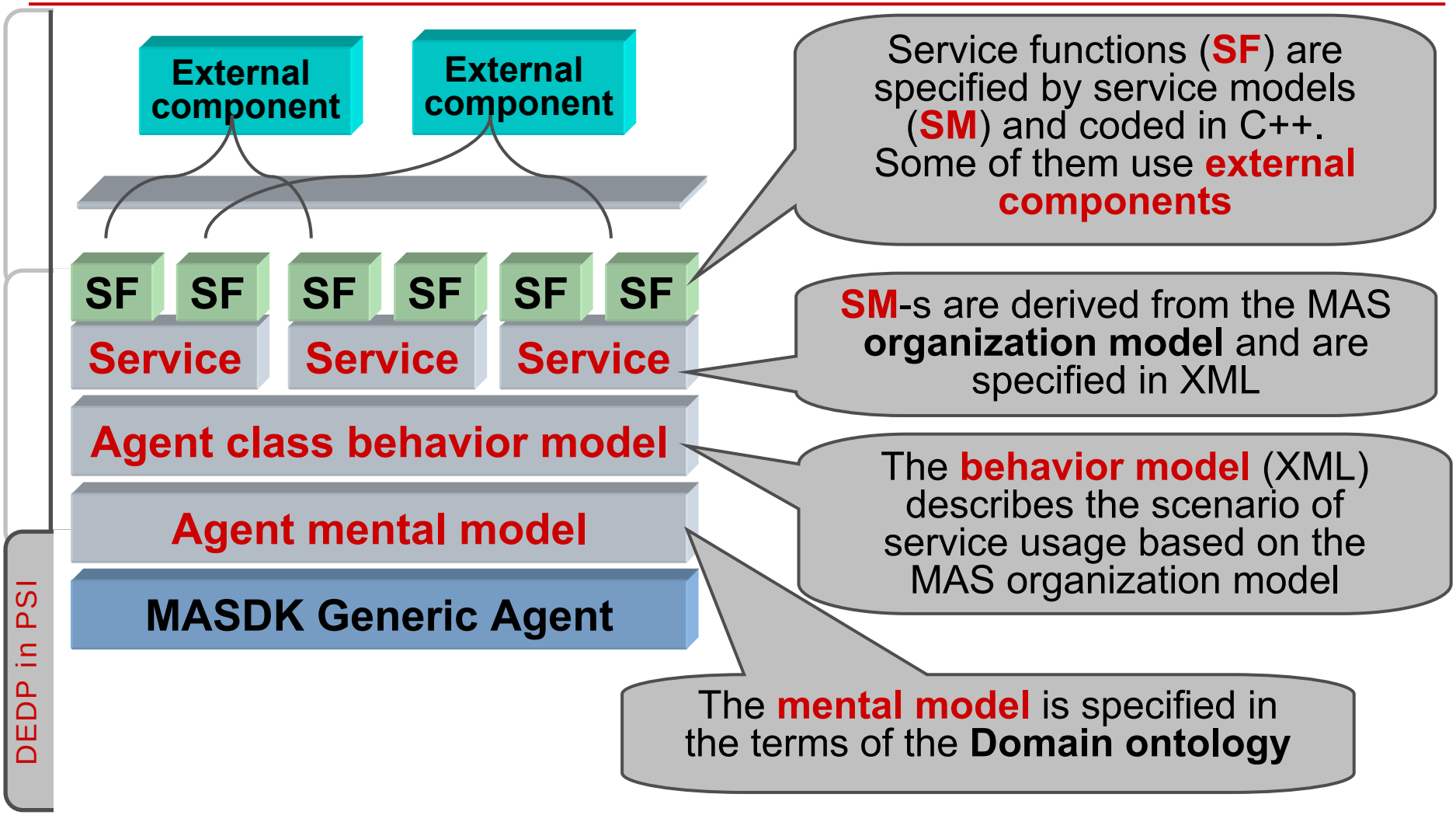
Software agent builder

- **SAB** is based on XSLT technology. It generates the agents' source code in C++ from XML-based specification (System Core) and compiles the agents' executable code using MS Visual C++ 7.0 environment

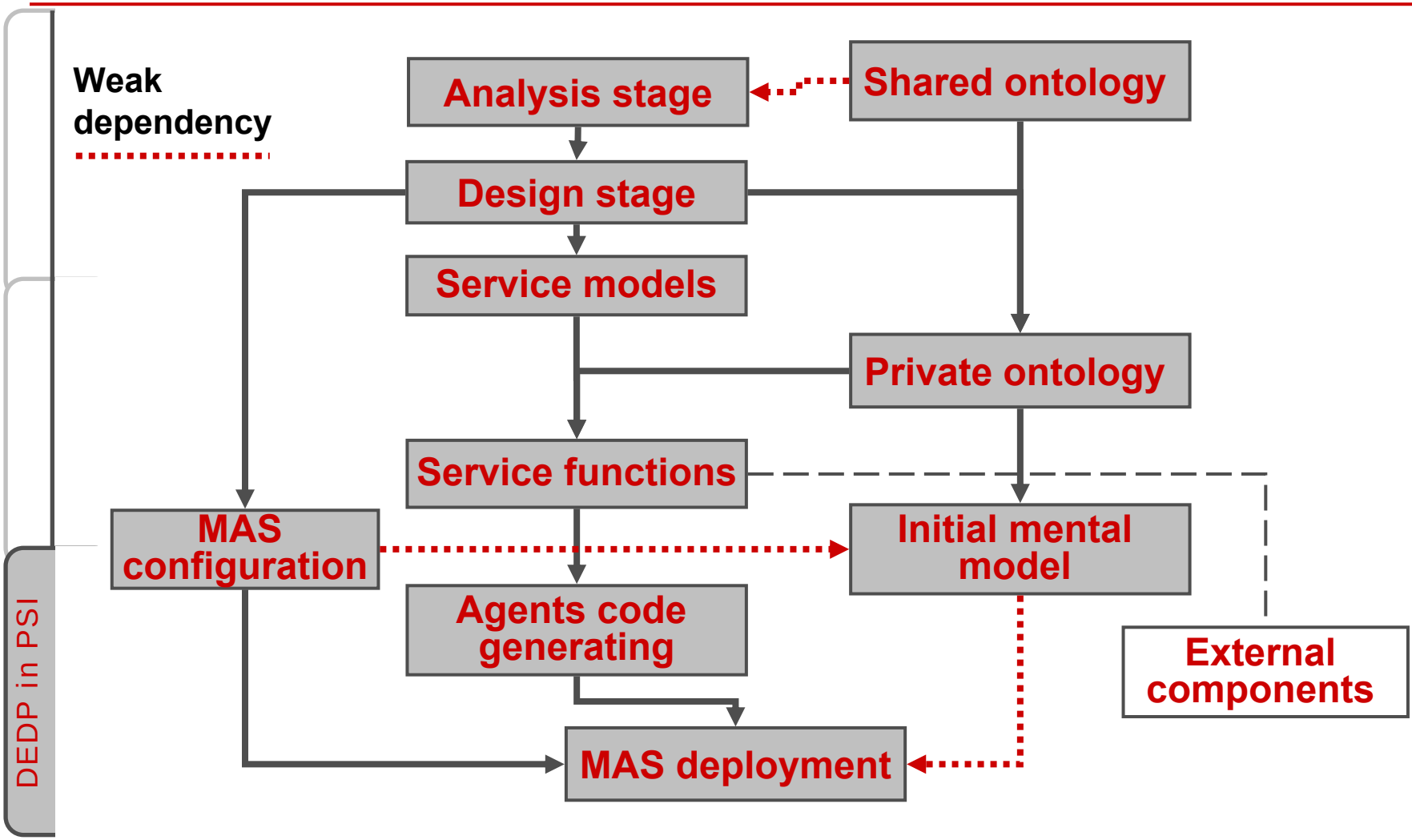
Communication platform

- **CP** is the distribution on the respective hosts of the **Portal** component. **CP** provides the transport level of agents' communication

MASDK: Application Agent Architecture



MASDK: the Methodology of MAS design



The Outlook

- Modeling and Simulation of DEDP-s in PSI
  - PSI Motivation – emerging collaboration in a DEDP is super-additive ☑
  - PSI project lines and partners: research, prototyping, industrial implementation ☑
  - The factors providing D in EDPs ☑
  - Productivity assessment and process run-time optimization (PSI DEDP modeling framework) ☑
  - PSI simulation modes (playback, predictive) ☑
  - MASDK as the rapid prototyping toolkit ☑

To Read More

- 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
- 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/psi-public/PSI-DEDP-MF-v10-Feb-2004.pdf

MORE INFO

The Outlook

- **Motivation**: ☑
  - Why to model Design?
  - Are there difficulties?
  - How to? Agents?
- Agent-based approaches and models in IPD 1993-2005+ ☑
- Modeling and Simulation of DEDP-s in PSI ☑
- Conclusions and Discussion

Conclusions …

- Hope that Modern Times in Engineering Design Automation have the flavor of the **Renaissance** …
- The Assets:
  - Agent metaphor is very likely an appropriate thing
  - The proper focus is on DEDP arrangement and management – Humans are “intellectual resources”
  - DEDP productivity may be assessed by using the Utilitarian approach
  - Simulations are the means to reach (sub-) optimal DEDP configurations – in dynamics, at run-time, by predictions-corrections
- Any indications of a Silver Bullet?
  - Our personal feeling after the 2,5 rounds of evaluation experiments in PSI
  - Back up from the audience?

ER2005

24-28 October 2005

cadence

That's all folks

CERTICON