Tutorial on

INTEGRATED WEB- AND AGENT-BASED COMPUTER SUPPORTED COLLABORATIVE DESIGN

Gosier, Guadeloupe, France - February 23, 2011
The Fourth International Conference on Advances in Computer-Humans Interaction

Bernadetta Kwintiana Ane

Institute of Computer-aided Product Development Systems
Universität Stuttgart, Germany

Outline

1. **Introduction: Today's CSCD application**
2. History: CSCW, Collaborative Engineering, and CSCD
3. Basic Concept of CSCD
4. CSCD Methodology
5. CSCD Tools
6. CSCD Architecture
7. Web and Agent Technologies for Collaborative Design
8. A Case Study:
   Tracking of Design Changes In Mechanical Product
9. Current Application

1. Introduction: Today’s CSCD application

Design in Automotive Industry

1886
Benz Motor Car

2010
Mercedes-Benz F-8000

2009
Mercedes-Benz Formula 1
with KERS hybrid energy recuperation system.

Free-flowing Ideas

“***Advanced Design*** *is a creative oasis — an island in the sea of production, where you can let your ideas flow freely without worrying about how to get them to the series production stage”.*

(Gorden Wagener, VP for Global Advanced Design at Daimler AG).

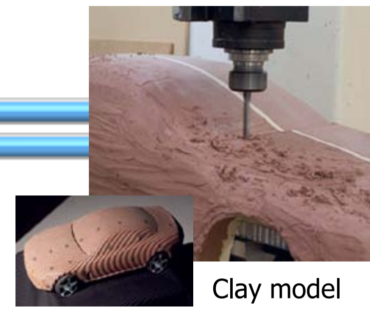
Image courtesy of Daimler AG, 2010.

1. Introduction: Today's CSCD application

Design creates connections

Sketch

CAD modelling



Clay model

Optimization

> *"Every good idea I have begins with a pencil and a blank piece of paper. I just sit down and let my imagination run free."*
>
> (Nicolar Garfias, Senior Designer Daimler AG)

Testing

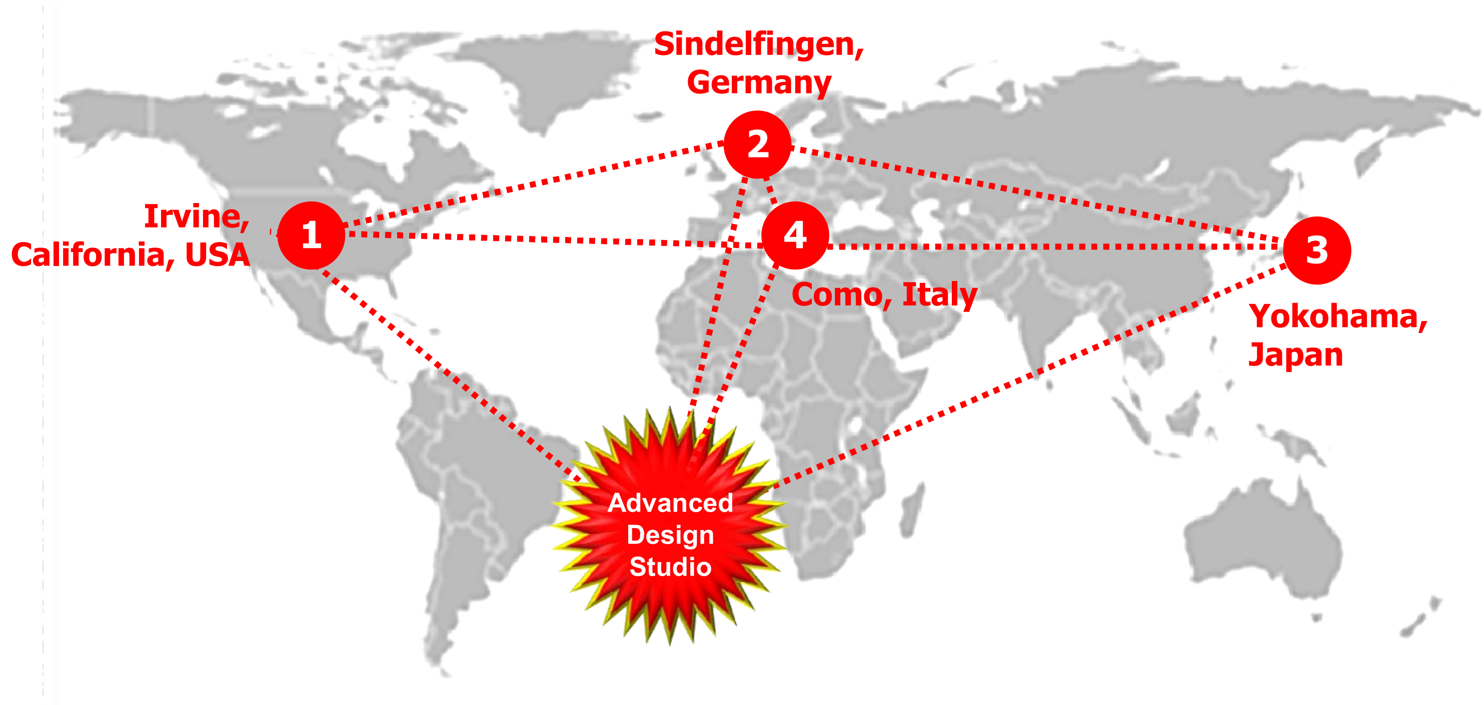
Production and assembling

Ramp-up

Images courtesy Daimler HighTech Report, 2008.

1. Introduction: Today's CSCD application

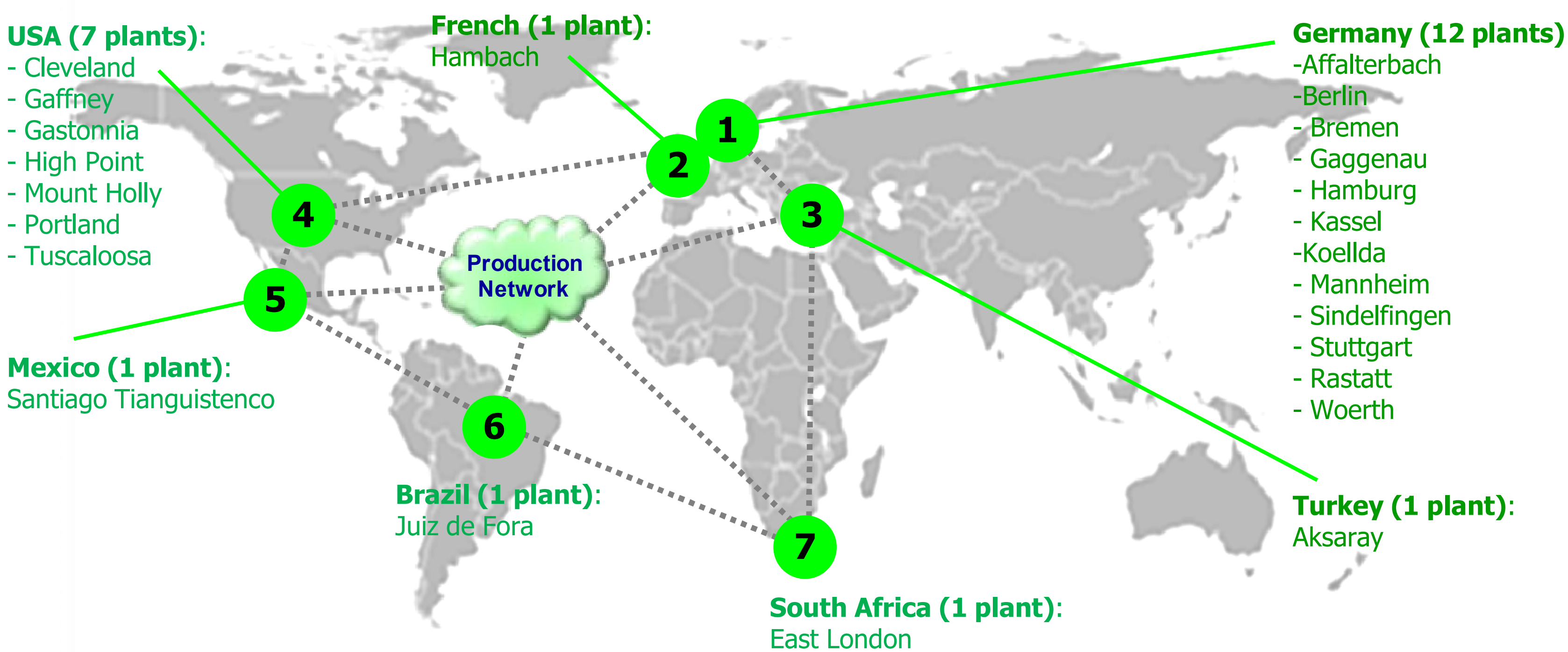
Collaborative Design in Daimler AG



Source: Daimler HighTech Report, 2008.

1. Introduction: Today's CSCD application

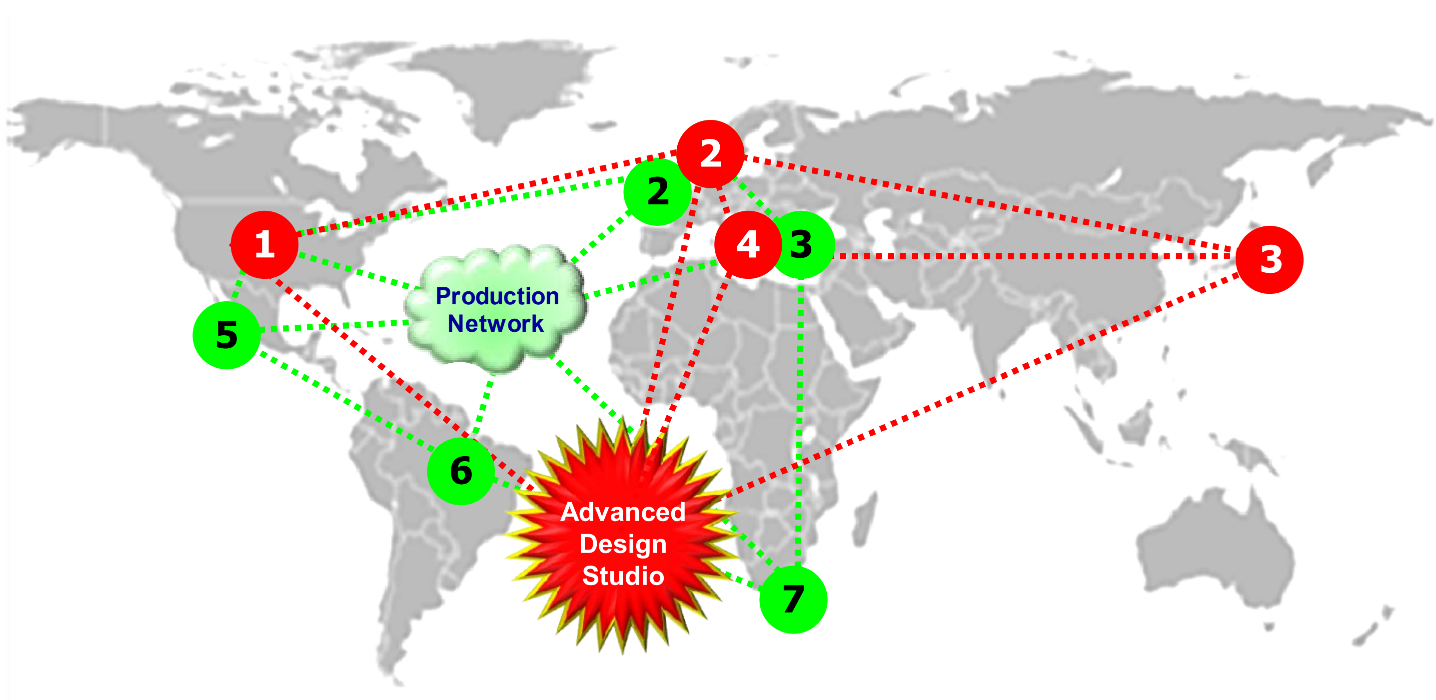
Daimler AG's Manufacturing Network



Source: Daimler HighTech Report, 2008.

1. Introduction: Today's CSCD application

Daimler AG's Integrated Design and Manufacturing Network

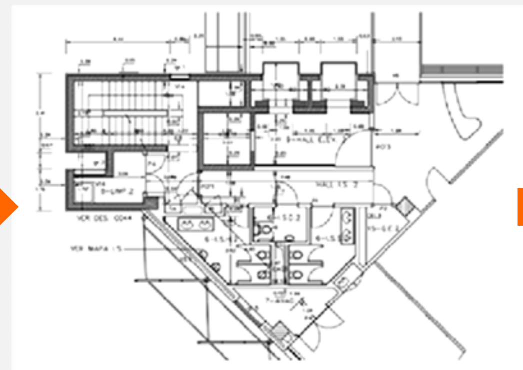


Source: Daimler HighTech Report, 2008.

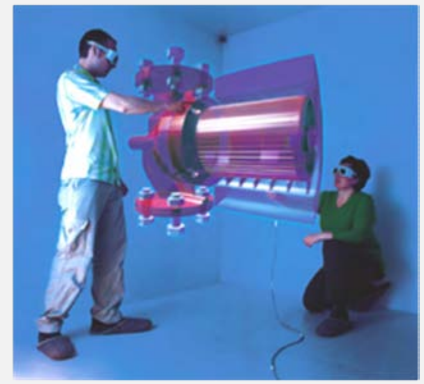
1. Introduction: Today's CSCD application

Design in Architectural Engineering

Sketch by hand (F. Jun, 2004).



CAAD drawing (by AutoCAD).



Collaborative design
in virtual environment
(VR Media Lab., Aalborg Univ., Denmark).

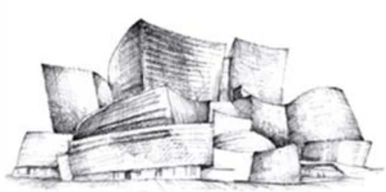
Creative process.	Generating details and dimensions.	Real-time coordination and visualization.

Source: M. A. Schnabel, University of Hongkong, 2004.

1. Introduction: Today's CSCD application

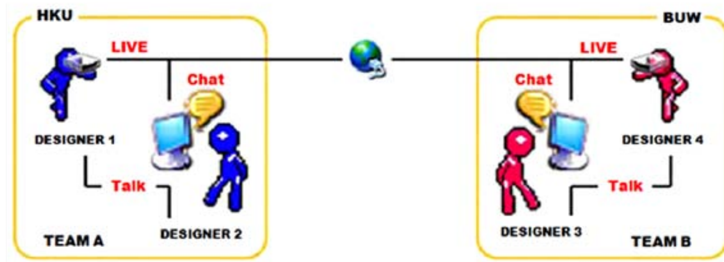
Design in Architectural Engineering

Guggenheim Museum, Spain
(by Hanson Wong)

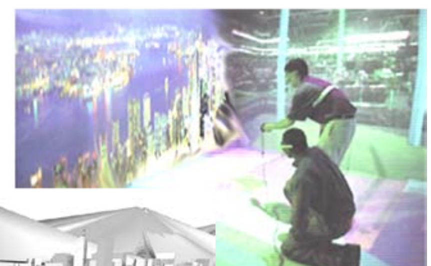


Pencil rendering

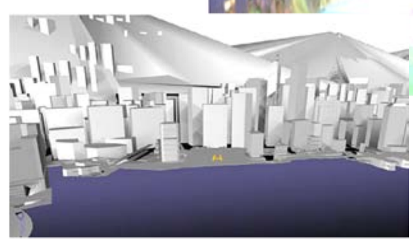
CAD rendering



Collaborative Architecture Engineering



Virtual environment



3D Screenshot model

Source: M. A. Schnabel, University of Hongkong, 2004.

1. Introduction: Today's CSCD application

Design has coined meanings

- as a product:

Courtesy of Zappos.com

Shoes

Courtesy of Treehuger.com

Exclusive chair

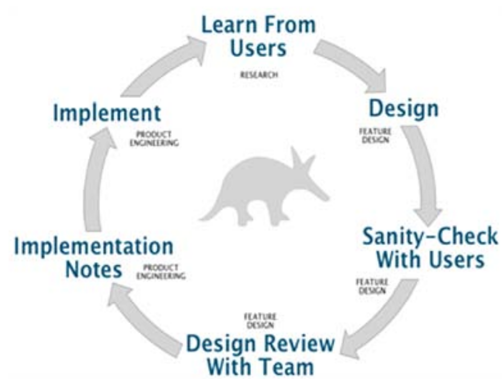
Courtesy of BusinessWeek

Taxi

Courtesy of IIDA, 2008.

Interior Design

- as a process:



Courtesy of blog.vark.com

CAD system can be either *"product structured"* or *"process-oriented"*.

1. Introduction: Today's CSCD application

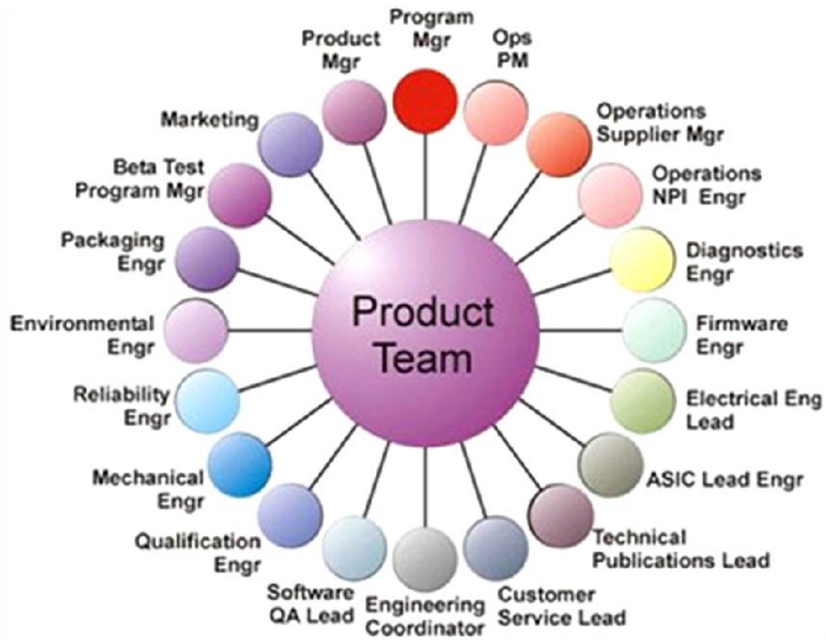
Design

- an activity of *modelling complex objects*,
- as a design project develops, the "parts-whole" relations are under *constant interpretations* by different team members from different design perspectives,
- triggers a fresh viewpoint to study *design communication and coordination* amongst multiple tasks and individuals.

One important aspect of collaborative design is ***"human communication"***.

1. Introduction: Today's CSCD application

Currently, a CAD system has been developed towards a new direction of ***"groupware"*** to support ***a wide range of group design activities***



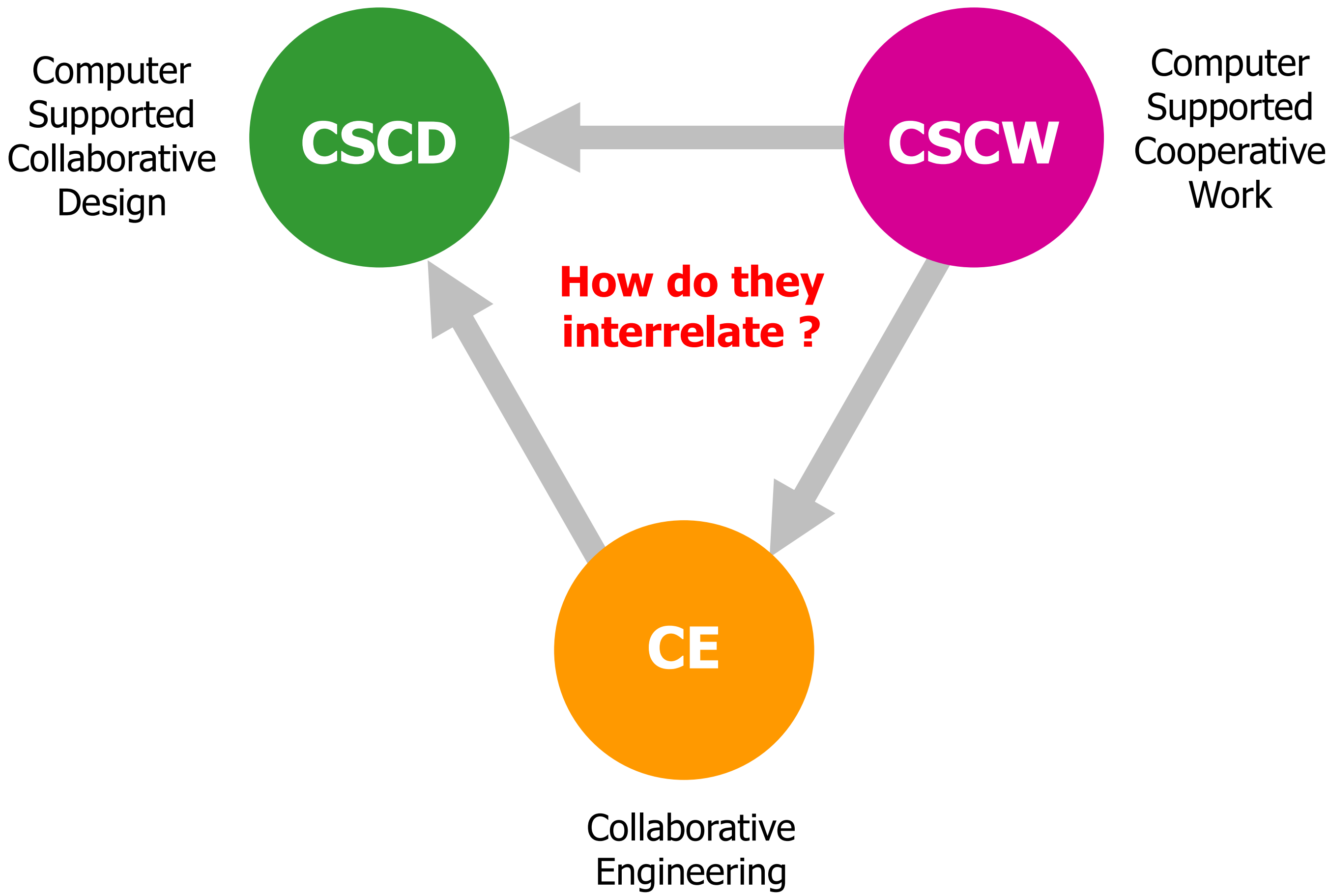
e.g., design information sharing, collaborative design decision making, virtual design conferencing, on-line product data access, etc.

Image courtesy of Fastcompany.com.

Outline

1. Introduction: Today's CSCD application
2. **History: CSCW, Collaborative Engineering, and CSCD**
3. Basic Concept of CSCD
4. CSCD Methodology
5. CSCD Tools
6. CSCD Architecture
7. Web and Agent Technologies for Collaborative Design
8. A Case Study:
   Tracking of Design Changes In Mechanical Product
9. Current Application

2. History: CSCW, Collaborative Engineering, and CSCD



2. History: CSCW, Collaborative Engineering, and CSCD

1984 - The term ***computer supported cooperative work (CSCW)*** was first coined by Irene Greif and Paul M. Cashman at a workshop attended by individuals interested in using technology to support people in their work.

1987 - Charles Findley presented the concept of ***Collaborative Learning-work***:

"Processes, methodologies and environments in which professionals engage in a common task, in which individuals depend on and are accountable to each other".

1987 - Louis S. Richman and Julianne Slovak introduced the concept of ***groupware*** that aims to place the computer squarely in the middle of communications among managers, technicians, and anyone else who interacts in groups, revolutionizing the way they work, to achieve their goals."

Source: J. Grudin, University of California, 1994.

2. History: CSCW, Collaborative Engineering, and CSCD

1991 - P. Wilson defined CSCW as a generic term, which combines the understanding of the way *people work in groups* with the enabling *technologies of computer networking*, and associated hardware, software, services and techniques.

1999 - P. H. Carstensen and K. Schmidt addressed *groupware as part of CSCW*:

"*CSCW*", *and thereby* "*groupware*" *addresses how collaborative activities and their coordination can be supported by means of computer systems.*

- While, many authors considered *CSCW and groupware* are *equivalent*.

Source: J. Grudin, University of California, 1994.

2. History: CSCW, Collaborative Engineering, and CSCD

CSCW is an identifiable research field focused on understanding characteristics of *interdependent group work* with the objective of designing adequate computer-based technology to support such *cooperative*.

Bringing together *social psychologists*, *sociologists*, *anthropologists* and *computer scientists*, among others.

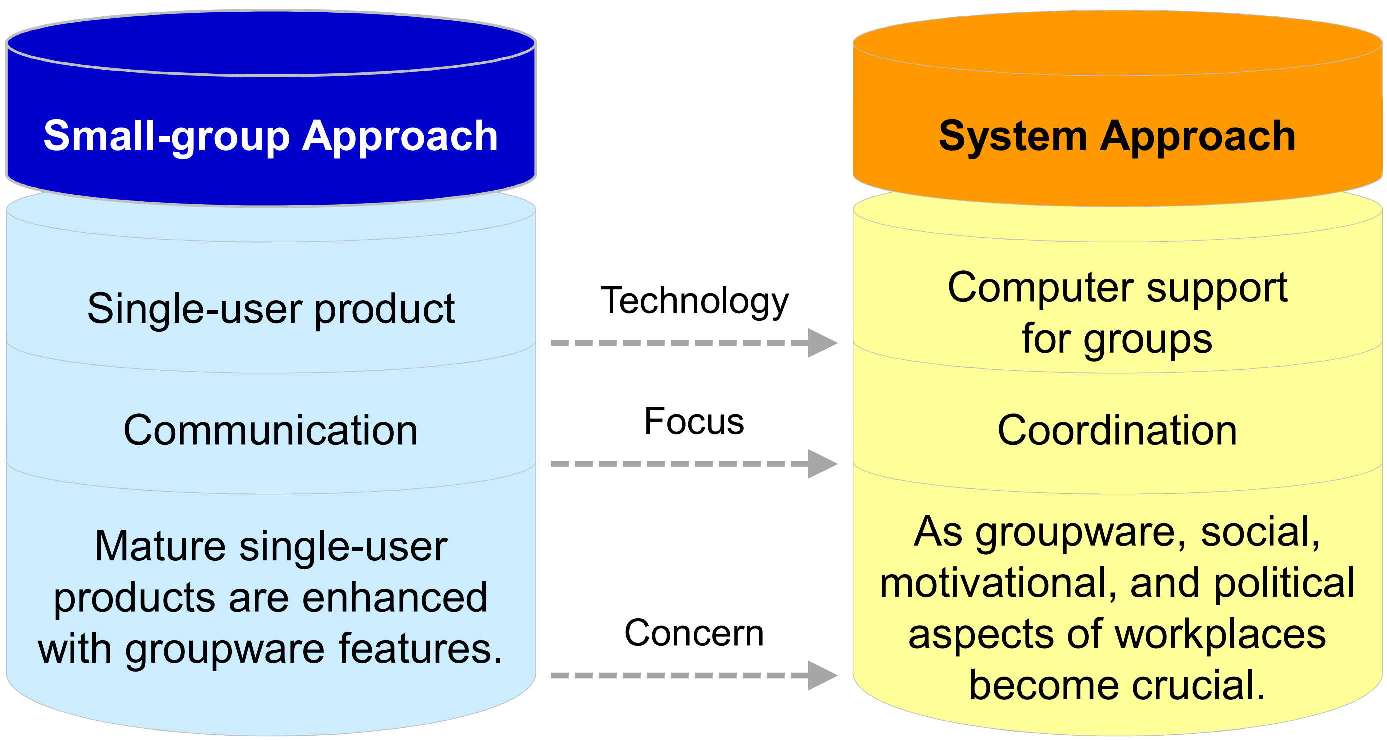
Core dimensions of cooperative work:

- *Awareness*
  gaining some level of shared knowledge about each other's activities.
- *Articulation work*
  partitioning work into units, and, after the work is performed, reintegrating it.
- *Appropriation (or tailorability)*
  how an individual or group adapts a technology to their own particular situation.

Source: J. Grudin, University of California, 1994.

2. History: CSCW, Collaborative Engineering, and CSCD

CSCW: Small-group Approach vs System Approach



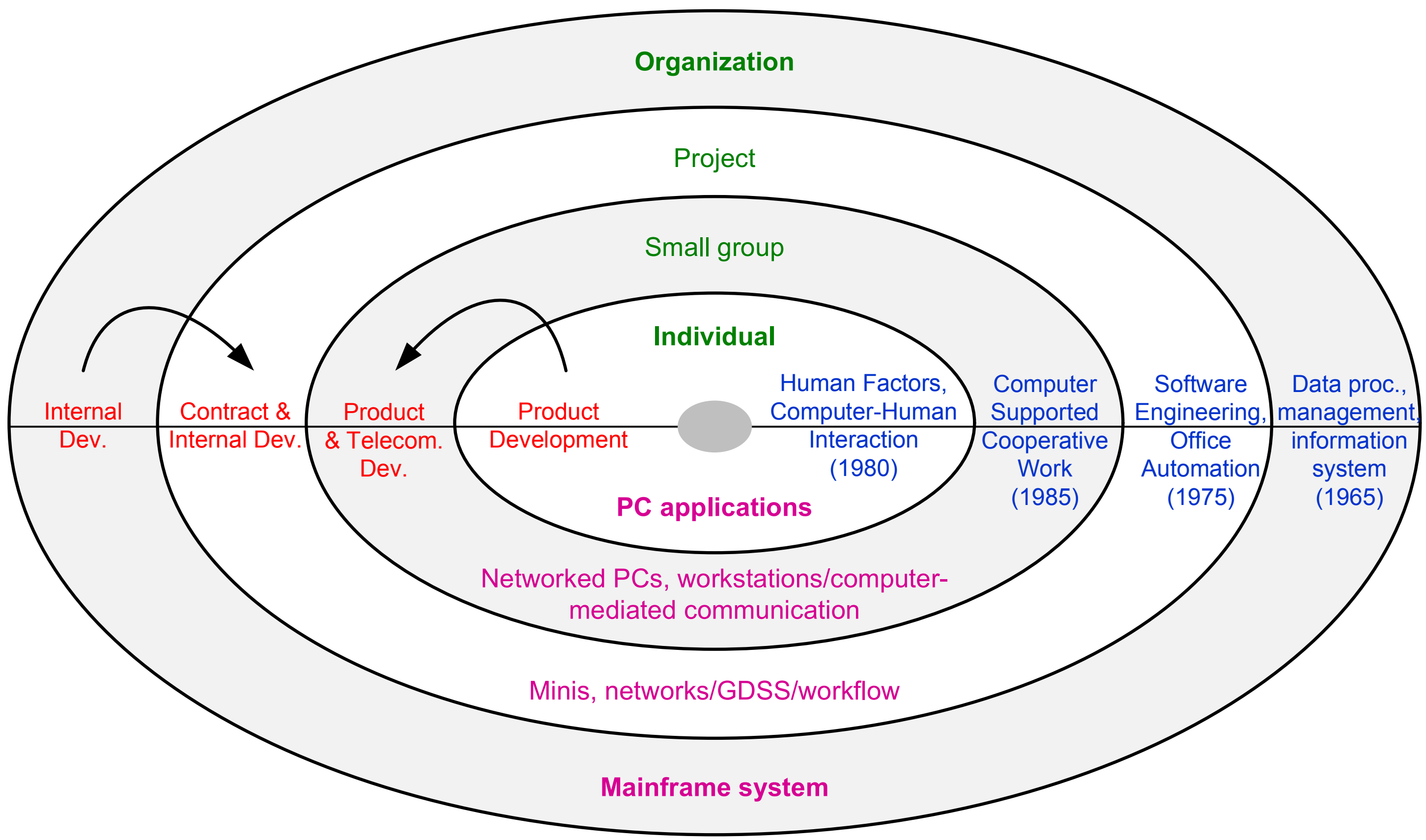
CSCW: US vs European Practices

	US Practice	Europe Practice
R&D activities	R&D between industries' research labs. and universities are more intertwined.	R&D are less intertwined.
Research funds	Mostly US government sponsored.	Supported by enormous European research sponsors; majority funded by ESPRIT, RACE, CO-TECH, and government sponsors.
Research focus	Large scale systems (i.e., experimental, observational, sociological) and technology development.	Formulation of system requirements (i.e., philosophy, social economy, politics) and implementation of platforms for organizational support.
Orientation	Empirically-oriented (e.g., cognitive task analysis)	Philosophically-oriented (e.g., task analysis of organization)

Source: J. Grudin, University of California, 1994.

2. History: CSCW, Collaborative Engineering, and CSCD

US research in CSCW



Source: J. Grudin, University of California, 1994.

2. History: CSCW, Collaborative Engineering, and CSCD

CSCW Matrix

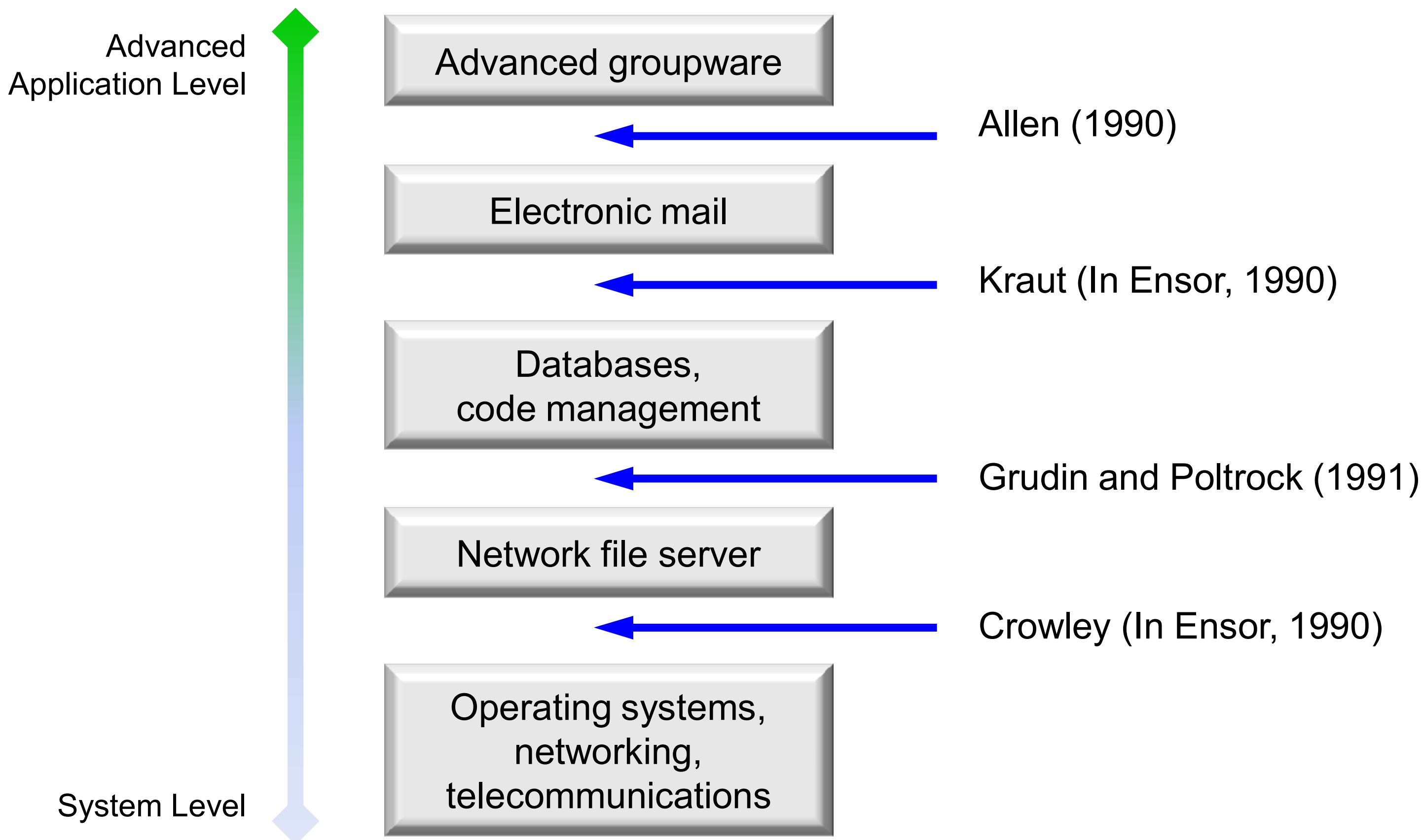
Time/Space Groupware Matrix

	Same Time synchronous	**Different Time** asynchronous
Same Place colocated	**Face-to-face interactions:** decision rooms, single display groupware, shared table, wall displays, roomware.	**Continuous task:** team rooms, large public display, shift work groupware, project management.
Different Place remote	**Remote interactions:** video conferencing, instance messaging, chats/MUDs/virtual worlds, shared screens, multi-user editors.	**Communication and Coordination:** Email, bulleting boards, blogs, wikis, asynchronous conferencing, group calenders, workflow, version control.

Source: R. Johansen, 1988.

2. History: CSCW, Collaborative Engineering, and CSCD

Demarkation of Groupware and Its Subtrate



Source: J. Grudin, University of California, 1994.

2. History: CSCW, Collaborative Engineering, and CSCD

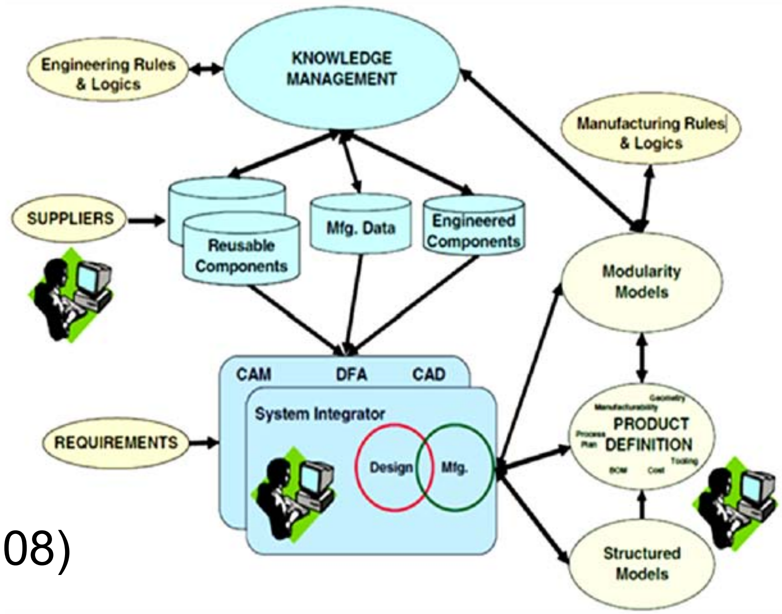
A 3x3 map of Groupware Options

Options		TIME		
		Same	**Different but predictable**	**Different and unpredictable**
PLACE	**Same**	Meeting facilitation	Work shift	Team rooms
	Different but predictable	Teleconferencing, videoconferencing, desktop conferencing	Electronic mail	Collaborative writing
	Different and unpredictable	Interactive multicast seminars	Computer boards	Workflow

Source: J. Grudin, University of California, 1994.

2. History: CSCW, Collaborative Engineering, and CSCD

Computer Supported Collaborative Design (CSCD) is considered by many researchers as an application of CSCW in design.



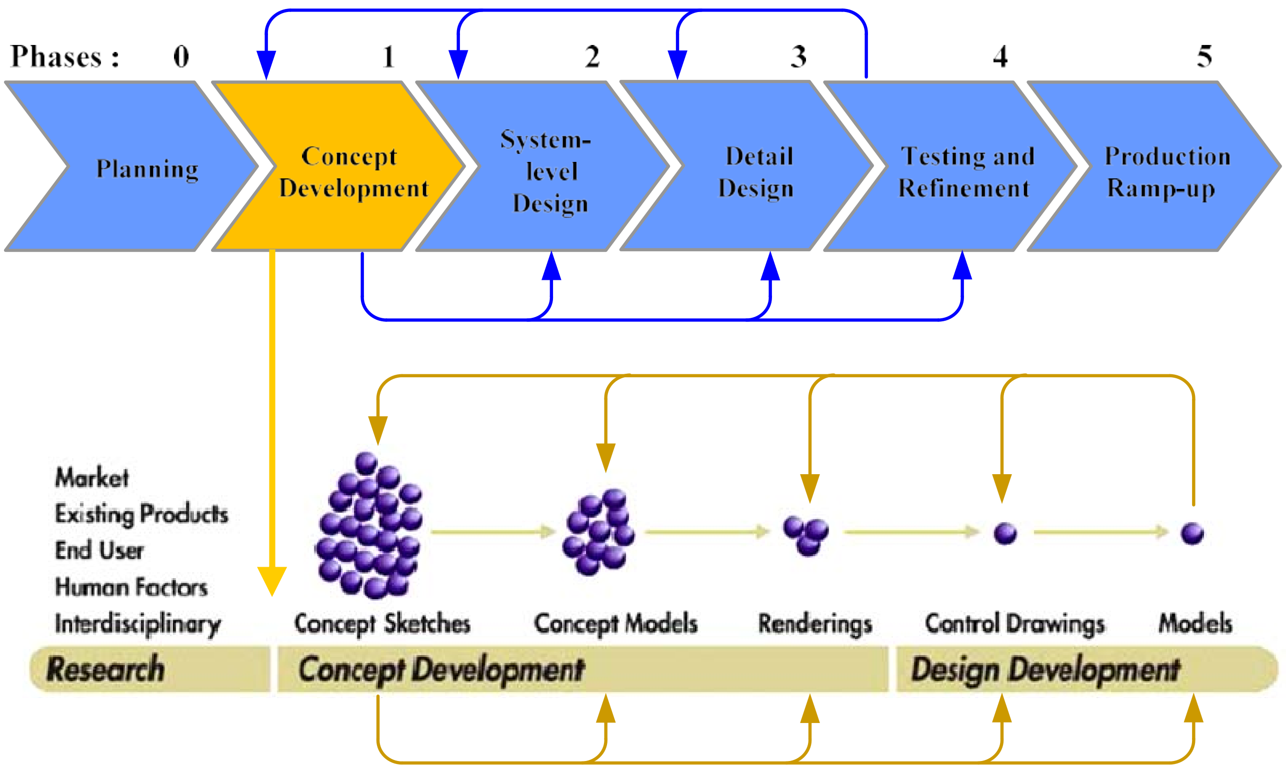
Huang et al. (2001) and Shen et al. (2008)

- it is ***"not just CSCW in design"***,
- CSCD is an application of ***"Collaborative Engineering to product design"***.

Source: A. K. Kamrani and E. A. Nasr, 2008.

2. History: CSCW, Collaborative Engineering, and CSCD

Traditional product design systems use a sequential mode of design generation, which breaks a design task into a number of sub-tasks that can be sequentially executed in a predefined workflow.



Source: K. Ulrich and D. Eppinger; B. K. Ane, 2005.

2. History: CSCW, Collaborative Engineering, and CSCD

Collaborative Engineering (CE)

- *Shen, Hao and Li (2008)*

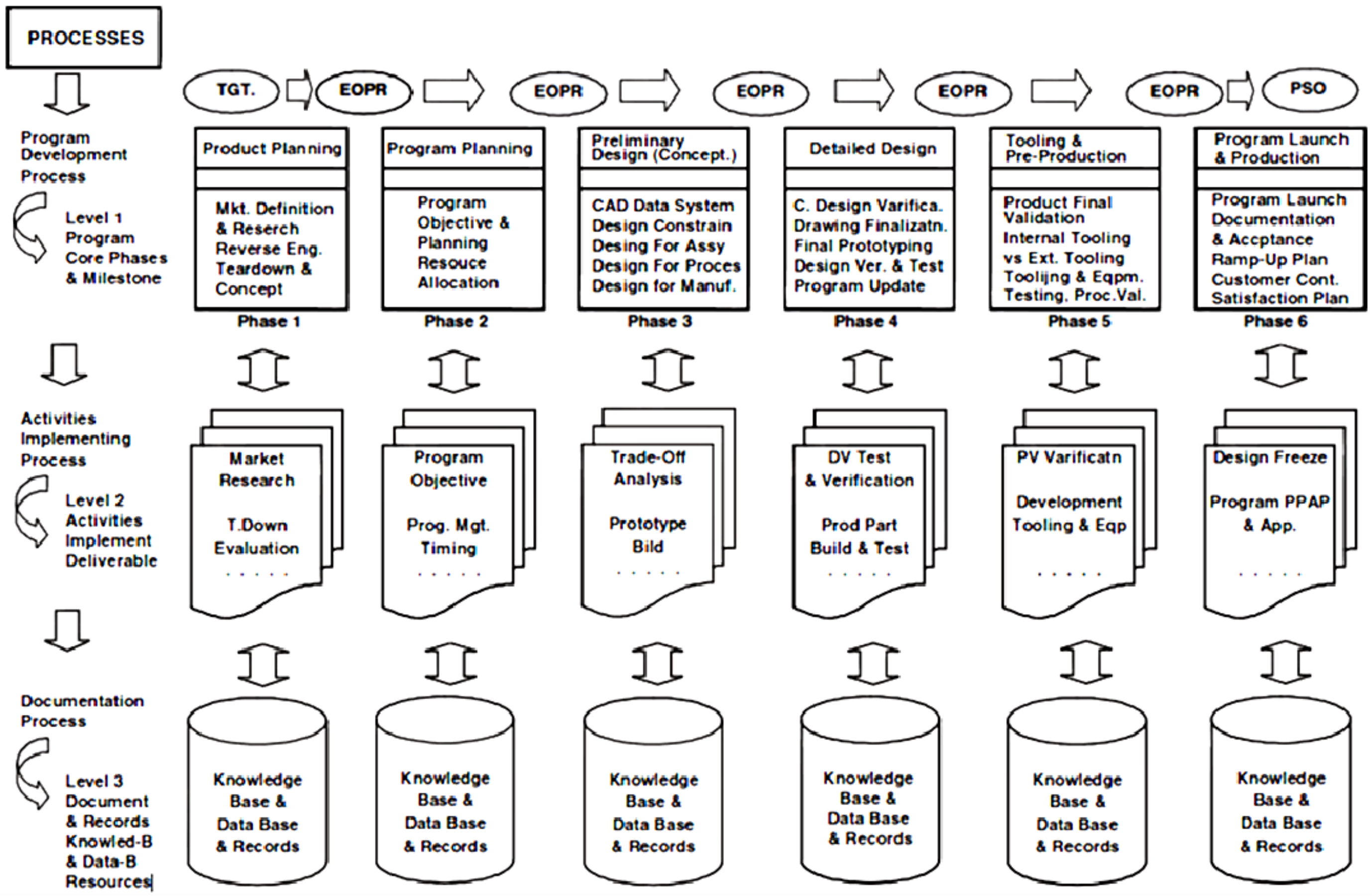
  a concept of *optimizing engineering processes* with objectives for better product quality, shorter lead-time, more competitive cost and higher customer satisfaction.

- *Kamrani and Nasr (2008)*

  a *systematic approach* to the integrated, concurrent design of products and related processes, including manufacturing, product service, and support.

2. History: CSCW, Collaborative Engineering, and CSCD

System Engineering Design Paradigm



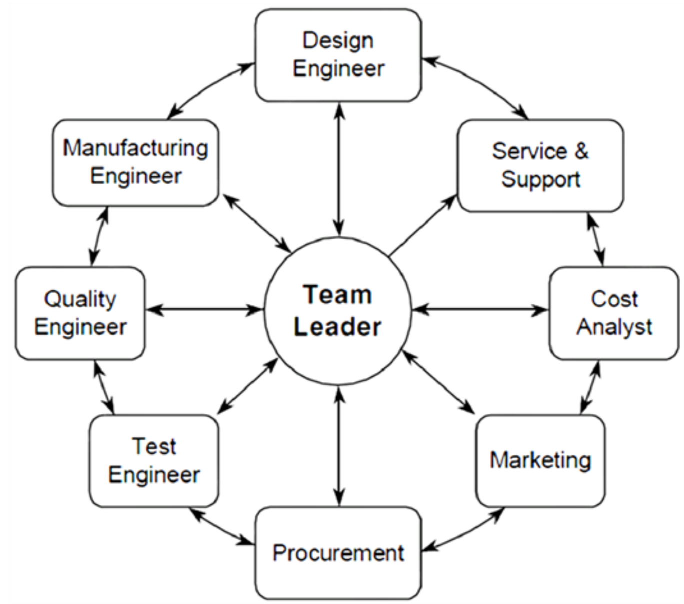
Source: A. K. Kamrani and E. A. Nasr, 2008.

2. History: CSCW, Collaborative Engineering, and CSCD

***Collaborative Engineering* (*CE*)** approach is intended to cause developers to consider all elements of the product life cycle from *conception through disposal*, including quality, cost, schedule, and user requirements.

The objective of ***CE*** is to *reduce the development cycle time* through a better integration of resources, activities, and processes.

As ICT advanced, CE application to product design, so-called ***Computer Supported Collaborative Design* (*CSCD*)**, becomes more promising.



Integrated and Collaborative Team Environment

Source: A. K. Kamrani and E. A. Nasr, 2008.

Outline

1. Introduction: Today's CSCD application
2. History: CSCW, Collaborative Engineering, and CSCD
3. **Basic Concept of CSCD**
4. CSCD Methodology
5. CSCD Tools
6. CSCD Architecture
7. Web and Agent Technologies for Collaborative Design
8. A Case Study:
   Tracking of Design Changes In Mechanical Product
9. Current Application

3. Basic Concept of CSCD

Computer Supported Collaborative Design (CSCD)

Definition (Sprow, 1992)

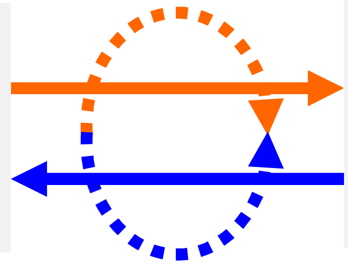
- also called *Cooperative Design*, *Concurrent Design*, or *Interdisciplinary Design*.
- the process of *designing a product through collaboration* among multi-disciplinary product developers associated with the entire product lifecycle

  (i.e., preliminary design, detailed design, manufacturing, assembly, testing, quality control, and product service, as well as those from suppliers and customers).

3. Basic Concept of CSCD

Basic Issues in Collaborative Design

Multiple individuals with different perspectives and design specialisms acting on heterogeneous worlds of design objects.



Development of integrated design schemes that support and satisfy all participating views and domain requirements.

Interplay between ***"integration"*** and ***"distribution"***

- Design is better characterized by spontaneous occurrences of *putting parts into wholes*, as well as *breaking existing wholes down* into smaller parts.
- There is a reciprocal effect of how *integration* is interrelated with *distribution.* arising from constant reinterpretation or negotiation of "parts-and-whole" relations.
- *Design integration* may lead to new runs of *design distributions*, and vice versa.

3. Basic Concept of CSCD

Fundamental Problems

Accomplishing a design task and delivering the results to manufacturing requires huge and complex information.

- Inner-side : Obstacle in products design

  A potential problem remains that mainly lies on the *geometric shape of products*; exists inherent in mass-customization.

- Outer-side: Obstacle in design system

  Most CAD/CAM/CAE technologies govern independent authoring tools that creates *incompatibility* and has significant impact to the product costs.

3. Basic Concept of CSCD

Coping With Fundamental Problems

- Design as an activity of modelling complex objects has attracted more interests to study "*design communication and coordination*".

- To meet the requirements of direct communication, the provision of *real-time computing supports* for *group interaction in design* amongst geographically distributed participants, is of primary concern.

3. Basic Concept of CSCD

The Objective of CSCD

- Supporting users to *work cooperatively in design* and, at the same time, harnessing the technology of computing communications.

- Creating effective communication and coordination to prevent the *insufficient* or even *absent of manufacturability checks* concurrently by detecting and considering design constraints and conflicts at earlier design stages.

3. Basic Concept of CSCD

Two Major Resources For Research Into CSCD

- ***Collaborative Computing***

  rapid combination of various computing and communication technologies.

  i.e., networking, communications, concurrent processing, and windowing environments (Grudin, 1991).

- ***Design Computing***

  diverse approaches to the making of design tools and environments in CAD.

  i.e., media of real-time graphical conversation, tools to manage design ideas by participants, media of performing “team-room” activities, etc.

3. Basic Concept of CSCD

Discovering Design Metaphors

Scenario 1: Bottom-Up Information Flow	Scenario 2: Top-Down Information Flow
The Theory of Situation (J. Barwise and J. Perry on formal semantic of natural language; 1983, 1989) "There are certain framework of information flow and constraint on information flow".	
▪ The situation types are conceptual construct that associated to an internal structure of design communication.	▪ The situation types are to follow the structure of products.
▪ Online access to digital visual references are useful for communicating common images or design metaphors.	▪ There are properties of flexible generic structures with which group dynamics can be associated.

3. Basic Concept of CSCD

What is design metaphors ?

A metaphor is "a device for seeing something *in terms of* something else. It brings out the *thisness* of that or the *thatness* of this."

-- Kenneth Burke, 1945

Source: D. Saffer, The Role of Metaphor in Interaction Design, 2005.

3. Basic Concept of CSCD

Theory of Design Metaphor

Hung-Hsiang Wang and Wan-Ju Liao (2009)

What is metaphor ?

In Georg Lakoff and Mark Johnson (1980) book entitled "Metaphors We Live By", *the essence of metaphor is understanding and experiencing on kind of another*. Consider the sentence "Architecture is a living machine". As "architecture" is to be understood in terms of "living machine", it is a metaphor. Most theories of metaphor focus on the ***verbal aspects***, and accordingly on *thinking process in language*, *regardless visual or pictorial aspects*.

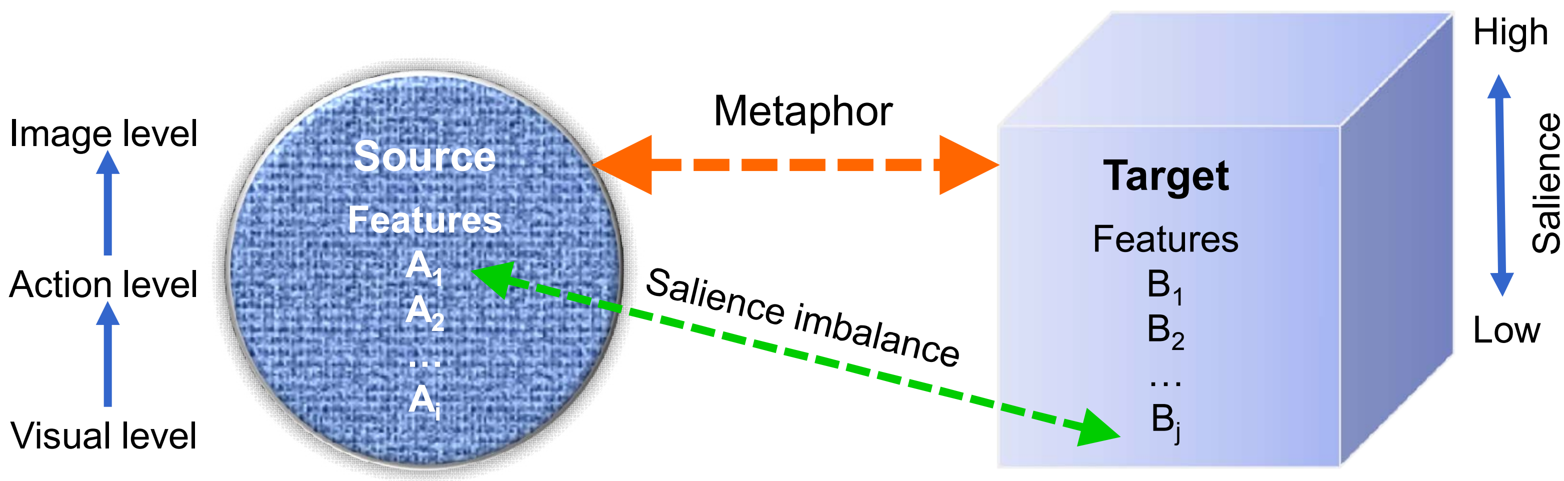
Why design metaphor ?

It is obviously that metaphor can be found in non-verbal objects such as pictures, photography, videos, films, as well as products. Noel Caroll (1994) called this kind of metaphor as *"visual metaphor"* in the book "Aspects of Metaphor", edited by Jaakko Hintikka, while Charles Forceville (1996) named it *"pictorial metaphor"*. We hereby coin the terminology *design metaphor to refer the metaphor exclusively generated by or interpreted from products in industrial design discipline*. For instance, from the appearance of Senseo coffee machine produced by Philips, a design metaphor that *"the coffee machine is servant"* is created and thus experienced.

Image courtesy of Ideastorming.tw.

3. Basic Concept of CSCD

Salience imbalance model with various levels of metaphors

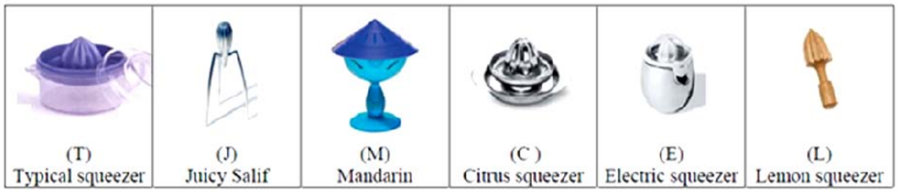


Metaphor is not just about language; *it's really about thought*. We conceive of *things* in terms of *other things*.

Source: H-H. Wang and W-J. Liao, IASDR, 2009.

3. Basic Concept of CSCD

Metaphors for Team Working In Design



SOURCE		J	TARGET			
Spider			Juicy Salif			
Feature	Salience		Feature	Salience		
Hunt	8.5		Twister	8.7	High	Action
Spin	6.5		Press	8		
Move	6		Collect water	7.5		
Jump	5.5		Hold	7		
Stand	5.5		Stand	6.3	Low	
Creep	5.5					
Twister	3					
Spider web	8		Squeezer head	8.7	High	Visual
The body	7		Filter	7.1		
The legs	6.7		Color	7	Low	
Squeezer head	5.7		Container	6.7		
			The holder	6		

The successful design teamwork seems closely related to common design concepts or metaphors"

- serve as *shared reference frameworks* for combining distributed designs into emerging design solutions.
- interacts continuously with individual interpretations of "*what parts are*" and "*how parts are*" interrelated with one another.

Source: H-H. Wang and W-J. Liao, IASDR, 2009.

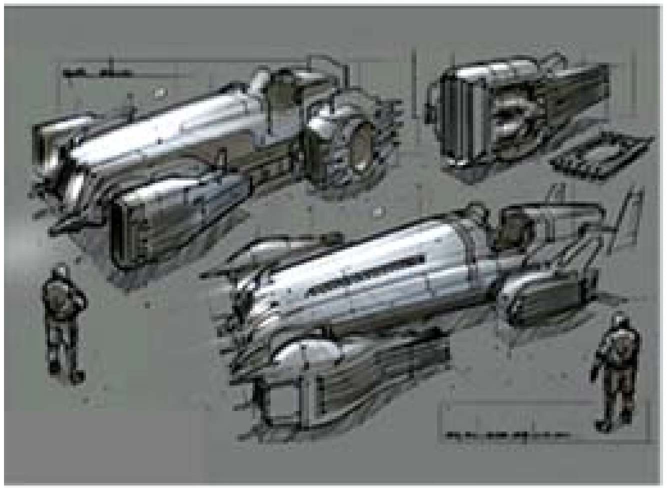
Outline

1. Introduction: Today's CSCD application
2. History: CSCW, Collaborative Engineering, and CSCD
3. Basic Concept of CSCD
4. **CSCD Methodology**
5. CSCD Tools
6. CSCD Architecture
7. Web and Agent Technologies for Collaborative Design
8. A Case Study:
   Tracking of Design Changes In Mechanical Product
9. Current Application

4. CSCD Methodology

Engineering projects involve a large number of components and the interaction of multiple technologies:

- the components included in the product are decided in an iterative design process,
- iteration proceeds towards increasing detail,
- design personnel may change, and their numbers expand with increasing level of detail.

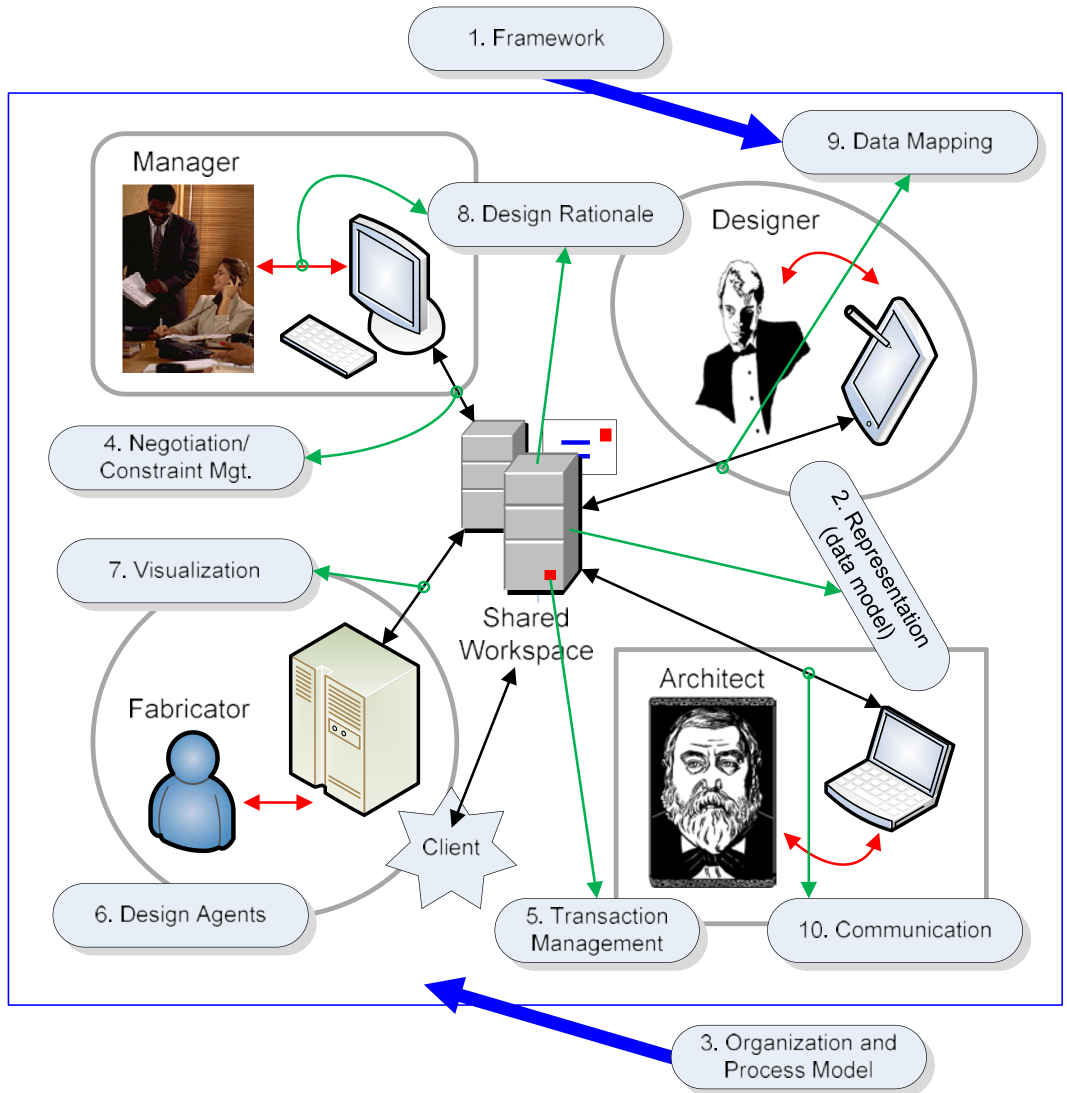


These problems demand considerable communication and coordination between various participants.

Image courtesy of www.carbodydeseign.com, 2011.

4. CSCD Methodology

General framework and issues in the collaborative design.



4. CSCD Methodology

Developing computer-supported design systems for collaborative design:

- Step 1 : Frameworks
- Step 2 : Data Models and Representation
- Step 3 : Organization and Process Models
- Step 4 : Negotiation and Constraint Management Techniques
- Step 5 : Transaction Management
- Step 6 : Design Methods
- Step 7 : Visualization Techniques
- Step 8 : Design Rational Records
- Step 9 : Data Mapping and Query Processing
- Step 10: Communication Protocols

4. CSCD Methodology

Step 1: Frameworks

Dealing with problem solving mechanism and computer architecture

- most current computer systems utilize the Blackboard architecture
- the shared workspace (Blackboard) is divided into three partitions, i.e., *solution*, *coordination*, and *negotiation*.
- a large part of the shared workspace is being implemented as an object-oriented database,
- the key components of the database include *Storage Manager*, *Object Manager*, *Transaction Manager*, and *Query Manager*.

Step 2: Data Models and Representation

Dealing with the development of product models and their representation needed for communicating information across disciplines.

Source: D. Sriram, 1993.

4. CSCD Methodology

Step 3: Organization and Process Models

Investigating strategies for organizing the engineering activities for effective utilization of computer-aided tools.

Step 4: Negotiation and Constraint Management Techniques

Dealing with conflict detection and resolution between various agents.

Step 4.1: Negotiation

Conflicts can occur either due to interface constraint violations or due to contradictory modifications of a single object.

Techniques to address the negotiation problems:

- Constraint relaxation
- Goal re-specification.

Source: D. Sriram, 1993.

4. CSCD Methodology

Step 4.2: Constraint Management

Constraints are continually being added, deleted and modified throughout the development of a new product. The resulting constraints are often numerous, complex, and may contain conflicting and/or unrealizable requirements.

Focus of constraint management:

- Keeping track of all relevant constraints and parameters
- Understanding the basic design relationship and trade-offs.

Source: D. Sriram, 1993.

4. CSCD Methodology

Step 5: Transaction Management

Dealing with the interaction issues between the agents and the central communication medium.

Comparative study:

- Traditional database management systems include notions of *atomicity* and *serializability* of transactions
- Collaborative engineering design include notions of *units of work must interact*, so that the results are usable together.

Key features of transaction management model include:

transaction nesting and grouping, exploitation of application semantics to ensure data consistency, communication facilities, version management for parallelism, and documentation of design evolution.

Source: D. Sriram, 1993.

4. CSCD Methodology

Step 6: Design Methods

Dealing with techniques utilized by individual agents.

The design shells of agents tend to be domain independent and incorporate certain problem solving methods, e.g., case-based reasoning, hierarchical refinement, constraint propagation, qualitative reasoning, first-principle reasoning, etc.

Key activities in the development of design shells:

- **Problem solving support** : What kind of techniques need to be incorporated in the shell ?
- **Representation** : How does one represent structure, function, behavior, geometry, design rationale, etc. ?
- **Index and Retrieval** : What schemes are required for using function, structure, sub-graph matching, behavior, etc., for indexing and retrieving past designs?
- **Data Quality** : How is the fuzzy engineering data represented and processed?

Source: D. Sriram, 1993.

4. CSCD Methodology

Step 7: Visualization Techniques

Dealing with user interfaces and physical modeling techniques.

To enable scientific information/data to be conveyed by visual diagrams (images), the computer-aided systems should have the ability to:

- generate diagrams
- recognize and understand diagrams

Key features in the visualization of engineering information:

- symbol to structure mapping
- interpretation of engineering drawings
- geometric modelling

Source: D. Sriram, 1993.

4. CSCD Methodology

Step 8: Design Rationale

Keeping track of the justifications generated during design, or other engineering activities.

- The CSCD system should incorporate techniques to encode design rationale about both the overall process and individual choice points.

Step 9: Data Mapping and Query Processing

Dealing with information transfer between various agents.

- The CSCD system should have the ability for translation of information (both syntactic and semantic mappings) between the shared workspace and local applications.
- Supporting facilities are needed for status checking and monitoring, communication between the shared workspace and users, and coordination.

Source: D. Sriram, 1993.

Step 10: Communication Protocols

Facilitating the movement of objects between various applications.

- The CSCD system should incorporate techniques to encode design rationale about both the overall process and individual choice points.

- Protocol can be designed in two categories (Dorst and Dijkhuis, 1995):
  - → *Concurrent protocol*, focus on the process-oriented aspect
  - → *Retrospective protocols*, focus on the content-oriented or cognitive aspect.

Source: D. Sriram, 1993.

Outline

1. Introduction: Today's CSCD application
2. History: CSCW, Collaborative Engineering, and CSCD
3. Basic Concept of CSCD
4. CSCD Methodology
5. **CSCD Tools**
6. CSCD Architecture
7. Web and Agent Technologies for Collaborative Design
8. A Case Study:
   Tracking of Design Changes In Mechanical Product
9. Current Application

5. CSCD Tools

Requirements for Networked-based Collaborative Design Tools

Req. 1 : Able to support designers to capture, store retrieve knowledge efficiently and effectively throughout the design process.

Req. 2 : Able to supply clear and complete design knowledge and, at the same time, facilitate designers intervention and customization during the decision-making activities in the design process.

Req. 3 : Able to support information flow between participants for a heterogeneous computing environment.

Source: X. Zha and H. Du, 2005.

5. CSCD Tools

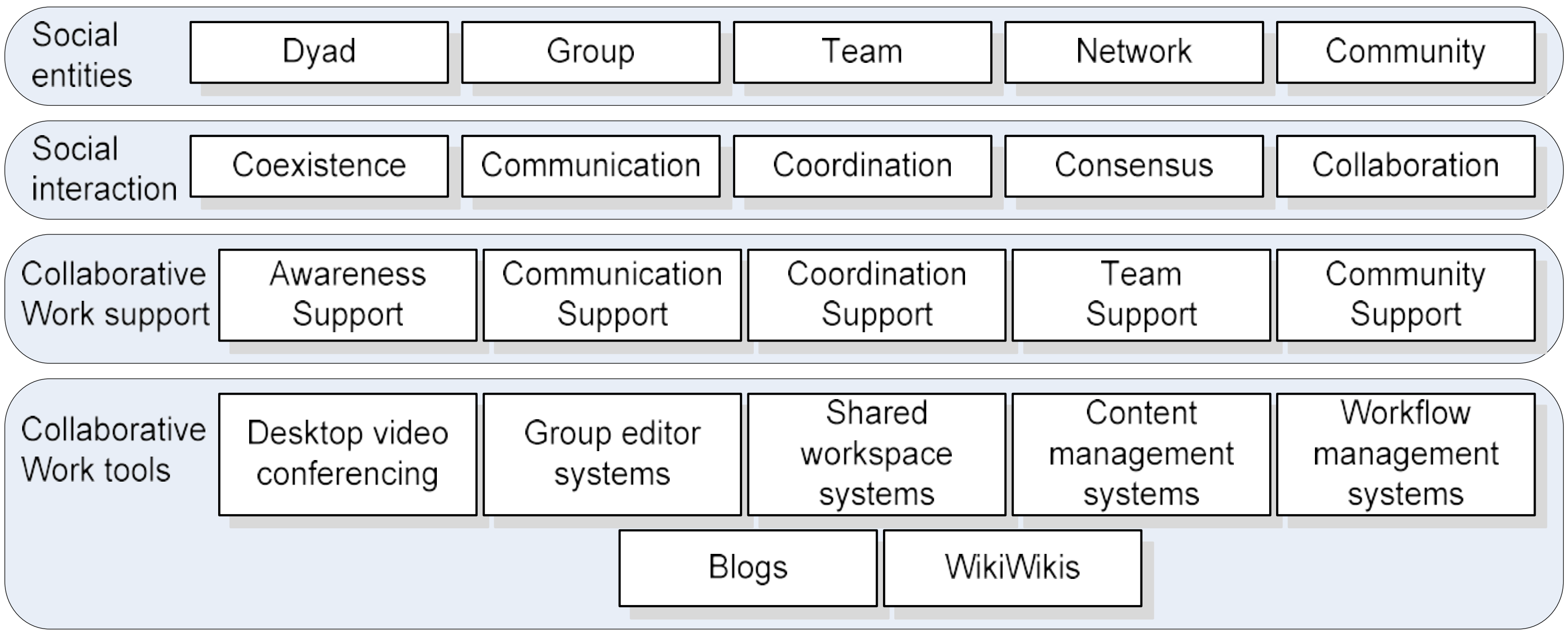
Req. 4 : Exist in an open environment so that models from new participants can be added, allowing the design model and tools to evolve with the design problem.

Req. 5 : Function as a means or mechanism for encapsulating expert knowledge or know-how using an object-oriented approach, comparing design solution alternatives and supporting the decision making process.

Req. 6 : Embodiment of a decentralized architecture that the coordination between design participants and models is not centrally modeled or controlled.

Source: Zha, X. and Du, H. 2005.

5. CSCD Tools



Source: T. Gross, Bauhaus-University Weimar, 2008.

5. CSCD Tools

Desktop Video Conferencing

Synchronous Conferencing Systems

(i) Audio conferencing systems

- telephones, i.e., the simplest form
- several more recent systems:
  - ☑ Skype [Skype 2007]
  - ☑ in some OS, audio conferencing is integrated (e.g., iChat on Mac OS X supports audio conferencing among up to ten users [Apple Computer 2007])

(ii) Desktop video conferencing systems

- MERMAID system [Sakata 1994]
- Supports multiple video connections
  - ☑ iChat [Apple Computer 2007]
  - ☑ NetMeeting [Microsoft 2006]
  - ☑ iVisit [iVisit LLC 2007]

Source: T. Gross, Bauhaus-University Weimar, 2008.

5. CSCD Tools

Group Editor Systems

GROVE [Ellis et al. 1991]

- Multi-user group outline editor for synchronous drafting of texts.
- Imposes hardly any system constraints, i.e., roles are not supported and default locking mode is no locking, all users can read and update any part of shared document.

Group Decision Support Systems (GDSS)

Aim

- Decision making process can be structured and recorded, in order to becomes more transparent.
- Users can analyze records to see how decisions are being developed.

The synchronous GDSS are often based on:

- issue-based information system method (IBIS)
- graphical IBIS (gIBIS)
- real-time group hypertext system (rIBIS).

Source: T. Gross, Bauhaus-University Weimar, 2008.

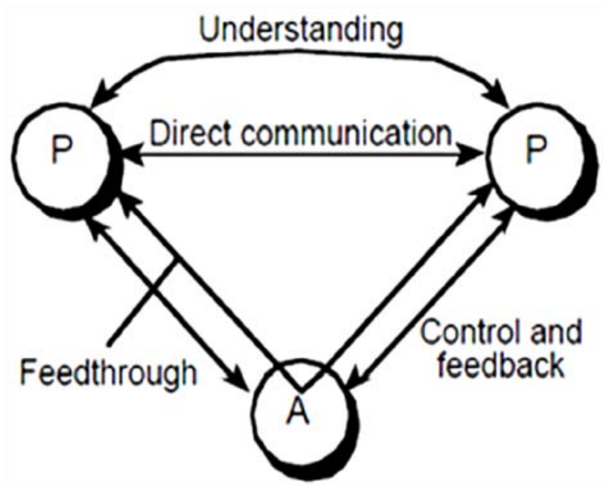
5. CSCD Tools

Shared Workspace Systems

(Dix, A., 1994)

- Support cooperative usage of shared data.
- Users can communicate through shared artefacts (i.e., explicit communication through direct communication & implicit communication as feedthrough).

P: person, ***A***: artefact.



(Ehrlich & Cash 1994, IBM 2007)

- Security features, i.e., notes are encrypted when transmitted over network.
- Access control is managed by different privileges, i.e., reader (read only), author (read, create new documents, manipulate own documents), editor (change documents of other users), depositor (save new documents but not read), designer (who can develop new applications).

Source: T. Gross, Bauhaus-University Weimar, 2008.

5. CSCD Tools

Content Management Systems

Collection of procedures used to manage work flow in a collaborative environment.

These procedures can be manual or computer-based to:

- allow for a large number of people to contribute to and share stored data
- control access to data, based on user roles
- aid in easy storage and retrieval of data
- reduce repetitive duplicate input
- improve the ease of report writing
- improve communication between users

Source: T. Gross, Bauhaus-University Weimar, 2008.

5. CSCD Tools

Workflow Management System

A **workflow**

- a sequence of connected steps.
- a depiction of a sequence of operations -- declared as work of a person, a group of persons, an organization of staff, or one or more simple or complex mechanisms.

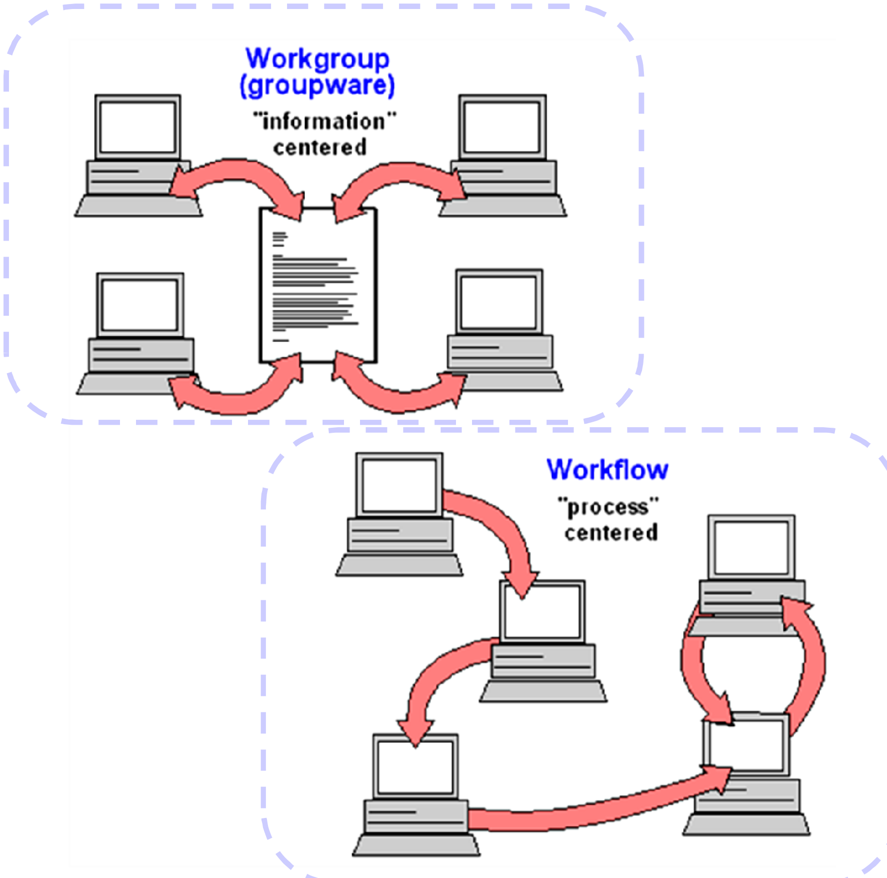


Image courtesy of Computer Desktop Encyclopedia, 1998.

5. CSCD Tools

Workflow Management System

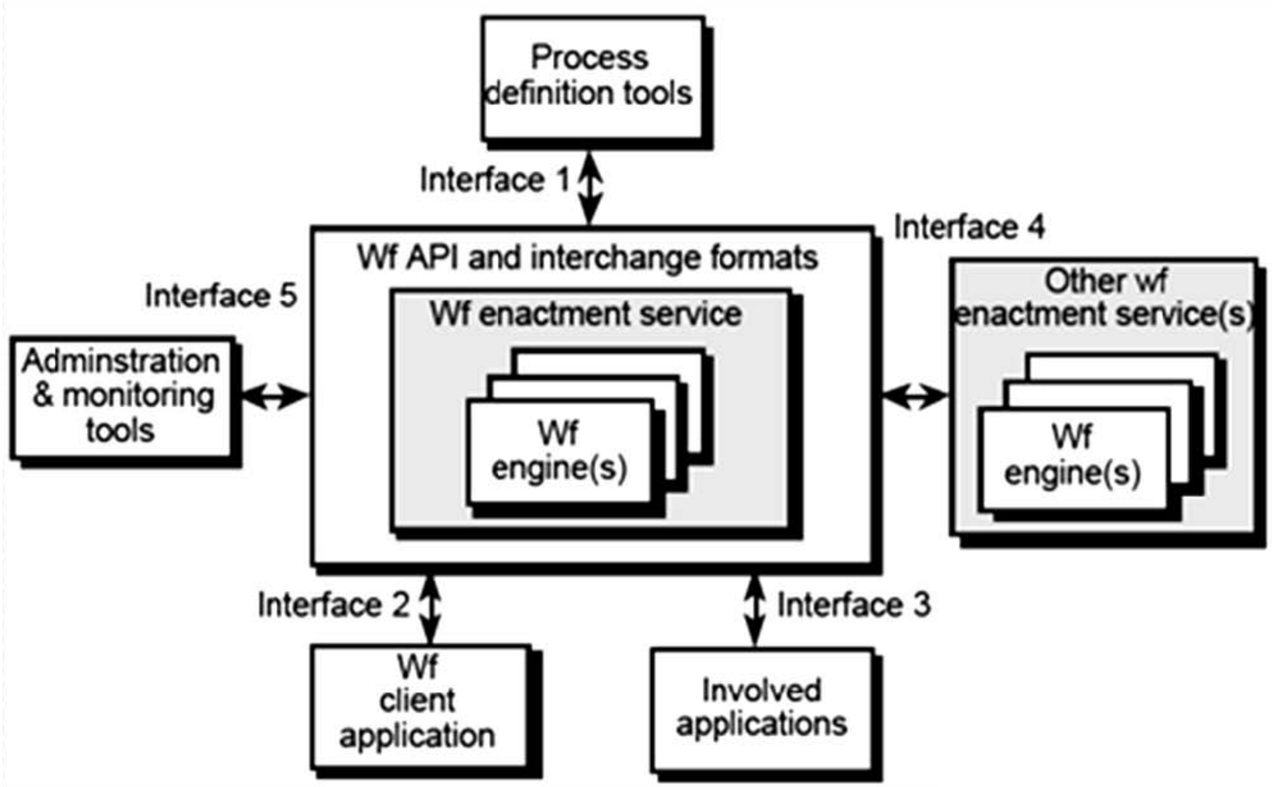
Four generations of WFMS

- **1st generation** : Very application-specific with hardcoded process definitions, closed and proprietary.
- **2nd generation** : Extracted workflow capabilities from application domain, WFMS were treated as separate applications, process definitions were tailorable through script languages.
- **3rd generation** : Offer generic workflow services that are accessible to other applications through APIs, architecture is open and based on standards, interchange formats are already defined.
- **4th generation** : Embedded enablers, integrated with other middleware services like Email or desktop management, ubiquitous, invisible.

Source: T. Gross, Bauhaus-University Weimar, 2008.

5. CSCD Tools

Workflow Management System



Source: T. Gross, Bauhaus-University Weimar, 2008.

Outline

1. Introduction: Today's CSCD application
2. History: CSCW, Collaborative Engineering, and CSCD
3. Basic Concept of CSCD
4. CSCD Methodology
5. CSCD Tools
6. **CSCD Architecture**
7. Web and Agent Technologies for Collaborative Design
8. A Case Study:
   Tracking of Design Changes In Mechanical Product
9. Current Application

6. CSCD Architecture

Collaborative Design Information and Support Systems

According to the purposes and focuses, the distributed computing systems can be grouped into the following categories:

Cat.-1 : Collaborative product data/information management systems for engineers to timely obtain the necessary product data and knowledge.

Cat.-2 : Network-based collaborative design systems, which can be further divided into web-independent, and web-dependent systems.

Cat.-3 : Process-centered collaborative design and workflow management systems.

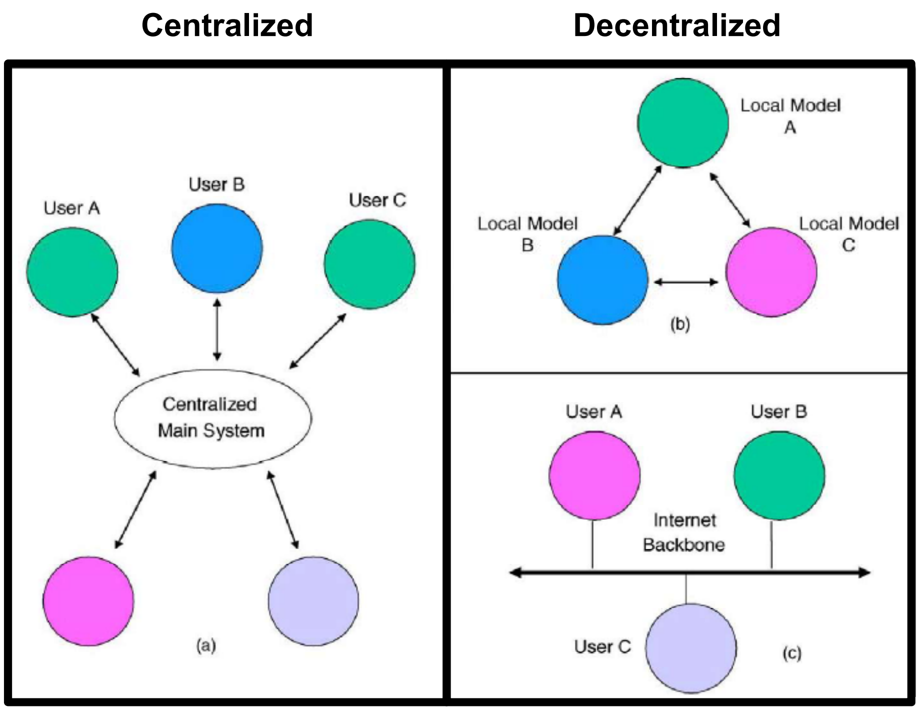
Cat.-4 : Conflict detection, management and resolution systems for collaborative design.

Cat.-5 : Flexibility and security focused collaborative design system .

Cat.-6 : Interoperability approaches in heterogeneous collaborative design systems.

6. CSCD Architecture

Interactions Between Modules Exchange Service



(a) Centralized multi-user system model.

(b) Integrated concurrent system model.

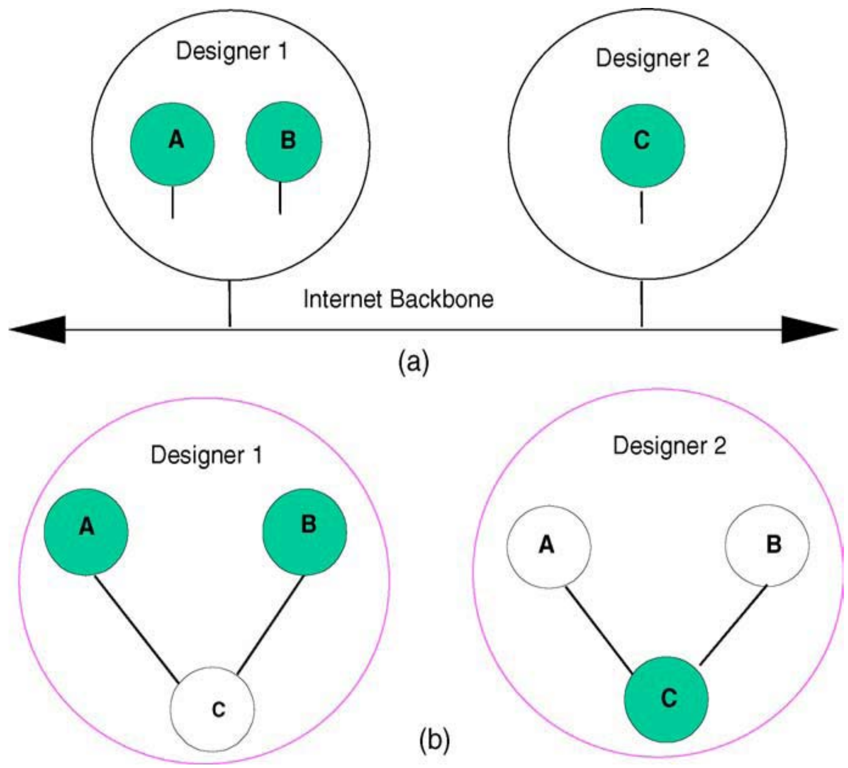
(c) Multi-agent based distributed system model.

- Wide-area networks and the internet-based WWW allow developers to provide remote web-based design servers.
- CAD systems running on these servers can support a large-scale group of users to communicate over the network.

Source: X, Zha and H. Du, 2005.

6. CSCD Architecture

Distributed Modules



Design module modelling and evaluation (DMME) :

- Involving two designers and three modules.
- Designer 1 defines modules A and B.
- Designer 2 defines module C.
- The two domains communicate through an Internet connection.

Source: X, Zha and H. Du, 2005.

6. CSCD Architecture

Module Network Modelling and Implementation

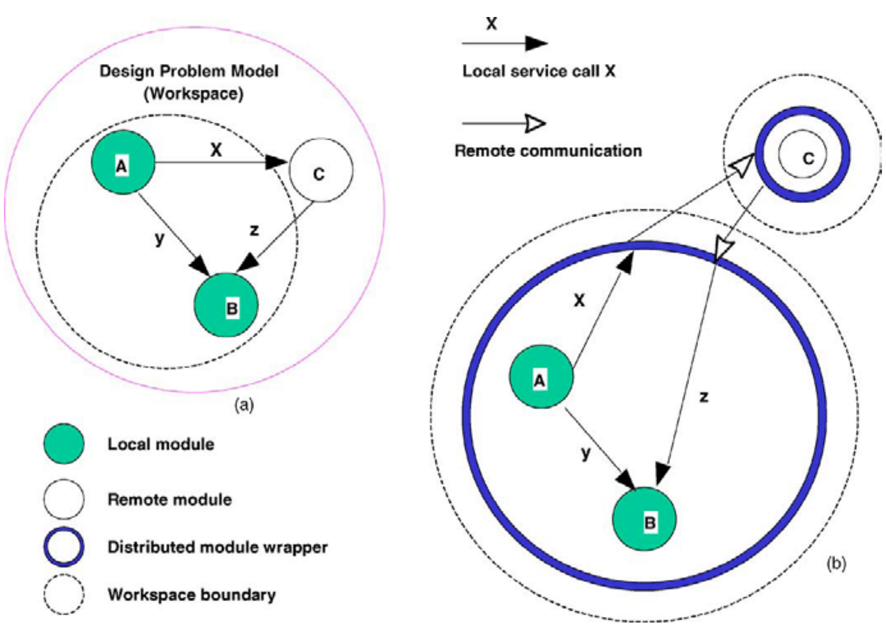


Fig. (a):

At the modeling layer the designer defines the problem in terms of modules and interactions among modules .

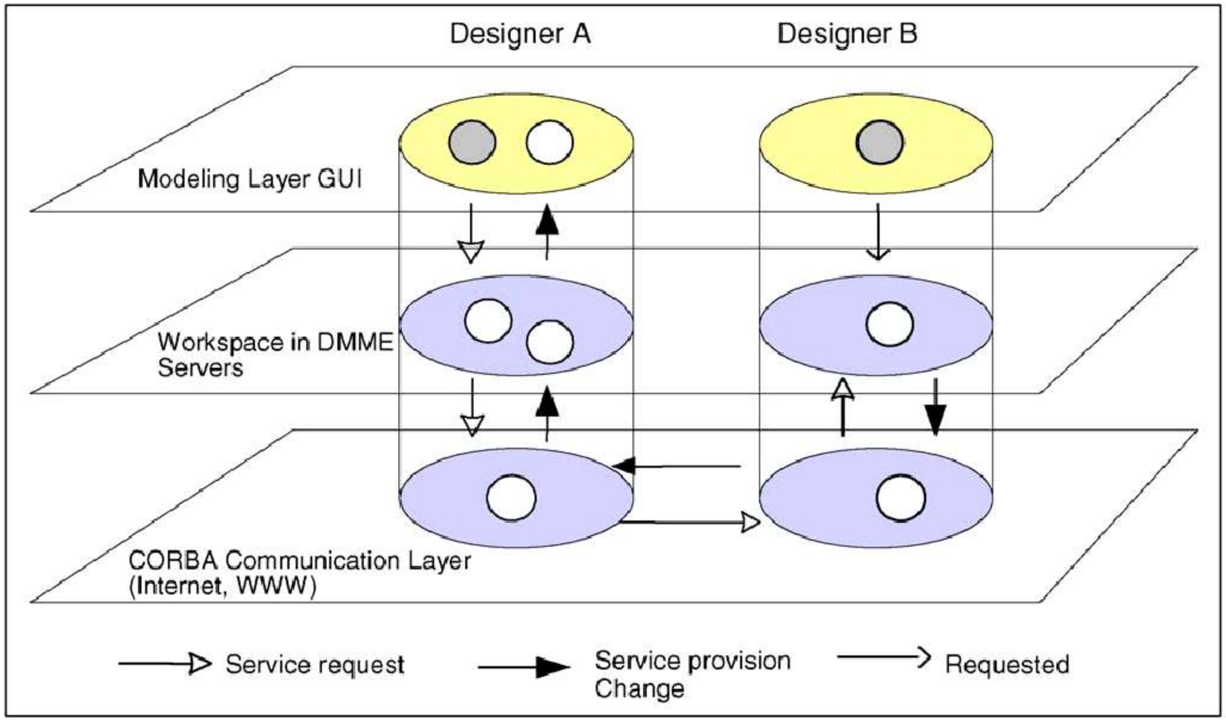
Fig. (b):

- The unseen implementation layer is created to provide the functionalities described in the modeling layer.
- This layer locates remote modules.
- The remote modules must be distributed objects capable of communicating via a standard communication protocol.

Source: X, Zha and H. Du, 2005.

6. CSCD Architecture

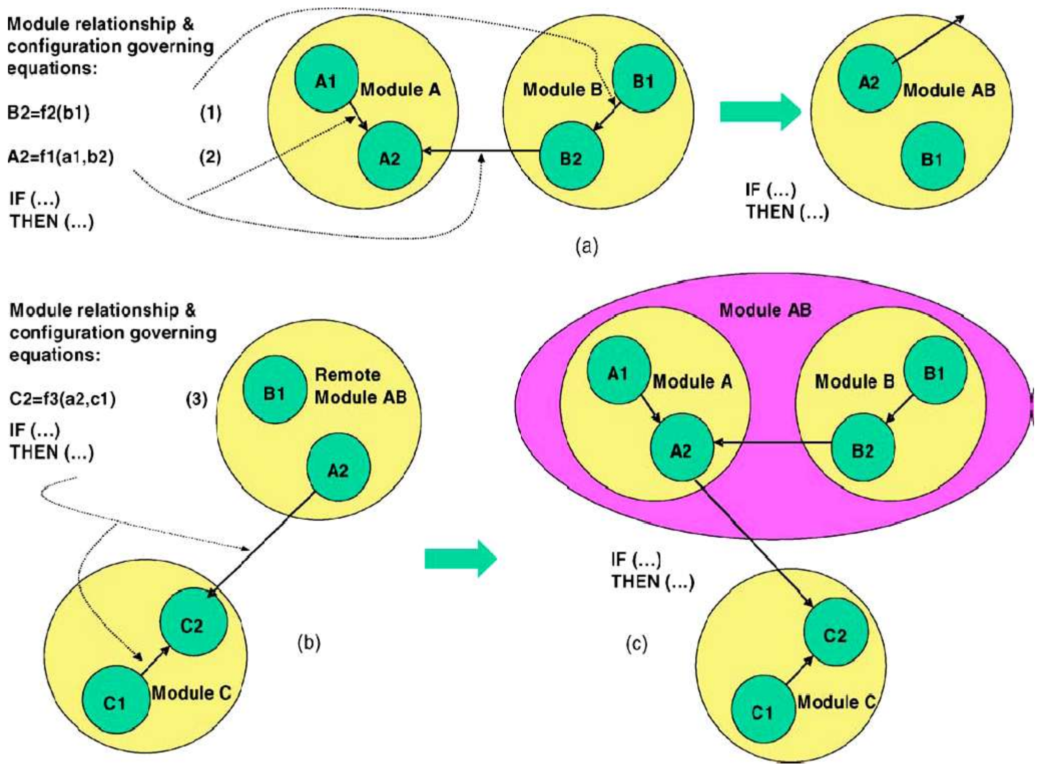
Service Exchanges Between Distributed Modules



Source: X, Zha and H. Du, 2005.

6. CSCD Architecture

Distributed Design Model



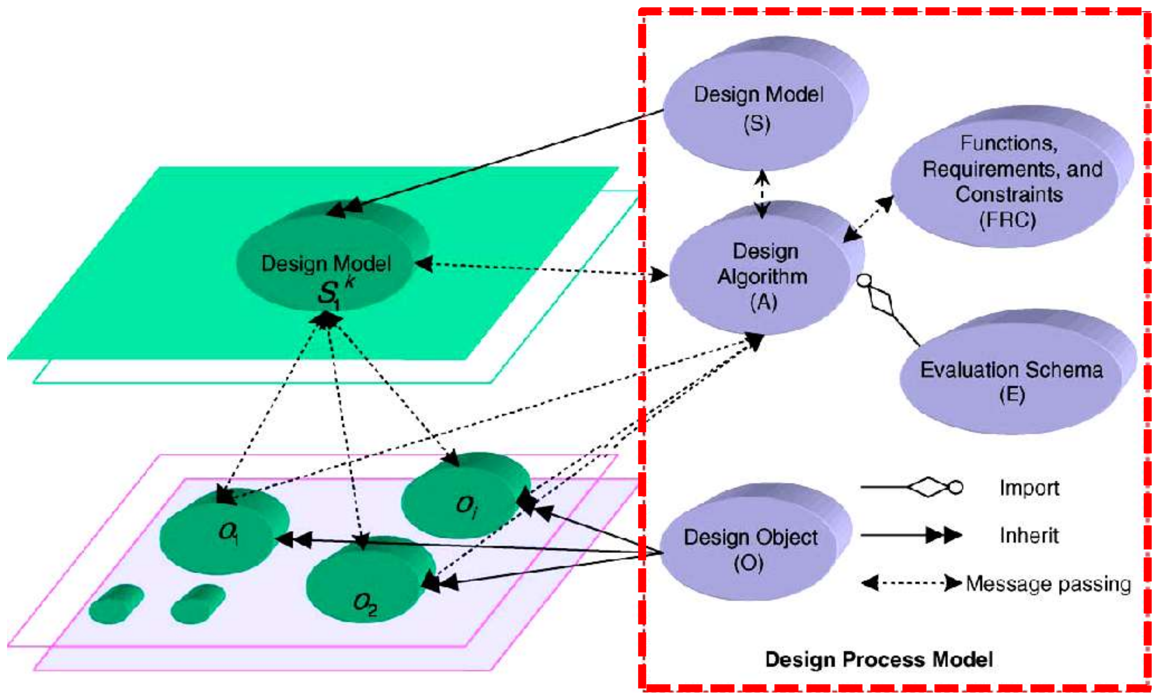
Simple distributed design model with two modules and a remote module:

- **Fig. (a)**
  A distributed module with the outward appearance .
- **Fig. (b)**
  The model from the viewpoint of the ABC designer. Module C is local to the designer.
- **Fig. (c)**
  The true integrated model created when the remote module AB and the local module C are connected.

Source: X, Zha and H. Du, 2005.

6. CSCD Architecture

Architecture of Design with Modules (*X. Zha and H. Du, 2005*)

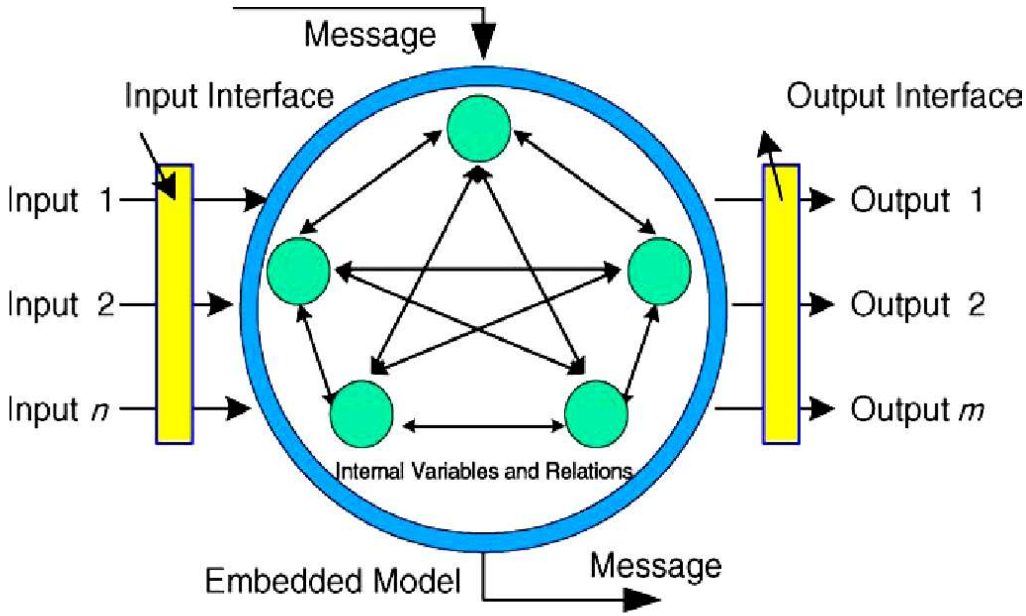


- The modular design concept has been widely used in product design for flexibility, rapid responsiveness, ease of maintenance and rapid deployment.
- During the design process, information processing is inherently model-based because design object is structural in type.

Source: X, Zha and H. Du, 2005.

6. CSCD Architecture

Module Definition and Embedded Model

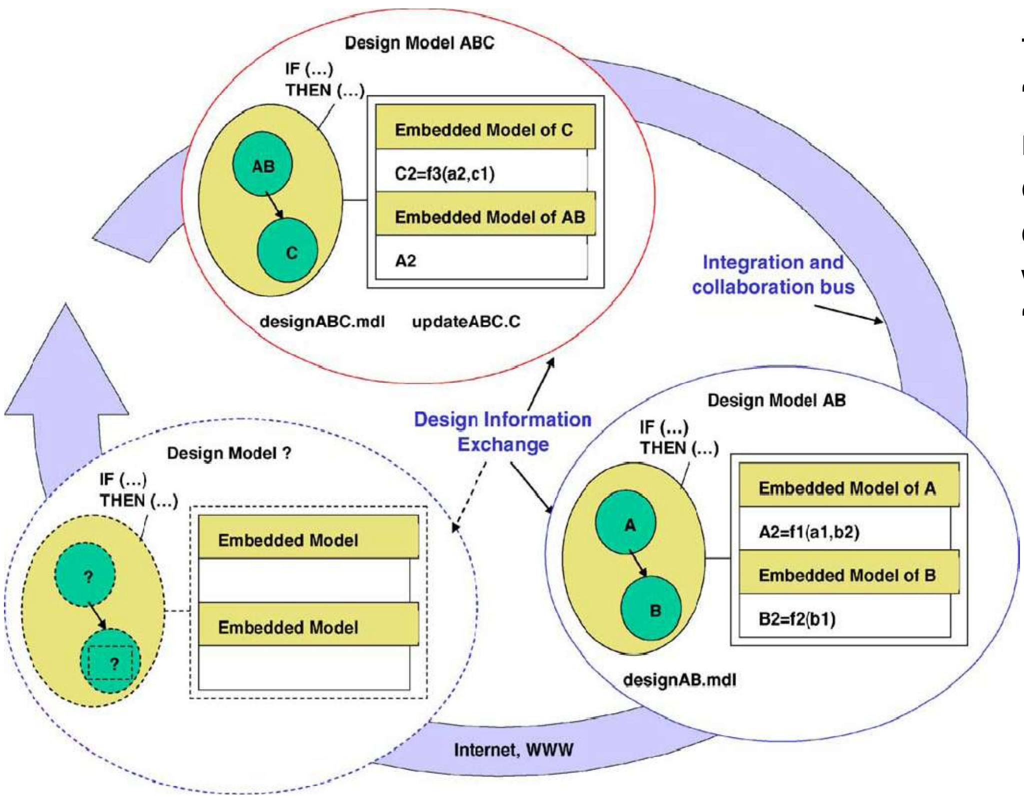


- Modules interact with each other by exchanging information and services, reacting to each other's changes for an integrated system model.
- Modules whose dependencies are to be analyzed are tasks and parameters contained in design tasks, and their dependencies are constraints among parameters.

Source: X, Zha and H. Du, 2005.

6. CSCD Architecture

Module Network Configuration

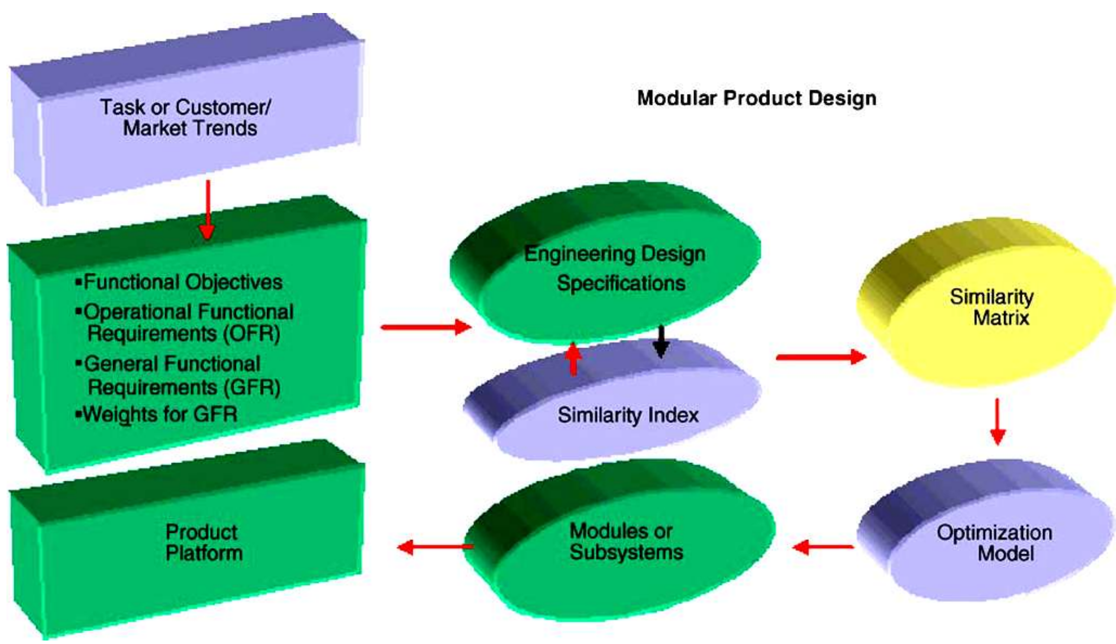


The embedded model of the "module AB" in design problem model ABC contains module connector that manages the design information exchange with the distributed design "module AB".

Source: X, Zha and H. Du, 2005.

6. CSCD Architecture

Modular Design and Modularization Process



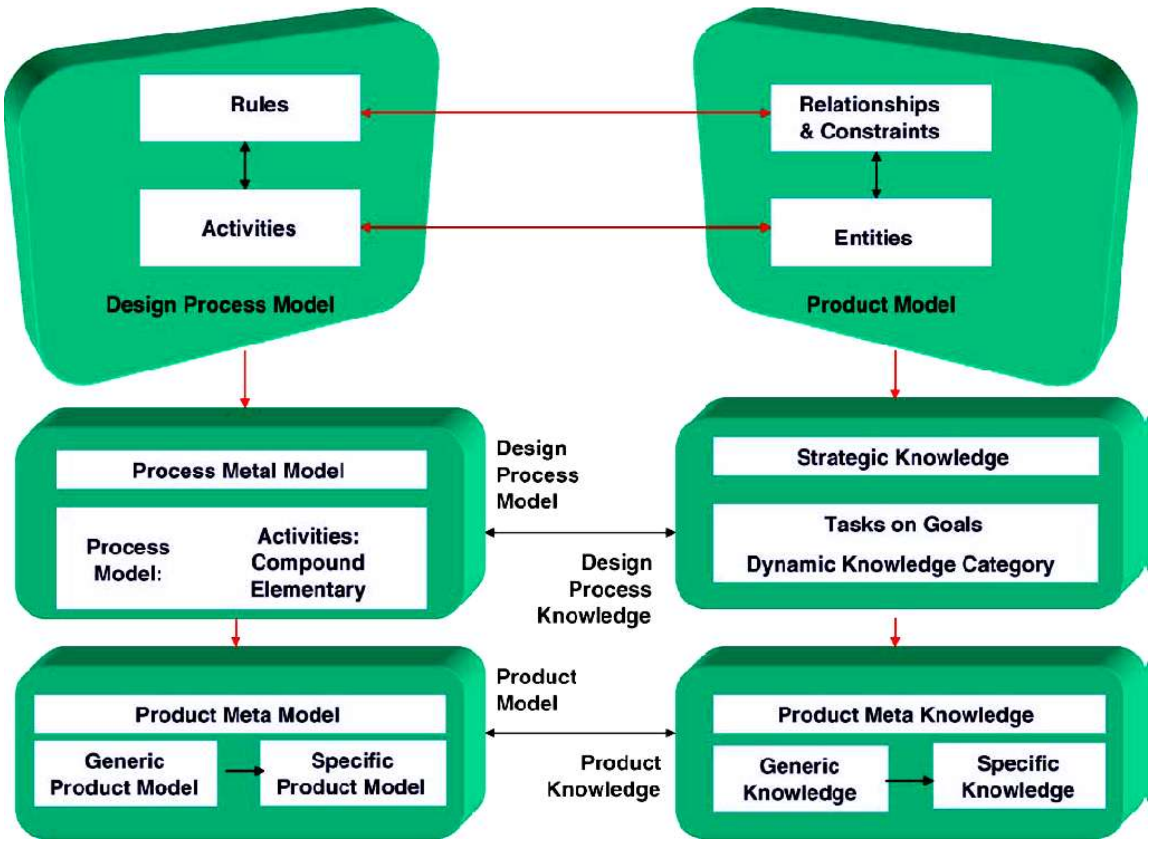
Modularization process

- **Step 1:**
  Requirement analysis and modelling.
- **Step 2:**
  Modularization of product architecture.
- **Step 3:**
  Module classification.
- **Step 4:**
  Mapping between functional modules and structural modules.
- **Step 5:**
  Representation of product architecture in hierarchical building blocks (modules).
- **Step 6:**
  Optimization of product architecture.

Source: X, Zha and H. Du, 2005.

6. CSCD Architecture

Product, Design Process and Knowledge



- Design knowledge is very extensive.
- Both on-line and off-line design knowledge representations are dealt with the modularization process.

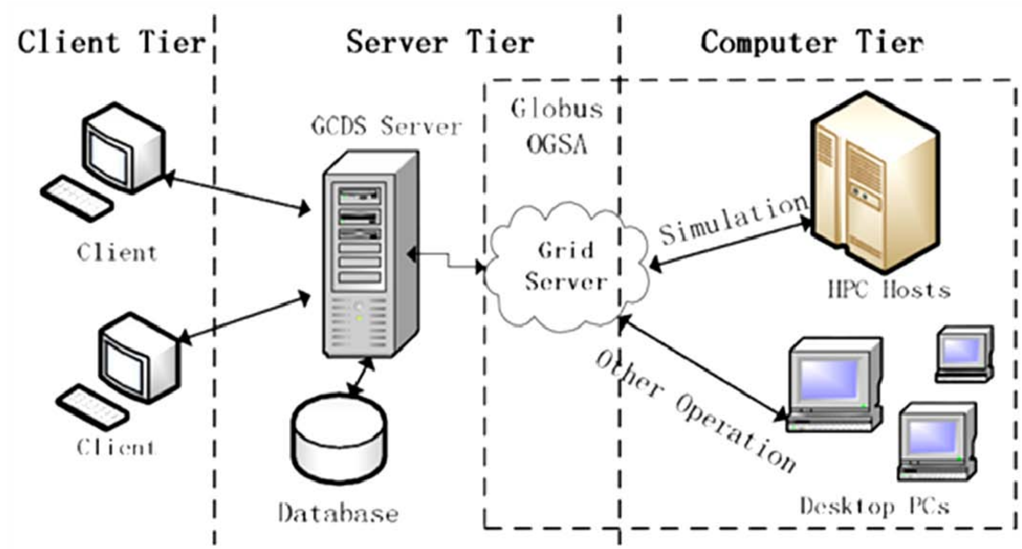
Source: X, Zha and H. Du, 2005.

Outline

1. Introduction: Today's CSCD application
2. History: CSCW, Collaborative Engineering, and CSCD
3. Basic Concept of CSCD
4. CSCD Methodology
5. CSCD Tools
6. CSCD Architecture
7. **Web and Agent Technologies for Collaborative Design**
8. Case Study:
   Tracking of Design Changes In Mechanical Product
9. Current Application

A. Web-based Collaborative Design Framework

- Wide-area networks and the internet-based WWW allow developers to provide intelligent knowledge servers.
- Knowledge-based expert systems running on servers can support a large-scale group of users to communicate with the system over the network.



Three-tier Communication Framework

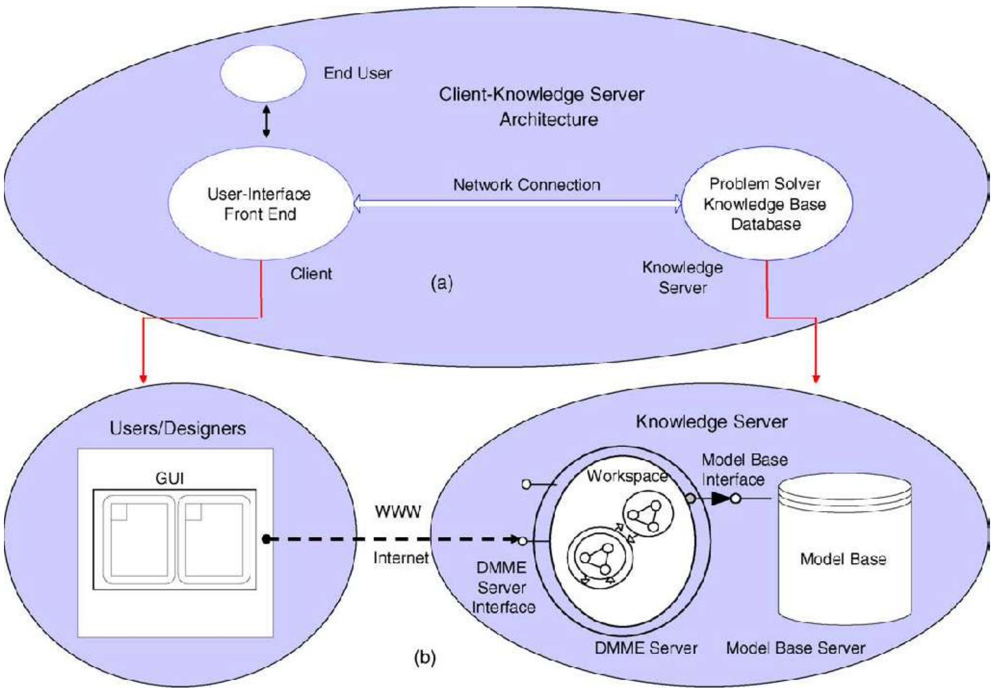
Image courtesy of Xuan et al., 2009.

7. The Web and Agent Technologies for Collaborative Design

A. Web-based Collaborative Design Framework

The approach:

- To make knowledge servers available, developer distributes the software front-ends that allow users communicate with the servers.
- The user interfaces based on Web protocols provide access to the knowledge servers.
- Users communicate over the network without special hardware or software to consult the services with appropriate web browsers.



Source: X, Zha and H. Du, 2005.

A-1. Collaborative Design on Grid (CDG)

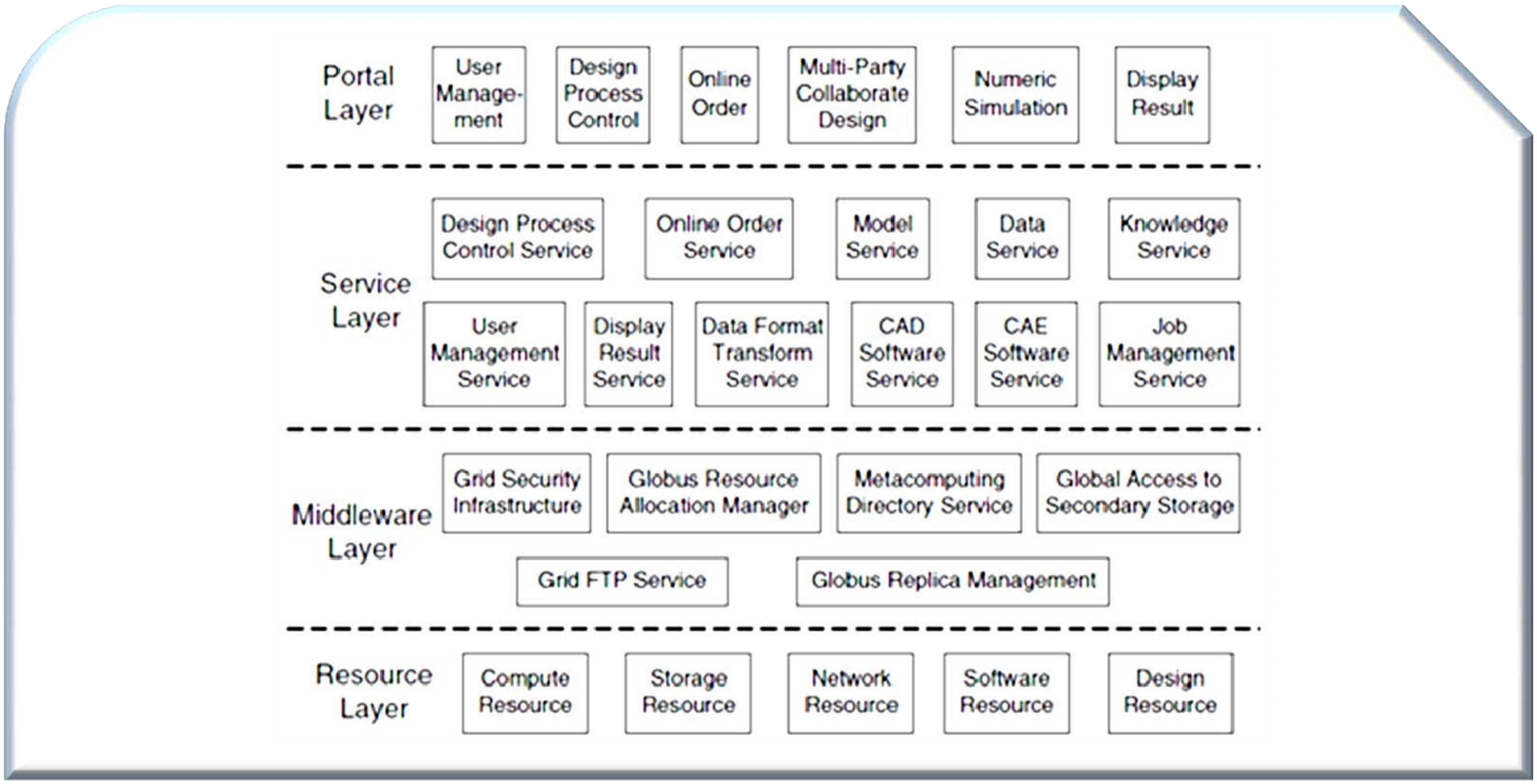
- The Grid Computing framework has been developed as an infrastructure of connecting and unifying various remote resources.
- Its basic characters are resource sharing and eliminate resource separation, but not scale of massiveness.

Concept of CDG

"CDG gathers geographically distributed, heterogeneous computational resources, storage resources, equipment resources, data resources and human resources to form a dynamic virtual organization for the design task of same product in WAN environment."

Source: Lie et al., 2007.

A-2. Structure of CDG



Grid services are combined with web services and grid technology, and are suitable for realizing all resources sharing, cooperative interaction of multi-party designers, multiple communication modes (peer-to-peer, end-to-end, multi-broadcast, etc.) and communication requirement (massive data transfer, stream data, group broadcast, etc.) in dynamical environment.

Source: Lie et al., 2007.

A-3. Key Technologies of CDG

A. Task Management

Task management takes charge of all tasks including routines management and dispatch, disintegration, distribution, result merge, report generation, etc.

B. Resource Management

Resource management deals with the problems how to describe resources for resource registry and how to discover resources in the CDG environment.

(i) Resource Description

The resources can be classified into four types: *data resources*, *computing resources*, *service resources*, and *storage resources*.

(ii) Resource Discovery

Resource discovery refers to the process of locating resources in a grid environment needed for a computational activity.

Source: Lie et al., 2007.

A-4. Workflow of Calling Collaborative Design Services

Execution process

Step 1: A service provider publishes services on a service proxy.

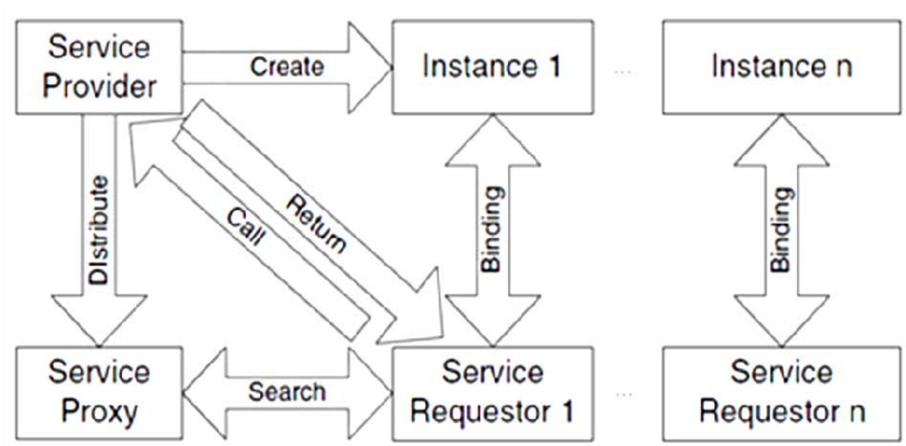
Step 2: A service requestor searches a requisite service from the service proxy.

Step 3: The service requestor calls the corresponding service according to the searched service address.

Step 4: The service provider responds the service requestor's calling and creates a service instance.

Step 5: The service provider returns address of the service instance to the service requestor.

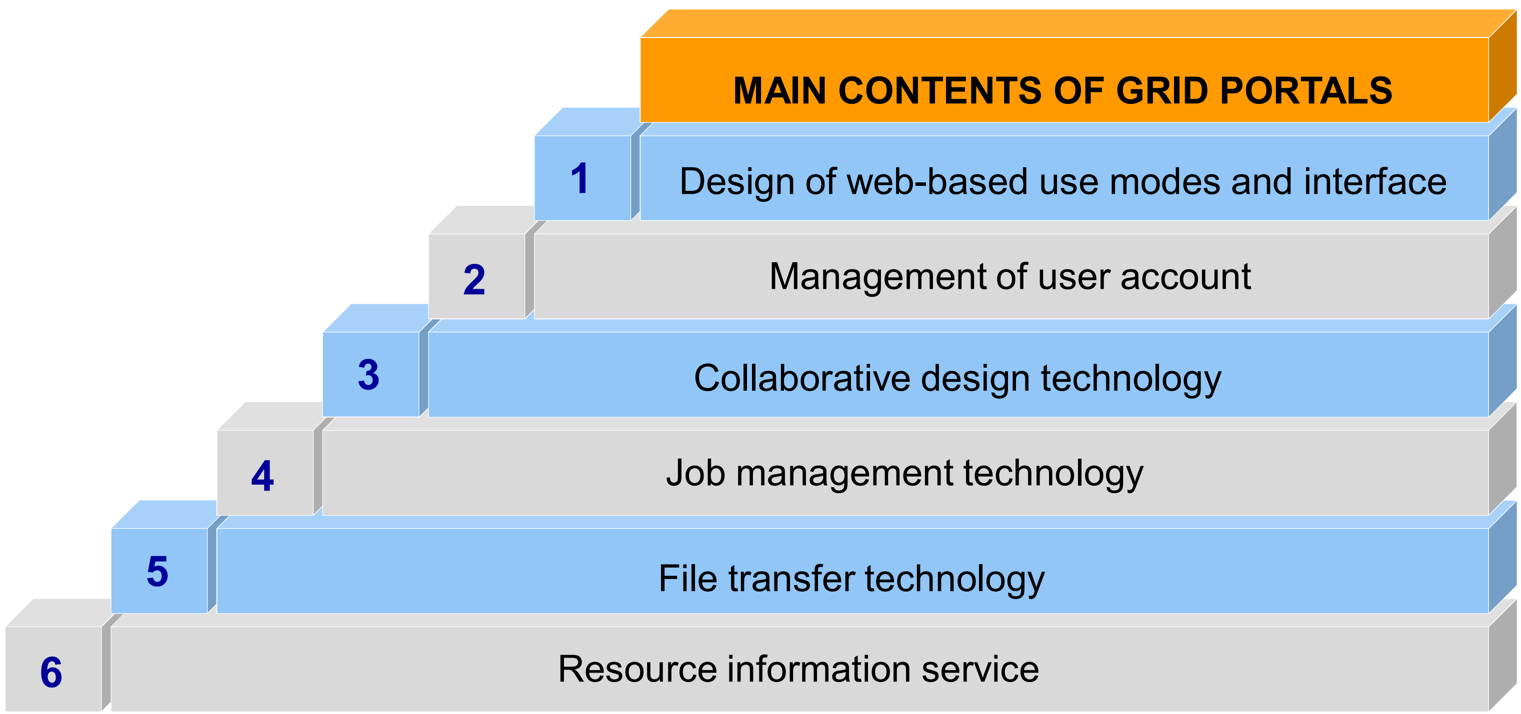
Step 6: The service requestor and the service instance are bound.



Source: Lie et al., 2007.

A-5. Design of Grid Portal

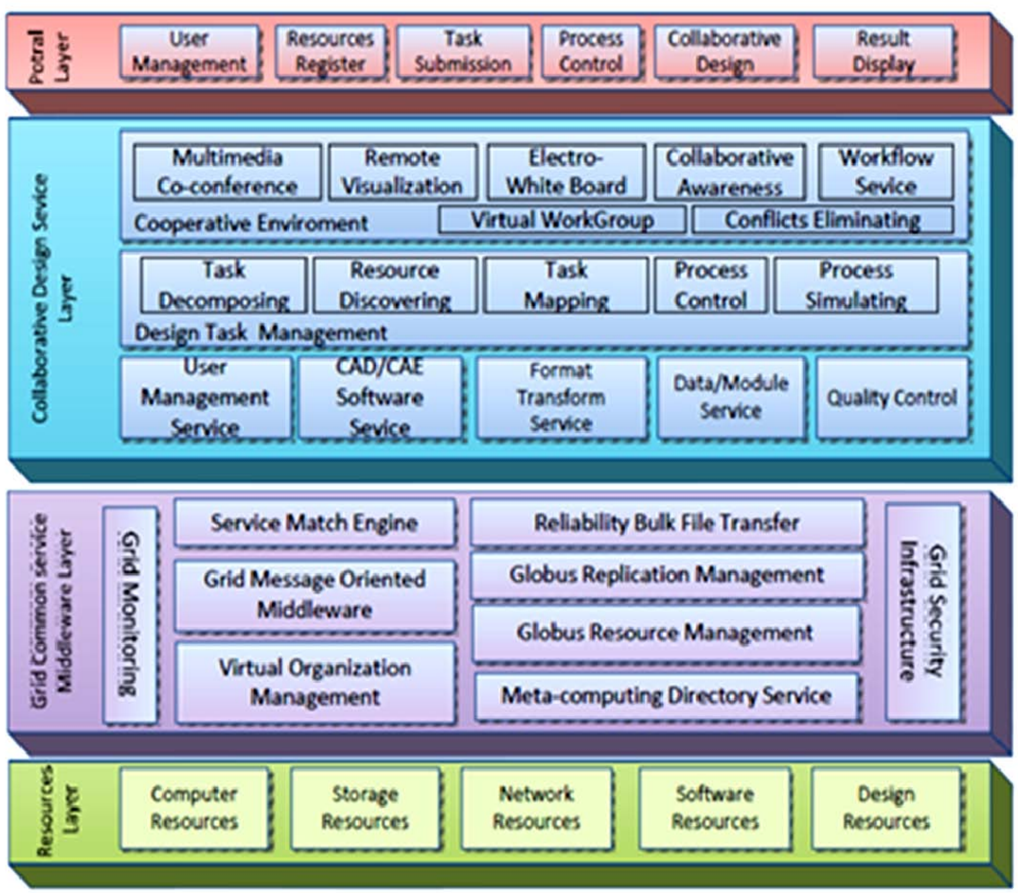
Grid Portal is an effective means of utilizing grid resource and provides high-end graphic interface for potential users.



Source: Lie et al., 2007.

7. The Web and Agent Technologies for Collaborative Design

A-6. Layered Software Structure of CDG



Source: Xuan et al., 2009.

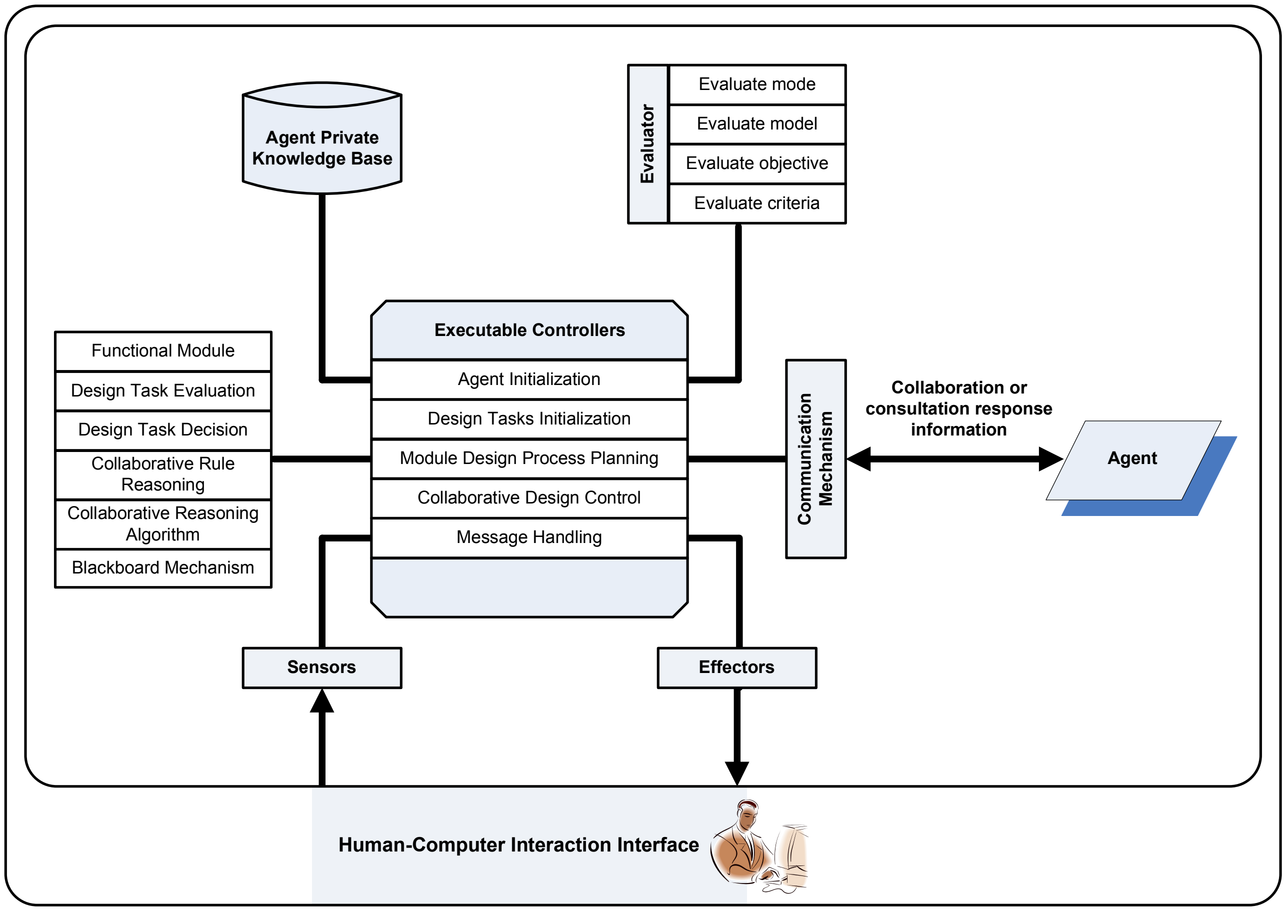
B. Agent-based Collaborative Design Framework

- Collaborative product design requires that the design *links of different modules* and different design aspects could be carry out in parallel and coordinate with each other in order to detect and solve design problems.
- The *module*, respectively *"agent"*, as the design units in collaborative design are independent and connected by the standard interface.
- An agent *plans* and *solves* the local problems of modular design by the design-related knowledge.
- Each agent is a *knowledge system* which solves the modular design problem based on their knowledge base.

B. Agent-based Collaborative Design Framework

- Collaborative product design requires that the design *links of different modules* and different design aspects could be carry out in parallel and coordinate with each other in order to detect and solve design problems.
- The *module*, or called *"agent"*, as the design units in collaborative design are independent and connected by the standard interface.
- An agent *plans* and *solves* the local problems of modular design by the design-related knowledge.
- Each agent is a *knowledge system* which solves the modular design problem based on their knowledge base.

B-1. Agent Structure Model for Collaborative Design



Source: Ming et al., 2010.

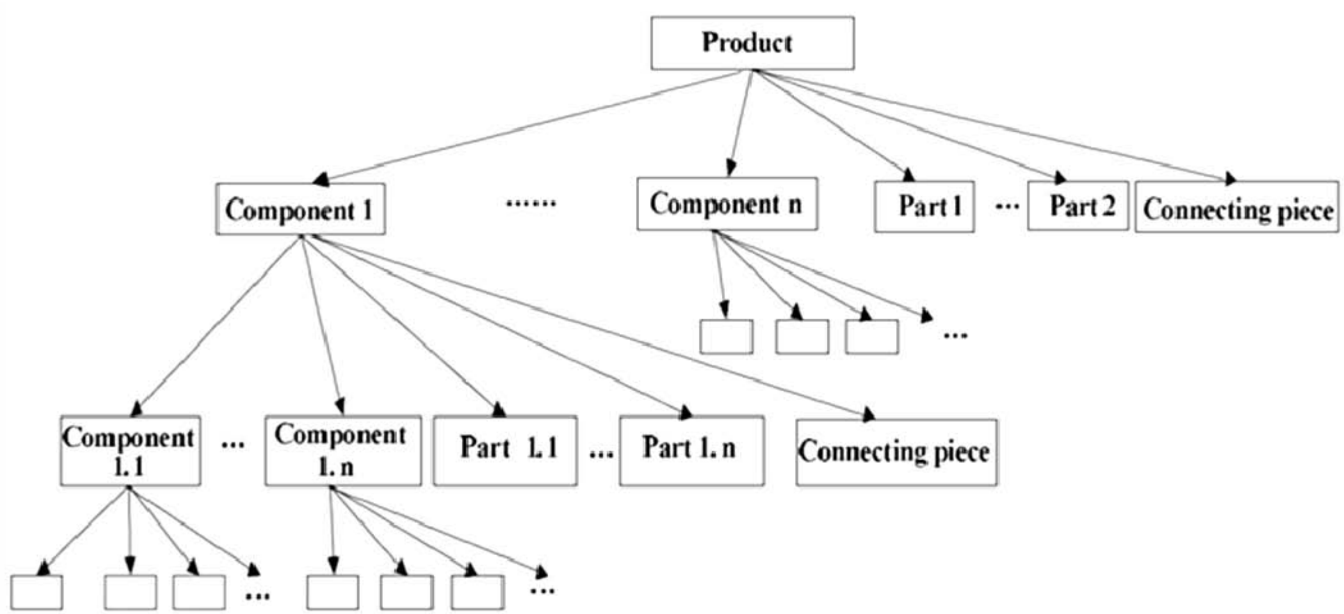
B-2. Modularity Strategy Based on Multi-Agents

Strategy 1: Decomposition-based Modularity

- The concept of product modularity is generally the planning and organization for the products or systems.
- Cell decomposition of the product is the intention of product modular decomposition technology.
- These units have the same function and the same combination elements with different performances or uses, even different structural features but can swap when they belong to one group.
- Methods for product modular decomposition:
  - modular decomposition based on structure
  - modular decomposition based on function
  - modular decomposition based on task.

B-2. Modularity Strategy Based on Multi-Agents

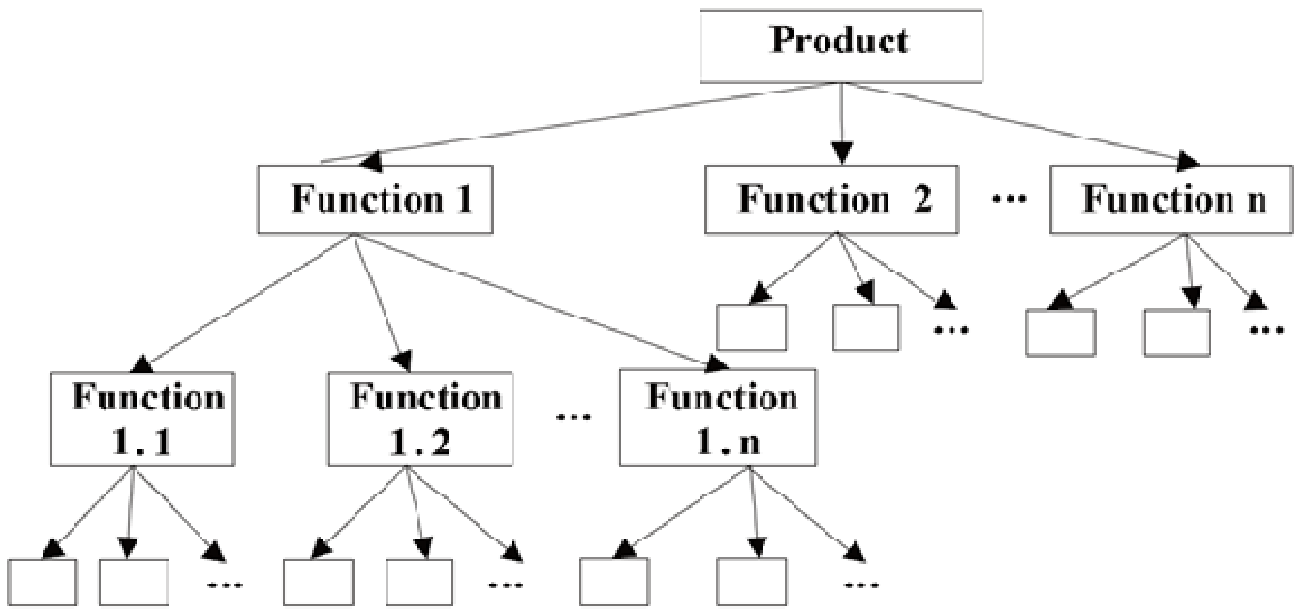
Modular Decomposition Based on Product Structure



Source: Ming et al., 2010.

B-2. Modularity Strategy Based on Multi-Agents

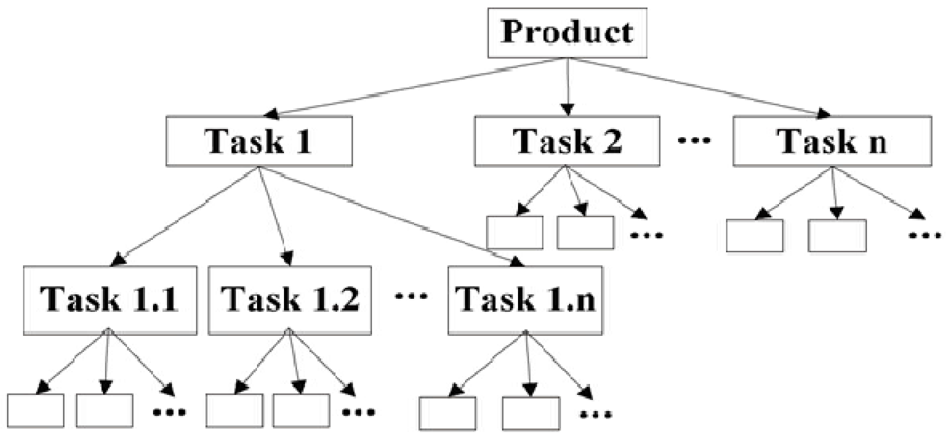
Modular Decomposition Based on Product Function



Source: Ming et al., 2010.

B-2. Modularity Strategy Based on Multi-Agents

Modular Decomposition Based on Product Task



Source: Ming et al., 2010.

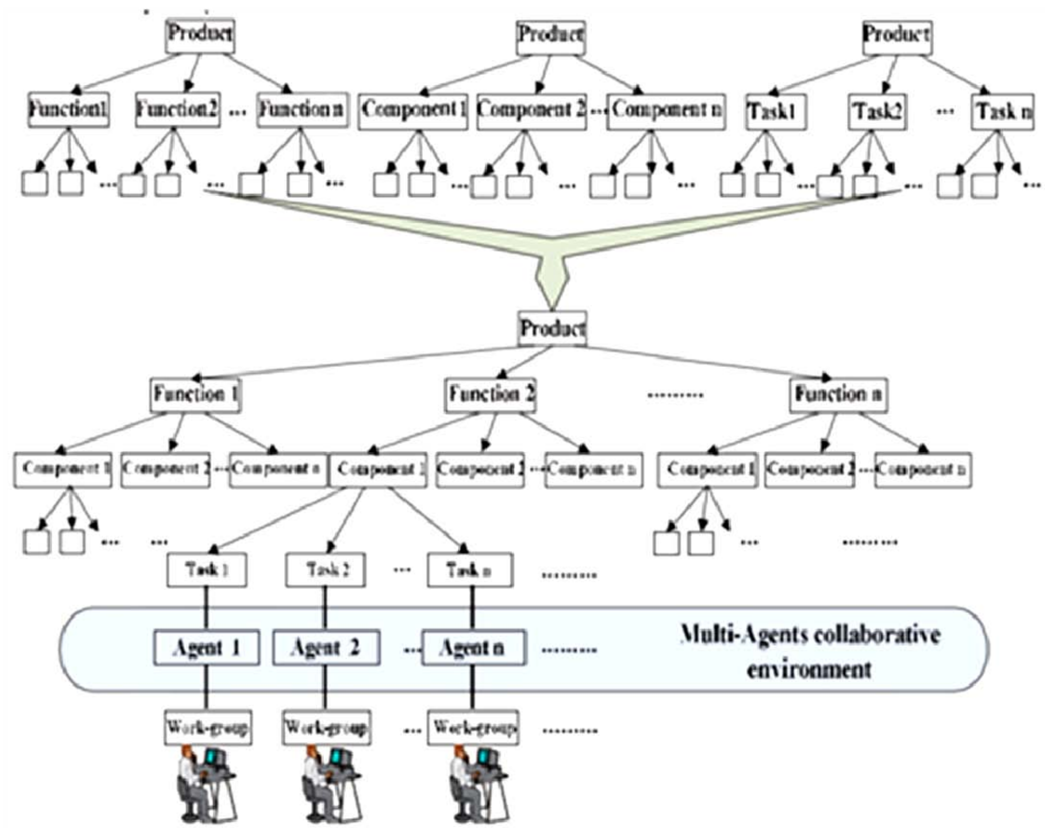
B-2. Modularity Strategy Based on Multi-Agents

Strategy 2: Modular Decomposition Based on Multi-Agent

- Collaborative product design regards each activity in the product development process as a *unity*.
- The whole process is managed and controlled from the perspective of *overall optimization*.
- The operational objective of collaborative design is the product *information structure*, which is linked by the *logical* and *temporal relation constraint* in the process.

B-2. Modularity Strategy Based on Multi-Agents

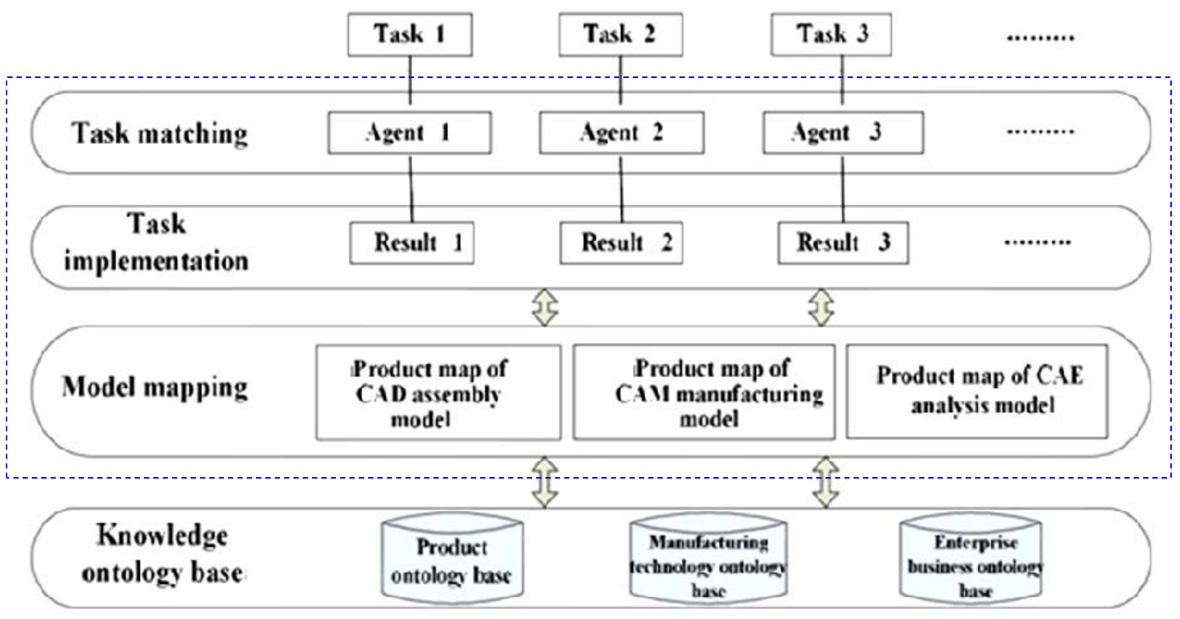
Modular Decomposition Based on Multi-Agent



Source: Ming et al., 2010.

B-3. Multi-Agents Collaborative Mechanism

Multi-Agent Blackboard Model

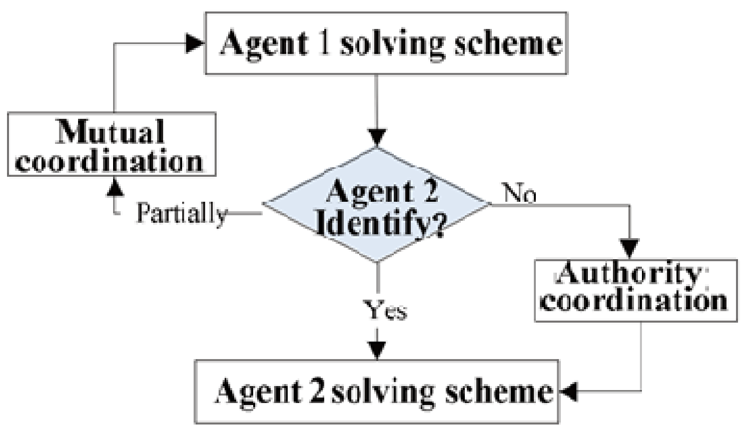


In the blackboard mechanism, all the agents are equal and take the tasks together through distributed blackboard.

Source: Ming et al., 2010.

B-3. Multi-Agents Collaborative Mechanism

Multi-Agent Negotiation Mechanism



Step 1: Agent 1 proposes a kind of solution by knowledge.

Step 2: Agent 2 decides whether to accept this proposal after consideration. If there are no objections, Agent 2 accepts the recommendations . Then, the two sides reach an agreement;

Step 3: If Agent 2 thinks the proposal from Agent 1 is different with its own and has no room for cooperation at all, it could reject the proposal and seek for other solutions like authority.

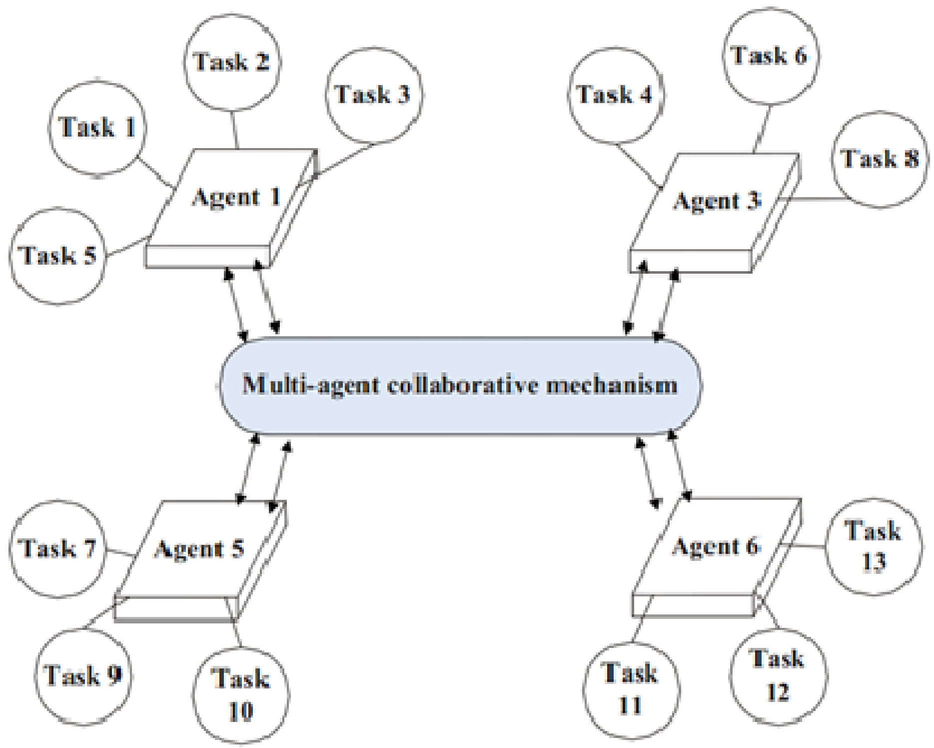
Step 4: If Agent 2 could partially accept the proposal from agent 1, it could put forward the solution about the inappropriate part in the proposal and ask Agent 1 for the advice.

Step 5: By negotiated for many times, the difference will be smaller and could be accepted by the two sides in the end.

Source: Ming et al., 2010.

B-3. Multi-Agents Collaborative Mechanism

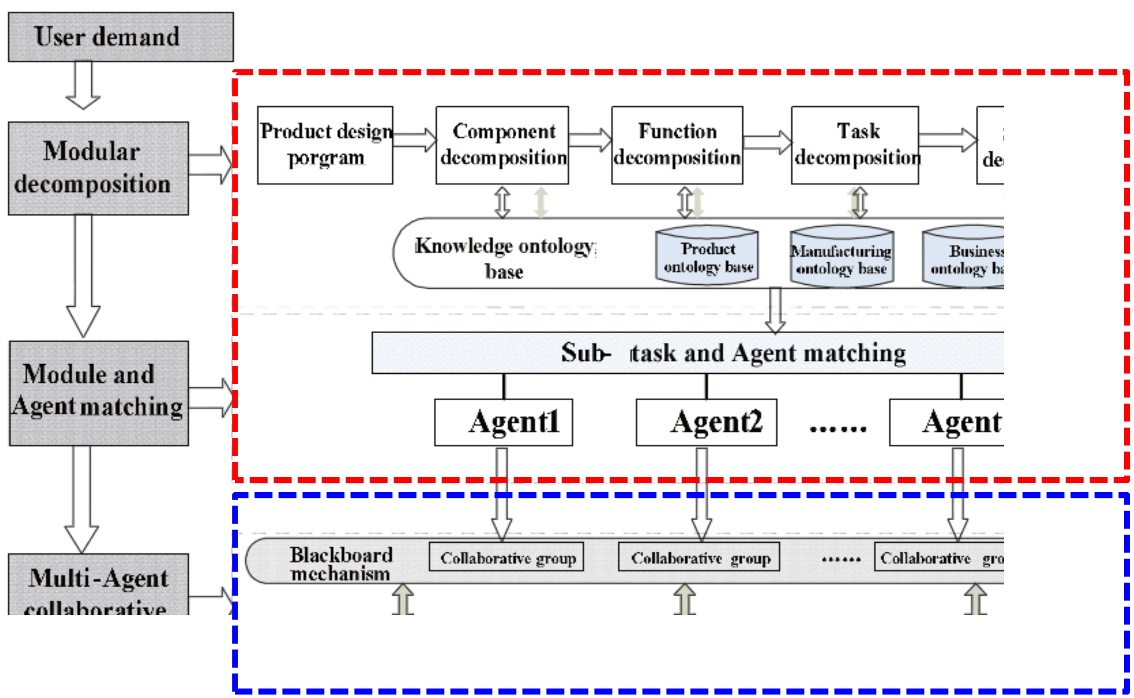
Solving Task In Multi-Agent Collaborative Design



Source: Ming et al., 2010.

B-3. Multi-Agents Collaborative Mechanism

Multi-Agent Collaborative Product Design Flow



Source: Ming et al., 2010.

Outline

1. Introduction: Today's CSCD application
2. History: CSCW, Collaborative Engineering, and CSCD
3. Basic Concept of CSCD
4. CSCD Methodology
5. CSCD Tools
6. CSCD Architecture
7. Web and Agent Technologies for Collaborative Design
8. **A Case Study:**
   Tracking of Design Changes In Mechanical Product
9. Current Application

8. Case Study:

Tracking of Design Changes in Mechanical Product Development

8.1. Background

- One of the consequences of the demand for *mass customization* of products has been the necessity for a team of designers to work together in a collaborative design environment.

- By exploiting each individual designer's expertise, the design team can generate design solutions quickly in response to customer requirements.

- By *analyzing conflicts* and *constraints* earlier from different perspectives, the design team is able to achieve *design objectives* through optimizing product performance, minimizing manufacturing costs, and ensuring that the product can easily and economically be serviced and maintained

Source: H. Xie, 2001.

8. Case Study:

Tracking of Design Changes in Mechanical Product Development

8.2. Problem

One of the issues in collaborative design is that one must assess the impacts of a design change on other design objects and notify other parties promptly:

- Due to *the lack of tracking of design changes* in most collaborative design environments, designers have to manually perform a consistency check for a proposed design change to identify all the impacts of the change.

- As a result, *design consistency and accuracy are not guaranteed* and design team productivity is compromised.

Source: H. Xie, 2001.

8.3. Approach to Tracking of Design Changes

One of the approaches to automatic tracking of design changes is an *expert system* approach.

- *Design change rules* can be stored in a knowledge base so that all the impacts of a design change can be retrieved through an *inference engine*.

- A generic mechanism for the tracking of design changes can be developed using the product data (i.e. *product specification*, *function decomposition structure*, *solution principles*, *layout design*, *assemblies*, and *parts*) driven approach.

Source: H. Xie, 2001.

8. Case Study:
Tracking of Design Changes in Mechanical Product Development

8.4. The Change Tracking Model

The tracking model consists of entities, attributes, and relationships.

A. Entities and Description

Entity	Description
Specification	Product specification generated from customer requirements.
Function	*An individual function composed of a* function structure.
Principle	A specific function implementation.
Design Object	A super class over entity "Assembly" and "Part". It describes common characteristics of assemblies and parts.
Assembly	A subclass of the entity "*Design Object".* It describes the specific characteristic of an assembly.
Part	A design object that cannot be further decomposed. A subclass of the entity *"Design Object". It describes the specific* characteristic of a part.

Source: H. Xie, 2001.

8. Case Study:
Tracking of Design Changes in Mechanical Product Development

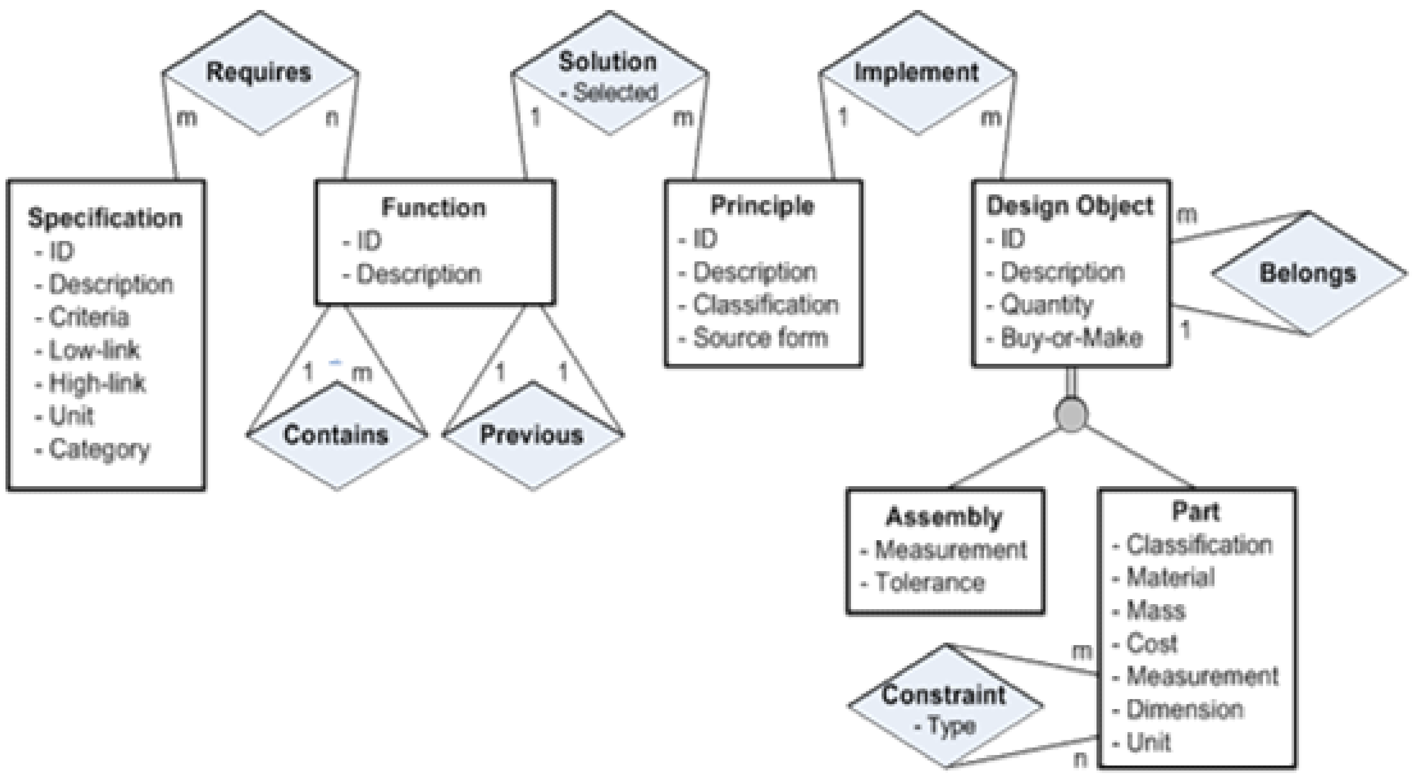
B. Relationships

- The *relationships* represent a set of associations among entities.
- The *relationships* limit the possible combinations of entity instances.
- Cardinality ratio constraints specify three common combinations for binary relationship types:
  - one-to-one (1:1)
  - one-to-many (1:M)
  - many-to-many (M:N).
- The relationships in the change tracking model include *Requires*, *Contains*, *Previous*, *Solution*, *Implement*, *Belongs*, *and Constraint.*

Source: H. Xie, 2001.

8. Case Study:
Tracking of Design Changes in Mechanical Product Development

C. The Associated Attributes in an Entity-Relationship Diagram



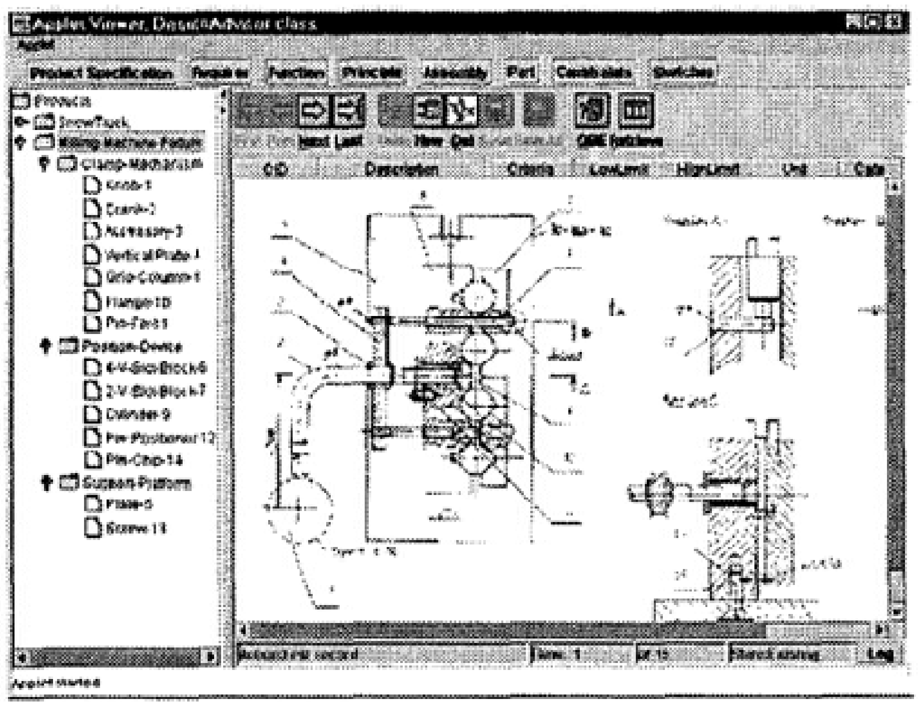
Source: H. Xie, 2001; B. K. Ane and D. Roller, 2010.

8. Case Study:

Tracking of Design Changes in Mechanical Product Development

D. Case 1: Milling machine fixture

- Tracking is done using a Web-based software called **Design Advisor.**
- Product information is shared from a back-end database through a Web server.
- The client-side user interface is developed on the basis of Java Applet:
  - It provides an enhanced functionality to display a product structure tree at different levels.
  - It also provides designers the ability to manipulate the product database.
- Database access used the JDBC API.
  - From the user interface, a designer is able to switch to different design views using the buttons on the toolbars.



Source: H. Xie, 2001.

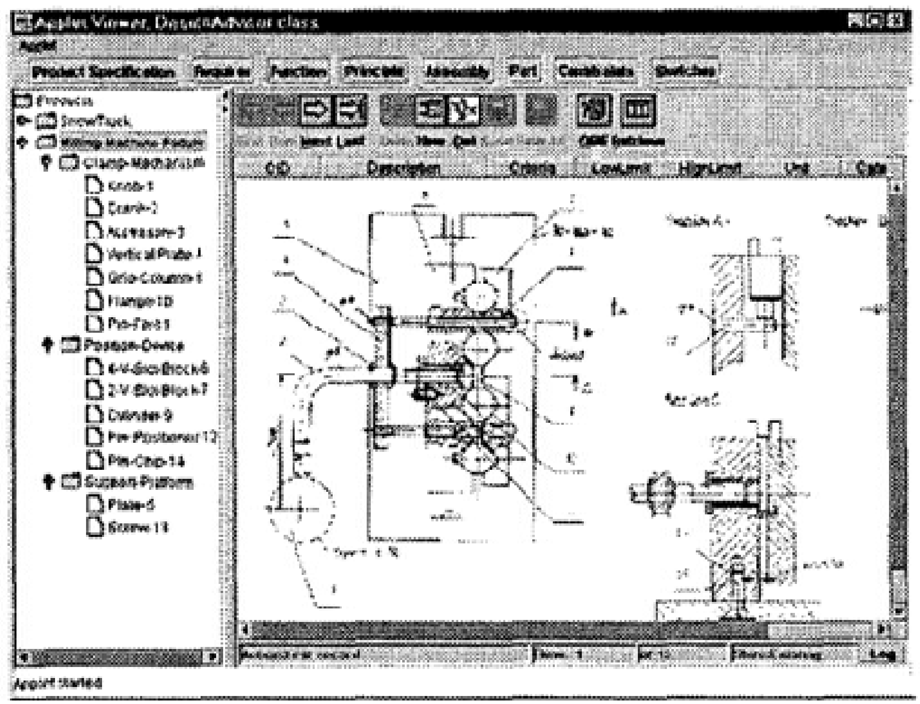
8. Case Study:

Tracking of Design Changes in Mechanical Product Development

D. Case 1: Milling machine fixture

The fixture is designed as three assemblies:

- The *clamp mechanism* consists of knob, crank, accessory, vertical plate, grid column, flange, and pin-fix.
- The *position device* consists of 4-V-slot block, 2-V-slot block, cylinder, pin positioner, and pin chip.
- The *support platform* consists of plate and screw.



Source: H. Xie, 2001.

8. Case Study:

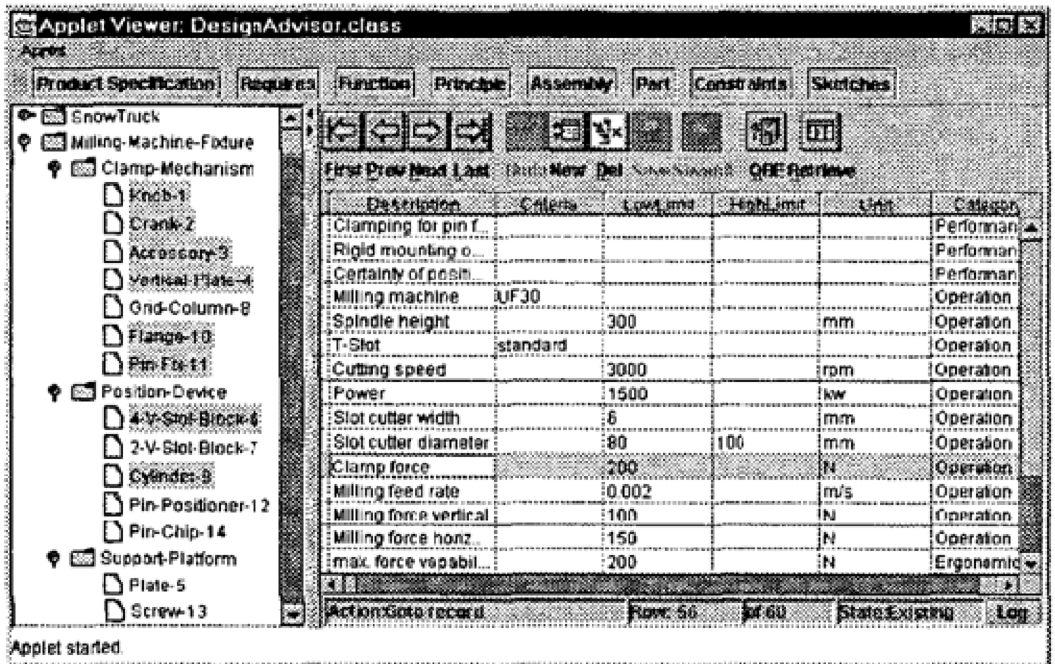
Tracking of Design Changes in Mechanical Product Development

D. Case 1: Milling machine fixture

Forward Tracking Mechanism

Suppose a clamping force in the specification has to be changed to meet a customer's requirement:

- The clamp force is related to function ***ForceAccept*** and ***ForceAmplify.***
- The function ***ForceAccept*** is implemented by principle ***Handle with x-axis rotation.***
- The function ***ForceAmplify*** is implemented by principle ***Screw/Nut.***
- The principle of ***Handle with X-axis rotation*** corresponds to Knob and Crank.



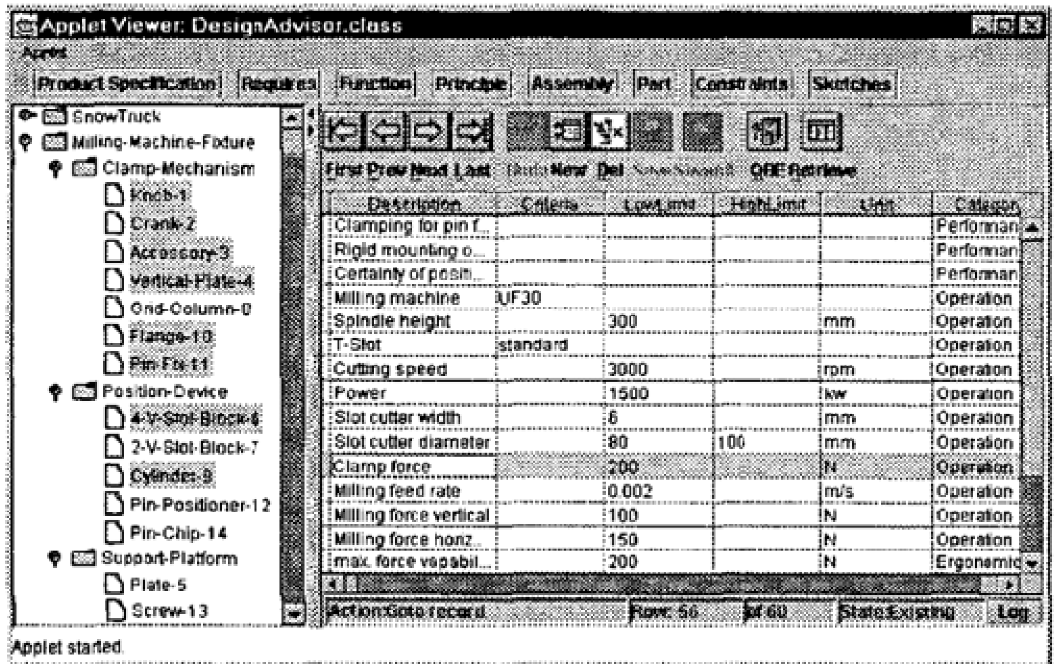
Source: H. Xie, 2001.

8. Case Study:

Tracking of Design Changes in Mechanical Product Development

D. Case 1: Milling machine fixture

- The principle of ***Screw/Nut*** corresponds to Cylinder and Flange-.
- The Knob, Crank, Cylinder, and Flange- are constrained by Accessory, Vertical plate, 4-V-slot-Block, and Pin-fix.
- Hence, If the clamp force is to be changed, then, Knob, Crank, Accessory, Vertical plate, 4-V-slot-block, Cylinder, Flange-, and Pinfix should also be changed .



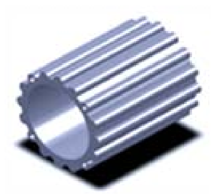
Source: H. Xie, 2001.

8. Case Study:

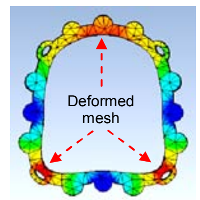
Tracking of Design Changes in Mechanical Product Development

E. Case 2: Motor Body

Initial Design | **Improved Design**

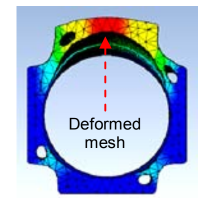


(*a*) cylinder shape



(*b*) estimated displacement

(*a*) block shape



(*b*) reduced displacement

Forward Tracking of Design Object Change

ENTITIES	ATTRIBUTES	STATUS	DESIGN VALUES	
			INITIAL	IMPROVED
Function	ID	No change	part#1	part#1
	Description	No change	Motor-body	Motor-body
Principle	Classification	No change	Motor protection	Motor protection
	Source Form	No change	Production house	Production house
Design_Object	Quantity	No change	100 pieces	100 pieces
	Buy-or-Make	No change	make	make
Specification	Criteria	No change	mechanic – static	mechanic – static
	Low-limit	Change	3.04976e-006 N/mm^2	3.43238e-007 N/mm^2
	High-limit	Change	0.000572322 N/mm^2	0.00139487 N/mm^2
	Unit	No change	1	1
	Category	No change	automotive part	automotive part
Part	Material	No change	alloy steel (SS)	alloy steel (SS)
	Mass	No change	min 1.750 – max 1.860 kg	min 1.750 – max 1.860 kg
	Cost	No change	USD 367.82 - USD 375.00	USD 367.82 - USD 375.00
	Measurement	No change	millimeter (mm)	millimeter (mm)
	Dimension	Change	d∅: 82.5mm, ℓ: 110 mm	w: 82.5 mm, h: 82.5 mm, ℓ: 110 mm

Source: B. K. Ane and D. Roller, 2010.

Outline

1. Introduction: Today's CSCD application
2. History: CSCW, Collaborative Engineering, and CSCD
3. Basic Concept of CSCD
4. CSCD Methodology
5. CSCD Tools
6. CSCD Architecture
7. The Web and Agent Technologies for Collaborative Design
8. A Case Study:
   Tracking of Design Changes In Mechanical Product
9. **Current Application**

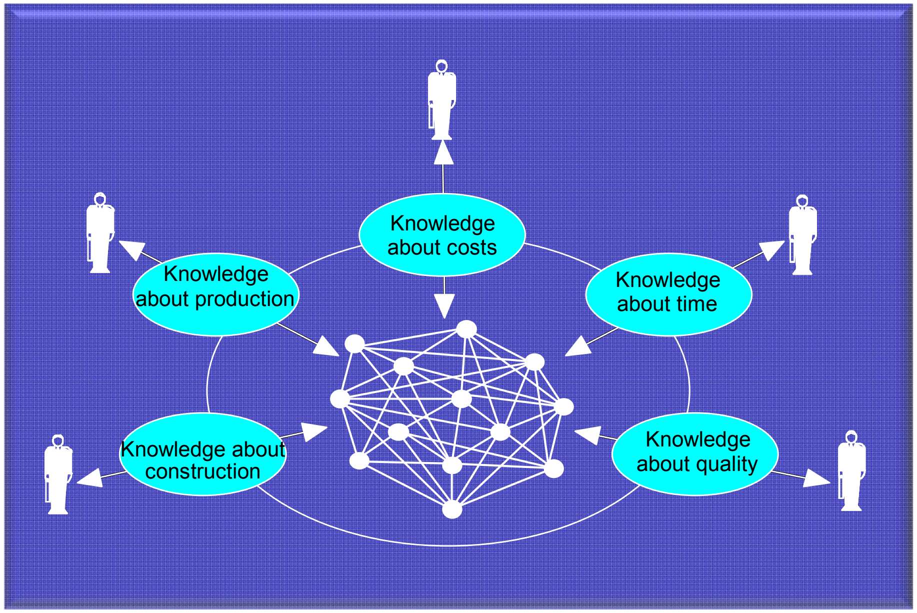
9. Current Application

An Active Knowledge Base for Supporting Product Development

Research institution : Institute of Computer-aided Product Developments Systems, Universität Stuttgart, Germany
Researcher : Prof. Dr. Dieter Roller

Goals

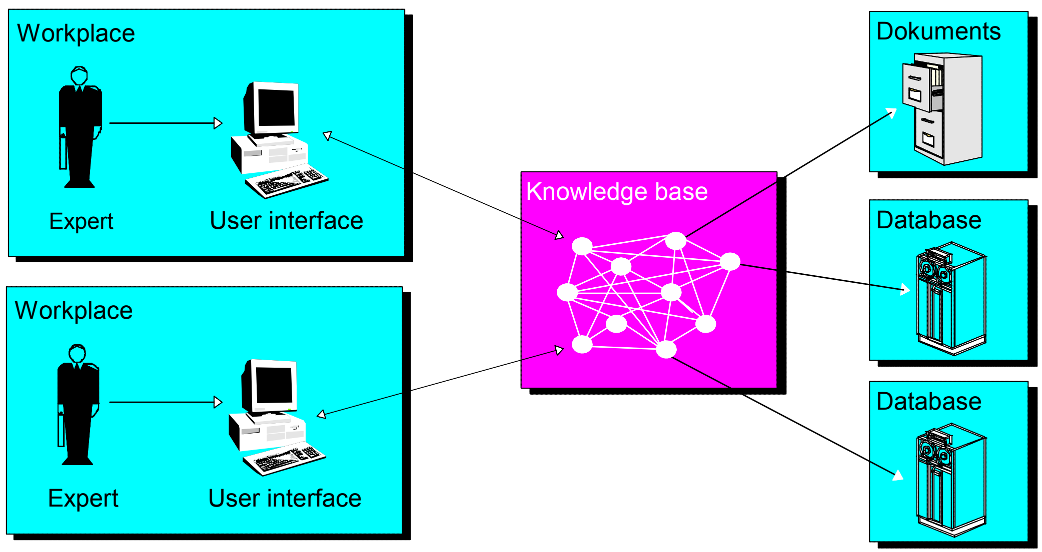
- Design of a knowledge base for the representation of all knowledge relevant to product development.
- Increase of the reusability of product data.
- Support of collaborative work.
- Support of distributed design teams.



Distributed Data Processing In the ASN

Access of distributed information in the Active Semantic Network (ASN):

- Integration of isolated information resources.
- Use of modern means of communication for "human information resources„.
- Support of the flow of information among designers.



9. Current Application: An Active Knowledge Base for Product Development

Definition of the Active Semantic Network

The Active Semantic Network (ASN) is a network of nodes and links

- The nodes represent objects of the real world
- The links represent relations between these objects

The ASN is able to handle and manage the increasing amount of knowledge during the product development process.

The active component is able to

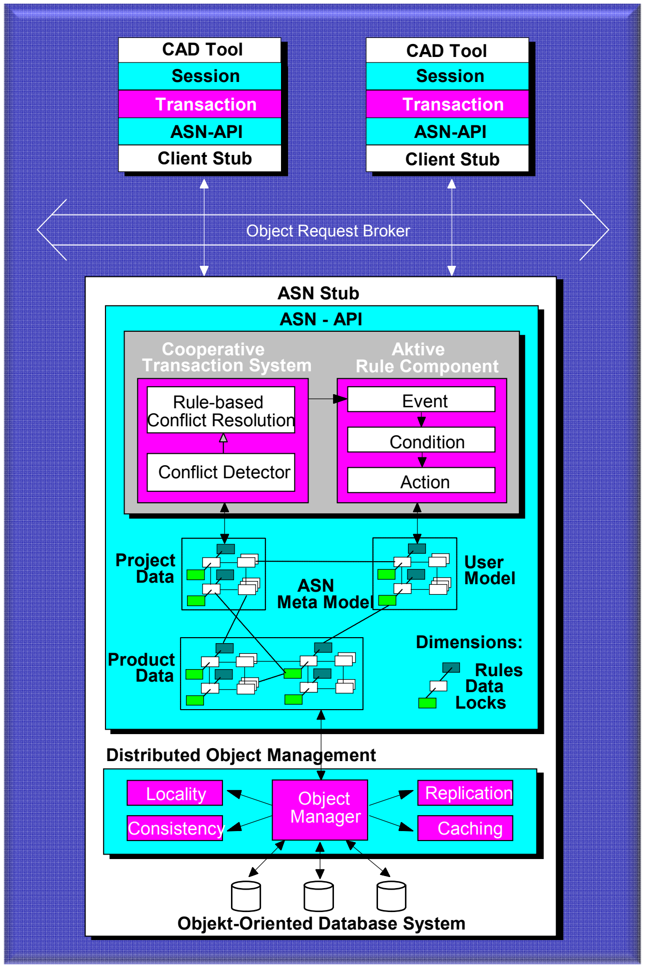
- make inferences
- start external actions (e.g., at a graphical user interface)

9. Current Application: An Active Knowledge Base for Product Development

Architecture of the Active Semantic Network

Realization of the knowledge base

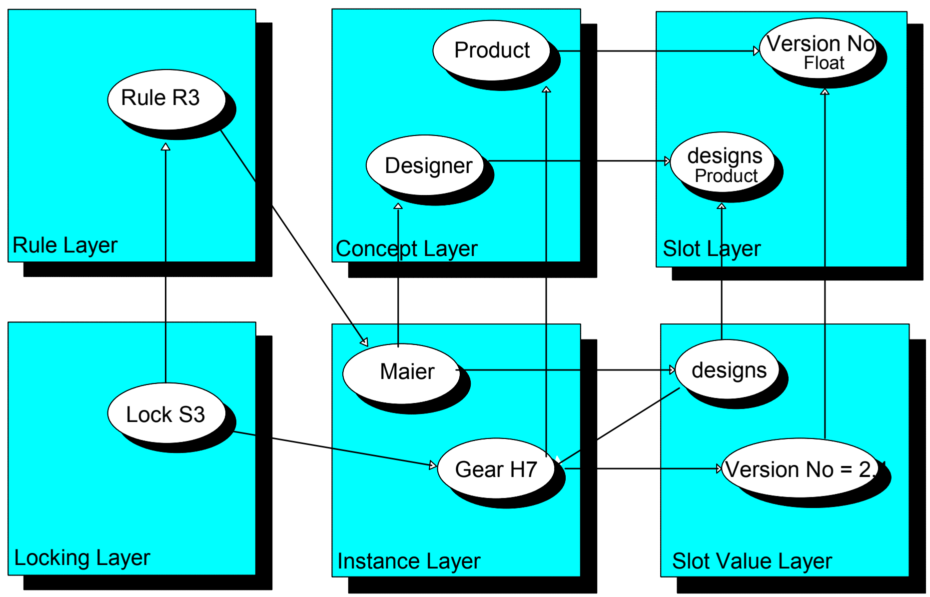
- Use of an object-oriented database system.
- Representation structures defined by meta models.
- Definition of a database-independent programming interface.
- Integration of ECA-Rules of active database systems.
- Interpretative rule processing.
- Cooperative transaction system.
- Distributed data access by RPC and CORBA technology.



Representation Structures of the ASN

The ASN meta model

- Specification of the ASN representation structures
- Representation of entities as instances of meta classes
- Representation of objects as triples: data, rules, locks
- Schema evolution at runtime

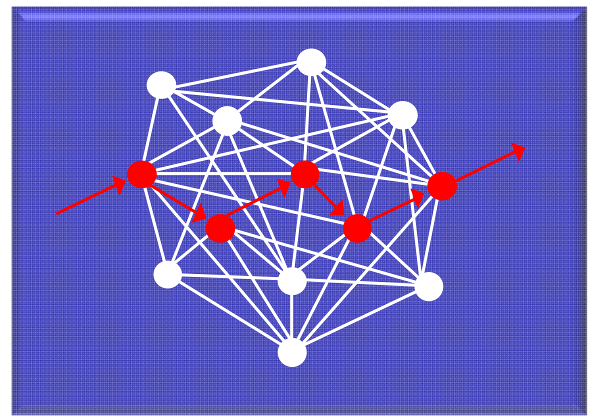


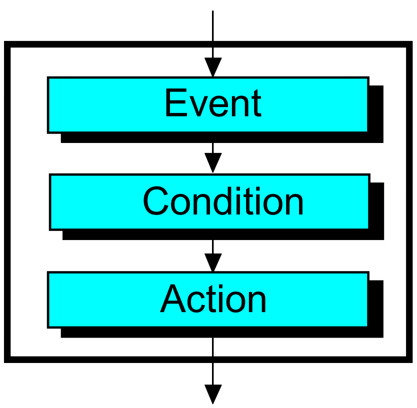
9. Current Application: An Active Knowledge Base for Product Development

The ASN Rule Component

Goals of the active rule component

- Inferences
- Modelling of integrity constraints
- Constraint Propagation
- Automatic information delivery





Use of ECA rules

- Event : An indicator that triggers a rule
- Condition : A collection of database queries
- Action : Actions executed when the event is triggered and the condition is satisfied

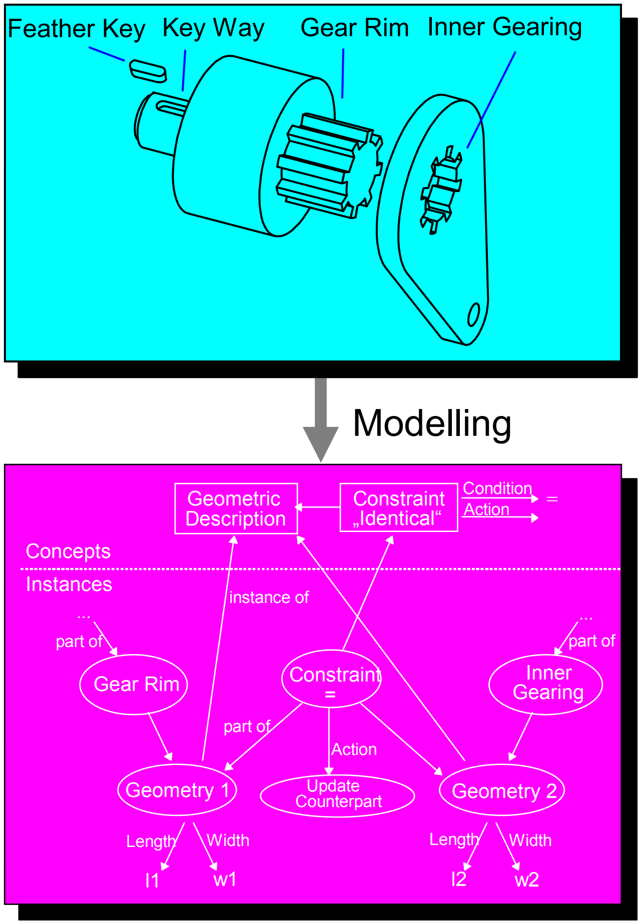
Constraint Propagation in the ASN

Constraint Representation

- Unique object-oriented representation of data and constraints.
- ECA-Rules as attributes of active objects.

Constraint Propagation

- Rule-based, interpretative evaluation of constraints.
- "Constraint Based Cooperative Design".



Knowledge Representation in the ASN

Modelling of product data in the ASN

- Representation of the semantics of data
- Increase of the reusability of product data
- Integration of application fields such as
  - Knowledge about *construction*
  - Knowledge about *production*
  - Knowledge about *costs*
  - Knowledge about *quality*
- Incomplete, prototypical, and dynamically increasing knowledge
- Integration of knowledge representation approaches
- Information access by providing means of communication to experts

Modelling Within the ASN

Access to ASN models

- Modelling language KML
- ODMG 2.0 - standard
- Conversion to STEP
- Database independent programming interface

Modelling language KML

```
Design_Object: Concept
PARTS:
  Identifier:     Text
  Construction:   Construction_Drawing
  Costs:          Costs
  Quality:        Quality_Information
```

Data query in OQL

```
select m.Identifier
from Design_Object m
where m.costs.target_costs > 10000
order by m.Identifier asc
```

9. Current Application: An Active Knowledge Base for Product Development

Support of Collaborative Work

Characterization of "CAD Transactions":

- Long transactions
- Interactive user control
- Cooperation among designers

Characterization of traditional transaction systems:

- ACID properties:
  atomicity, consistency, isolation, durability
- Serializability
- Rollback in case of failures

ASN transaction model:

- Rule-based concurrency control
- Cooperative transactions
- Extended locking table
- Interactive rollback

References

A. K. Kamrani and E. A. Nasr, "Collaborative Engineering: Theory and Practice," Springer Science+Business Media, New York, 2008.

B. K. Ane and D. Roller, "An Architecture of Adaptive Product Data Communication System for Collaborative Design," In: Proceedings of the Fourth International Conference on Advances in Computer-Human Interactions (ACHI), 2011, pp. 182-187.

C. Ming and L. Jinfei, and K. Qinghua, "Study on Collaborative Product Design Based on Modularity and Multi-agent," In: Proceedings of the Fourth International Conference on Computer Supported Cooperative Work In Design, 2010, pp. 274-279.

D. Roller, "An Active Knowledge Base for Supporting Product Development," Research Report and Presentation, Institute of Computer-aided Product Development Systems, Universität Stuttgart, Stuttgart, Germany.

D. Saffer, " The Role of Metaphor in Interaction Design," presented in the Information Architecture Summit, 2005.

D. Sriram, "Computer Supported Collaborative Engineering," AAAI Technical Report WS-93-07, 1993.

Daimler AG., "HTR Daimler HighTech Report: Designs For Tomorrow," Issue 01, 2008.

Dix, A., "Computer Supported Cooperative Work: A Framework," In: Rosenberg, D. and Hutchinson, C. (eds.), Design Issues in CSCW, Springer-Verlag, London, 1994. pp. 9-26.

C. A., Ellis, S. J. Gibbs, and G. L., "Groupware: Some issues and experiences," Communications of the ACM vol. 34(1), 1991, pp. 38-58.

E. Sprow, "Chrysler's Concurrent Engineering Challenge," Manufacturing Engineering, vol. 108, issue 4, 1992, pp. 35-42.

H. Xie, "Tracking of Design Changes for Collaborative Product Development," Computer Supported Cooperative Work In Design, 2010, pp. 175-180.

H.-H. Wang and W.-J. Liao, „Applications of Metaphor Theory to Product Design," In: Proceedings of the International Association of Societies of Design Research (IASDR), 2009.

References

IBM, ”IBM Software - IBM Lotus Notes Home Page,” http://www-142.ibm.com/software/sw-lotus/products/ product4.nsf/ wdocs/noteshomepage, 2006 (Accessed 02/10/2011).

J. Barwise, “The Situation In Logic,” CSLI Lecture Notes, CSLI Publications, Stanford, 1989.

J. Barwise and J. Perry, “Situation and Attitudes,”, MIT Press, 1983.

J. Grudin, “Computer-Supported Cooperative Work: History and Focus,” Journal of Computer vol. 27(5), 1994.

J. Grudin, “Interactive Systems: Bridging the Gaps Between Developers and Users,” IEEE Computer vol. 24(4), 1991, pp. 59-69.

K. Ehrlich and D. Cash, “Turning Information into Knowledge: Information Finding as a Collaborative Activity,” In: Proceedings of the Digital Libraries Workshop - DL’94, Springer-Verlag, Heidelberg, 1994.

K. Ulrich and D. Eppinger, “Product Design and Development,” McGraw-Hill, Boston, 2004.

M. A. Schnabel, “Architectural Design In Virtual Environments,” Doctoral dissertation, Department of Architecture, The University of Hongkong, 2004.

R. Johansen, “Groupware: Computer Support for Business Teams,” The Free Press, 1988.

T. Gross, “Computer-Supported Cooperative Work: Concepts and Tools,” Computer-Supported Cooperative Work Group, Faculty of Media, Bauhaus-University Weimar, Germany.

W. Shen, Q. Hao, and W. Li, “Computer Supported Collaborative Design: Retrospective and Perspective,” Computers in Industry vol. 59, 2008, pp. 855-862.

W. Xuan, S. Xue, and T. Ma., “Architecture of Collaborative Design Based on Grid,” In: Proceedings of the Third International Conference on Genetic and Evolutionary Computing, 2009, pp. 244-247.

X. F. Zha and H. Du, “Knowledge-Intensive Collaborative Design Modelling and Support – Part I: Review, Distributed Models and Frameworks,” Computers In Industry vol. 57, 2006, pp. 39-55.

Z. Li, X. Jin, Y. Cao, X. Zhang, and Y. Li, “Architecture of Collaborative Design Grid and Its Application Based on LAN,” Advances In Engineering Software vol. 38, 2007, pp. 121-132.

