



# Improving collaborative work and project management in a nuclear power plant design team: A Human-Centered Design approach

Guy Andre Boy, Gopal Jani, Annalisa Manera, Matthew Memmott, Bojan Petrovic, Yassine Rayad, Lucas Stephane, Neha Suri

## ► To cite this version:

Guy Andre Boy, Gopal Jani, Annalisa Manera, Matthew Memmott, Bojan Petrovic, et al.. Improving collaborative work and project management in a nuclear power plant design team: A Human-Centered Design approach. *Annals of Nuclear Energy*, 2016, 94, pp.555-565. 10.1016/j.anucene.2015.12.039 . hal-02956574

**HAL Id: hal-02956574**

**<https://hal.science/hal-02956574>**

Submitted on 2 Oct 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Improving collaborative work and project management in a nuclear power plant design team: A Human-Centered Design approach

Guy André Boy<sup>1</sup>, Gopal Jani,  
Annalisa Manera, Matthew  
Memmott, Bojan Petrovic, Yassine  
Rayad, Lucas Stephane and Neha  
Suri

## ABSTRACT

This article presents a collaborative system, called SCORE, useful for a multi-disciplinary team designing a new nuclear power plant (NPP). It was developed during the first phase of the I<sup>2</sup>S-LWR project (Integral Inherently Safe Light Water Reactor). SCORE enables the generation of design cards (DCs). A DC includes four main spaces (Boy, 2005): (1) a rationalization space where the various components of the system being designed (SBD) are described in terms of design rationale, integration and requirements; this space includes declarative and procedural descriptions and statements; (2) an activity space where the current version of the SBD is displayed; it includes static and dynamic features; this space enables SBD manipulation; (3) a structure space where the various components and their inter-relations are formally and declaratively described as systems of systems; (4) a function space where the various functions of the SBD are described in terms of procedural knowledge and dynamic processes involved; this space includes qualitative and quantitative physical and cognitive models. The rationalization space is informed using an adapted version of the QOC method (Questions, Options, Criteria), which was tested within the I<sup>2</sup>S-LWR design team. The activity space contains 3D models developed using AutoDesk Inventor, and transferred into the Unity game engine web player in order to facilitate integration within the DC spaces and enable intuitive manipulation of objects in the activity space. Two additional spaces were added: an instant messaging capability that allows design team members (DTMs) to exchange with one another on a DC; and a structured evaluation space. DCs are cooperatively created and refined by DTMs, and synthesized during periodic design meetings, the frequency of which may vary. Incrementally combining abstract explanations of designed elements and integration with their explicit visual representation improves mutual understanding among DTMs, and consolidates design decisions. This human-centered design (HCD) approach also improves individual and collective familiarization with the complexity of mixing several expert contributions in NPP design. In this case, HCD is not focused on end-users, but rather on the designers themselves.

## KEYWORDS

Human-Centered Design; Computer-Supported Cooperative Work; Knowledge Management; Modeling and Simulation; Design Rationale; Project Management; Nuclear Power Plant.

---

<sup>1</sup> Human—Centered Design Institute, Florida Institute of Technology, 150 W. University Blvd., Melbourne, FL 32901, USA. Phone: 1 (321) 674-7631. Fax: 1 (321) 674-7175, gboy@fit.edu.

## 1. Introduction

Human factors and ergonomics (HFE) methods have been extensively used for the evaluation of systems already developed. In contrast, this article presents a human-centered design approach (HCD) that takes into account designers' human factors during the design process. The approach is then called **HCD for designers**. Designers need to collaborate to elicit, assemble and integrate complex design concepts as well as workable objects. They usually manipulate and articulate interrelated abstract concepts and concrete systems, which they need to visualize. In other words, designers need to incrementally conceptualize systems being designed.

This work started during the first design phase of the **I<sup>2</sup>S-LWR project** (Integral Inherently Safe Light Water Reactor). We quickly became aware of some human factors issues emerging from a design team of 35 people having different backgrounds. Project management is not simply a matter of designing and using a linear Gantt chart, where tasks are well assigned. In creative projects, such as I<sup>2</sup>S-LWR, things are more non-linear. Ideas and viewpoints need to be incrementally synthesized and integrated. There are intertwined periods of creativity and phases of rationalization. Software engineering already practices agile development. The collaborative approach that we propose in this article is strongly based on agile systems engineering. Essentially, “**agile**” means that at each step during the design process, punctuated by a design review meeting, a version of the product is established and is the starting point of a new step.

Design team members (DTMs) collaborating through various kinds of media need to understand each other. This is why it is required to have a common **frame of reference (CFR)** (i.e., a kind of music theory and overall score prescribing the symphony). This CFR is incrementally developed. CFR is also a product in itself. This article describes a CFR support system that we have called SCORE. SCORE is a mediating tool that enables design team members to generate, modify, refine and validate both physical and figurative tangibility of the complex system being designed (e.g., the I<sup>2</sup>S-LWR nuclear reactor).

Collaborative work has been extensively studied and supported in the Computer Supported Cooperative Work (CSCW) community. **CSCW** is used to address “research into experimental systems and into the nature of workplaces and organizations” (Grudin, 1994). Computing is used to mediate human-human interaction. Team members may interact with one another at a same or different time, and in a same or different location. There are then four cases that can be supported, combining time and space. CSCW focuses on meeting environments (e.g., interactive walls and tables, and more generally visualization systems), written and video media systems (e.g., Skype), social media systems (e.g., Facebook) and knowledge management systems (e.g., Cmaps). Our goal was to combine these CSCW capabilities for the sake of the design of a nuclear power plant (NPP), such as I<sup>2</sup>S-LWR.

**Visualization** of concrete physical objects is an important part of collaboration support to share situation awareness (i.e., design team members should understand the same things). It is therefore crucial to better understand and integrate appropriate modeling and simulation (M&S) tools into CSCW means. Commercial M&S software, such as CATIA or Inventor, can be used to generate useful sources for populating the collaborative tool being currently designed. Gaming engines, such as Unity, can support dynamic and interactive attributes. Combining these tools contributes to sharing meaning among design team members (i.e., shared situation awareness).

Design is also about abstract things that need to be represented, shared and used. The field of **knowledge management (KM)** has already produced tools and methods to this end (Tuomi, 1999). KM is an important part of human-centered design, and more specifically **participatory design** (Muller, 2003). It enables design rationale capture and provides design team members with design history traceability at any time. A design team is usually a (relatively small) group of experts, from various backgrounds. They work in close cooperation, but not necessarily in the same location. They fulfill different roles, and strongly need to continuously communicate. A leader often coordinates the work of the design team. A design team is usually supported by technologies such as instant messaging; virtual team workspaces where information is

managed (e.g., Dropbox); video conferencing; data conferencing; shared calendars; and discussions through a web browser (e.g., Skype, Google Hangout, VSee).

In this article, we present a collaborative tool, called SCORE, which enables a design team to create, refine and share design knowledge while designing a life-critical system such as a NPP. This tool acknowledges and supports the articulation of the duality between physical components and related abstract concepts that improve perception, comprehension and projection of the complex system being designed. Since systems are now designed as pieces of software from the beginning, using modeling and simulation tools, their functions can be tested very early during the design process. One of the main issues is their **tangibility**. The notion of tangibility is at the center of this human-centered design approach. Tangibility is taken in the physical sense (i.e., a physical object is graspable, and then tangible) and the figurative sense (i.e., a abstract concept is credible, and then tangible). It helps design team members to incrementally figure out and better understand components, interconnections and the whole concept of the system being designed. Therefore, collaborative HCD supports the design of tangible interactive systems (TISs), where structures and functions are incrementally articulated (Boy, to appear). The SFAC model (structure-function-abstract-concrete) is proposed to this end. SCORE is based on the creation and progressive refinement of design cards (DCs) that are shared by design team members. Simplified examples of the use of SCORE are given for the design of the I<sup>2</sup>S-LWR nuclear reactor.

## 2. System design is also knowledge design: The rationalization space

A life-critical system (LCS), such as a nuclear power plant (NPP), has specific properties that directly impact the way knowledge management is carried out. A LCS is a system where loss of life could result from non-normal operations. A NPP is a complex system, which needs to be described as completely as possible, using most mature domain knowledge. Control and management of a LCS typically involves three main objectives and issues: safety, efficiency/effectiveness and comfort. These issues need to be addressed at three levels: technology, organization and people.

Table 1. Life-critical system's attributes

|               | Safety  | Efficiency and Effectiveness                 | Comfort     |
|---------------|---|--|-------------|
| Technology    | Redundancy, Resilience, Socio-cognitive stability | Observability, Controllability, Traceability | Usability   |
| Organizations | Safety Culture                                    | Coordination, Communication                  | Cooperation |
| People        | Competence, Skills, Knowledge                     | Training, Experience                         | Education   |

NPPs fall into the category of life-critical system of systems. Complexity needs to be handled at various levels in such systems. First, product complexity requires a great deal of identification and articulation of LCS's technological components. Second, since LCS design requires multiple background and competencies, design team complexity analogously requires a great deal of identification and articulation of design team member's roles, contexts of validity (of these roles), and resources (required to accomplish these roles). Third, organizational complexity needs to be taken into account, and we will propose a model to achieve this. The tool being proposed in this article helps handling these complexity issues.

### *Design knowledge and knowledge design*

Designs are incrementally verified and validated (i.e., formative evaluations). In the end of the design and development process, certification by different authorities and certification bodies is typically carried out (i.e., summative evaluations). In order to get a system certified, one has to be able to justify the choices that were made, to prove, as much as possible, that extensive amount of theoretical and operational domain knowledge has been taken into account, and that the system will function safely in a large variety of difficult and even disastrous scenarios. **Complex system design concurrently involves design of related knowledge.** Typically, systems being developed have to be validated with respect to current regulatory principles and rules. However, when new systems are being designed and further developed, it is required to also develop additional regulatory principles and rules that accredited authorities need to certify. Knowledge design mirrors the design of the system that it supports. The system being designed and knowledge that supports it influence each other. Several iterations are necessary to reach acceptable maturity. Such maturity can be interpreted as maturity of the technology being developed, and maturity of the underlying principles defining and governing this technology.

Up to now, descriptions of **design knowledge** used and generated during the **design process** are stored in documents (Boy & Barnard, 2003). These design documents are incrementally finalized and approved. Designers have to deal with draft documents, addressing questions such as how one can recognize that a document is on-going, how versioning is taken into account, and how revisions are managed. The validation of a document is related to the appropriate list of signatures. When a document is validated it becomes “official”. Each design rationale description should be appropriately contextualized including its status (i.e., mainly the revision and approval dates) and background information (where it emanates from and who issued it). In order to follow appropriate guidelines to edit and publish such a document, training may be necessary and guidelines should be easily available. From a broader standpoint, our investigations led to the distinction between private and public spaces of a document, i.e., each technical document has a private space where it is invisible outside of a specific community, and a public space where it is visible by a wider community.

### *Human-centered dimensions for knowledge design*

Therefore, even if we would like to rationalize safety (i.e., safety-critical technology and organization) in a systemic sense, the field remains a matter of people. They are of course human operators (i.e., users of safety-critical technology), but also designers, manufacturers, maintainers, certifiers, trainers and (not to forget) managers. Most human factors approaches attempt to correct and adapt engineering systems to users’ needs after these systems are developed; whereas *Human-Centered Design* (HCD) takes into account people from the very beginning of design (Boy, 2011). HCD must consider not only the system components, but also interactions among human and machine agents. A proper analysis will holistically take into account five entities and their interrelations:

- the Artifact being designed (i.e., technology that is being designed);
- possible Users or human operators (i.e., a categorization of user profiles is necessary).
- the various Tasks that are anticipated (i.e., inputs of the various cognitive functions that the various agents will have to use);
- the Organization in which users will perform tasks using the artifact (i.e., typically a set of human and machine agents);
- the various Situations (i.e., various kinds of context patterns that characterize the environment).

The **AUTOS<sup>2</sup> pyramid** framework describes the various interrelations among the above five entities (Boy, 2011). It addresses the operational basis of resilience engineering (Hollnagel et al., 2006) as well as technology and practices' complexity, by a combined study of complementary perspectives of human-system integration. These interrelated perspectives should be understood as the framework to support completeness in *Human-Machine System* (HMS) analysis, design and evaluation. This approach has been successfully applied in aviation in order to determine that airline pilots rely on skills and knowledge over procedures (Boy & deBrito, 2000). Indeed, *Cognitive Function Analysis* (CFA) and the AUTOS pyramid were successfully used as an appropriate approach to organize experience feedback while designing cockpits that improve situational awareness (Boy and Ferro, 2003).

### *Capturing design rationale*

HCD goes from purpose to means. Therefore, the first thing to do is to get an initial purpose of what we want to achieve, knowing that it will evolve with the product being incrementally developed and used. **Design team members (DTMs)** need to work on meaningful things. Meaning comes from both elicitation of design rationale (setting up tasks) and formative evaluations (figuring out activities). This is not only important for developing a great product, but also create and keep a soul in the project. Indeed, DTMs need to clearly perceive and understand what they are doing and why, and they need to project themselves in the future to "see" the product they are developing at use.

### *Documenting the design process and its solutions*

HCD for designers promotes documenting design work incrementally during the design process and not only when product manufacturing is completed. An example of such a tool is the Computer Integrated Documentation (CID) system developed at NASA (Boy, 1991). Another example can be found in the IMAT (Integrating Manuals and Training) system developed for designing learning material (de Hoog et al., 2002). Also in the WISE (Web-enabled Information Services for Engineering) workspace, the engineer is enabled to make annotations to all different kinds of knowledge objects and choose whether to share them with DTMs or other relevant stakeholders (Boy & Bernard, 2003). Crisp and clearly understood design rationale is a good indicator of design maturity of the final product. Formalisms have been developed to describe design rationale such as gIBIS (*graphical Issue-Based Information System*) (Conklin & Begeman, 1989) or QOC. QOC stands for Questions, Options and Criteria (MacLean et al., 1991). They support the elicitation of design rationale and enable the documentation of design decisions, development plans and systems that are effectively developed.

We are using QOC to capture design knowledge. QOC is a semiformal notation that enables a design team to represent the design space around the system being developed. Questions are key design issues. Options provide possible answers to the Questions. Criteria guide the assessment and comparison of the Options. MacLean and his colleagues defined QOC to develop the Design Space Analysis approach, which enables the documentation of design rationale including possible alternative designs. QOC notation is very useful later on when we need to understand design history and rationale (Moran & Carroll, 1996). Design space analysis also enables design team members to take into account consistency, models and analogies, and relevant data and theory. Indeed, there are design decisions that are based on experience and expertise, and others on theoretical constructs (e.g., on phenomena described by equations in the nuclear domain). QOC enables the documentation of explanations, argumentations and decisions during the design process.

In practice, design rationale documentation is done using graphical tools such as Cmaps (Concept Maps), which enable the definition of concepts and their inter-relations (Novak & Cañas, 2006). There are of course other knowledge management tools, such as Protégé, which enable design and refinement of ontologies that define concepts in a domain and their relationships. (Gennari, Musen et al., 2003). We choose CmapTools for their simplicity of use. CmapTools provides features for construction and navigation

---

<sup>2</sup> AUTOS is defined as *Artifact, User, Task, Organization and Situation*.



through different concept maps. These maps can be shared enabling critical thinking of the knowledge model emerging from the concept maps. A Cmap is typically constructed with the intent to answer a specific question. It is used for modeling “tacit” knowledge captured from experts. Experts know many things that they often cannot express in layman’s terms. For example, designers can construct their knowledge models using CmapTools on their local computer, which can then be shared with anyone using Cmap servers. Designers can also create links between different Cmaps and have web pages created based on the model. They can also edit their Cmaps at the same time. An example of QOC design rationale represented using a Cmap is provided in Figure 1.

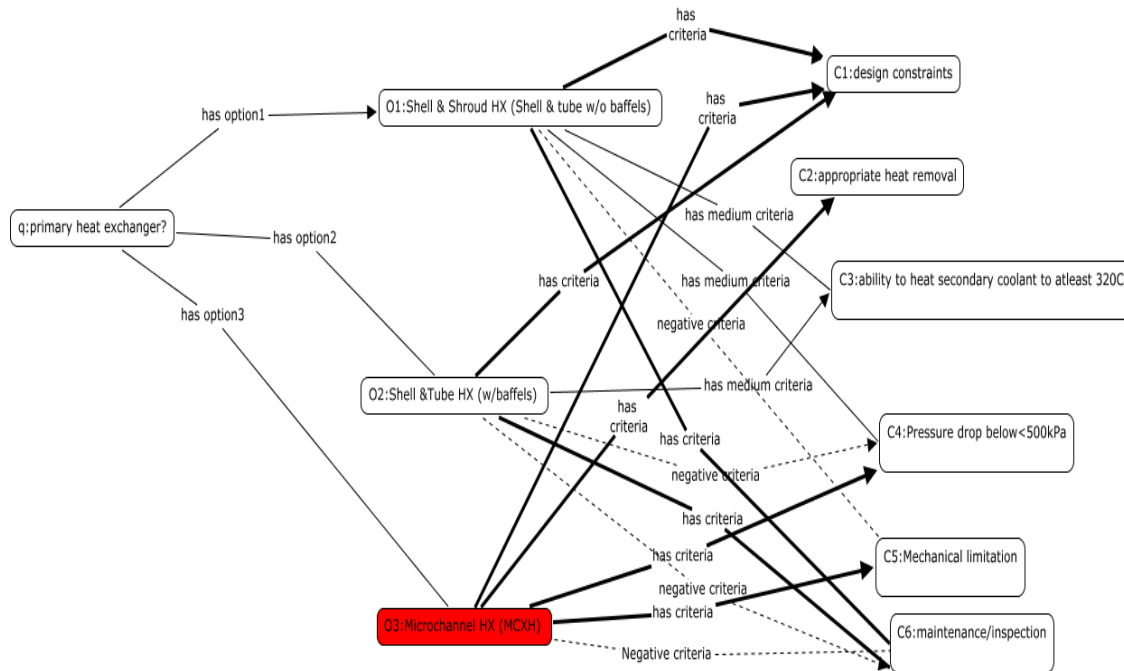


Figure 1. I²S-LWR primary heat exchanger QOC rationalization (i.e., design rationale).

### 3. Visualization and manipulation of physical objects: The activity space

NPP components and structural details are important and should be represented concretely and accurately in the activity space. Interactive 3D visualization provides more insight, since components can be easily manipulated as (visual) virtual objects. This improves designers’ awareness and decision-making.

#### *Modeling 3D static objects*

In addition to mainstream traditional engineering that focuses on delivering technologically-working products, HCD focuses on engaging DTMs and creating value. Designing static objects, such as an architect would do, is the first step to capture subject matter experts’ experience (i.e., user’s activity). There are many computer-aided design (CAD) tools that enable designers to produce virtual 3D static objects, such as Dassault Systèmes’s CATIA and AutoDesk Inventor. Visualization of concrete objects provides intuitive anticipation of possible activities (i.e., domain experience) and impacts on the real world (i.e., emergent properties).

Easy manipulation of 3D static objects provides DTM with capability of testing configurations, also called declarative scenarios. They can construct and deconstruct objects, as well as assemble them with one another. For example, a nuclear reactor 3D model, visualized on a display, can be decomposed into components, such as core assemblies and geometrical dimensions of rods<sup>3</sup> (Figure 2). This is made possible through the link between visually tangible objects and their formal representations in the corresponding rationalization space. Typically, a DTM selects a visually tangible object and obtains a description pop-up window that enables selection of sub-components or attributes.

This kind of modeling capability provides them with endless trial-and-error possibilities. In addition, when this graphical capability (activity space) is connected with a rationalization space, it tremendously increases the production of meaningful tangible interactive objects (Boy, 2014). This graphical tool can be implemented as an annotation mechanism on top of the 3D-static-objects visualization. This type of feature provides meaningful interactivity with the objects being designed, and enables traceability among the various versions of these objects as well as connectivity among the various components of the systems being designed.

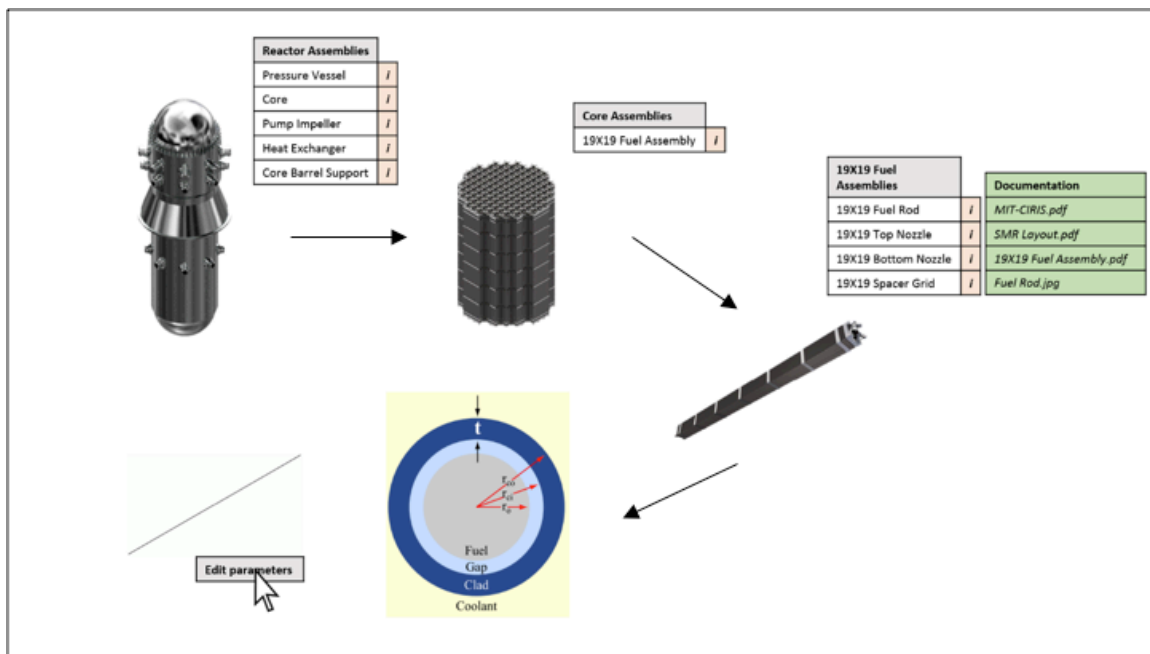


Figure 2. Example of a decomposition of a nuclear reactor into components, such as core assemblies, to geometrical dimensions of rods.

Visually tangible objects enable DTM to immediately capture salient features that either confirm design choices or suggest modifications. Design is an iterative process (i.e., visualization suggests confirmations or modifications); the rationalization space enables creating and refining design rationale descriptions. Linking visually tangible objects to their abstract descriptions (i.e., design rationale, as well as structure and function abstractions) enables designers to proceed with a convergent design process.

<sup>3</sup> This example illustrates a human-centered design student project carried out at the Human-Centered Design Institute of Florida Institute of Technology in the Fall 2014, involving the following graduate students: Saad Almesalm, Nicholas Kasdaglis, Joan Savage, Golnoosh Torkashvand, Ruthvik Adloori, and Joseph Torkaman.



### *Modeling dynamic processes*

Visualization of 3D static objects is an important step for handling their tangibility. However, these objects also have internal and/or external dynamics. In the case of a nuclear power plant, we typically focus on internal fluid dynamics, which include momentum, mass and thermal transfers. In the case of aircraft, an important part is their external dynamics (i.e., kinematics of and fluid dynamics around aircraft). Nevertheless, it is also useful to introduce a pseudo-natural dynamics of the way component assembly is, or can be, done.

Aesthetics guide our construction of patterns that help us make sense of the real world. According to Tufte (2001) in his description of visual display of quantitative information, “graphical elegance is often found in simplicity of design and complexity of data.” Klein and Lachize-Rey (2001) claimed that, “both Einstein and Dirac<sup>4</sup> felt that the aesthetic mathematical appeal of a physical theory was not just to please the mind. It was also an indication—indeed perhaps the best there is—of its validity.” For example, if one sees a car, the angle at which this car is seen does not matter, one will say that it is a car. This will be true until one sees an attribute of the object that we watch that disconfirms the concept of a car, for example. This means that we learn patterns that help us recognize objects (e.g., a car), no matter from what viewpoint we observe it from.

The whole question is then to visualize the right patterns to represent dynamic processes. Indeed, even if we have equations to represent these dynamic processes, the human centered designer needs to provide the best graphical representation of their complexity. According to Tufte, “not the complication of the simple; rather the task of the designer is to give visual access to the subtle and the difficult – that is, the revelation of the complex.” (Tufte, 2001, p. 191).

In addition, the user of the visualization system is part of the problem to be solved. This user should be able to easily manipulate visually tangible objects and associated dynamic processes. This is why gaming software engineering is very useful to this end. Aesthetics is again part of the “game.” Aesthetics in human-computer interaction deals with qualitative attributes such as interaction minimality, usability, usefulness, simplicity and cooperation. The easier human-system interaction is in terms of number of interactions, efficiency, success or pleasure, the more aesthetic the system will be.

### *Using a gaming engine for improving tangible object manipulation*

Computer gaming was chosen because current advancement in rendering, interaction methods and graphical capabilities are becoming mature. The computer game engine tool, Unity, was chosen to support SCORE. Therefore, SCORE provides capabilities, such as 3D representation, cross-platform, interoperability, and flexibility of coding in many different programming languages. It allows us to have integration of codes written in different languages and process manipulation accordingly (e.g., one part could be programmed in C# and another in Java) which makes it cross platform and easy to be maintained and/or update the system. SCORE inherits the following features from Unity (see example on Figure 3):

- **Rendering:** The Unity graphics engines use OpenGL, Direct3D, OpenGL ES for mobile platform (iOS, Android) and various APIs. There is also support of reflection, parallax and bump mapping. It provides features to render text and use of shadow maps for dynamic shadows. Various file formats of different software are supported. For instance, Adobe Photoshop, Blender and 3ds Max.
- **Scripting:** Scripting is built on Mono, the open source platform for .NET Framework.
- **Asset Tracking:** Unity has an asset server-control solution for developer scripts and visualization assets. It uses PostgreSQL as a backend system for audio and the Theora Codec for video playback.

---

<sup>4</sup> Paul Dirac was a physicist who contributed to the development of quantum physics.

- Physics: Unity engine provides built-in support for PhysX engine with real time simulation.

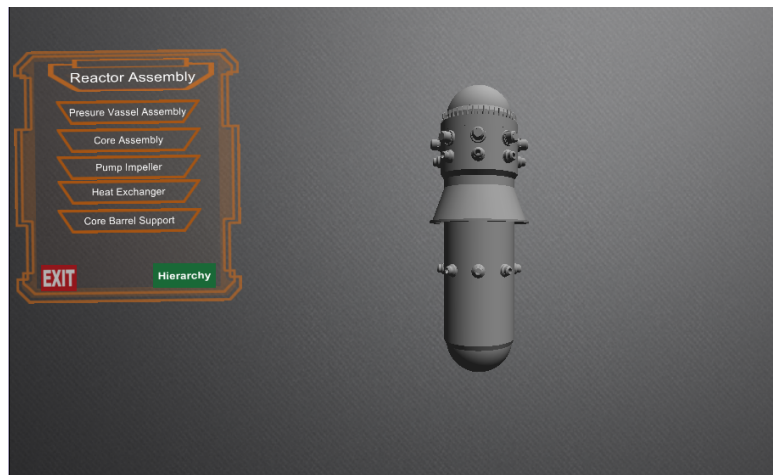


Figure 3. Example of visualization of I<sup>2</sup>S-LWR reactor assembly (activity space of a design card).

#### 4. The SCORE system for collaborative design

##### *Shared situation awareness supports incremental integration of rationalization and activity spaces*

Collaborative work is a crucial activity in a HCD team. **Shared Situation Awareness (SSA)** is a key issue in LCS design. SSA has been, and still is, studied at operations time (Stanton et al., 2006), but it requires more attention at design and development times. People may make errors because they are not aware of the current state of the design process. We will denote a design team member (DTM). We should provide solutions to answer the following questions: Is DTM<sub>1</sub> aware of current actions and productions of DTM<sub>2</sub> at any time? Is DTM<sub>1</sub> aware of what DTM<sub>3</sub> did at some point in time on the same topic he/she is currently working on or a similar one? How can we create and maintain the best SSA in the design team? We choose to develop a computer-supported cooperative work system, called SCORE, which supports SSA. The SCORE overview is provided in Figure 4.

SCORE uses components and procedures models. It is implemented using a web based application mechanism, which allows secure and trusted communication via VPN (Virtual Private Network). In addition, effective search mechanisms should provide the necessary means to pull appropriate information when needed. It would also be nice to have the appropriate information pushed to the front so potential users are aware of its existence. In both cases, context-sensitive information should be available at any time.

For this reason, at all stages of the design process, any DTM needs to know salient reasons that pushed other people to design systems the way they are. One author carried out an exhaustive study on traceability within a large aircraft manufacturing company (Boy, 1999). **Traceability** is not only information retrieval, it also deals with awareness that potential knowledge exists somewhere, and finally when this knowledge is found, it must be understood correctly.

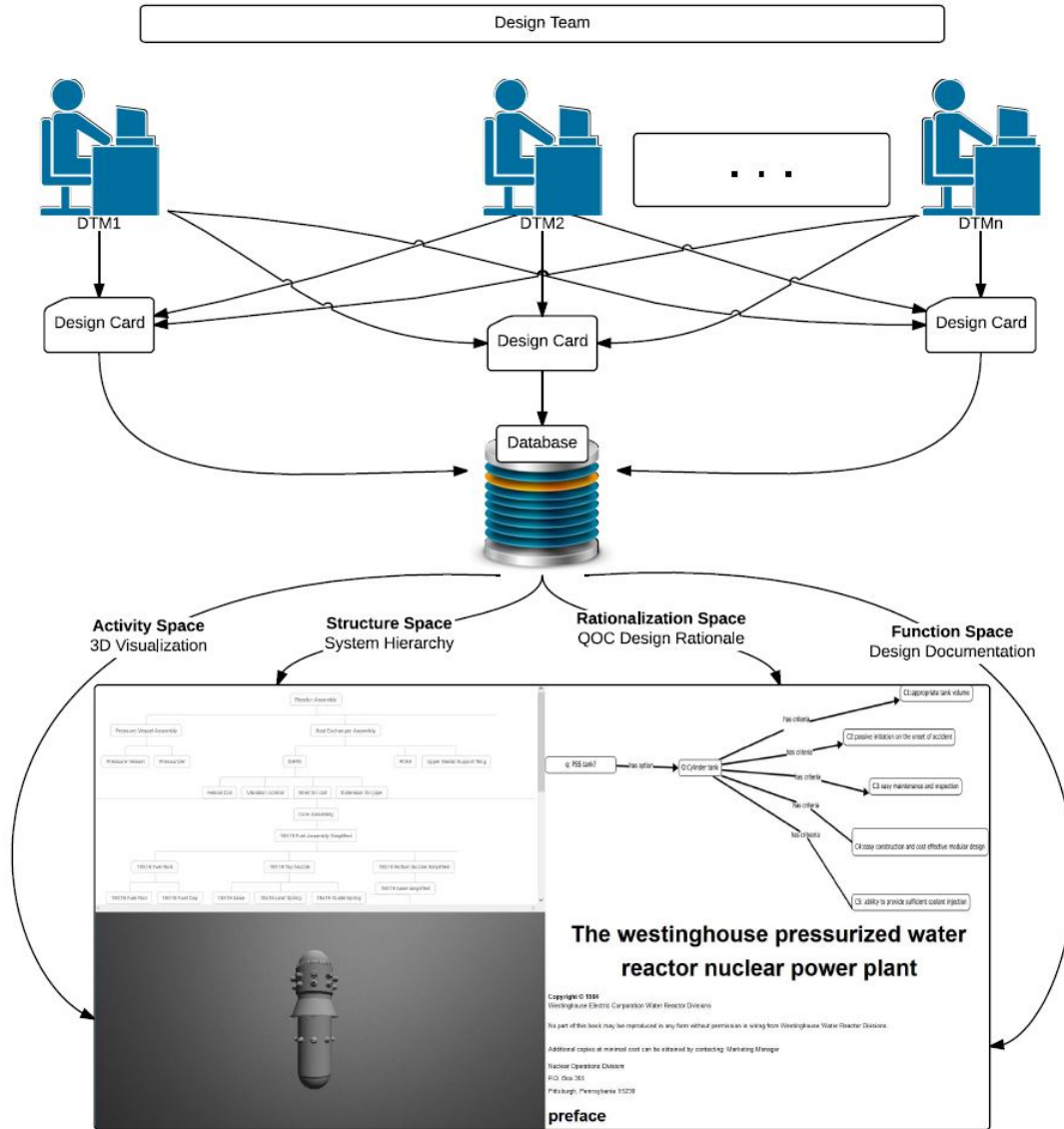


Figure 4. SCORE system overview.

#### *Design cards: Looking for the emergence of structures and functions*

The concept of active design document (ADD) was developed to support design knowledge capture and traceability (Boy, 1997, 2005). The ADD concept was recently extended to the concept of design card (Boy, to appear). A design card (DC) keeps both activity and rationalization spaces already developed in the ADD concept. Instead of solely focusing on end-users as the ADD does, the DC concept encapsulates all kinds of stakeholders dealing with the system being designed. For this reason, the **structure-function** concept is introduced. Each component of a system of systems can then be represented abstractly and concretely such as visualized in the SFAC (Structure-Function-Abstract-Concrete) model (Figure 5). Abstract rationalization includes design rationale and various kinds of justification of the system being designed. Concrete interactive visualization enables manipulation of virtual objects on a display for example. This can be done in the activity space. In addition, each system (or component) can be described in terms of structure (i.e., declarative descriptions of static objects) and function (i.e., procedural knowledge and dynamic processes).

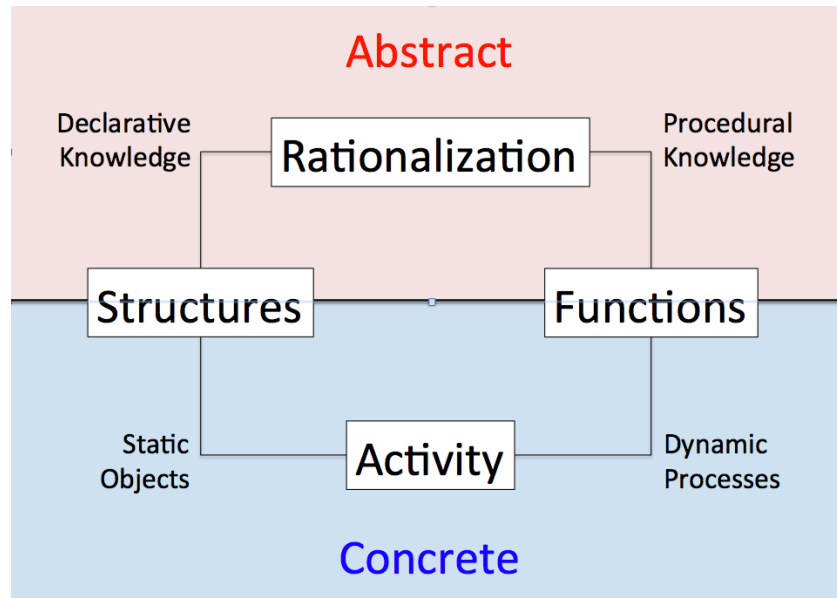


Figure 5. Combining abstract concepts and concrete objects through structures and functions.

A design card (DC) is defined by four entities (Figure 6):

- a rationalization space where the various components of the system being designed (SBD) are described in terms of design rationale, integration and requirements; this space includes declarative and procedural descriptions and statements;
- an activity space where the current version of the SBD is displayed; it includes static and dynamic features; this space enables SBD manipulation;
- a structure space where the various components and their inter-relations are formally and declaratively described as systems of systems;
- a function space where the various functions of the SBD are described in terms of procedural knowledge and dynamic processes involved; this space includes qualitative and quantitative physical and cognitive models.

The rationalization space is informed using the QOC notation, which was tested within the I<sup>2</sup>S-LWR design team. The activity space contains 3D models developed using AutoDesk Inventor, and transferred into the Unity game engine web player in order to facilitate integration within the DC spaces and enable intuitive manipulation of objects in the activity space. Two additional spaces were added: an instant messaging capability that allows design team members (DTMs) to exchange with one another on a DC; and a structured evaluation space that contains metrics that DTMs can inform to express their formative evaluations. The instant messaging capability also enables storage of old written contributions. It is very helpful as a DTM to be able to review at some later point a previous conversation to understand or even build upon the design methodologies. DCs are cooperatively created and refined by DTMs, and synthesized during periodic design meetings the frequency of which may vary with respect to various kinds of constraints. Incrementally combining abstract explanations of designed elements and integration with their explicit visual representation improves mutual understanding among DTMs, and consolidates design decisions. This human-centered design (HCD) approach also improves individual and collective familiarization with the complexity of mixing several expert contributions in NPP design. In this case, HCD is not focused on end-users, but rather on the designers themselves.

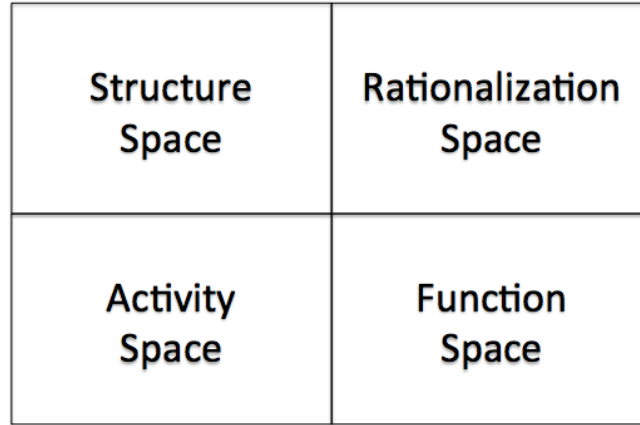


Figure 6. A generic Design Card (DC) (Boy, to appear).

A given DC presents the state of the design of a tangible interactive system at a given time for a given DTM. It is formally represented by  $DC(t, DTM_i)$ , where  $t$  is time and  $DTM_i$  is the design team member  $i$  (could be a person or a group of persons).

The concept of TIS fits very well with the DC concept. A DC enables designers to describe the various components of a system and the integrated whole (i.e., the TIS itself) in the rationalization space, display and manipulate them in the activity space, describe and use the navigation and control features in the operational space, and fill in the evaluation space as required after assessment of the system being designed.

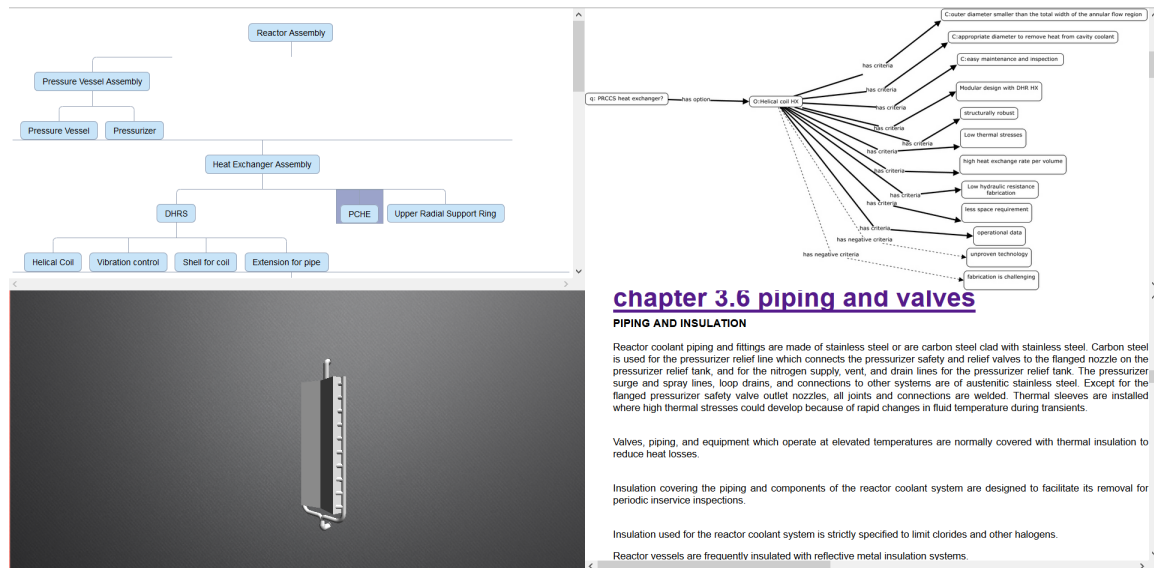


Figure 7. Example of SCORE design card showing: a structure space (e.g., describing the various components and their links), an activity space (e.g., visualizing a reactor), a rationalization space (e.g., QOC), and related functional documentation.

An example of DC is provided for the I<sup>2</sup>S-LWR design project (Figure 7). The upper-left part is the structure space where the system is described in terms of abstract concepts and their interrelations. The lower-left part is the activity space where the system and/or its components and dependencies can be visualized and manipulated as virtual objects. The upper-right part is the rationalization space where design rationale can be stored and related to the three other parts. The lower-right part is the function space where

physical and cognitive functions can be defined, refined and connected to the three other parts. In addition, any DTM can interact with another DTM using the instant messaging space of the DC. All DC parts are inter-related. For example, a DC user can easily navigate from one part to another. The current version is in progress. Later versions will enable any DTM to generate any component and describe it in the various spaces.

Using design card (DC) supports solving several problems such as geographical spread-out of experts of these groups, speed of technology evolution, high personnel turnover, and lack of documentation of the design process. DC generation happens during design. No matter how long DC generation can take, it is mandatory to ensure shared situation awareness of systems being designed, and associated knowledge. DC quality contributes to the quality of design.

#### *Design card generation*

Each DC ( $t$ , DTM<sub>*i*</sub>) corresponds to a version of the system being designed and developed. Each time design management has a design review meeting at time  $t_1$  (Figure 8), all DTMs analyze the work done by each DTM and create a synthetic ADD ( $t_1$ , DT), where DT is the whole design team. DCs are like scores that musicians use to play a symphony in an orchestra, with the peculiar difference that, unlike scores, DCs are being incrementally defined to get a sound symphony in the end of the design process.

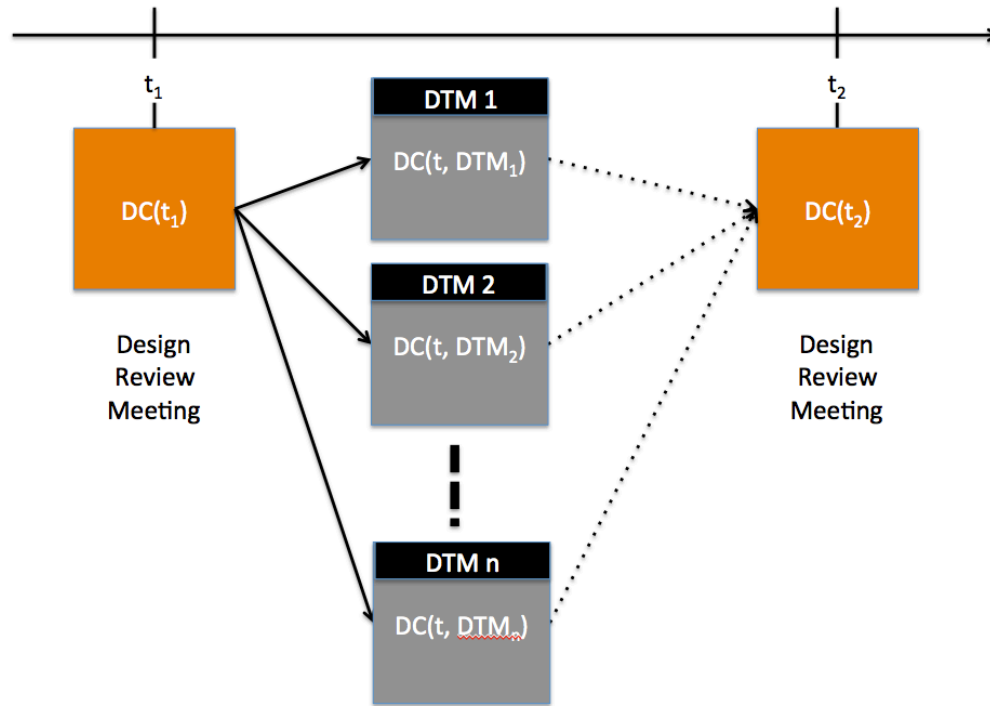


Figure 8. Design cards generation.

After a design review meeting at time  $t_1$ , each DTM<sub>*i*</sub> works on the premises of  $DC(t_1, DT)$ , and produces their own  $DC(t, DTM_i)$  until the next design review meeting is organized at time  $t_2$ , where a new  $DC(t_2, DT)$  will be produced from the integration of all active design documents created and/or modified by each design team during the time interval  $[t_1, t_2]$ .

Each DC is stored into a design database, and can be retrieved at any time by any member of the design team (although some restrictions could be implemented and applied if necessary). Various DC traceability mechanisms can be implemented such as via:

- rationalization space (by date of creation, design rationale keywords);



- activity space (by selection of components graphically);
- structure space (as a table of contents);
- function space (using scenarios and criteria as indices).

## 5. Discussion

### *The issue of tangibility*

During the 20<sup>th</sup> century, mechanical engineering was the central engineering discipline. We designed systems such as power plants, aircraft and cars by developing mechanical things and assembling them, for example. Structure was the main issue of this technology-centered engineering approach. Systems were designed as a set of physical parts that were assembled before test and delivery. Control mechanisms, the functional components, were then developed to command and control these systems. Manual and automatic control was first based on electrical circuits and then electronics. During the last three decades, we have experienced an exponential growth of software for control and management of these systems. This evolution can be summarized as “incorporating software into hardware.” This functionalizing phenomenon was called “automation,” which became a real human factors issue because machines had cognitive functions of their own and people had to adapt to this revolution.

Since the beginning of the 21<sup>st</sup> century, software engineering has become the central discipline. We are designing systems as pieces of software that we can test very early during the design process. Function is now the main driver of this human-centered design approach. Systems are developed as software objects providing virtual structures and methods leading to appropriate functions. From a human factors point of view, the main problem is no longer automation since we are able to integrate structure and function as a piece of software from the beginning. In addition, modeling and simulation capabilities enable designers to involve test users to evaluate these functions very early in the design process. However, the main issue has become tangibility of the final product (i.e., making the final product both physically and figuratively tangible). This is why we need to incrementally develop tangible explanations, evaluation and descriptions of emergent properties of the product being designed and developed. This article proposes a method and a tool that support this approach. Physical tangibility can be supported by 3D visualizations of the structural components in the beginning of the design process. We can even 3D print these components to improve physical tangibility assessment. Figurative tangibility can be supported by the development of ontologies (i.e., networks of concepts), and tested using criteria such as consistency, complexity, maturity and other factors.

In addition, tangibility emerges from collaborative work of a variety of design team members. A single person cannot design a nuclear reactor. It requires several disciplines collaborating. Team members are like musicians who need to be coordinated in an orchestra (Boy, 2013). In this design framework, writing (i.e., designing) the symphony (i.e., the product) is not done by a single composer, but by the whole design team. DTMs need a common **frame of reference (CFR)**, which is their music theory and various coordinated scores that support the overall symphony. Design cards enable incremental development of SCORE’s CFR. This is being applied to the design of I<sup>2</sup>S-LWR nuclear reactor.

### *3D visualization production*

At design time, the concrete part is commonly represented using computer-aided design (CAD) software, which enables to generate 3D models of the various components of the system being designed. These 3D models include static objects and dynamic processes that enable the visualization of the way the various components being designed work and are integrated together. Later on during the design and development process, these 3D models can be 3D printed and lead to more graspable appreciation of the components being built and their possible integration. Testing occurs at each step of the design process by taking into account concrete parts together with their abstract counterparts (i.e., their design rationale, justifications, as well as various syntactical and semantic relationships that exist among them).

CAD software, such as AutoDesk Inventor, is used by a specialized community of 3D designers. The SCORE solution, based on Unity, captures the main interaction features with 3D content and thus enables a heterogeneous community to visualize and manipulate resulting 3D objects in the context of the overall design of the I<sup>2</sup>S-LWR.

#### *Computer Supported Cooperative Work (CSCW)*

SCORE, as a CSCW<sup>5</sup> system, provides organizational support for collaborative work of design team members (DTMs). The term “supported” in CSCW refers to both very simple, uncoordinated access to shared data and to complex, synchronized modeling and provision of team-internal relationships and interactions (like group processes).

SCORE not only helps in technological design, but it also helps in the development of organizational concepts and relations, such as strategies, policies, procedures, and organizational structures required to support the design team. The design card paradigm provides traceability possibilities both in time (i.e., any DTM can look backwards in time and review previous decisions, designs and improvements) and space (i.e., at any possible time, several DTMs can connect from anywhere, discuss current design cards, and make decisions).

## **6. Conclusion**

The way knowledge is exchanged during design and further during the life cycle of a life-critical system induces several factors related to systems (complexity, completeness, maturity, traceability, system maintainability, reliability and availability) and people (expertise, writing, simplicity, drafts, information credibility, uncertainty, awareness, trust and consensus). The SCORE system is currently developed to support a design team to coordinate accomplishment of tasks satisfying these factors.

Design of life-critical systems involves DTMs coming from several disciplines and backgrounds. Collaborative work is then a major issue, especially when there are budgetary and time constraints. DTMs need to understand each other correctly to provide the best design in the end. By developing, assessing and validating design cards, they can converge toward both good design solutions, as well as appropriate documentation of the design process and solutions.

Offering cross-platform capabilities and emphasizing frequent up-to-date technical communication is a key consideration in this case. SCORE satisfies these requirements by providing a collaborative tool well suited for agile development. Furthermore, its ability to track design changes in terms of chronology as well as design team member involvement allows all DTMs to obtain a complete and well-informed design status at any time.

## **Acknowledgements**

We would like to thank all members of the I<sup>2</sup>S-LWR project and its funding organization, the US Department of Energy, NEUP 12-4733, SRC#00132015, Under Prime Contract No. DE-AC07-05ID14517 (GT Project 2506J12, Research Agreement RD537).

---

<sup>5</sup> The term “computer-supported cooperative work” (CSCW) was coined in 1984 by Paul Cashman and Irene Grief to describe a multi-disciplinary approach focused on how people work and how technology could support them.

## References

- Boy, G.A. (To Appear). *Tangible Interactive Systems*. Springer, London, U.K.
- Boy, G.A. (2014). From automation to tangible interactive objects. *Annual Reviews in Control*, 38, 1–11.
- Boy, G.A. (2013). *Orchestrating Human-Centered Design*. Springer, London, U.K.
- Boy, G.A. (2011). Introduction of the Handbook of human-machine Interaction: A human-centered design approach. Ashgate, UK.
- Boy, G.A. (2005). Knowledge management for product maturity. Proceedings of the International Conference on Knowledge Capture (K—Cap'05). Banff, Canada. October; also in the ACM Digital Library.
- Boy, G.A. (1999). *Traceability*. EURISCO-Airbus report T-99-060-V3.
- Boy, G.A. (1998) *Cognitive function analysis*. Praeger. ISBN-13: 978-1567503777.
- Boy, G.A. (1997). Active design documents. *Proceedings of ACM DIS'97 Conference*. ACM Press, New York.
- Boy, G.A. (1991). Indexing Hypertext Documents in Context. *Proceedings of the Hypertext'91 Conference*, San Antonio, Texas, December.
- Boy, G.A. & Barnard, Y. (2003). Knowledge Management in the Design of Safety-Critical Systems. Proceedings of the European Cognitive Science Conference, Osnabrück, Germany.
- Boy, G.A. & deBrito, G. (2000). Towards a categorization of factors related to procedure following and situation awareness. Proceedings of HCI—Aero 2000, Cepadues, Toulouse, France.
- Conklin, E.J. & Begeman M.L. (1989). gIBIS: A tool for all reasons. *Journal of the American Society for Information Science*, 40, pp. 200-213.
- Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F. & Tu, S.W. (2003). The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-computer studies*, Academic Press, Vol. 58, Issue 1, pp. 89-123.
- Grudin, J. (1994a). Computer-Supported Cooperative Work: History and focus. *IEEE Computer*, 27, 5, 19-26.
- Grudin, J. (1994b). Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM*, 37, 1, 92-105. Republished in Baecker, et al. (1995).
- Hoog, R. de., Kabel, S., Barnard, Y., Boy, G., DeLuca, P., Desmoulins, C., Riemersma, J. & Verstegen, D. (2002). Re-using technical manuals for instruction: creating instructional material with the tools of the IMAT project. In Y. Barnard, M. Grnadbastien, R. de Hoog & C. Desmoulins. *Ingrating Technical and Training Documentation, Proceedings of the ITS2002 workshop*, p. 27-38. 6<sup>th</sup> International conference on Intelligent Tutoring Systems, San Sebastian, Spain/Biarritz, France.
- Klein, E. & Lachièze-Rey, M. (2001). *The Quest for Unity: The Adventure of Physics*. Oxford University Press, ISBN-13: 9780195120851.
- MacLean A., Young R.M., Bellotti V.M.E. & Moran, T.P. (1991). Questions, options and criteria: Elements of design space analysis. *Human-Computer Interaction*, 6, pp. 201-250. This article describes the background for designing space analysis and the QOC notation. It explains how it can be used in the design process, and to study reasoning in design. Other articles of interest on this topic also appear in this issue of the journal.
- Moran, T.P. & Carroll, J.M. (Eds.) (1996). *Design rationale: Concepts, methods and techniques*. Hillsdale, NJ: Erlbaum.
- Muller, M.J. (2003). Participatory design: the third space in HCI. In *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc., pp. 1051–1068
- Novak, J.D. & Cañas, A.J. (2006). *The Theory Underlying Concept Maps and How to Construct and Use Them*. Institute for Human and Machine Cognition, Pensacola. Technical Report IHMC CmapTools 2006-01 Rev 2008-01. <http://Cmap.ihmc.us/docs/theory-of-concept-maps>, Retrieved on May 1, 20015.
- Poltrock, S.E. & Grudin, J. (2001). Collaboration Technology in Teams, Organizations, and Communities. Tutorial. 8<sup>th</sup> IFIP TC13 IFIP INTERACT Conference on Human-Computer Interaction, Tokyo, JAPAN July 9-13, 2001.
- Stanton, N.A., Stewart, R., Harris, D., Houghton, R.J., Baber, C., McMaster, R., Salmon, P., Hoyle, G., Walker, G., Young, M.S., Linsell, M., Dymott, R. & Green, D. (2006). Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology. *Ergonomics*. Oct 10-22; 49 (12-13), pp.1288-311.

Boy, G.A., Jani, G., Manera, A., Memmott, M., Petrovic, B., Rayad, Y., Stephane, A.L. & Suri, N. (2016). Improving collaborative work and project management in a nuclear power plant design team: A Human-Centered Design approach. *Annals of Nuclear Energy*. Elsevier, ANE4864.

Tufte, E.R. (2001). *The Visual Display of Quantitative Information*. Graphics Pr, 2nd edition. ISBN-13: 978-0961392147.

Tuomi, I., *Corporate knowledge: Theory and practice in intelligent organizations*. Helsinki: Metaxis, 1999.  
WISE: [www.ist-wise.org](http://www.ist-wise.org)

## Authors

Dr. Guy Boy is University Professor and Director of the Human-Centered Design Institute at Florida Institute of Technology. He has worked on design for safety for more than 35 years, mostly in the aerospace domain. He was the Principal Investigator of a 4-year AREVA research grant on the design of new nuclear power plant control rooms. He was the director of the European Institute of Cognitive Sciences and Engineering (EURISCO) for 16 years, where he organized two summer schools in 1995 and 2001 on knowledge management, as well as human-centered organization design and management and design for safety topic that gathered most recognized contributors such as Robert DeHoog, Jonathan Grudin, Kari Kuutti, Saadi Lahlou, Dan Shapiro and Norbert Streitz. Several references to their work are in this article. In addition, he gave several tutorials organized by the HUMANIST European network of excellence. These tutorials enabled him to synthesize distributed knowledge and experience on design for safety.