ICKT support for the Building Industry: A Virtual Partner

S.S.Ozsariyildiz, I.S. Sariyildiz and R. Stouffs
Delft University of Technology
Delft
The Netherlands

ABSTRACT

We are now at a stage in which ICKT techniques allows us to develop knowledge intensive systems, such as intelligent decision support systems, to support collaboration and cooperation. This paper describes a theoretical approach, in which *collaborative agents* take on the role as a partner during the decision making process, in order to support various actors in the building process. The *group-decision making* set-up will be discussed and we will give an overview on the state of art of this subject. We will give some insights into their application in the building practice. In addition, we will provide some examples of use-case-scenarios as inception and early design support and evaluate it.

1 INTRODUCTION

The building sector is entering a new era. ICKT (Information, Communication, and Knowledge Technology) and Internet technologies enable a closer link between participants in the building process, their activities, knowledge, and information throughout the entire life cycle of a building. Collaboration and communication, such as in collaborative and cooperative engineering, using ICKT techniques will be the future of the building practice. These rapid advances in information technologies give rise to a demand for more intelligent systems for ordering data. This led to the birth of *computational intelligence,* a science mainly involved with systems which are able to process complex, uncertain, even incomplete or contradictory information. This paper provides an overview of these and other future developments of information and communication technology within the building sector as a *virtual partner*.

2 ICKT IN THE BUILDING DESIGN PROCESS

The building and construction process can be subdivided into four main categories of activities, namely, *initiation*, *creation*, *usage*, and *demolish*. Correspondingly, we can distinguish four main domains of applications of ICKT in the building process.

- *Initiation* oriented ICKT (applied in the inception or briefing)
- *Creation* oriented ICKT
    - *Conception* oriented ICKT (conceptual design phase)
    - *Materialization* oriented ICKT (building physics and building technology aspects such as calculating bearing structures and detailing)
    - *Pre-realization* oriented ICKT (linking the first two categories; specifies work,

detailed design, schedule and resource.)
  - *Realization* oriented ICKT
- *Usage* oriented ICKT
- *Demolition* oriented ICKT

As a result, of these developments in ICKT, the role and work processes of the people who are involved in the building process are changing. Until now, this process was divided into a few stages. When the architect designed the concept, this then went to the constructor to be worked out and materialized and afterwards to the contractor to be built. Additionally, there was always a supervisor, manager who led this process. We are now experiencing an evolution of this process where it is no longer sequential (co-operation) but more typical of a *network*, which we call *information, communication and collaboration networking in the building process*.

Looking back at developments of ICKT in the building sector, we notice that computers were first put into practice as a *tool***,** as an instrument for achieving a specific result; either to produce a final drawing, an animation, a simulation, or an interactive visualization. Already computers have taken on a different role within the architectural design process as a new *medium* besides the existing media. Especially the widespread use of the Internet and the developments of the Web have pushed the computer into the role of a medium.

In the very near future, we can expect another shift in the role of computers in the design and building process, namely, as a partner (Schmitt 1999; Sarýyýldýz et al. 1998; McCullough 1996). We are now at a stage in which ICKT allows us to develop new techniques and methodologies where the computer can be used as a *partner* by means of knowledge integration, decision support, and artificial intelligence. Decision support systems allow the computer to support the user through knowledge provided by experts or by the user. The computer can also be a partner when we teach it things it can reason with. It can even be a valuable and reliable friend when we let it solve problems that are not clearly defined, fuzzy, or uncertain. It can also assist us in generating shapes by processing information that influences the shape, supported by self-learning techniques. Here, artificial intelligence techniques such as fuzzy logic, genetic algorithm and neural networks play an important role. ICKT as tool, medium and partner has the following support in the design process:

1. *Tool*
   - CAD (Computer Aided Drafting)
   - Shape modeling 2D/3D
   - Presentation (animation, simulation, rendering, etc.)
   - Analysis
2. *Medium*
   - Interactive visualizations (virtual reality, cyberspace)
   - Information processing
   - Communication (Internet technology)

- Collaborative and concurrent engineering, CSCW (Computer Supported Collaborative Work)
- CAD-CAM, CAE, EEM (Enterprise Engineering Management), etc

3. *Partner*
   - Knowledge integration (artificial neural networks, fuzzy logic, intelligent agents, etc.)
   - Decision support systems
   - Advanced modeling (genetic algorithms, grammars, etc.)
   - Intelligent management

Finally, ICKT is meant to support the designer in the design process to achieve the intended goal. This goal can be much differentiated, depending on the user. The flexibility and the efficiency of these ICKT means are an important issue for the future.

3 VIRTUAL PARTNER

In the near future, we will see increasingly a use of ICKT tools as virtual partner. Virtual partners may assist us in making decisions or in supporting decision-making. They can also serve to support design coordination among different design stages. Agent technology will be central to the development of virtual partners; these will have artificial characters and the freedom to act independently in order to support the creation process.

3.1 **Decision Support**

During the creation, process experts may make decisions related to project management, engineering, or client requirements, etc. Currently decisions can be supported in two ways; using decision software or decision support systems.

3.1.1 *Decision Software*

Decision software is special kind of algorithmic software designed to help individuals make decisions. These programs examine data given to them and, in a similar way to expert systems, suggest an optimum decision or conclusion.

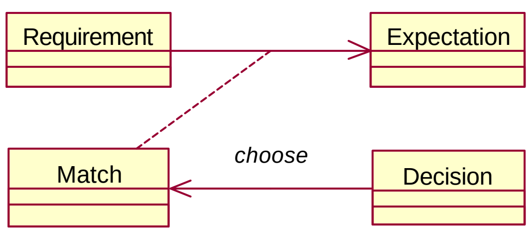


Figure 1: **A simplified decision is: matching the requirements to expected results and choosing the possible solution among the matched solutions.**

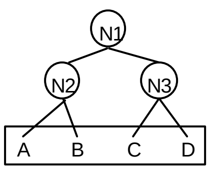
Decision software can help one make a decision on a complex problem. Typically, the goal is to choose the best of several different alternatives (figure 1). The criteria for selecting a particular outcome are defined in numerical terms. Their importance or priority is weighed, and by converting the criteria into numerical values, mathematical algorithms can be used to process them and select the most desirable outcome. Decision software provides a precise solution, whereas knowledge based system may only offer an approximate solution or a guess, or no solution at all.

3.1.2 *Decision Support Systems*

A decision support system is collection of programs used for decision-making. Such programs help management in forecasting, planning, and managing large enterprises. Decision Support Systems (DSS) usually incorporate some kind of decision programs. Many DSS also include a modeling capability that enables a mathematical simulation of a situation to be built in order to test various tactics and strategies. Classical DSS use conventional procedural data processing to achieve their results. The modeling of creation decision processes can be formulated in both decision tables and decisions trees (figure 2). The tables are used to store the relevant knowledge in a formal way so that the agents can re use this knowledge.

3.2 **Creation Process and Coordination**

Many partners are involved in the creation process, such as the architect, constructor, government, users, and each is concerned with specific knowledge and information during this process. The increasing actual costs, time, and the high demand for amounts of information and knowledge require better communication and co-ordination for collaboration between these partners. Among others, to meet these requirements between various partners, the virtual partners could fulfill the role of the missing partner or act as part of the total process without any cost.



S1	P1	P2	P3

Figure 2**: The modeling of creation decision processes: (left) a simple decision tree; (right) a simple decision table**

The coordination process is essential for the total outcome of a project. The coordination covers the act of *cooperation*, *completion* and *collaboration* (figure 3). Firstly, cooperation requires planning. The planning should be both central and distributed. Secondly, completion is an important part of the coordination process, which can be achieved by negotiations and biding. Finally, collaboration is an act of searching for a solution with common a goal. These three acts of coordination are critical for a successful virtual partner.

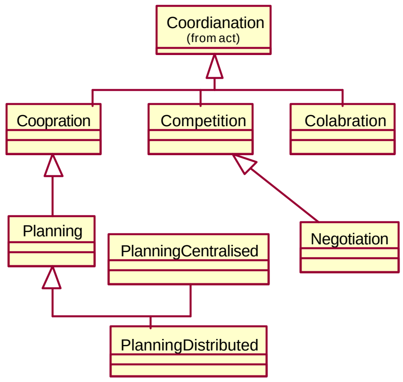


Figure 3: **Creation process coordination for virtual partners**

3.3. **Agents**

After searching various articles, papers, and web pages that attempted to described an agent; it has become increasingly clear that a concise definition of an agent has yet to be reached. An exemplar definition of an agent is “Someone who acts on your behalf” (Agent 2001). However, *software agents* commonly share three main characteristics, that is, they are *adaptive*, *reusable*, and *autonomous* (figure 4).

Agents are *reusable* if they are able to perform their task many times for different purposes and goals. An *autonomous* agent is able to take initiative and exercise a non-trivial degree of control over its own actions. It is goal-oriented, collaborative, flexible, and self-starting. An *adaptive* agent automatically customizes itself to the preferences of its user, based on previous experience. An adaptive agent also adapts to changes in its environment. Agents may also be *intelligent*, which is desired but not essential, or *mobile*, such that it is able to transport itself from one machine to another and across different system architectures and platforms. A *communicative* agent is able to engage in a complex communication with other agents, including people, in order to obtain information or enlist their help in accomplishing its goals (Hebrew 2001). Agents in the building and construction industries can be used as a modification or extension of product models. Agents could also simulate cost engineers’ tasks or mechanical engineers’ tasks, etc.

However, the major bottleneck with this approach is not related to the technology itself, but to the *communication* and *knowledge modeling* of the real world actors. Therefore, in building and construction projects, most of the recent agent-based tools did not go further than research prototypes. With respect to agent communication, we can distinguish three main approaches: KQML, KIF and DAML+OIL.

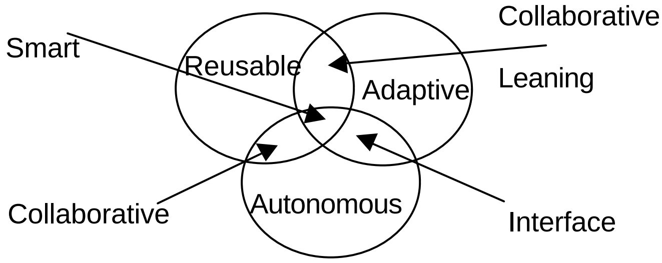


Figure 4**: Common software agent characteristics: reusable, adaptive, and autonomous different types of agents such as smart, collaborative, interface, and learning agents**

3.3.1 *KQML Approach as Communication Language*

The knowledge query and manipulation language (KQML) is a protocol for exchanging information and knowledge, as illustrated in figure 5. The KQML is part of a broader research effort to develop a methodology to distribute information among different systems (KQML 1992). The elegance of KQML is that all information for the understanding of the content of the message is included in the communication itself. The basic protocol is defined by the following structure: (*KQML-performative* :*sender* <word> :*receiver* <word> :*language* <word> :*ontology* <word> :*content* <expression> …) (Finin 1994).

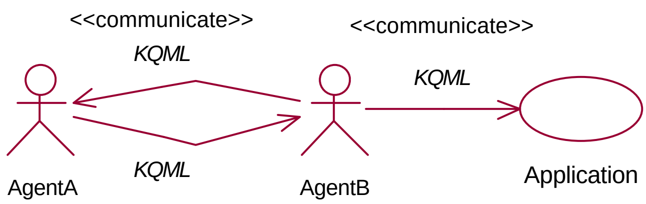


Figure 5: **KQML is a protocol for communications among both agents and application programs**

The syntax is similar to Lisp; however, the arguments -identified by keywords preceded by a colon- may be given in any order. The semantics of KQML-performatives (see table 1) are domain independent, while the semantics of the message is defined by the fields: *content* (the message itself),: *language* (the language in which the is expressed) and :*ontology* (the vocabulary of the words in the message.)

However, KQML must operate within an infrastructure that allows agents to locate each other. The infrastructure is not part of the KQML, and implemented systems use custom-made utility programs called routers or facilitators, to perform this function. Basically, all the calls are requested from the facilitator in order to find the capabilities of the agents.

Table 1: **KQML performatives organised into seven basic categories.**

Performative Categories	Performatives
Basic query	Evaluate, ask-one, ask-all, …
Multi response	Stream-in, stream-all, …
Response	Reply, sorry, …
Generic informational	Tell, achieve, cancel, untell, …
Generator	Standby, ready, next, rest, …
Capability definition	Advertise, subscribe, monitor
Networking	Register, forward, broadcast, …

3.4 **Artificial Characters**

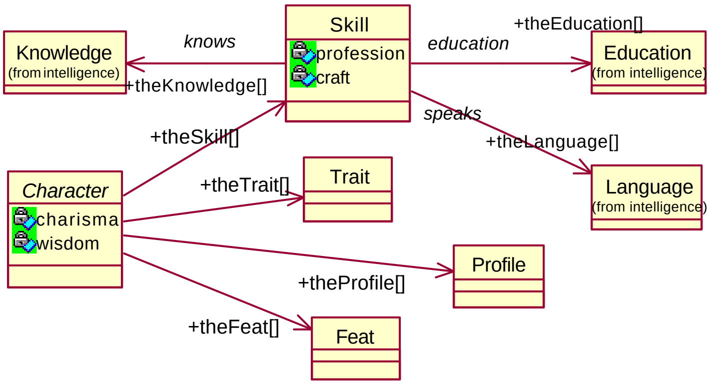


Figure 6**: Knowledge and artificial character**

Virtual agents have *character*, charisma, and wisdom. The *character* holds the information about the physical properties such as trait, feat, and profile. In addition, the character has a set of *skills* related to the agent's profession. The skills are a set of *languages* it understands and uses, and the capability to learn and *educate* it self. Furthermore, it performs a certain intelligence that knows what *knowledge* it has. As an example, the company Virtual Personalities developed a character available via the Internet (figure 7). The agent is capable of understanding neural languages. It is capable of making dialogs about certain subjects. In the future, we could expect more applications of agents as virtual partners in building and construction process.

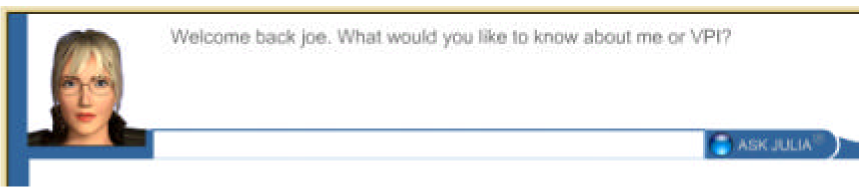


Figure 7**: An exemplar artificial character provided by company Virtual Personalities Inc.**

4 CONCLUSIONS

As in any other sciences, ICKT developments influence the building process. The newest developments for the building sector will be in the field of *knowledge integration* and *decision support environments*, leading finally, to ICKT support in the entire building process, from initiative until demolition. Collaborative, cooperative and competitive engineering will pervade the building design process. Because the technological developments on Computer Science technically make it possible that the architect can be supported by the ICKT means *as a partner* during the decision making process, by using computational intelligence techniques. To comply with the associated requirements for a building design in a consistent and optimal way, careful design deliberations have to be carried out before a final decision is made. Due to comprehensive considerations that are necessary, the decision process is very much time consuming. To facilitate such a design process, a systematic approach with the application of artificial intelligence (AI)-based *intelligent agent technology* in information processing and support is necessary. To achieve the intelligent environment for collaboration and communication and cooperation for building design agents which have artificial characters is described. In the near future, virtual agents with *character*, charisma, and wisdom that are capable of generating solutions and able to support decisions will be part of actual design process.

5 REFERENCES

Agent (2001) *http://www.msci.memphis.edu/~franklin/AgentProg.html*

Finin, T.; McKay, D.; Fritzson, R.; and McEntire, R. (1994). The KQML information and knowledge exchange protocol. In *Third International Conference on Information and Knowledge Management*.

Hebrew (2001) *http://www.cs.huji.ac.il/labs/dai/wkshp/index.html*

KQML Spec2 ARPA Knowledge Sharing Initiative. (1992). Specification of the KQML agent--communication language. *ARPA Knowledge Sharing Initiative, External Interfaces Working Group working paper. Available as http://www.cs.umbc.edu/kqml/.ps*.

McCullough M. (1996). Abstracting Craft: The Practised Digital Hand, *MIT Press, Cambridge, Mass.*

Schmitt G. (1999). Information Architecture: Basics of CAAD and its future, *Birkhäuser, Basel.*

Sarýyýldýz S., van der Veer P., Schwenck M. and Çiftçioðlu Ö. (1998). Computers as reliable and valuable partner, *Cyber-real Design, 5th Conference on Computer In Architectural Design, TU Bialystok, Poland.*

Sariyildiz I.S., Çiftçioglu Ö., Stouffs R. and Tunçer B. (2000). Future developments of ICT in the building sector, Construction Information Technology 2000 (ed. G. Gudnason), *Icelandic Building Research Institute, Reykjavik, Iceland, vol. 2, 790-800.*