International Journal of Design Sciences & Technology

Volume 16 Number 2

ISSN 1630 - 7267

Editor-in-Chief:
Reza Beheshti
Khaldoun Zreik

Van de Ruitenbeek, M. and Beheshti, R. (2009). Towards developing ubiquitous design environments, International Journal of Design Sciences and Technology, 16:2, pp. 17-37

Europia PRODUCTIONS

ISSN 1630 - 7267

15, avenue de Ségur,
75007 Paris, France.
Tel (Fr) 01 45 51 26 07 - (Int.) +33 1 45 51 26 07
Fax (Fr) 01 45 51 26 32- (Int.) +33 1 45 51 26 32
E-mail: dst@europia.fr
http://www.europia.fr/DST

Towards developing ubiquitous design environments

Martinus van de Ruitenbeek* and **Reza Beheshti****
* Delft University of Technology, The Netherlands. E-mail: hkm.vanderuitenbeek@tudelft.nl
** Delft University of Technology, The Netherlands. E-mail: m.r.beheshti@tudelft.nl

The objective of this research project is to provide structural engineers with an intuitive human-computer interfacing in order to transparently use modern digital design tools that often have useful but hard to find features. Every design task is a close combination of conceptual and detailed design. It is hard to catch both modes in a convenient digital design tool. This paper reports a prototype that allows both Imprecise (such as sketching and discussing) and Constraint (such as typing) Human Computer Interaction, in a ubiquitous design environment (UDE). For constraint but low-level (precise) human-computer interfacing, UDE is equipped with a multi-touch interactive screen. For imprecise but high-level human-computer interaction the prototype implements a rudimentary android called Levente that can 'talk, hear and see' in order to interact with both the structural engineers and the design systems. Its physical existence is meant to give a structural engineer the illusion of having a colleague and to demonstrate how humans and computers can interact mutually without the need of precise protocols by using only speech and vision. Obviously Levente can also exist as an avatar. UDE might prove to be a valuable design tool relieving the structural engineer from cumbersome functionality in software applications and time consuming tasks in order to focus on the actual design. Recent developments in formalizing backbone technology for information and communication handling as well as proper definitions of how to define and solve problems in the BC industry contribute to the implementation of UDE.

Keywords: precise and imprecise, human computer interaction, structural engineering, multi-touch, android

1 Introduction

Structural engineers are frequently confronted with a paradox: complex mathematical problems are easy to solve with today's software tools whereas the seemingly far less complicated and off-topic tasks such as gathering, managing, combining, inputting, manipulating, reporting and archiving information turn out to be most time consuming, error-prone and hard to automate. The most interesting work for a structural engineer is to find creative solutions for a given problem, make rough estimations on dimensions and cost, pass it on to specialists for detailed analysis, interpret the results and tweak the solution accordingly. There are many excellent tools for virtually every thinkable task (and if not, it is fairly easy to rapidly create them using programming tools, spreadsheets etc.). Moreover some tools can integrate other tools (think of a service oriented architecture and the semantic web). The question is where to find an easy to use overall 'application' that can take over the annoying off-topic tasks of information handling, finding, installing, maintaining and learning appropriate tools, integrating those tools, creating new tools and so on? Similar to a human such a tool must be capable of understanding a problem from a variety of formats (mainly visual auditory), delegate it to dedicated lower level tools in any requested format and prepare the tool's output in any requested format.

The current state-of-the-art solution is simple and effective: provide design leaders with qualified secretaries, draftsmen, structural engineers and document controllers, allowing them to delegate their annoying tasks to those sophisticated human 'tools'. What is the reason for those human tools' effectiveness and how can their secret be automated? Brooks posed an interesting solution: Intelligence without Representation[1] and Intelligence without Reason.[2] His experiments with robots show that representing the real world in a virtual world can get in the way of intelligence because it is too rigid in a dynamic world. The Darpa Grand Challenges, with the Darpa Urban Challenge held in 2007 vividly demonstrate the recent success of robot vehicles in real-life traffic situations and the rapid developments since the first challenge in 2004 during which none of the vehicles reached the finish.[3] [4] Moravec also stresses that the development of true intelligence must depend on mobility, acute vision and the ability to carry out survival-related tasks in real-world dynamic environments.[5] He also predicts that computer hardware will be sufficient by 2020 to handle the requirements.[6] Based on these ideas Brooks reported successful Creatures built on incremental parallel layers of intelligence that are organized by activity (or objective) rather than by function.[7] Perception, representation and other functions are consequently not defined in certain locations but appear scattered throughout the architecture. There are many similarities with the human brain in this approach.

The building and construction industry now heavily focuses on Building Information Modelling (BIM) in order to explicitly represent the outside world in a virtual one. Tolman was involved in BIM developments and concludes that the most viable solution is to stop slow standardization attempts and focus on a distributed web based project information service in which partners can publish and download information in a variety of formats.[8] The service provider must take care of information management and interoperability (essentially a BIM), and if he succeeds then market parties will be willing to pay for the service.

Tolman's conclusion is similar to Brooks': try to be compatible with the real world rather than setting up a new representation that requires changes in the real world and its actors. The actual challenge in the authors' opinion is at the front-end where people and tools interface. When human-computer interfaces become intelligent, it is likely that the necessity for standardization and representation efforts fade out of human scope. In fact communication and representation standards already exist: visual and auditory communication and the real world representing itself.

2 Problem description

The Building and Construction industry demands new solutions to compensate for the ever increasing complexity of projects with respect to project size, regulations, quality and risk requirements and juridical responsibilities. Structural engineers find themselves in a crucial role as they are expected to design safe, reliable, low-cost structures that are easy to build, maintain and reuse. Traditional peer to peer like design approaches fail in complex, distributed and multidisciplinary projects partially because structural engineers have difficulties to manage the large and heterogeneous information flows, comply with continuously changing quality control systems register all actions and inform relevant project partners.

2.1 Background

The BC industry is trying to overcome these problems by adopting modern design methodologies such as systems engineering that prove to work well in

1 **Brooks, R.** (1998). Intelligence without representation, Cognitive Architectures in Artificial Intelligence: The Evolution of Research Programs

2 **Brooks, R.A.** et al eds (1991). Intelligence without Reason, Morgan Kaufmann publishers, San Mateo, USA, Sydney, Australia

3 **DARPA** (2007), [www.darpa.mil/grandchallenge/overview.asp]

4 **DARPA** (2004), [www.darpa.mil/grandchallenge]

5 **Moravec, H.** (1984). Locomotion, Vision and Intelligence, MIT Press, Cambridge, MA

06 **Moravec, H.** (1998). When will computer hardware match the human brain, Journal of Evolution and Technology 1, pp. 1-12

7 **Brooks, R.** (1998). ibid

8 **Tolman, F.** (1999). Product modelling standards for the building and construction industry: past, present and future, Automation in Construction, 8:3, pp. 227-235

complex mass production environments like the aircraft and automobile industries. A commonly agreed design methodology together with a transparent set of open, standardized protocols for information storage and exchange will contribute to cooperation, design quality and knowledge management. Systems engineering, supported by a Building Information Model (BIM) and connected worldwide through fast internet connections gain implementation momentum. In the Netherlands the BIR (building information council) stimulates BIM related developments such as COINS,[9] CHEOBS[10] VISI.[11] Most of the big building and construction companies in the Netherlands are involved. A combination of the above mentioned developments is attractive for structural engineers: neutral BIM information management (COINS), formalized problem & solution definitions (CHEOBS) and formalized message exchange (VISI), Market parties such as Autodesk, Nemetschek and Itannex are working to make their products compatible with the emerging COINS standard.

Structural engineers base their designs on information, communication and knowledge. In the near future BIM technology is likely to facilitate a proper ICKT backbone. Structural engineers use specialized tools such as other human experts, pen & paper, spread sheets and Finite Element Method software to manipulate the BIM. These tools will be compatible with specific BIM implementations but the associated human-computer interfaces remain unchanged. Although the current human-computer interaction allows for more humanoid access to information manipulation, they also limit the possible actions to the programmed. The quality of a structural engineering design, level of detail and such largely depend on the human-computer interfaces' flexibility to facilitate abstract high level commands into low level instructions for BIM related information manipulation.

2.2 Problem

In the near future projects actors will have access to 1) a commonly agreed neutral information layer (BIM), 2) a business layer (such as VISI) and 3) a human-computer interfacing layer. Many initiatives address the information and business layers to support efficient information exchange which is a correct approach for BIM development. Meanwhile the human-computer interface layer seems to remain largely unaddressed and decentralized; structural engineers must still use the individual tools' (limited) human-computer interfaces in order to manipulate information.

There is progress in the right direction such as Autodesk's NavisWorks and attempts to put design environments in a shared game setting.[12] Still, the presentation layer does not receive the thorough redesign it deserves in the emerging BIM design environment.

2.3 Objective

'If you buy a drill, you actually want a hole' - Henning Kagermann.[13] In the words of this one-liner the ultimate objective in this UDE (Imprecise and Constraint Human-Computer Interaction) project is to create a digital design supporting environment in which structural engineers have easy access to a variety of information and computational resources (including but not limited to BIM) through natural human-computer interfaces to query and instruct the

[9] **CUR** (2008). Building objects and virtual construction, [www.coins web.nl]

[10] **CROW** (2008). CHEOBS object libraries, [www.crow.nl/Cheobs]

[11] **CROW** (2008b). VISI communication standard, [www.crow.nl/visi]

[12] **Hubers, J.** (2008). Collaborative architectural design in virtual reality, PhD thesis, Delft University of Technology

[13] **Kagermann, H.** (2007). Technisch weekblad 10, pp. 13

environment in both informal (using for example natural language or sketching) and constraint (using for example formal systems such as FEM software) multimodal modes, without the need to know any specifics about individual tools' human-computer interfaces and hidden features. BIM is being addressed by multiple international standardization efforts and it is likely to root within structural engineers' design environment within the next decade; therefore the first objective in the UDE project is to develop a generic presentation layer that fits in the current BIM developments.

New rapid programming environments such as the Borland ECO framework but also the semantic web provide all necessary plumbing to apply business rules on various data sources and to generate a default presentation layer (for debugging purposes). The traditional dependency breakdown structure from presentation layer down to data layer is thus less strict because presentation layer and data layer evolve when business rules change. The second objective in this project is to break the presentation layer's dependency on subsequent layers in order to support lower level systems instead of depending on them (Figure 1). One might call it a fat client because it requires its own set of meta-business rules and meta-data in order to understand the actual business rules and data. Therefore the presentation layer in a sense still depends on subsequent layers, but it is a loosely coupled level that can support multiple BIM environments, not only a specific BIM implementation or a single tool.

Loose coupling is a contradiction between coupling and decoupling two systems.[14] In the case of a structural engineering design environment tight coupling may be expected with design codes but that would require a standard format for saving design codes. Loose coupling can only be expected when UDE can interpret design codes regardless if it is on paper, screen or disk. Loose coupling is on the level of language and understanding; it is fairly independent of the representation's format because a specific format can be decoded by using understanding and language.

Every structural engineering design step is a marriage of conceptual and precise engineering. The third objective in the UDE project is to create a presentation layer that simultaneously supports precise interaction for detailed design and more abstract humanoid interaction for conceptual design. For example NEC Corp developed a home robot 'PaPeRo' that receives spoken instructions in addition to input through a touch screen. The robot interprets the spoken commands and sends the optimal route to a car navigation system. An avatar on the car navigation screen indicates it correctly understood the instructions. The robot was demonstrated at the AT International 2008 at Makuhari Messe in Chiba Prefecture, Japan.[15]

Summarizing, the objectives in the UDE project are:

- Develop a generic presentation layer that fits in the current multiple BIM developments
- Break the presentation layer's dependency on subsequent layers - Fat client (Figure 1)
- Simultaneously support precise interaction for detailed design and more abstract humanoid interaction for conceptual design

---

[14] **Orton, J.** & **Weick, K.** (1990). Loosely Coupled Systems: A Re-conceptualization, The Academy of Management Review, 15:2, pp. 203-223

[15] **AT International** (2008). [techon.nikkeibp.co.jp/english/NEWS_EN/20080724/155369]

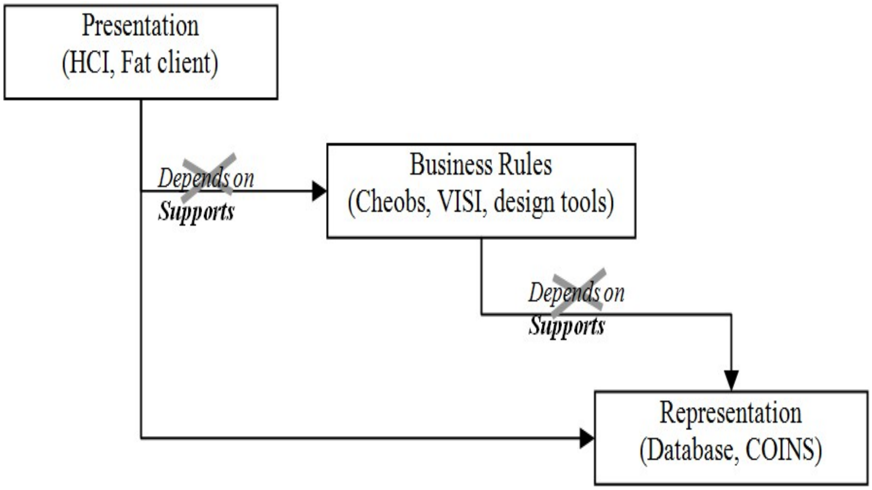

Figure 1 Presentation and business layers support subsequent lower layers rather than depending on them

2.4 Analysis

This section addresses the aforementioned objectives and proposes prerequisites to achieve them. The prerequisites will be applied to the future prototype developments. The current prototype is reported in the next section.

Objective #1: *Develop a generic presentation layer that fits in the current BIM developments*

Prerequisite #1.1: *UDE must have enough knowledge of BIM and structural engineering tools*

According to Reich a machine learning system must strongly rely on an information management system to be successful.[16] UDE operates in a BIM environment, in particular COINS. However a BIM is intended to manage project related information whereas UDE must support higher level tasks such as the process of setting up and managing a BIM. As COINS is a specific BIM implementation, UDE must have general knowledge about BIM and structural engineering related tools; in fact it is an agent that with meta-BIM, meta-tools etc. Its own information management is not a BIM but it is based on machine learning techniques in order to continuously adapt to a changing project environment and provide a useful intermediate layer for a structural engineer (Figure 2).

The structural engineers' and UDE's roles are similar; both attempt to translate imprecise instructions into precise actions (concept to detail). The structural engineer is a technical specialist who translates specifications into designs. UDE is an ICKT specialist that translates the structural engineer's specifications into the digital environment. Besides its programming and other ICKT related skills it must be self and social aware, communicate with the structural engineer using natural language and see the world through visible light. It therefore needs two cameras, a speaker and two microphones. Other senses such as touch and smell are regarded less important for a structural engineer assistant. The observer is not expected to connect to the computer through various protocols; a connection with keyboard and mouse is sufficient.

[16] **Reich, Y.** (1997). Machine learning techniques for civil engineering problems, Microcomputers in Civil Engineering, 12:4, pp. 295-310

17 ibid

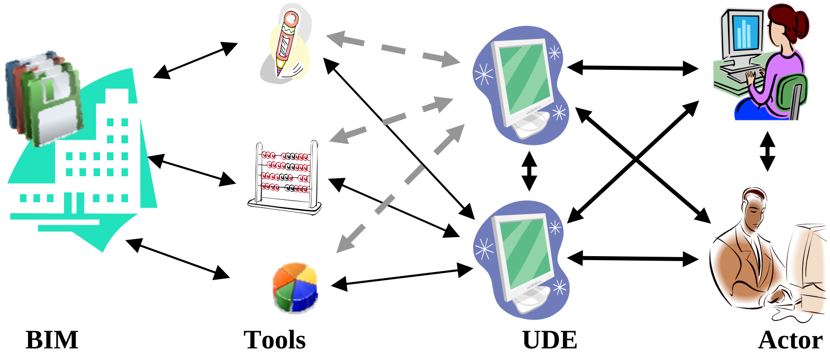

Figure 2 BIM (in our case the COINS implementation) is the backbone for information management; UDE is the front-end

Prerequisite #1.2: *UDE must be a loose coupled fat client system to support a structural engineer in a changing design environment*

Reich also concludes that machine learning in civil engineering is difficult to implement and use, and that most implementations use simple robust decision trees to find the best computational solving algorithms for specific problems.[17] In that respect it is useful for structural engineers to implement general purpose fat clients that have knowledge of most profound problems and additionally provide a general platform – connected with but not relying on the BIM – to distribute and reuse new knowledge without explicitly depending on them.

The objective of BIM is to formalize and unify information exchange between structural engineering tools. Similarly UDE attempts to formalize and unify interfacing between structural engineering tools and structural engineers. It will have much in common with Service Oriented Architectures (SOA) use loose coupling, distributed computing and modularity. In addition a SOA provides a flexible, high-level business layer. The same flexibility might be dangerous as well. In the BC industry this danger will partly disappear with the introduction of BIM; a BIM will ensure low-level model integrity. Therefore UDE may be implemented as a loose coupled fat client system that implicitly relies on a BIM's integrity checking.

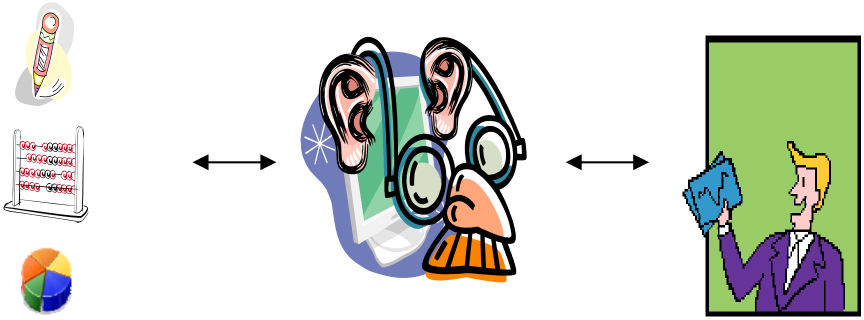

Figure 3 UDE must provide loose coupling both to humans and to structural engineering tools in order to fulfil its intermediate role and to form a meaningful actor in a project

Prerequisite #1.3: *UDE must use general invariant protocols to communicate with structural engineering tools*

It is likely that BIM environments and structural engineering tools will change continuously. A software update in a structural engineering tool may result malfunctioning low-level protocols such as an Application Programming Interfaces (API) or a Software Developer Kit (SDK). However humans generally appreciate updates especially when the frontend has improved. Apparently the Graphical User Interface (GUI) is a more or less invariant way to exchange information with humans. GUI designs share much in common, and in some badly designed applications business logic even resides within the GUI layer. Therefore a safe approach for UDE to tie up all structural engineering tools without bypassing any business logic is to understand and interact with GUI even though interacting with a GUI is slower than through an API. Software developers will continue to create GUIs for their (stand-alone) tools, which will ensure compatibility with UDE without requiring special purpose protocols.

Humans are compatible with each other through their presentation layer. In the case of structural engineering the profound interfacing is through speech and vision. If UDE is to be an actor (agent) it must at least incorporate speech and vision. The importance of talking and drawing during conceptual designs has been stressed by Gustafson et al.[18]

Hoc proposed to change the term Human-Computer Interaction (HCI) into Human-Computer Cooperation.[19] This is true for UDE; it is more than a simple presentation layer, it is an actor that participates in a project. Without any domain specific knowledge it would fail (Figure 3).

Prerequisite #1.4: *UDE must solve single instance, non-standard problems*

With UDE having a rudimentary knowledge of interacting with both humans and computers, it may solve interesting problems as well as enriching existing GUIs. For example, an engineer put 20 forces in a FEM application but needs to change every second force into a line load of which the magnitude depends on the floor thickness. This is a time consuming task and useless to automate for a single instance. UDE will be capable to solve such single instance problems fairly easy through a GUI after being instructed verbally by a structural engineer.

Objective #2: *Break the presentation layer's dependency on subsequent layers (Fat client, loose coupling)*

Prerequisite #2.1: *UDE must rely on strong back-propagation through a cerebellum*

Structural engineers start making designs and calculations depending on a particular situation (problem). It is as if the knowledge in the structural engineer's mind unfolds in that specific situation, and causes him to unchain specific actions. His very own reaction changes the actual situation, and triggers a new design loop. For instance an engineer might write down the formula $M=F*l$ when he sees a cantilever with a load on it. The formula he wrote changed the actual situation within the office and might cause him to fill in the values for F and l. Again the situation changed, causing him to compute the outcome of his own formula. In Artificial Intelligence (AI) this is back propagation. The human brain has a cerebellum which is a sophisticated back-propagation mechanism for high-level commands such as intentions and motor instructions. It compares the initial commands with the actual outcome which it gathers from sensors in the body (and mind), and adjusts the initial plans

---

18 Gustafson, B. et al (2007). Using Talking and Drawing to Design: Elementary Children Collaborating With University Industrial Design Students, Journal of Technology Education 19:1, pp. 19-34

19 Hoc, J. (2000). From human-machine interaction to human-machine cooperation, Ergonomics 43:7, pp. 833-843 accordingly.

In interaction with structural engineering tools' GUI, such a cerebellum might be particularly useful. Having domain-specific knowledge, UDE expects certain graphical results on screen. Error messages or a system crash must trigger it to solve the new sub-problems instead of returning an error itself.

Prerequisite #2.2: *UDE must support situation unfolding*

The unfolding process is common in nature, and formal (non-linear) back propagation systems have such properties. It is similar to a double DNA helix that unfolds under certain chemical conditions to allow an mRNA copy which ultimately produces a single protein or even complete individuals. On top of DNA there are many layers such as proteins, muscles, nerves, brains and personalities. Interestingly enough the layers that actually result from the original DNA cause new chemical conditions for the DNA (back propagation). The enormous potential of unfolding is visible during embryonic growth. Only one of the many stunning examples is the optic nerve. It starts to grow from the eye and diverges into millions of exact locations in the visual cortex of the occipital lobe in both hemispheres. It finds its way through a natural pruning process, which depends on the actual chemical conditions. The process is coded implicitly in the DNA but it certainly does not rely on low-level DNA unfolding at the moment of pruning. Thoughts are other examples of high-level processes with implicit DNA-coding. The final design has very effectively been compressed into the lowest level, based on several assumptions for the unfolding process:

- stimulating situations during the actual unfolding process are present
- the unfolded products or 'bodies' create the necessary situations for the next unfolding step

There is a major difference between biological high-level 'modules' and software modules. A biological module can make decisions on its own level because it is more or less independent of the lower levels it stems from and it has its own body (of course it does rely on its own building blocks, but it does not stem from them). A biological nerve will fire under certain electrical and chemical conditions regardless of its DNA information at that particular moment. At the contrary a software module will fail to execute if only the smallest bit of low-level machine-code it stems from is incorrect or missing. A software module has a strong hierarchy; it heavily depends on machine code for each action. Nevertheless there are many examples in the digital world that support the idea of embodied level of computation. One of them is Google, who indexed almost all of internet to enable reactive and intelligent searching features.

Regard basic knowledge (a formula) as DNA, a situation as the chemical conditions, the intended design as mRNA and the achieved result as a set of proteins. Similar to DNA, knowledge is rather static (as is classicalism in terms of AI). It takes our entire childhood and more to mould basic knowledge such as language and mathematics into our brains. Once achieved, our future learning starts to speed up because of self-reference. We use our own language to explain new words within the same language.

A dictionary is highly cyclic and could cause computer programs to end up in an endless loop when searching for the meaning of a word. Nevertheless humans seem to pile up knowledge without relying explicitly on the underlying lower levels. This has been demonstrated by researching chess professionals. They do not actually foresee hundreds of future moves but instantly see a situation as a familiar chunk of information that falls into place in their brains without knowing the exact positions and without verifying all of the game rules.[20] The same applies to structural engineering situations.

Prerequisite #2.3: *UDE must provide invariance*

One could regard a situation as a query on available knowledge. The knowledge must be suitable to be triggered in a particular situation; it must be invariant. This explains why people get confused when situations change rigorously; the knowledge simply does not fit anymore, and they might have to revert to slower lower levels for a solution. The top layers have the duty to translate high-level external situations back to low-level situations. It causes many simultaneously more or less automated (low level) reactions by the source code, DNA or knowledge. Now the low-level situation has changed, and may cause higher level situations to change again. Note that it is not every internal or external situation causes a full reaction chain. The process is highly non-linear because of recursion. An example is the nerves in our visual cortex that react to light pulses. The external situation causes chemical and electrical reactions to occur at high rates. Such reactions bypass the DNA; they do not require the DNA to instantly unfold and produce a low level instruction for further execution. However, the DNA is still required for the cell maintenance. In a sense it produced a faster mechanism. This is a fundamental difference with computer source code.

Computer systems merely try many alternative options or end up in endless loops when things go wrong. However, people try to find out what went wrong and based on error information. A useful experiment would be to teach a computer to program. It would need some rudimentary knowledge about programming, computer interfaces and such. Then instruct it to program a ‘hello world’. Obviously the first attempts would fail. Now make the computer use that knowledge in order to learn how the programming environment actually works.

Prerequisite #2.4: *UDE must provide continuous learning and training environment*

Dubucs calls design compressing and prediction decompressing. His variant of emergentism is indeed compatible with situation *un*folding and he attempts to add the idea of situation folding.[21] He concludes that ‘there is no shortcut to knowledge of what happens before it happens’. In other words, we must still rely on knowledge extraction (invariance) during projects.

UDE is an interesting platform to provide continuous learning and training. Its specialism is in the presentation layer and interaction, making it an ideal platform to deliver and present relevant just-in-time learning and training material during a project.

Objective #3: *Simultaneously support precise interaction for detailed design and more abstract humanoid interaction for conceptual design*

Prerequisite #3.1: *UDE must combine connectionism and classicalism*

Artificial Intelligence (AI) has two main streams being classicalism and connectionism. Classicalism (also called expert systems) is a top-down approach

[20] **Hofstadter, D. R.** (1979). Godel, Escher, Bach: An Eternal Golden Braid, Basic Books

[21] **Dubucs, J.** (2006). Unfolding Cognitive Capacities, Keio University Press

and assumes that the human mind is computational in a serial way. Some examples are mini-max trees, pre-programmed databases, and prewritten code. Classicalism is quite rigid in the sense that it cannot learn, expand and change easily. It is based on symbol systems and logic. Any flaws in the symbols or logic cause malfunctioning within the system. However it is well suited for precise theorem proving and such. Connectionism (also called neural networks or parallel processing) is a reaction to the classicalism drawbacks. It is based on the mechanism of neurons in our brains. It is very successful in face recognition, motor skills and such but fails to emulate higher level tasks like understanding language. For these tasks they tend to be slow and inaccurate compared to classicalism. One reason is that they often start from scratch whereas classicalism starts with a filled repository. Hadley proposed a logic combination of both: use connectionism at the low neuron-levels, but classicalism at higher levels.[22] It is similar to the ideas of Hofstadter who states that lower levels group up to more intelligent 'chunks'.[23] Indeed it is well known that connectionism models can implement explicit symbol manipulation capabilities (classicalism).

When reflected on a BIM environment, connectionism could facilitate the flexible business layer that wires all information together. Especially when used in conjunction with back propagation it is suited to simulate dynamic non-linear systems superfluously found in the BC industry. However current BIM implementations are still based on databases rather than on neural networks.

Classicalism on the other hand could be the set of tools connected with the BIM that will group information in useful chunks and use it for specific purposes (calculations, drawings, planning, knowledge management etc).

A combination of classicalism and connectionism is necessary to efficiently manage and manipulate a BIM. Humans hardly develop language (classicalism) before knowing relations (connectionism) within their surroundings. This is evident for children; they start using an explosive amount of words after about two years. With BIM being introduced as a neutral connectionism framework it is time to develop classicalism extensions, of which CHEOBS is an example.[24]

Prerequisite #3.2: *UDE must use eigenknowledge*

Computer scientists often use eigenvalues to identify specific images such as faces from a known database of images. For the purpose of faces they use the term 'eigenfaces'. A computer analyzes a large amount of faces and determines the invariance in the faces, resulting in approximately 150 invariant images. All original images are a result of a certain combination of those images. For new faces the system combines the invariant images to form a best lookalike for the new face. Such a system generally achieves 95% accuracy. Similarly 'eigenknowledge' could provide a way to invariantly store and use project knowledge for structural engineers. One can continuously extract invariance from active projects and reuse it in new projects.

In systems engineering and many other design approaches a design starts with a concept and finishes in detail after several design loops. This is true for a complete system (validation) but also for every decomposition level (verification). Connectionism is suited for conceptual design; it can adapt to reflect the actual situation and use 'eigenknowledge'. CHEOBS is an attempt in this direction.

---

22 **Hadley, R.F.** (1999). Connectionism and Novel Combinations of Skills: Implications for Cognitive Architecture, Minds Match. 9:2, pp. 197-221

23 **Hofstadter, D. R.** (1979). ibid

24 **CROW** (2008). ibid

Prerequisite #3.3: *UDE must facilitate precise and imprecise interaction trough vision and hearing*

When observing designers, one can conclude that:[25]

- Structural engineers often use both precise and imprecise interaction, for example writing, drawing, pointing, gesturing, expressing facial emotions, talking, making physical contact and other.
- Structural engineers sit behind a desk in an office environment during most of their contribution to the design process. In that case structural engineers use vision, hearing, talking and their hands rather than other sensory and motor capabilities.
- Structural engineers consult colleagues to discuss or delegate work.
- Structural engineers use design tools such as computers, calculators, measurement tools and writing tools.
- Structural engineers use structuring tools such as a large desk for big drawings, staples, paper piles, binders and notes.

Based on these observations one can conclude that structural engineers need a tool to manipulate information in a constraint environment such as BIM that as to ensure information integrity. On the other hand they need a tool on top of it to translate high-level informal humanoid voice and visual commands to and from the constraint environment. In this regard Ritter and Young report a human computer interfacing test tool based on simulated humanoid interaction.[26] The testing device has eyes and hands and operates on Graphical User Interfaces. The approach is interesting for other domains as well, including automatic car driving and in this case, operating structural engineering related software applications.

With the discovery of the main visual mechanisms by Hubel and Wiesel, it is tempting to regard the visual system as being an incremental line up to understanding scenes.[27] However humans can decode natural scenes extremely fast and in near absence according to Li et al.[28] [29] [30] It seems that the more complex a scene, the better the brain functions. Also since the publication of a paper titled ‘Put that there’, speech recognition has made major improvements but has not yet broken through to everyday human-computer interfacing by the masses.[31] Maybe an activity-based approach in which vision, hearing and talking are not separate units but rather used for specific goals as successfully demonstrated by Brooks, speech and vision might contribute to self-awareness.[32]

Prerequisite #3.4: *UDE must have a clear understanding of models*

Computers and software are not very self-aware. For example a computer program does not understand its own graphical user interface. Maybe the problem lies in the way we create models by assuming that objects exist outside the scope of understanding, of which slow standardization processes might be a proof (e.g. the Industry Foundation Classes initiative by the International Alliance for Interoperability took more than 20 years to evolve and it is still far from complete). Take as an example the definition for a simple cylinder, shown in Figure 4.

Most modellers will regard a diameter plus height representation sufficient. A computer might perform impressive operations on the model (visualization, cost estimation, quantities, planning and other BIM related operations) without knowing what it really is. The diameter and height are in fact concepts, not

25 Also refer to Gustafson, et al (2007). ibid

26 Ritter, F. & **Young, R.** (2001). Embodied models as simulated users: introduction to this special issue on using cognitive models to improve interface design, International Journal of Human Computer Studies, 55:1, pp. 1-14

27 Hubel, D. & **Wiesel, T.** (1968). Receptive fields and functional architecture of monkey striate cortex, The Journal of Physiology 195:1, pp. 215-243

28 Li, F. et al (2002). Rapid natural scene categorization in the near absence of attention, Proceedings of the National Academy of Sciences, 99:14, pp. 95-96

29 Attention and data processing are two different systems according to Posner & Petersen[30]

30 Posner, M. & **Petersen, S.** (1990). The Attention System of the Human Brain, Annual Review of Neuroscience, 13:1, pp. 25-42

31 Bolt, R. A. (1980). Put-that-there: Voice and gesture at the graphics interface, Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques, pp. 262-270

32 Brooks, R. (1998). ibid

merely properties. At our level of knowledge and understanding we are capable to extract the relevant invariant concepts of a cylinder, find a suitable way to model these and create the model itself or new representations (such as this text and the picture above). Not surprisingly, if we decide to add a hole or an extension to the cylinder, the computer is unable to do so even if a hole is defined. First the relation between holes and cylinders must explicitly exist. In other words, the computer does not understand the 'knowledge'; it merely stores compressed human knowledge. Understanding models clearly is a combination of connectionism and classicalism. The Greek philosopher Heraclitus (ca 540–ca 480 B.C.) already found that 'everything flows and nothing abides; everything gives way and nothing stays fixed'.[33]

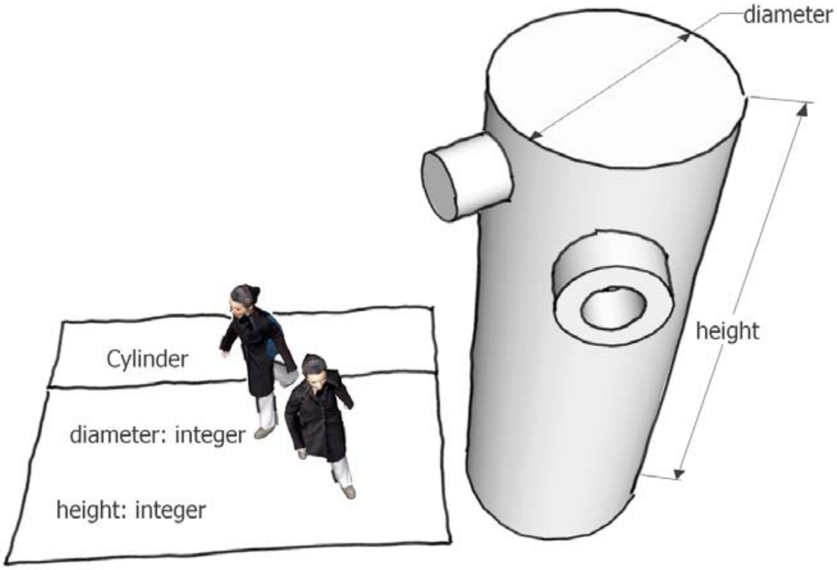

Figure 4 A simple property such as diameter is a complete concept that deserves a model itself. Moreover a single representation for an object or a system will often prove insufficient for a different context. Multi-perception might be a better modelling approach

Within BIM initiatives the term Computable Data is used. Data could be digital but in the wrong format and thus incomputable, such as a 'static' table in a word processor opposed to the same table in a spreadsheet. Unfortunately a BIM is not going to make all data computable because one BIM has a wrong format to another BIM making them incomputable towards each other. A translation definition between the two is as weak a solution as is a translation between a word processor and excel.

Linhares proposed to use multi-perception similar to the rich variety in which humans use specific 'objects' and their ontology (i.e. context).[34] The remaining question is how humans manage to create multi-perception; clearly they must rely on a form of meta-perception. Some argue that this meta-perception in which consciousness is of importance has a non-biological basis because people with near-death experiences report detailed descriptions of their surroundings during their periods of zero brain function.[35] Others attempt to explain these phenomena on a biological basis.[36] Whilst both may be true, the author believes that at least during normal brain function there is sufficient biological basis to

[33] **Wikipedia** (2009). Heraclitus [en.wikiquote.org/wiki/Heraclitus]

[34] **Linhares, A.** (2000). A glimpse at the metaphysics of Bongard problems, Artificial Intelligence 121:1-2, pp. 251-270

[35] **Moody, R.** (1978). Reflections on Life after Life, Bantam Books

[36] **Saavedra-Aguilar, J. & Gomez-Jeria, J.** (1989). A neurobiological model for near-death experiences, Journal of Near-Death Studies, 7:4, pp. 205-222

meet the limited requirements of actors in structural engineering.

3 Summarizing the analysis

The following list summarizes the prerequisites in the analysis:

- Objective #1: Develop a generic presentation layer that fits in the current BIM developments.
  - Prerequisite #1.1: UDE must have enough knowledge of BIM and structural engineering tools
  - Prerequisite #1.2: UDE must be a loose coupled fat client system to support a structural engineer in a changing environment
  - Prerequisite #1.3: UDE must use general invariant protocols to communicate with structural engineering tools
- Objective #2: Break the presentation layer's dependency on subsequent layers (Fat client, loose coupling)
  - Prerequisite #2.1: UDE must rely on strong back-propagation through a cerebellum
  - Prerequisite #2.2: UDE must provide invariance
  - Prerequisite #2.3: UDE must support situation unfolding
  - Prerequisite #2.4: UDE must provide continuous learning and training environment
- Objective #3: Simultaneously support precise interaction for detailed design and more abstract humanoid interaction for conceptual design
  - Prerequisite #3.1: UDE must combine connectionism and classicalism
  - Prerequisite #3.2: UDE must use eigenknowledge
  - Prerequisite #3.3: UDE must facilitate precise and imprecise interaction trough vision and hearing
  - Prerequisite #3.4: UDE must have a clear understanding of models

For the prototype implementation most of prerequisites have not yet been fulfilled. Future improvements are to follow.

4 Prototype setup

This prototype aims to demonstrate a combined Informal and Constraint Human-Computer Interface system (UDE: read 'I See' 'He Sees' in the sense of understanding) while wrapping up tools' specifics and hidden features. Additionally it can integrate learning and training in the production environment. The implementation described in this paper is not the only solution for an UDE; it solely serves to demonstrate the beneficial combination of a precise and imprecise human-computer interface for structural engineers supported by appropriate backbone technology.

4.1 Precise low-level HCI: Multi-touch screen

The first author has built a 1000x500 mm multi-touch screen shown in Figure 5 to serve as a convenient precise HCI. A 3D tacit room-size environment would be more realistic, but the current 2D setup is certainly adequate for the first experiments with basic tabletop interaction: precise information manipulation, tool using and work organizing.

The cheap and easy to setup multi-touch screen is based on Frustrated Total Internal Reflection (FTIR), shown in Figure 6.[37] Light coming from the 940nm IR LEDS enters the acrylic and gets captured within the acrylic because of total internal reflection. Any object on the acrylic, such as a finger, frustrates the total internal reflection. A modified webcam picks up the scattered IR light. The silicone rubber compliant surface layer causes brighter light blobs even with gentle finger pressure. Without this layer little or no FTIR effect will occur since dry fingers or unsuitable material on top of the acrylic hardly bond to the acrylic well enough to change the breaking index between acrylic and air. The Rosco Grey projection sheet is a rear projection screen which blocks some unwanted ambient IR light and displays images coming from the rear beamer.

Figure 5 Multi-touch table impressions, featuring digital equivalent for analogue tools

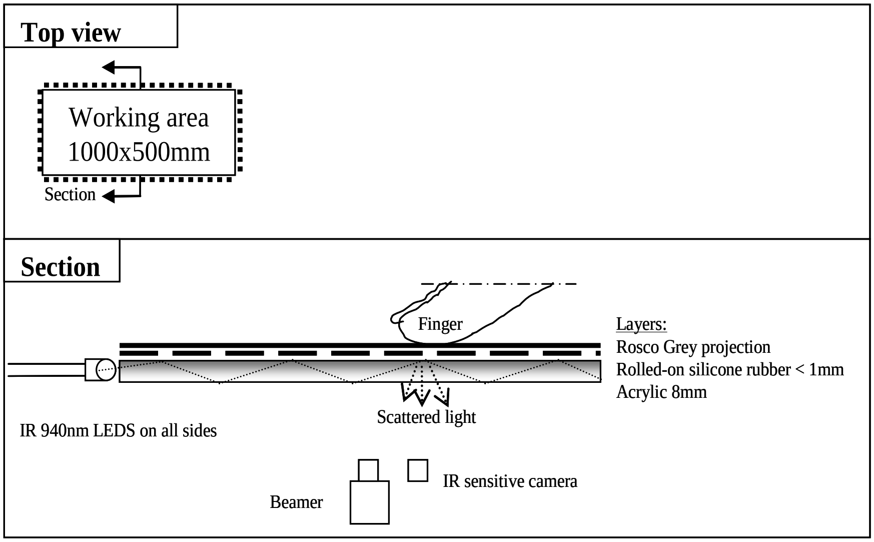

Figure 6 The multi-touch screen based on FTIR

---

[37] **Jefferson, H** (2005). Low-cost multi-touch sensing through frustrated total internal reflection, UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology, pp.115-118

There are many other multi-touch techniques such as Diffused Illumination, Laser Light Plane and Diffused Surface Illumination. Diffused Illumination is the right choice for object, fiducially, and hovering recognition whereas FTIR is better for reliable blob detection. It is possible to combine both techniques or to increase performance by using multiple IR wavelengths. For the prototype the author chose only FTIR because of the reliable blob detection. Fiducial and object tracking is interesting for structural engineers (e.g. for using a physical ruler), and the author might add it in a later stage. More information on (FTIR based) multi-touch screens is on the NUI group website.[38]

The open source Touchlib software processes the webcam images to identify blobs (spots of frustrated IR light). It assigns unique identifiers to the blobs, calculates their coordinates and area, and keeps track of them as long as they remain on the screen (Figure 6).

Figure 7 shows two pictures with the original IR capture and the final result after noise and background removal, which only leaves the blobs. The identity, area and location of the multiple blobs are the basic input for multi-touch enabled applications. Some demos included in the Touchlib software are paint, physics, and photo and music applications.

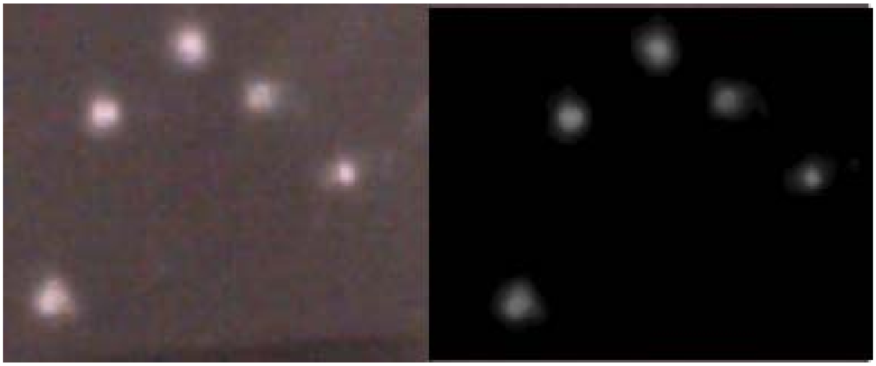

Figure 7 Original image and image after filters and background removal

A large multi-touch screen provides a convenient surface for large or multiple documents such as drawings and calculations. The structural engineers can use natural finger gestures to zoom, pan and rotate through the content. In addition it is possible to use a stylus with a small infrared light mounted in front.

Both Apple and Windows have implemented multi-touch capabilities in their new operating systems. Apple introduced multi-touch on the iPhone, Microsoft sells Surface, and there are various open source (hardware and software) initiatives such as CUBIT. In future multi-touch screens will become flat and may be integrated in electronic paper.[39] Already several tablet PC's have multi-touch capabilities.

4.2 Imprecise high-level humanoid interaction: Levente

In the Journal of Technology Education, Gustafson et al emphasize the importance of combined drawing and talking to develop, express and change ideas.[40] They conclude that talking and drawing are tools for collaboration and for thinking about design. Furthermore Shuji Hashimoto, director of the

[38] **NUI Group** (2008). Natural User Interface (NUI) Group [www.nuigroup.com]

[39] **iRex** (2007). iLiad Electronic Paper Display [www.irextechnologies.com/ products/iliad/features]

[40] **Gustaaffson, B.** et al (2007). ibid

humanoid robotics centre at Waseda University in Tokyo, states that what robots need is *kansei* which is Japanese for a raft of emotional notions, including feeling, mood, intuitiveness and sensibility. Kansei is the link with humans and is essential for understanding interaction properly.

Figure 8 Levente, our digital assistant for structural engineer

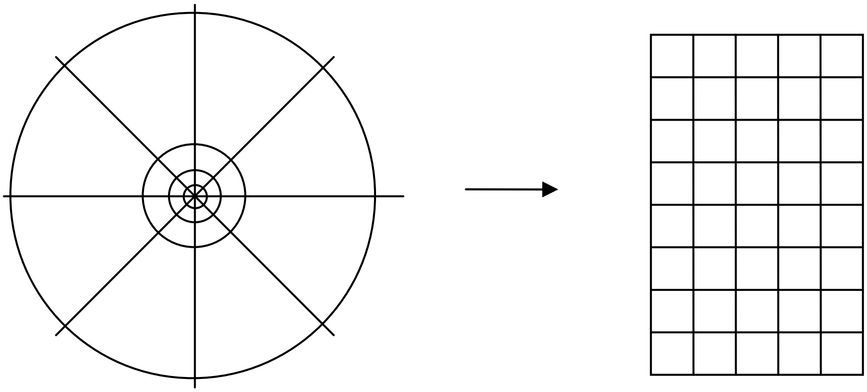

Figure 9 Log-Polar Mapping enables foveal vision and allows easy extraction of edges and movements. When cut open and unfolded, one can map this representation onto a Cartesian grid

This necessity for humanoid, imprecise interaction was the motivation to introduce Levente, my first humanoid prototype (Figure 8) into the setup. The reason to choose for a physical android is that its existence emphasizes humanoid presence, which is beneficial for interaction. Besides, it can interact with other actors and with the real world using motor and sensor components. Other researchers have built digital characters with realistic facial expressions which will admittedly result in more realistic facial expressions.[41] In Canada the project Aiko resulted in a female android. Aiko shows facial expressions, can mimic pain, react to touches, talk and listen, see and even solve mathematical problems displayed visually. It uses the Biometric Robot Artificial Intelligence Neural System (B.R.A.I.N.S.) software to learn. The android in my project is far from being that sophisticated, but only demonstrates the benefits of UDE for structural engineers. Again other researchers built an anthropomorphic robot head for studying autonomous development and learning of vision systems.[42]

[41] **Burford, D.** & **Blake, E.** (2001). Face-to-Face Implies no Interface, 2nd South African Conference on Human-Computer Interaction (CHI-SA2001)

[42] **Kim, H.** et al (2004). Design of an anthropomorphic robot head for studying autonomous development and learning

Levente’s head is made of epoxy fibres (for the time being non-deformable, thus without expression) to allow for two cameras, two microphones and a speaker. Levente must be able to turn his head and eyes (not yet implemented) in order to observe and focus on specific parts of the structural engineer’s working area as well as his own.

4.3 Vision using log-polar mapping

Levente will use Log-Polar Mapping (LPM) for vision (Figure 9). The primate retina has a high resolution fovea surrounded by incrementally lower resolution towards the perimeter. This space variance maps to a space invariant representation in the visual cortex. LPM is known to approximate the space variance, and it is particularly suitable to equip active vision robots with a detailed focus while maintaining a wide view.[43] Compared to a Cartesian representation it consumes far less processing resources because of the logarithmic increment in pixel size, and it is fairly easy to apply algorithms that result in invariantly finding straight lines, circles and movement. A binocular setup is beneficial in a 3D environment when tracking objects in the horopter. LPM uses a fovea and sacrifices peripheral resolution, so saccades (rapid eye movements) are needed to build up the complete image representation. Kim et al produced a low cost but well performing active vision system based on two firefly cameras and a 9600 bits/s servo controller that saccade-like eye movements.[44] Saccades are small movements that last about 20 to 200 milliseconds and reach velocities of up to 1000 degrees per second. One reason for saccades is that a static image on the retina would vanish from vision after a few seconds since rods and cones only respond to variations in illumination. Micro-saccades are constant eye vibrations of about 60 Hz that ensure constant image refreshing. Larger coordinated saccades allow the fovea rather than the outer less sensitive areas of the retina to receive interesting parts of a scene. Several types of complex cells in the visual cortex only respond to moving objects, and will therefore benefit from saccades. Kikuchi and Fukushima implemented a model that allows for saccadic eye movements.[45] LPM is available in various mathematical packages such as Matlab. Bidirectional log-polar mapping using back propagation (adjusting weights of internal hidden nodes) is a better mimic of the visual system but obviously takes more computational resources.

Various vision VLSI (Very Large Scale Integration) research groups (MIT, SYNERGY) built advanced vision chips that are based on primate or less complex insect vision concepts.[46] The company Synaptics created a silicon chip which is an artificial replication of the retina visual processing layer, including digitally controlled analogue processing.[47] This company recognizes the necessity of massive distributed parallel processing to analyze the approximately fifty billion bits per second (6 GB/s!) of binocular data flow.

Cheaper but less performing alternatives use standard webcams and extract LPM representations from them. In order to avoid mechanical movements it is an option to apply a fish-eye camera and read only the LPM part of interest pixels to simulate eye rotation, tilt and even the fast saccade movements. For moving cameras the *Plustek Opticam M1* or the *Typhoon Motion Cam* are an option. Head rotation and tilt requires two motors.

[43] **Jurie, F.** (1999). A new log-polar mapping for space variant imaging, Application to face detection and tracking, Pattern Recognition 32:5, pp. 865-875

[44] **Kim, H.** et al (2000). ibid

[45] **Kikuchi, M.** & **Fukushima, K.** (2000). Pattern Recognition with Eye Movement: A Neural Network Model, Neural Networks, IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference

[46] **Etienne-Cummings, R** (2001). Biologically Ins-pired Visual Motion De-tection in VLSI, Interna-tional Journal of Compu-ter Vision 44:3, pp 175-198, Kluwer Academic Publishers

[47] **Moini, A.** (1998). Vision chips or seeing silicon, A. Moini

4.4 Speech using automated speech recognition and text-to-speech

For speech interaction the author uses Dragon Naturally Speaking (DNS) together with a Software Developer Kit and some additional tools to manipulate the active vocabulary and commands. This commercially available software can 'listen' to voice input and 'talk back' through Text-To-Speech commands. A silent environment, high quality microphones and a trained user contribute significantly to better speech recognition. DNS does not include any software for intelligent conversations.

DNS performs well even though it uses large vocabularies. Performance is particularly good in a noise free environment, proper noise cancelling microphones and a properly trained user.

4.5 Hardware problems

The software implementation for Levente is a far bigger challenge than the software for the multi-touch screen. In fact the ultimate goal is to create a human equivalent android or better. But for such a system, modern computers are far from adequate, mainly because of serial processing. The human brain houses about hundred billion parallel operating neurons, and each neuron is wired to 1000 other neurons on average with each connection firing at a maximum of about 200 times a second. According to Forrester Research, there are approximately 1 billion personal computers in the world by the end of 2008. In order to create an internet brain, that would require us to wire each computer in the world to 100.000 other computers according to the brain architecture through the internet and maintain a maximum of 200 bytes a second on each connection. This sums up to 20 million bytes = 2,5MB/s for each computer, which is quite plausible with modern fast ADSL connections. Maybe it is even possible to use the concentration of countries to represent cortex areas. With the help of Moore's law and ever increasing amount of computers that are connected to the internet through ADSL, this scenario might gain plausibility. However, the result would disappointingly be only a single or very few virtual personalities. Neurocomputers are better fit for neural networks as they are based on distributed parallel processing.

5 Benefits for structural engineers

The design environment may reduce the complexity of today's design environments to a large electronic paper-like display and a humanoid assistant. Since Levente is a separate observer equipped with hearing, speech and vision, it can receive commands such as 'please put these sketches in a new document with our standard cover sheet and report the force distribution as well'. The sketches may reside on a piece of paper, a computer screen or on the touch screen. Format is only of partial importance. Levente must search for the appropriate tools to accomplish the tasks and not bother the structural engineer with details such as crashes, wrong versions and such.

Summarized, the main benefits are:

- hidden complexity of specific tools
- both abstract humanoid interaction and constraint precise interaction depending on the task
- no need to comply with specific quality systems, design approaches, BIM

and such since Levente hides that complexity

- problems with exchanging information are dealt with by Levente. Specific formats and standardization are dealt with by Levente and his ‘colleagues’

6 Conclusions and recommendations

Despite the lack of time, budget and computing power for my own project, it still is possible to build a very rudimentary responsive humanoid which is not the state of the art but still serves demonstration purposes well enough.

For both the multi-touch table and Levente, computing power is one of the major issues when it comes to vision. The Touchlib software consumes most of the computing power to process the IR images. However it can send the blob coordinates through a UDP connection, which allows using a dedicated computer for blob detection. Each demanding component in the prototype can reside on a dedicated computer. A minimum of 5 computers with reasonable specifications are needed for the system to be sufficiently reactive: one for blob tracking, one for applications that run on the multi-touch table, one for DNS, one for LPM and one (most probably many more in future) for Levente’s reasoning.

A multi-touch setup is easy to scale up. The impression in figure 10 shows three large screens with three beamers that allow multi-touch and rear projection. This setup is a fairly simple and cheap way to create a combined virtual reality/multi-touch environment based on open source. The user can easily switch between 2D and 3D design environments, which often occurs in structural engineering. Multi-touch applications that are designed for structural engineers are very rare, if they exist at all.

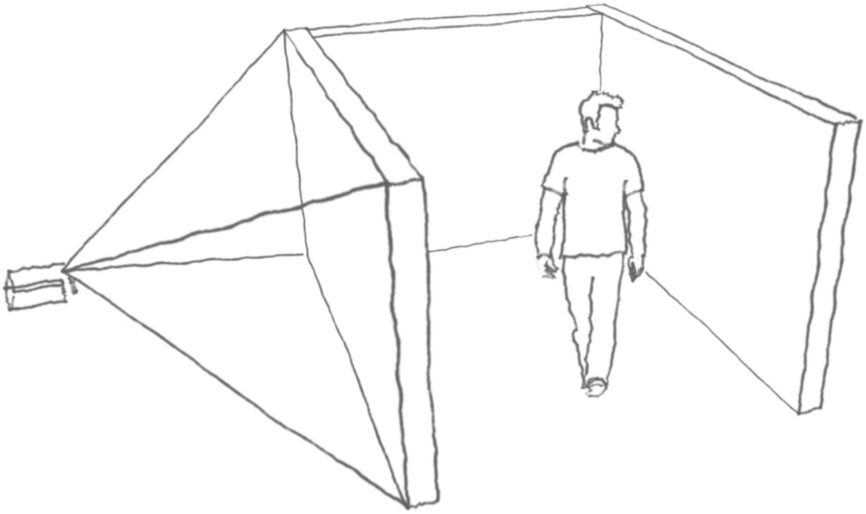

Figure 10 Vertical multi-touch screens combined with virtual reality

The most important part of a human friendly design system is not a reliable information backbone neither is it a fancy precise human-computer interface device. It rather is the intelligent, sensitive and responsive humanoid actor that can read precise or imprecise information from various sources, including humanoid.

The prototype is still a far way from solving Bongard et al problems.[48] Basically it only implemented the conceptual idea of a loose coupled generic and humanoid human-computer interface for structural engineers, both for conceptual and detail designs. However a lot of work must be done to fulfil all of the proposed prerequisites.

This discussion did not address artificial intelligence nor does it give any detailed implementation guides for the UDE design environment. The multi-touch screen is fairly easy to build using established technology while the humanoid Levente is far more complex. There are many difficult issues to be solved including consciousness, multi-perception, incremental layers of intelligence, intelligence without representation and hardware. Basic hardware such as cameras and microphones can still not compete with their biological equivalents.

[48] **Bongard, M.** et al (1970). Pattern Recognition, Spartan Books

Acknowledgement

An earlier version of this paper is presented at the 'Innovations for Building and Construction' international conference, organised by the Design Research Foundation and EuropIA.[29] This paper presents ideas developed by the first author within the framework of a PhD research project at the Design and Construction Process section of the faculty of Civil Engineering and Geosciences, Delft University of Technology.

Bibliography

Bolt, R.A. (1980). Put-that-there: Voice and gesture at the graphics interface, Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques, pp. 262-270

Bongard, M., Hawkins, J. and **Cheron, T.** (1970). Pattern Recognition, Spartan Books

Brooks, R. (1998). Intelligence without representation, Cognitive Architectures in Artificial Intelligence: The Evolution of Research Programs

Brooks, R.A., Myopoulos, J. and **Reiter, R.** eds (1991). Intelligence without Reason, Morgan Kaufmann publishers, San Mateo, USA, Sydney, Australia

Burford, D. and **Blake, E.** (2001). Face-to-Face Implies no Interface, 2nd South African Conference on Human-Computer Interaction (CHI-SA2001)

CROW (2008). CHEOBS object libraries, available at: http://www.crow.nl/cheobs

CROW (2008b). VISI communication standard, available at: http://www.crow.nl/visi

CUR (2008). Building objects and virtual construction, available at: www.coinsweb.nl

Dubucs, J. (2006). Unfolding Cognitive Capacities, Keio University Press

Etienne-Cummings, R (2001). Biologically Inspired Visual Motion Detection in VLSI, International Journal of Computer Vision 44:3, pp 175-198, Kluwer Academic Publishers

Gustafson, B., MacDonald, D. and **Gentilini, S.** (2007). Using Talking and Drawing to Design: Elementary Children Collaborating With University Industrial Design Students, Journal of Technology Education 19:1, pp. 19-34

Hadley, R.F. (1999). Connectionism and Novel Combinations of Skills: Implications for Cognitive Architecture, Minds Mach. 9:2, pp. 197-221

Han, J.Y. (2005). Low-cost multi-touch sensing through frustrated total internal reflection, Proceedings of the 18th annual ACM symposium on User interface software and technology, pp. 115-118

Hoc, J. (2000). From human-machine interaction to human-machine cooperation, Ergonomics 43:7, pp. 833-843

Hofstadter, D. R. (1979). Godel, Escher, Bach: An Eternal Golden Braid, Basic Books

Hubel, D. and **Wiesel, T.** (1968). Receptive fields and functional architecture of monkey striate cortex, The Journal of Physiology 195:1, pp. 215-243

Hubers, J. (2008). Collaborative architectural design in virtual reality, PhD thesis, Delft University of Technology

iRex (2007). iLiad Electronic Paper Display, available at: http://www.irextechnologies.com/ products/iliad/features

Jurie, F. (1999). A new log-polar mapping for space variant imaging. Application to face detection and tracking, Pattern Recognition 32:5, pp. 865-875

Kikuchi, M. and **Fukushima, K.** (2000). Pattern Recognition with Eye Movement: A Neural Network Model, Neural Networks, IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference

Kim, H. et al (2004). Design of an anthropomorphic robot head for studying autonomous development and learning

Li, F., VanRullen, R., Koch, C. and **Perona, P.** (2002). Rapid natural scene categorization in the near absence of attention, Proceedings of the National Academy of Sciences, 99:14, pp. 95-96

Linhares, A. (2000). A glimpse at the metaphysics of Bongard problems, Artificial Intelligence 121:1-2, pp. 251-270

Moini, A. (1998). Vision chips or seeing silicon, A. Moini

Moody, R. (1978). Reflections on Life after Life, Bantam Books

Moravec, H. (1984). Locomotion, Vision and Intelligence, MIT Press, Cambridge, MA

Moravec, H. (1998). When will computer hardware match the human brain, Journal of Evolution and Technology 1, pp. 1-12

NUIGroup (2008). Natural User Interface (NUI) Group, available at: http://www.nuigroup.com

Orton, J. and **Weick, K.** (1990). Loosely Coupled Systems: A Reconceptualization, The Academy of Management Review, 15:2, pp. 203-223

Posner, M. and **Petersen, S.** (1990). The Attention System of the Human Brain, Annual Review of Neuroscience, 13:1, pp. 25-42

Reich, Y. (1997). Machine learning techniques for civil engineering problems, Microcomputers in Civil Engineering, 12:4, pp. 295-310

Ritter, F. and **Young, R.** (2001). Embodied models as simulated users: introduction to this special issue on using cognitive models to improve interface design, International Journal of Human Computer Studies, 55:1, pp. 1-14

Saavedra-Aguilar, J. and **Gomez-Jeria, J.** (1989). A neurobiological model for near-death experiences, Journal of Near-Death Studies, 7:4, pp. 205-222

Tolman, F. (1999). Product modelling standards for the building and construction industry: past, present and future, Automation in Construction, 8:3, pp. 227-235

Instructions for Authors

All papers are reviewed by at least two reviewers. All reviewed and accepted papers have to be resubmitted, implementing reviewers and editors comments or suggestions. Only accepted papers conforming to instructions will be considered for publication in the *International Journal of Design Sciences and Technology*.

The first page of the paper must contain the full title of the paper as well as the name, affiliation, address, telephone, fax and email of the main author and co-authors (if applicable). Also mention the name, postal address, telephone, fax and email of the author to whom all correspondence to be directed.

The second page should contain the full title of the paper, the sub-title (if any), an abstract of 100 to 150 words summarising the content of the paper and 3-5 keywords for the purpose of indexing (the use of references in the abstract is discouraged). Maximum length of a long paper is 7000 words (4000 words for short papers). Please note that the use of Footnotes and endnotes are discouraged. The layout of the journal allows the use of 'side notes' (see a sample issue of the journal). Where appropriate give information for the 'side notes' (maximum length 60 words) between double square brackets (such as full bibliographic reference, page numbers, remarks and notes). All side notes should be numbered consecutively. For instance: [[17 A 'side note' reflects additional information, a reference or the URL of a website]]

The paper will be written in (UK) English. It will be single-spaced with 30 mm margins on all sides (paper size A4). Use Times New Roman for the main body of text (size 10), figures (size 8) or tables (size 8). The use of **Bold**, *Italics*, ALL CAPS, SMALL CAPS, etc. is discouraged. All chapters should be numbered consecutively (more than one level sub-headings are discouraged). All Figures and Tables with their respective captions should be numbered consecutively. They should each, be placed on a separate page, at the end of the paper. All figures, tables and equations should be mentioned in the body of text. Give an approximate insertion point for figures and tables, between double square brackets. For instance: [[insert Figure 5]]. You will be asked to resubmit your drawings if necessary. **Do not layout your paper**. *Do not use any styles or any automatic layout system.* Please do not use 'Track Changes'.

All Tables should be referred to in the text as Table 1, Table 2, etc. All Figures should be referred to in the text as Figure 1, Figure 2, etc. Line drawings should be of good quality. Use light background if possible (white is preferred). Photographs and screen-shots should also be submitted separately as JPEG files (use high resolution for better results). Authors should prepare high quality figures and drawings. Avoid the use of colours in your illustrations, as the journal is not published in colour. Maximum width and height of a figure are respectively 115 (150 mm if absolutely necessary) mm and 190 mm. Maximum width and height of a table are respectively 115 mm (150 mm if absolutely necessary) and 170 mm. All Equations will be numbered consecutively and should be mentioned in the text.

Use 'Harvard System of Reference'. Bibliography (references) is collected at the end of the paper, arranged in alphabetical order by the first author's surname, followed by initials. All authors should be mentioned. Dates will appear between brackets after the authors' name(s). This is followed by the title of the book, name of the publisher, place of publication and page numbers (if applicable). To refer to a journal paper, add the full title of the journal followed by Volume:Issue Number and page(s). Examples of references to a book, a journal or a website are shown below:

> **Dixon, N.M.** (2000). Common Knowledge: how companies thrive by sharing what they know, Harvard Business School Press, Boston, MA
> **Buxton, W.** (1997). Living in Augmented Reality: Ubiquitous Media and Reflective Environments. In: Finne K., Sellen A. and Wilber S. (eds). Video Mediated Communication, Erlbaum, Hillsdale N.J., pp. 363-384
> **Beckett K.L. and Shaffer D.W.** (2004). Augmented by Reality: The Pedagogical Praxis of Urban Planning as a Pathway to Ecological Thinking, University of Wisconsin, Madison
> **Djenidi H., Ramdane-Cherif A., Tadj C. and Levy N.** (2004). Generic Pipelined Multi-Agents Architecture for Multimedia Multimodal Software Environment, Journal of Object Technology, 3:8, pp. 147-169
> **Gorard, S. & Selwynn, N.** (1999). Switching on to the learning society? Questioning the role of technology in widening participation in lifelong learning, Journal of Education Policy, 14:5, 523-534
> **Blackman, D.A.** (2001). Does a Learning Organisation Facilitate Knowledge Acquisition and Transfer? Electronic Journal of Radical Organization Theory, 7:2 [www.mngt.waikato.ac.nz/Research/ ejrot/Vol7_1/Vol7_1articles/blackman.asp]
> **World Bank** (2002). Social assessment as a method for social analysis, World Bank Group [www.worldbank.org/gender/resources/assessment/samethod.htm]

Papers in their definitive version will be submitted as a MS Word file for the PC (MS Word RTF format for Macintosh). In addition, a formatted version of the paper (including images and tables) will be submitted in PDF format. Submit your paper as an email attachment to the Editor-in-Chief [M.R.Beheshti@ TUDelft.NL]. You can also send your paper and images on a CD-ROM by an International Courier to:

The Editor-in-Chief
International Journal of ***Design Sciences and Technology***
Europia Productions
15, avenue de Ségur
75007 Paris, France

Author(s) have to complete, sign and return a *Copyrights Transfer Form* to the publisher. This copyrights transfer assignment will ensure the widest possible dissemination of information. Papers published in the International Journal of *Design Sciences and Technology* cannot be published elsewhere, in any form (digital, paper-based or otherwise) without a prior written permission from the publisher.

The author(s) are responsible for obtaining permission to utilise any copyrighted material. For more details about this subject, please contact the publisher at an early stage.

The decision of the Editor-in-Chief on all matters related to the International Journal of Design Sciences and Technology including the review process, publication of papers, etc. is final and cannot be disputed.

The leading author of a paper published in the International Journal of *Design Sciences and Technology* will receive a digital copy of the author's paper free of charge. Printed copies of the paper (minimum 50) and the journal can be purchased from the publisher (ask for an invoice from the address above or DST@europia.fr).

Subscription Order Form

International Journal of ***Design Sciences & Technology***

Please use BLOCK letters

Title, Initials and Name:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Organisation:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Postal Address:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Postcode:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

City:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Country:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Telephone number:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Fax number:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

E-mail:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Institutional (libraries and organisations):
☐ 1 volume (2 issues per year) € 90.00 (Only in France)
☐ 1 volume (2 issues per year) € 100.00 (Outside France)

☐ 1 volume (2 issues per year) € 55.00 (Personal Subscription Only)

☐ Payment enclosed by cheque drawn on a French bank
☐ Payment to be sent in Euros via a banker's draft

Please ensure that your bank covers any transfer charges.

Signature: Date:

Please make cheques payable to: **Europia Productions**
Account N° 30002 00442 0000006991Z 58 Banque CREDIT LYONNAIS
Agence Paris Marceau 44 Avenue Marceau, 75008 Paris, France

Complete and return this Order Form to:
Europia Productions, 15, avenue de Ségur, 75007 Paris, France
Tel +33 1 45512607 - Fax +33 1 45512632 E-mail: dst@europia.fr

International Journal of
Design Sciences and Technology

Editor-in-Chief: **Reza Beheshti** and **Khaldoun Zreik**

Volume 16 Number 1

Issue Editor: **Edwin Dado**

Table of Contents

1630-7267