DEVELOPING A FRAMEWORK FOR LINKING DESIGN INTELLIGENCE FROM MULTIPLE PROFESSIONS IN THE AEC INDUSTRY

DOMINIK HOLZER, YAMIN TENGONO
RMIT, Australia

and

STEVEN DOWNING
Arup, Australia

Abstract. The research presented in this paper addresses the issue of bridging conceptual differences between the theories and practice of various disciplines in the AEC industry. The authors propose an application logic that works as a framework in assisting the evaluation process of multidisciplinary design. Parametric design templates assist in channeling and integrating information generated during the design process.

1. Introduction

The authors of this paper draw from experience in practice as well as academic discourse. They illustrate the development of a collaborative design framework which is extending the capabilities of a basic model which has previously been developed in the professional organisation (Arup) they are currently embedded in. It is assessed if synergies can be found between the proposed framework and existing efforts in the field – in particular the Building Information Model (BIM) through the application of Industry Foundation Classes (IFCs). The questions raised in this paper are of serious concern for building practice as well as building up epistemological knowledge in the field.

1.1. COMBINED PROFESSIONAL / ACADEMIC RESEARCH

It has shown highly beneficial in the conception of this paper that the authors are part of a team of professionals and researchers from architectural, IT/engineering and computer science background. Next to their commitment

A Dong, A Vande Moere & JS Gero (eds), CAADFutures'07, 303-316.

within academia, they are associated with a large multi-disciplinary design firm, which is acting on a global level. By working in this collaborative environment, the authors have been able to conduct applied research in order to bridge the gap between experimental academic discourse and requirements from everyday practice.

Due to the nature of the AEC industry, where teams reconfigure on a project-basis, few systems are in place to help transfer knowledge gained on projects across disciplines to foster systemic innovations. (Taylor and Levitt 2004) Professional organisations often face difficulties in developing procedural as well as organisational memory from the projects they carry out for building up epistemological knowledge in practice (Acha, Gann and Salter 2005). Preliminary outcomes of the research undertaken by the authors suggest that being in the centre of often contradicting layers of interest from different parties is not an easy position to be in, but it is at this cusp where the most challenging issues get raised.

1.2. RESEARCH AIM

One problem identified by the multidisciplinary organisation the authors are embedded in, is the insufficient interfacing capability of digital tools within the company and across their industry partners. As described in the 2006 *Report on integrated practice* of the American Institute of Architects (AIA), feedback from specialists to the designers in the AEC industry only occurs *at discrete points with varying frequency* (Bedrick and Rinella 2006) which causes delays and discontinuities in the workflow and consequently is responsible for coordination errors and the necessity to rework. Current specialist software tools of the individual professions in the AEC industry are not well suited for facilitating collaborative design, planning and construction (Kannengiesser and Gero 2004). An integrated workflow is depending on data exchange formats and capabilities of communicating design intelligence in a clearly understandable fashion across teams for feedback and evaluation purposes. (Chaszar 2003)

In response to the requirements mentioned above, the authors have focussed their investigation on the development of a domain independent framework for linking information through an application logic that allows for exchange of design intelligence by profession specific modules.

2. Research Interviews

The development of the application logic has been preceded by a series of research interviews which have been conducted by the authors within Arup to ensure that the objectives of the framework are informed by the requirements of practice. The interviews address the current problems and future expectations of practitioners in various fields of the AEC industry in

regard to interoperability and the sharing of design intelligence. A representative cross-section of specialists has been chosen as interviewees, ranging from drafters, expert designers, team leaders, program developers and directors at Arup. The interviews were structured in three parts:
In the first part, practitioners were asked to comment on their design-deliverables and in particular elaborate on those tasks which appeared most repetitive and time-consuming to them.

– The more junior interviewees within Arup see their ‘deliverability’ mostly affected by with time- (and cost) pressure of getting information ‘out’ in the most streamlined fashion possible, while those in leading position understand a quest for broader collaboration across the AEC industry as the main potential of the integration of digital tools
– 70% of all interviewees refer to accommodating changes made by other parties as the most time-consuming part of their work, it is criticised that distinct applications require their own datasets.

The second part of questions dealt with the tools the practitioners were using and possible future requirements. In particular the integration of digital tools in the work-process and their data-interfacing-capability was investigated.

– Answers derived in this question give an indication that the practitioners are increasingly relying on 3D model information to produce their design output. In this regard a better integration of information within and across disciplines is mentioned as the main concern in order to get more intelligence out of the 3D model.
– It is acknowledged by all practitioners that increased interoperability of tools has to go hand in hand with a better dialogue with other parties involved. Only if the roles and contributions of others are understood, the knowledge transfer can be matched with according data-interfaces.

The third part of the interviews the practitioners were asked to comment about roles they would imagine an interoperability framework could perform in order to substantially foster collaborative design processes beyond the sharing of data.

– In response to the point made in section two, 30% of the interviewees state that it would be fundamental to the framework to implement a section that allows various users from the AEC industry to gain better understanding about the rules others are applying to finding their design solution.
– Addressing project specificity, some of the practitioners argue that the framework would have to be something that can be tweaked for different projects as requirements change from case to case
– In particular the more senior practitioners are critical about implications within the framework which aim at the automation of design. They argue that flexibility needs to be maintained so that expert input can occur at any time to avoid ‘black box’ scenarios.

– Those interviewees with an IT background and those being more technically inclined, point out that the framework should accommodate different plugin-modules to break things down into logical sub-schemas and to represent the profession-specific input of various user-groups.

Lessons learned in the evaluation process of the interviews have shown that the framework is expected to be open for change and individual adaptation from project to project. The users' capability of customising individual modules for interaction will be a necessary quality of the framework as well as transparent definition of rules for finding design solutions.

3. Linking Building Performance to Design Optimisation

3.1. INTERACTION AND AUTOMATION

The research team has investigated methods which assist in the creation of an iterative process where informed decisions can be made by professionals on the basis of performance aspects of the design. Methods for fostering the automation of geometry updates, data transfer, design analysis and code-checking of design were explored.

3.1.1. Decision Support

Previous investigations (Mueller 2006) illustrate that automated processes facilitated by computational means need to be integrated carefully with conventional (non computational) work methods as they can lead to *black box* results where team members have little insight in how any particular result was generated. When using automated processes in design optimisation, it often is crucial to understand how certain results were derived (how they evolved) for rapid interpretation and consequent design decision making (Baldock 2004). If a project team has access to automation routines at any point in the design process, members of that team can guide the direction of the optimisation to propose alternative design solutions. At the same time such methods enable the recording of *information trails* for showing how design decisions impact long term goals (Onuma 2006).

3.1.2. Parametric Geometry and Design Updates

In order to allow for designers to engage in an iterative process between performance analysis, optimisation and design decision making, the time factor is of high importance. The more immediate results can be communicated across a team, the better the information-flow and the collaborative capabilities. In this context, changes need to be adopted quickly and integrated into a flexible geometrical setup on the spot without requiring lengthy redraws. In the 2006 AIA *Report on integrated practice*

Eastman describes how parametric modelling enables to integrate and encapsulate the combined expertise of individuals into a design tool (Eastman 2006). Instead of working on a fixed geometrical template, parametric models allow incorporating various design intentions that *persist over geometric variations* (Schelden 2006).

Depending on the level of resolution required and the type of parameters chosen, parametric modelling offers manifold possibilities for addressing a range of issues at different levels of precision from the design ideation phase up to construction. In this context Aish speaks of parametric design as a way to *progress from intuition to precision*. (Aish 2005)

Once an initial desired form is agreed, it may be encoded within a parametric model and used to generate geometry through evolutionary means. Examples of this are present in the research undertaken by combining *EifForm* and *Custom Objects*, where a generative design tool is integrated with a parametric design environment to generate an array of possible solutions for a complex structure. (Shea, Aish and Gourtovaial 2003). An open source platform for collaboration which is taking reference to this principle is presented by the 'Open Source Architecture' group with the 'Hylomorphic' project. The *modes of operation* proposed for the project include the translation of knowledge into exchangeable data, the filtering of information into specific parameters for an architectural object and an iterative evaluation process (Sprecher, Chandler and Neuman 2006)

Recent approaches to standardise the exchange of parametric models and product data as described in the ISO 10303 Standard of the Exchange of Product Model Data (STEP) have shown that the attempt to do so is difficult even with the domain of exchanging between parametrically capable software, and within a single discipline due to accuracy problems and differences in modelling methodology (Eastman 1999) (Pratt and Kim 2006).

4. Precedence from within the Profession

The authors have taken into consideration existing investigations that have been undertaken by members of the professional organisation within which the authors are embedded. A previously developed Collaborative Design Framework (basic CDF) has successfully been applied within the organisation while working on a stadium project. The research team has analysed the basic CDF in regard to its content, its relevance to the research undertaken by the team, and possible changes/alterations to better fit the objectives of the team.

4.1. THE BASIC CDF

The basic CDF is a Windows based (VB.Net) application which has been developed to interact with CATIA/Digital Project as a parametric modelling engine and Oasys GSA as a structural analysis engine. It contains a number of inbuilt modules for checking and optimising of a structural design.

First, the CDF requires a parametric model-template to be created in CATIA with careful adherence to a predefined naming-scheme for the elements in the CATIA model. A series of attributes required for structural design are then applied to the model by importing these attributes from a spreadsheet (using the element names as identifiers in the spreadsheet). These attributes include such data as design loadings, design conditions for any particular element/node and appropriate checks to be applied to that element.

Once the CATIA model has been built, the basic CDF application interacts with it through a COM (Component Object Model) interface to extract that information and a structural engineering analysis model can be created using that geometry and the associated attributes. The structural analysis model is built in GSA which too can export results to the basic CDF through a COM interface. Using CATIA and the basic CDF application to generate the structural analysis model in this method allows the engineer to quickly re-create an analysis model whenever the defining geometry in CATIA is modified (within the parameters and constraints defined by the parametric model).

Once the analysis model has been built and analysed it can be run through a series of inbuilt modules for checking and optimising the design. The final output of this is a structure whereby all elements have been sized and optimised as per the targets set in the optimisation module.

The basic CDF stores its data on disk using a 'serialisation' technique to write a binary file which is only readable by the basic CDF application.

4.2. LESSONS LEARNED

As an application built to support a particular project, the creators of the basic CDF have not been able to fully explore the universal potential of the framework due to time restrictions. Any multi-disciplinary functionality was not investigated.

Chief amongst the improvements that the creators of the basic CDF were forced to omit was the ability for outcomes of the analysis and design (such as member sizes) to be properly transferred back to the CATIA/Digital Project model, which meant that the documentation process is 'divorced' from the design analysis model.

Use of the basic CDF also highlighted a number of shortcomings when interacting with 3rd party applications through COM. This includes such processes as recursively querying the application for related data, when the relationships between that data can be reconstructed within the controlling application. It was also determined that the basic CDF was able to extract information from a CATIA ‘Assembly’ which are an aggregation of a number of CATIA design files (Parts). This enabled multiple members of the design team to connect several CATIA Parts in a robust manner.

The authors of this paper scrutinised the underlying source code in order determine if it could be made less project and domain specific or open to linking with other parametric/3d modelling software and other analysis software. Due to the rich OOP (Object Orientated Programming) class structure used during the development of the basic CDF, the authors believe there is great potential for generalising this basic CDF.

The data storage used by the basic CDF was seen as a weak both by both the original developers and the authors of this paper, although for different reasons. It was agreed among the authors of this paper to investigate Industry Foundation Classes (IFC) compatibility which would increase the ability to export static data to other applications, although the range of data capable of being exported is limited by the capabilities of the IFC format.

5. Extending the basic Collaborative Design Framework

Based on the lessons learned from analysing the basic CDF and the aforementioned interviews the authors of this paper are currently developing an extended Collaborative Design Framework. The authors have scrutinised the current structure of the basic CDF to make it less project and domain specific and more open for interfacing design intelligence from multiple disciplines in the AEC industry.

The conception of the extended framework was informed by three insights resulting from project based work in practice as well as investigations from with the academy: The first insight suggested that one single generic tool can not facilitate a sensible integration of all possible design requirements, but that it would have to be an open, documented and extensible data structure without the requirement of expert programming skills. (Chaszar 2003) The second insight implies that a basic understanding of the production of code is about to become an integrative part of design culture and that designers will increasingly become more active in the development of small, tailor made scripts of code to fit project specific requirements of their design. (Silver 2006) The third insight addresses the situatedness of decision making based on observations and interpretations of results derived in an iterative process (Kannengiesser and Gero 2004). Rules need to be established to regulate trade-offs between the contributing parties and consequently add to the procedural memory and systemic knowledge within a project.

Four strategies have been developed by the authors in consideration of their previous investigation.

5.1. ECDF MODULAR FRAMEWORK

To satisfy the adaptation-requirement mentioned in the research interviews, a modular framework strategy has been developed by the authors which provides a balance between the required per-project flexibility and the desire to reduce unnecessary 'redevelopment' for each project. It is responsible for maintaining the data schema between different roles and hereby is acting as a bridge to connect between them (Figure 1). The framework supports a series of module roles which describe in the following:

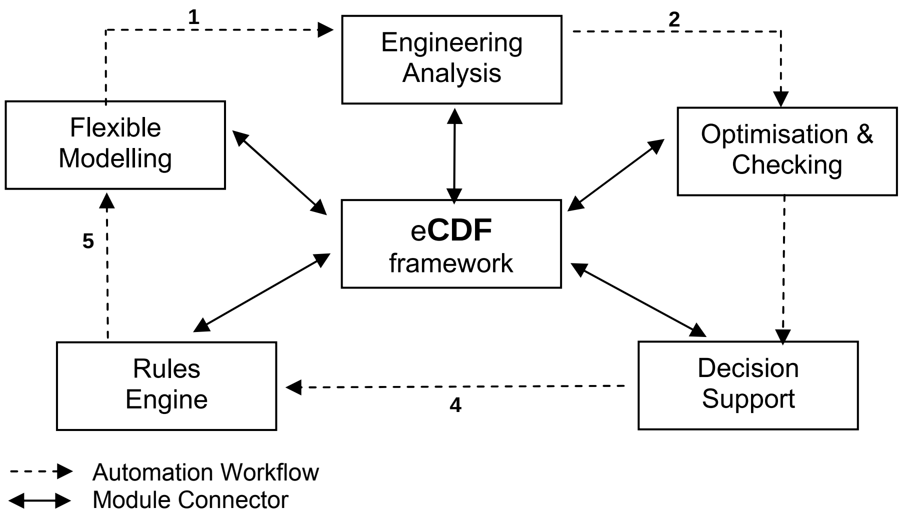

Figure 1. eCDF modular framework

5.1.1. Flexible Modelling Role

A design model is built either by flexible or generative geometry modelling. A static configuration of this geometry can then be extracted by application specific modules (acting in the flexible modelling role). Care should be taken to use a sensible naming convention as this will aid in the future identification of each element in the model.

5.1.2. Engineering Analysis Role

Within the engineering analysis module, the geometry and extra analysis attributes are used by different application specific modules (acting in the engineering analysis role). These may include structural, acoustic, mechanical, daylight-evaluation properties (or more) depending of the project-specific requirements.

5.1.3. Optimisation & Design Code Checking Role

The geometric model may also be passed through a material optimisation module (acting in the optimisation role) to minimise the material usage. The completed and analysed model may then be processed by a design checking module (acting in the design check role) so that the design produced conforms to applicable design codes. Both of these modules may initially be project specific, but over time generic modules can be derived.

5.1.4. Rule engine Role

When the framework is running in 'automated' mode, the rule engine module will extract relevant data and use that as a bases for evaluating a series of user defined rules, the outcomes of which may be used to modify the initial design model and allowing the system to recompute the new model until the desired model according to the rule is found. Rule weight could also be assigned to help guide the important principals of the design. The weighting has to be done by the designer or the design team after several sub-optimal solutions are proposed by an automated process. In this way the 'rule engine' becomes a negotiation environment for different design disciplines to encode the ways the wish to modify a design under given conditions, and the acceptable boundary conditions for that design. Combined with a decision support module that graphically displays the results of these optimisation attempts, it allows AEC designers to evaluate the objective design performance alongside the subjective design appearance.

5.1.5. Decision support Role

Once an optimised and compliant design has been found, this (and any other less-than-optimal designs) is passed to a decision support module. Using the module, a graphical display and comparison graph are created which allows design partners to review the relative merits of the proposed designs.

5.2. INFORMATION STORAGE STRATEGY

The authors have investigated various exchange formats such as STEP, IFC, IGES and other expedient project-specific data formats using Excel, Access or text files. The development of an extensible information schema is seen as one central element to the creation of the framework which supports the necessary project adaptation.

Research undertaken to this point suggests that the information schema must be able to support a superset of the information required by any tool acting as part of the framework, so that all the relevant information is available at every step during the design process. The information schema must be capable of containing the geometry, analysis requirements and analysis results for a particular design instance and when automation/optimisation is considered, the schema must be capable of co-ordinating and comparing multiple instances of these same data sets. When automation capabilities of the framework are being used, automation management information is stored in the schema, not just the information being manipulated by the process.

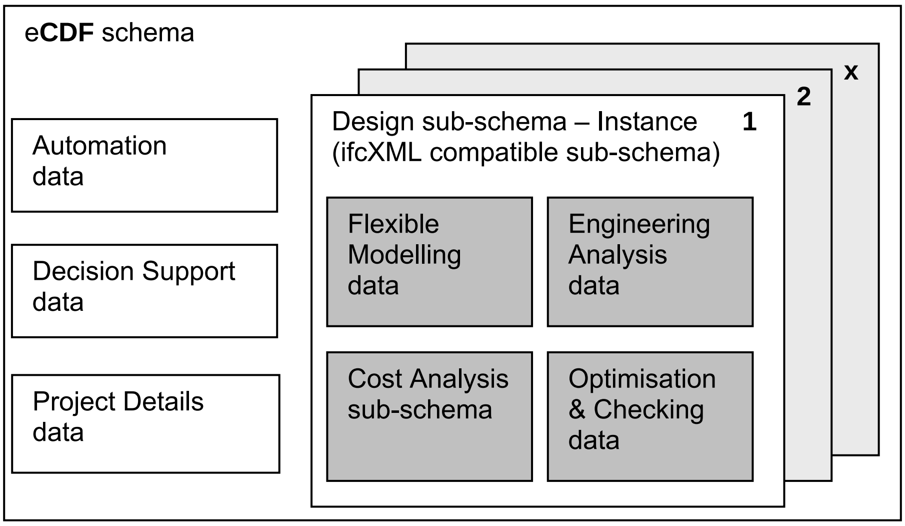

Figure 2. eCDF information structure

In order to support this, the overall eCDF schema is broken down into a number of sections which target particular roles supported by the framework. As shown in Figure 2, the main sections are Automation, Decision support, Project details and Design. The Decision Support section contains data used and captured by the Decision Support module, such as any analysis results extracted for use by the Rule Engine. The Automation section is primarily used by the eCDF controller application to store data required to construct the rules used by the Rules Engine and data pertaining to any automated processing, such as which particular plugins are used. The Project Details section contains information about the project, such as the source files that were used.

The Design sub-schema contains the information about geometry, analysis requirements and analysis results for a particular design instance. There may be multiple instances of the Design sub-schema, and this sub-schema is used by the Flexible Modelling, Engineering Analysis and Optimisation and Code Checking roles. The Design sub-schema also forms the basis of the data model used by the framework and the plugins fulfilling those roles, meaning that translation between the eCDF data model, and any application specific data models, only happens within that plugin. When introducing a new application to the framework, this limits the extent of changes required to only the plugin which interacts with the new application.

The data is stored in XML (eXtensible Markup Language) format which as a widely supported format offers well documented, strict syntax combining machine readability and (to a lesser degree) human readability. This in turn lowers the barriers for the primary developers using the framework who will be AEC designers, rather than professional software developers. It also offers related technologies such as XQuery to assist in the data management and XSLT (eXstensible Stylesheet Language Transformations) to enable transformation of the data into other text based file-formats. It is recognised that the XML format does suffer some weaknesses and limitations, such as it is a verbose format and is a hierarchical document based model.

5.3. FRAMEWORK FLEXIBILITY

To accommodate the required per-project flexibility, the framework was design with openness and pluggability in mind. This flexibility is allowed for in three ways:

1. Plug-in module: Framework roles will be paired with a suitable 'interface definitions' for the functionality which is required to support that role. Using these interface-definitions, a plug-in module can be written which either provides that functionality or allows the framework to interact with 3rd party software to provide that functionality.
2. Open data structure: The data schema allows for some extensibility and flexibility of the data and is stored in an industry standard XML Schema Definition format. Therefore, it is possible to write 'macros' or 'plug-ins' that run within 3rd party software (rather than within the framework) and can directly read and write the data used by the framework.
3. Rule engine: An embedded rules engine allows users to encode optimisation rules without needing to write a framework for processing the rules. If that proves unacceptable, the rules engine is itself a pluggable module and can be replaced.

5.4. STANDARDS COMPLIANCE

5.4.1. Information Schema Compliance

The International Alliance for Interoperability (IAI) was founded in 1994 and is developing a non-proprietary, multi-disciplinary data-exchange system based on the 'building product model' (Eastman 1999) (Malkawi

2004). The data-exchange system, known as Industry Foundation Classes (IFC), consists of object-based descriptions of building parts and their interrelation.

The authors acknowledge the value of IFCs as strong partner in augmenting the feasibility of the proposed framework and their possible compatibility-interfaces were examined. On the one hand the requirements for the application logic do differ from those of the IFCs while at the same time the extensive IFC structure does not provide enough flexibility to encompass all the data required by the framework. The IFC format does provide a good starting point for the requirement of an open, extensible data format as it includes the concept of "property sets" which allow for user defined data to be attached to any particular object within the schema.

The current data structure strategy is the use of a variant of the IFC schema in its XML format as the Design sub-schema within the larger CDF schema. (ifcXML - http://www.iai-international.org/Model/IfcXML2.htm). Although this format is less compact than the EXPRESS based IFC format, the XML based format allows developers to extend the schema as required, and still retain IFC compatibility by using XSLT transformations to convert the extended ifcXML back to pure ifcXML.

The eCDF data schema differs from the IFC schema in two significant ways. Firstly, the IFC schema is a published industry standard, and, as such, must be a stable, well-considered data format. The IFC format is still actively being drawn up, and it has been proven that even where IFC explicitly supports analytical data requirements for a particular design discipline, some data requirements may still be missing (Wan, Chen and Tiong 2004). As identified during the research interviews, ‘flexibility’ is a prime attribute of the framework, allowing end users to extend from the IFC data format when required. Using Structural Engineering optimisation as an example, investigation of the data formats used in the ‘basic CDF’ show that these extensions include new entities such as ‘Profile Group’ (a grouping of structural profiles, e.g. a group of IfcProfileDef’s) or new attributes such as ‘Target Utilisation’ (a target value applied to each element to guide the optimization process).

Secondly, the eCDF schema must be capable of supporting multiple instances of similar designs (i.e. multiple instances of the ‘Design’ sub-schema), whereas the IFC format has been designed to contain a single instance of a design. By storing the results of multiple design options, the eCDF Decision Support module is capable of showing those options in a multi-disciplinary format, increasing the ability of designers to communicate design intelligence across discipline boundaries, and collaboratively explore the ramifications of design choices.

5.4.2. Compliance with Industry Practice

Since the bulk of design software used by the AEC industry is software running on the Windows® platform, this 'Collaborative Design Framework' is implemented using Visual Basic.Net (or other ".Net" compliant languages). This allows the framework to interact with other design software using .Net interoperability methods or COM. This provides AEC designers who may be familiar with scripting, and are generally not professional software developers, an easy transition into being able to modify or develop modules for the framework.

6. Conclusions

The research presented in this paper illustrates the development of a collaborative design framework based on project-based investigations in practice, requirements from expert in the AEC industry and supported by academic research in the field. Our investigation has shown that such a framework needs to be designed to separate the requirements for project-specific adaptation from the core unchanging concepts of workflow interaction and the conceptual tasks the software tools of individual domains within the AEC industry perform. This is achieved by structuring the framework to accept modifiable plug-ins to provide a link to the tool which will perform any particular task.

Setting up the framework around a central controlling application allows the system to provide automation capabilities, which combined with a situated rules engine provide practitioners with the option to more easily perform multi-criteria, multi-discipline optimisation. Central to this is the use of a parametric, associative modelling tool, which is driven by the rules engine during an optimisation process. A user interface facilitating decision support through knowledge visualisation is built into the framework to allow for conditioning and finetuning of the rules engine. Compliance to current IFC standards can be achieved by using a variant of the IFC schema in its XML format.

Acknowledgements

The authors would hereby like to acknowledge the contribution of Prof. Mark Burry, Andrew Maher and Anitra Nelson from RMIT University, as well as Richard Hough from Arup Australia in the conception of this paper. We would further like to thank all members of staff from Arup Australia who have volunteered to participate in the interviews which are referred to in this paper and which make a substantial contribution to our research.

References

Acha,VI, Gann, DA, and Salter, AM: 2005, Episodic innovation: R&D strategies for project-based environments, *Industry and Innovation* **12**(2): 255-281.

Aish, RO: 2005 From Intuition to Precision, Digital Design: The Quest for New Paradigms, *23rd eCAADe Conference Proceedings*, Lisbon, Portugal, pp. 10-14.

Baldock RO: 2004 *Structural System Optimisation Techniques for the Building Industry*, Cambridge University Engineering Department Report.

Bedrick, JI and Rinella, TO: 2006, Technology, process, improvement, and culture change,

AIA Report on integrated practice. Retrieved January 11, 2007 from http://www.aia.org/SiteObjects/files/8_Bedrick%20Rinella.pdf.

Chaszar, AN: 2003, Bridging the Gap with Collaborative Design Programs, *Architectural Design* **73**(5): 112-18.

Eastman, CH: 1999, *Building Product Models: Computer Environments Supporting Design and Construction,* CRC Press, Boca Raton, Florida.

Eastman, CH: 2006 University and industry research in support of BIM, *AIA Report on integrated practice*. Retrieved November 12, 2006 from http://www.aia.org/SiteObjects/files/2_Eastman.pdf.

Kannengiesser, UD and Gero, JO: 2004 Using Agents in the Exchange of Product Data, *in* M Bramer and V Devedzic (eds), *Artificial Intelligence Applications and Innovations*, Kluwer, Dordrecht, pp. 129-140.

Malkawi, AM: 2004, Performance Simulation: Research and Tools, *in* B Kolarevic and AM Malkawi (eds), *Performative Architecture: Beyond Instrumentality*, Spon Press, New York, pp. 85-94.

Mueller, VO: 2006, Integrating Digital and Non-digital Design Work, *Blurring the Lines: Computer-Aided Design and Manufacturing in Contemporary Architecture*, Academy Press, pp. 38-46.

Onuma, KI: 2006, The twenty-first century practitioner: Transformed by process not software, *AIA Report on integrated practice*. Retrieved December 4, 2006 from http://www.aia.org/SiteObjects/files/6_Onuma.pdf.

Pratt, MI and Kim, JU: 2006, Experience in the exchange of procedural shape models using ISO 10303 (STEP) *Proceedings of the 2006 ACM symposium on Solid and physical modeling,* Cardiff, pp. 229-238.

Shelden, DE: 2006, Tectonics, Economics and the Reconfiguration of Practice: The Case for Process Change by Digital Means, *in* M Silver (ed), *Programming Cultures*, *Architecture, Art and Science in the Age of Software Development*, John Wiley & Sons, London, pp. 82-88.

Silver, MI: 2006, Towards a Programming Culture in Design Arts, *in* M Silver (ed), *Programming Cultures*, *Architecture, Art and Science in the Age of Software Development*, John Wiley & Sons, London, pp. 5-11.

Sprecher, AA, Chandler AH and Neumann, ER: 2006, Critical Practice: Protocol for a Fused Technology, *in* C Hight and C Perry (eds), *Collective Intelligence in Design*, John Wiley & Sons, London pp. 31-35.

Taylor, JO and Levitt, RA: 2004, Understanding and Managing Systemic Innovation in Project-based Industries, *in* DP Slevin, DI Cleland and JK Pinto (eds), *Innovations: Project management research*, Project Management Institute, Newtown Square, Pennsylvania, pp. 83-99.

Wan, C, Chen, P and Tiong, R: 2004, Assessment of IFC's for Structural analysis domain, *Electronic Journal of Information Technology in Construction*. Retrieved March 2, 2007 from http://www.itcon.org/cgi-bin/works/Show?2004_5.