Parametric and associative design as a strategy for conceptual design and delivery to BIM

Jeroen L. COENDERS*

*Arup, Delft University of Technology
Naritaweg 118, 1043 CA Amsterdam, Stevinweg 1, 2628 CN Delft
jeroen.coenders@arup.com, j.l.coenders@tudelft.nl

Abstract

In this paper the author will present and discuss several strategies and extensions to parametric and associative design to increase the applicability of this paradigm in structural design and engineering. The goal is to apply parametric and associative technology as one of the means to model design knowledge and logic as well as to generate and to communicate design information to the design stages where currently Building Information Modelling software is more appropriate to document and further carry, check and detail the information towards the execution stages of the building. Furthermore, the author would like to provide a brief insight in the underlying theory of the Structural Design Tools approach, its relation to structural conceptual design and computation.

Keywords: Building Information Modelling, Parametric and Associative Modelling, Structural Design.

1. Introduction

Building Information Modelling (BIM) is a term which has been around for quite some years and still remains a topic of heated debate. Not only debate on who invented the term and what it exactly means, but also debate on whether or not to apply it, its appropriateness (Holzer [10]) who should apply it, when it should be applied, who owns the model, etc. Autodesk [2] describes BIM as “3D, object-oriented, AEC-specific CAD” while Eastman [8] ties it to Building Product Models (data models for buildings).

The author immediately would like to propose the distinction between “BIM as a vision” and “BIM as a software technology”. Everybody will agree that “BIM as a vision” is a vision to be strived towards in the future. It can be observed in practice, especially in the more complicated projects, that computation, automation and digitalisation is becoming completely trivial for designers to assess and manage (part of) the building process. This vision includes that the different parties in the design, engineering, construction process and even later stages of the building life-cycle (maintenance and operation) apply digital, virtual or computational models to reduce design, construction and operation failures (clashes), improve cost estimates, connect scheduling, increase efficiency, deliver data to logistics, manufacturing, facility management, etc.

However, the author would like to argue that the current state of software, here referred to as “BIM as a software technology” or “BIM software”, is far from this vision and that it might even be plausible that current software applications are too simplistic or perhaps even based on too simplistic principles and paradigms to achieve this vision in full vigour.

One of the important reasons for this gap between vision and practice in current software will be further discussed in this paper. This reason is the lack of BIM supporting the design process, because this process has a different nature than the engineering, construction and operation process. In the latter process BIM might be more appropriate. In fact, this lack of use and availability of suitable computational approaches in the design process provides an additional problem for the application of computation in the later stages of design and construction as crucial initial information often has to be retrospectively modelled with all problems attached of information loss (of the current state but especially of the information generated in the process), oversimplification, unavailability of (historical) information, etc. Other reasons which can be mentioned are the facts is that many of the software applications only support limited complexity in objects and geometry, still lack an interoperability format which works in practice and still only have limited capabilities in professional drawing extraction, quantity derivation, clash detection of non-geometrical information, etc.

These problems can be overcome in practice however, for example by making use of the by Arup developed “BIM++ strategy” which overcomes technological hurdles by management and custom and project-specific workflows. In this paper the focus will lie on the fact that next to management and workflows, it would help in the future to also further develop the technology and tools.

Although the BIM software tools can be used in the design process and have successfully be applied in past projects as a data storage during the process, it does not make them “design tools”. The author would like to emphasize that BIM can be used to record design with a manual modelling approach (which is perhaps less manual than 2D paper drawings, but cannot be called “automatic”) or that some software can be applied as a useful analysis tool.

Design tools aid in the design and need to take into account the characteristics and core values of design, such as the fact that design’s nature changes fast, design by nature is a complex process, information is needed early (but not necessarily with a high precision), alternatives need to be considered in concurrency, design changes from indicative and low information density towards precision and control of high information density, etc. Another important aspect of design is the fact that design is a process to unravel complex problems with high information density and many design options and to gain confidence in the proposed solution alternatives regarding safety, efficiency, buildability, budget, risk, etc. The design process is a proven technology over the history of design and engineering as can be observed from the many successful buildings from the past and current. However, currently a trend can be observed of the uptake of computational tools by the designers of which actions and information need to be integrated in the design process. When a designer or engineer does not have any confidence in the tools or the methods used in the tools, he or she will simply choose not to adopt them. Confidence in tools is gained and enhanced by insight in the *behaviour* and internal workings, *control* over the methods used in the tool and of the information generated by the tools and *adaptability* of the tools to designers preferences. Considering adaptability of tools, design is a very personal process, unique to any building, person, team or firm. It can be observed that designers have a style or signature of sketching, drawing, calculation and reasoning. Furthermore, from accounts by various designers design is a process which also has an irrational side of hunches and gut-feelings (Coenders [4]). However, most computation systems enforce standardisation of calculation and little room for adaptation to gain confidence, insight and control. Most of the current systems act as black boxes which hide their internal workings from the user, choose simplicity over customisability and adaptability and have no mechanisms to control large amounts of data (except for storage, basic geometrical operations and data-field entry).

The author would like to propose that parametric and associative software, such as Bentley's GenerativeComponents (GC) by Robert Aish [1] or McNeel's Grasshopper [11] when properly adapted could provide a means to generate design data, store design knowledge and logic and carry the design data through the design process towards the current state of BIM software. If parametric and associative design software will become part of the BIM vision, or if a new term has to be invented, the author will let the readers decide. The essential difference between the parametrics in these systems and parametrics and objects of current BIM systems is the ability for the user to define, compose and modify the object's creation logic through the concepts of parameters, associations and definition defined below. Because the logic is essential open to the user, insight, adaptability and control are less of an issue in this approach.

However, parametric and associative systems are currently very much focussed on the definition and modification of geometrical logic rather than design, engineering and construction logic. In the past, the author has proposed several strategies to use and enhance these systems for structural design.

In this paper the author will present, discuss and provide overview over these strategies and novel extensions to parametric and associative design to increase the applicability of this paradigm in structural design and engineering. The goal is to apply parametric and associative technology as one of the means to generate and communicate design data to the stages where currently BIM software is more appropriate, by expression of design logic which currently exist in the design process.

2. Structural design (tools)

Structural design is the act of designing a structure from the initial conception to a detailed design which can be constructed on site. An important difference to note is the difference with analysis or calculation, which is part of the design process, but has a different nature. Analysis by definition needs a subject to analyse. This subject has to be created by the process of design. While this distinction is useful to understand why computation has difficulty in supporting design, in reality design slowly blends with construction as construction issues often are considered during the design and during construction still details of the design are being solved. This slow blending from limited information, large granularity and much design freedom towards massive amounts of data to be managed, fine granularity and (over)constrained problems is part of the problem why computation only is used as a tool for analysis in the design or as a tool to document the design, but rarely to fully carry the design data and execute the process of design. Therefore, it is crucial that sensible strategies for the application of computation in design are being found.

However, this does not mean that we should impose computational strategies on the design without considering design itself. This technology-centred approach has often failed in the past. As can be seen with BIM applications computation has a strong relationship with the data management and constrained problems but not with the loose nature of design. We should however not forget that design has successfully dealt with many successful building designs without computation and is a proven "technology" to conceive buildings, even with a high complexity. Lessons can be learned from design as until now it has dealt more with complexity than computational design. Design focuses on conception of the known rather than analysis, evaluation and optimisation of the known.

In the recent past the author has proposed the "Structural Design Tools approach" (SDT approach) as a solution direction for the application of more computational advanced technology in the structural design process. Instead of a centralised building model approach where all data is centralised or the definition (or standard) is centralised to one shared standard, the SDT approach proposes a federated tool approach. Tools can be applications, large and small, commercial, open-source or in-house developed, plug-ins, add-ins, networks, (development) frameworks, clusters, commands, features, components, objects, etc. etc. This federated tool approach also allows for a centralised approach, but this is the choice of the user for a particular application (in a project, organisation or software). Not only the data is decentralised, but also the data structure is decentralised. Furthermore, the approach proposes that all data and data structures are open for the user to review, compose, define and edit. An open approach allows for closed application based on the user's or developer's choice. Federation increases insight for the user instead of overwhelming systems. Also, software development is federated in this approach and assumes that designers should be aided in building their own tools (or choosing to use other people's tools) instead of being constrained in solutions by one or a few centralised software providers. Note that in all these cases, choice is the key to provide flexibility to the designer. As stated, design requires flexibility from tools as every design process is unique to its context. Also note that on availability of appropriate standards for the particular design problem the designer can also choose to adopt the route of applying standardised approaches or a combination.

Communication between the tools is a choice for the user by making use of a customisable data exchange formats based on a user-customisable interoperability framework. If a (open) standard is available, the user can choose to use this format, if not, he can develop (or extend) his own. Development and extension preferably happens through (knowledge) modelling, but can also be performed through API's, scripting and (visual) programming.

A similar approach has been proposed independently by Holzer, Tenogo and Downing [10] in the Delivering Digital Architecture in Australia (DDAA) research.

3. Parametric and associative design (PAD)

Parametric and associative design (PAD) systems come in many varieties and many definitions are available. In this paper the author will focus on systems which combine the following concepts: object-orientation, parameters (parametric), association (associative), definition and single-directional graph generation. These concepts will be defined below.

3.1. Object-orientation (OO)

Object-orientation as a concept in parametric and associative design aims on the fact that objects to be modelled with are exposed to the user, which have properties (parameters and variables), relationships (associations) and behaviour (definition). Representations of these objects can be used to visualise, derive other models, etc. Most contemporary computer applications are programmed by making use of object-orientation (Wikipedia [14]), but this type of object-orientation does not have to be exposed to the user. In case of this concept the essential part is the exposure of the object logic to the user.

3.2. Parameters (parametric)

Parametric as a concept simply indicates that the user can define and modify (input) parameters on the objects. Through association, definition and the solving method these parameters are transformed into model outputs. Parameters are also referred to as properties or attributes.

3.3. Association (Associative)

Association indicates that the user can define and modify associations between the objects. This association can be a simple input-output mapping, but also occur as mathematical expressions. In most parametric and associative systems, association can be thought of as directed edges in an acyclic graph which describes the generation process (single-directional graph parsing).

3.4. Definition

Definition is a concept which indicates that an object has a computer-processable, user-definable definition from input parameters to outputs, either numerical, in objects or visual. Definition often is labelled as update methods, but sometimes is hidden from the user through automated interpretation of the inputs. Definition can be defined by the user through packing object logic in new objects (“feature building” in GC, “clustering” in Grasshopper), by scripting or programming.

3.5. Single-directional graph parsing

Single-directional graph parsing is a solving or generation method to walk through the associations from model inputs to outputs as if walking through a directed, acyclic graph which represents the objects (and their update methods/definition) as nodes and the associations as edges. The method simply carries the output of prior objects to the input of subsequent objects. Since cycling is not allowed, the result is guaranteed if all intermediate steps guarantee a result. Other solving methods are available in other systems, but often only apply to extremely simplified cases of relationships and geometry.

Considering parametric and associative design as a paradigm these concepts make it possible for the user to define, compose and modify design logic on a level of components (objects), but also on the level of a complete building model. Design knowledge in the form of processable design logic potentially could be a very powerful aid in using computation in design and potentially could fill in the role of design tool or even communication framework in the Structural Design Tools approach.

Parametric and associative design currently is very much oriented on geometrical logic, but can be used to support a move to embed more design intelligence in the form of processable knowledge and logic in parameters and objects and to become knowledge-oriented. Currently, some systems support simplistic structural logic, but the move towards the support for structural design and engineering (and other disciplines) still needs to be made.

Parametric and associative design therefore shares a common problem with BIM being that at the moment there are two directions: "PAD as a vision" and "PAD as a software techology". The challenge for the future lies in pushing the boundaries of the software practice towards the vision and to embed parametric and associative design technology in the BIM vision for the design stages. It must be noted here that technologies like Building Information Modelling, parametric and associative design will not be the one and only solution, but that in the approach of structural design tools it will be one of the useful tool in a complete range of tools. Some of these tools will even be new tools, specific to projects, users or companies, inherently unknown to us now.

Furthermore, parametric and associative design is currently mainly applicable for designs with a repetitive nature as the concepts in the available system support this easily. However, in most building design this is not the case. Therefore, the author proposes that methods need to be found to easier support exception as a rule, rather than a complication. Rule-processing (briefly discussed below) is one concept which would make it easier for systems to react to the disorganisation caused by exception.

Below a number of strategies will be presented which have been developed by the author to enhance parametric and associative design for structural design and engineering.

4. Strategies for parametric and associative structural design (PASD)

The questions of which of the different strategies should be applied depends on a number of factors, such as the availability of computational and non-computational design models (the architect's model, models of other parties), the nature and amount of models which have to be used in concurrency, the suitability of the design for (the current state of) parametric and associative design (e.g. repetition), the design phase (and coming phases), the nature of the project and process, and the organisation and context which surrounds the design. Especially the design phase has a special influence on the used strategies. In the earlier phases of the design the emphasis will lie on conception and quick evaluation of structural concepts while in the later stages the emphasis will move in the direction of controlling design information and knowledge on a large scale and more detailed analysis of multiple models.

4.1. Insight: Early stage conceptual design tools: StructuralComponents

StructuralComponents (Breider and Coenders [3]) is an add-in extension of GenerativeComponents (but could easily be extended to support other parametric and associative design systems or other types of design systems used in practice) which allows the user to compose a conceptual structure for high-rise buildings by making use of large-scale building blocks. The conceptual structure can be analysed in real-time by making use of a simplified analysis method (modified SuperElements method [12]) which still is reasonably accurate and because the mechanical behaviour in each component can be described analytically, various kinds of results can be derived separately. By presenting the results on a dashboard view the user quickly gets an indication of the important aspects in the structural design and its feasibility. The real-time response, the elaborate results and ability to compose, tweak and assess the structure provides insight, adaptability and control for the earliest stages of design of high-rise buildings. Furthermore, this approach potentially can be extended to include more knowledge on building design and building types. This is currently under investigation as a joint research between Delft University of Technology and Arup.

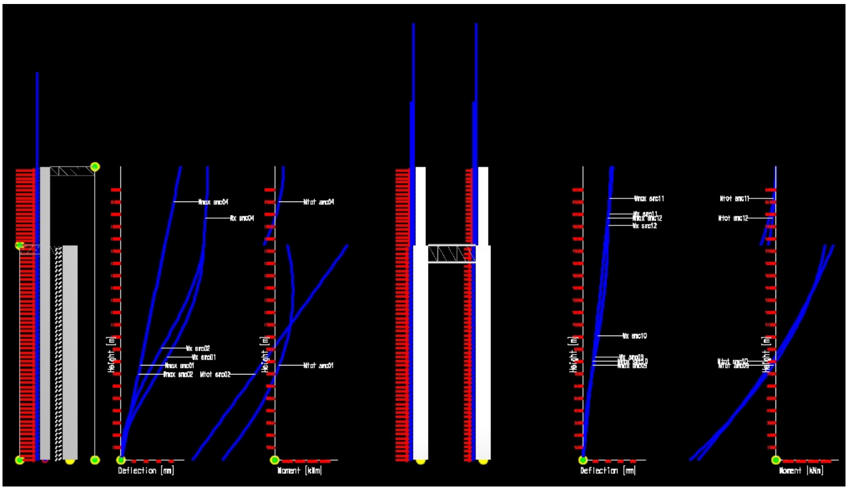

Figure 1: Typical dashboard view of StructuralComponents

4.2. Control: automated checking by rule-processing

A strategy which can be employed for control of large models, either parametric or non-parametric, is automated checking by making use of rules. Rule-processing and rule-based systems have been explored in the past. Currently, the rising problem of increased complexity and amount of information in building design, the increased use of large computational models and the potential move towards a heavily computationally augmented design process provides new relevance for this technology. In the industries, such as the financial industry, IT has also adopted rule-based systems, known as business-rules.

This strategy makes use of a specific case of rule-processing (discussed below) to verify two types of checks: (1) *content checks* and/or (2) *validity checks*. In the first case, the rule-based check verifies an assertion judging the content of the model (e.g. the stress of a beam should be less than 200 N/mm2) and reports a success or failure (IF..THEN..ELSE). In the second case the check verifies the validity of a rule (e.g. the stress-check of the beam should only occur for beams with a span larger than 1 m). Computationally both checks are the same, but for the engineering there is an essential difference (especially when these rules are combined in rule assemblies) as a collection of the first type will give the engineer evaluation information concerning the performance of the design while a collection of the second type will check if the rules applied are appropriate for the considered artefact. This is of course limited by the amount, complexity and content of the rules expressed by the designer. This strategy can be used to enhance the control of the engineer, not to try to replace him. However, if models become more complex, the engineer is able to ensure that certain rules always apply under certain conditions. This principle has been inspired on unit testing (Wikipedia [13]), which is a technology used for constraining and testing code in a continuous cycle in agile programming environments where code constantly changes but needs to stay robust and stable. Design preferably has similar qualities.

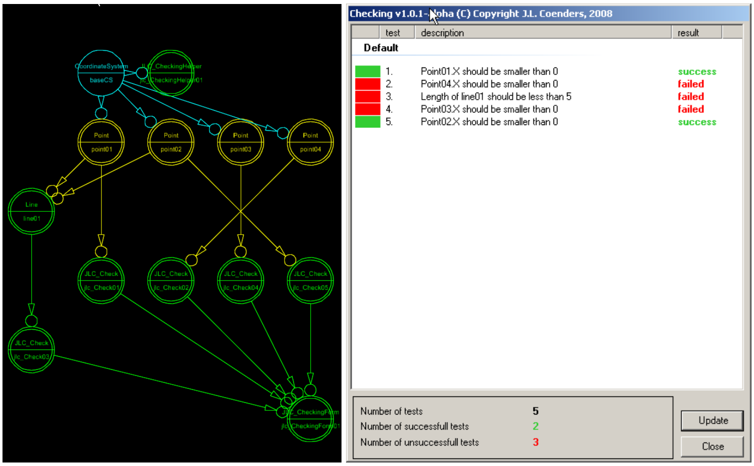

Figure 2: (left) Symbolic representation of the checking logic (right) Prototype graphical user interface of test report

4.3. Rule-processing

Rule-processing is the generalised strategy which can be applied to model an alternative way to define logic in parametric and associative systems. This alternative to single-directional graph parsing logic increases expressibility in the design rules defined by the user which increases the ease of modelling and therefore its power. Rule-processing can be used to express cross-cutting concerns in parametric and associative logic (Coenders [6]), such as shown in the example of automated checking. Cross-cutting concerns are aspects or parts of logic which cross cut the normal stream of logic and appears at multiple places. These principles come from aspect-oriented programming (AOP) (Coyler, Clement, Harley and Webster [7]). In the example of automated checking, the checks scatter through the entire logic model as shown in Figure 2.

Next to the automated checking, rule-processing can for instance be used to model context independently from the design artefact itself. For example, the wind loads can be modelled as independent entities which are imposed on the model by projection logic. Environmental conditions are essentially independent from the building logic. Rule-processing becomes particularly powerful when the rules are assembled in packages so that with little modelling effort a logical assembly consisting of a large number of rules can be applied in one action.

4.4. Automated analysis

The direct connection between design models and analysis has always been envisioned as an enriching capability as it will prevent mistakes in remodelling the structure. Especially in the case of parametric and associative models variations in the design can lead to quick regeneration and recalculation of the design. For this purpose the author has developed an interfacing strategy (Coenders [5]) based on a generalised framework in .NET which collects parametric and associative data, upgrades and cleans the data so that it becomes a proper structural model based on rule-processing logic and sends (through the software interface) the data to structural analysis applications, is able to reanalyse the model and post-process the output data.

However, the author would like to emphasize that, especially for the early stages of design, this type of interfacing and analysis strategy only works in the hands of experienced designers and for complex structures, where other types of analysis would be considered oversimplified. The author would like to stress that the value of structural engineering design does not lie in the analysis, but in the development of a clean structural concept. Often analysis of simplified models of 2D sections or stick models give excellent results for this purpose and give much more insight in the essential load carrying mechanisms which occur in the strucuture.

Computationally, analysis of simplified models instead of the available more advanced models seems unsophisticated but for insight in the design it is essential. The author would like to propose that it might be very beneficial for insight and control if computational systems would allow quick generation of simplified models, such as sections or selections of objects, which maintain a parametric and associative relationship with the other information in the models. Rule-processing would again be a concept which can be used to obtain this information, check it against rules and process the results.

StructuralComponents is an example of a strategy where simplified analysis is employed for conceptual design rather than for instance full scale Finite Element models. Finite Element models in combination with parametric and associative interface strategies do make sense when the structure is so complex that it cannot be analysed with simpler method (a case where engineers might have to rethink their design too) or in the later stages of design when more detailed, but also more constrained structural models are being considered.

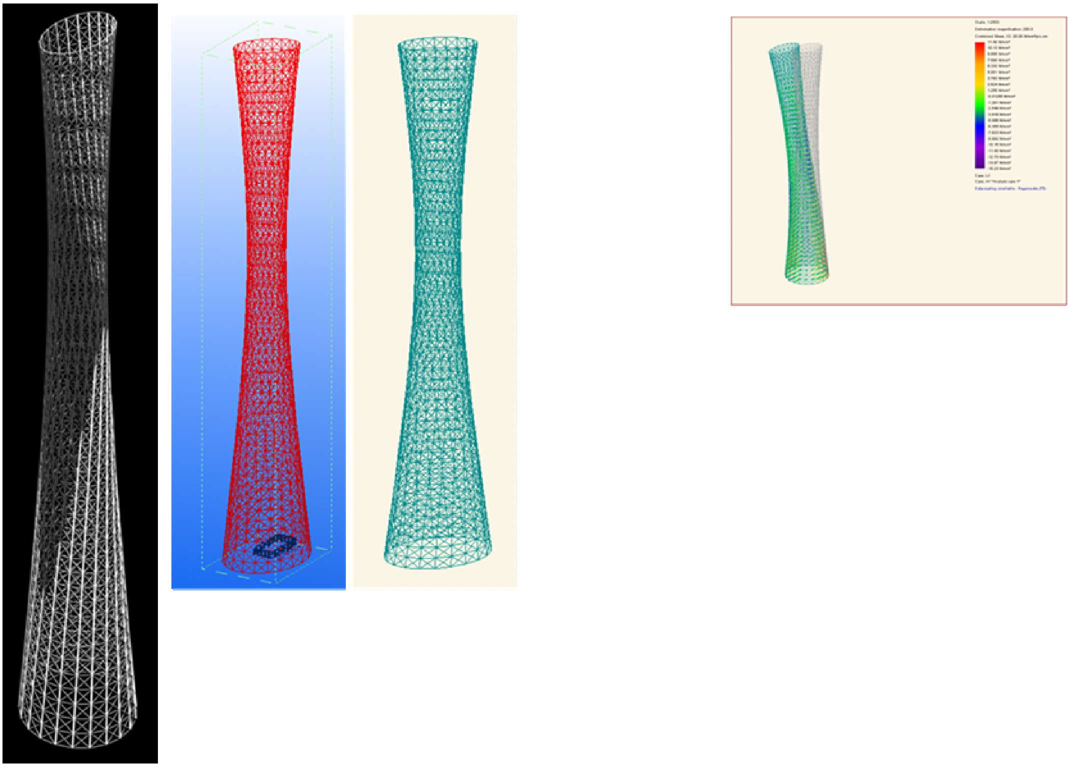

Figure 3: Several images from an interface strategy case where (from left to right) the parametric associative model in GenerativeComponents is interface to Tekla, GSA and analysed automatically.

4.5. Delivery to BIM

At the end of the design process the information generated in the design needs to be communicated to continue with further detailed analysis, planning, logistics, etc. towards the construction and operation stages of the building. Many of the current BIM tools are developing in this field towards becoming professional tools for these purposes and therefore much experiment takes place to apply these tools in these stages. However, these models do not provide strategies to generate design data in a form suitable for design, because in the stage where these tools natively focus on the model logic does not deal with design changes on that level. Therefore, the model logic is more focussed on data management rather than change. For this purpose the author has developed an extension of the interface strategy to deliver design data generated by parametric and associative systems (in this case GenerativeComponents) to Building Information Modelling software (Tekla and Revit). The concept which is used is mapping which means that the parametric and associative objects are mapped to an equivalent in the BIM software. The mapping can occur on objects of similar nature (a parametric wall becomes a BIM wall object), but also on simplified representations (a parametric line becomes a BIM beam object). The author would like to propose for the future that parametric and associative objects potentially also could be used in BIM systems as they essentially are very similar. In the case of BIM objects the parametric objects could be simplified by removing unnecessary logic or blocking inputs.

5. Discussion

The developments discussed in this paper are part of a research project which is performed at the Structural Design Lab at Delft University of Technology as well as research and development projects at international consulting firm Arup. The research at university focuses bringing forward the Structural Design Tools approach while the projects at Arup focus more on providing practical tools which can be used in the day-to-day practice of design and engineering.

The developments presented in this paper are part of a larger effort to provide one or more strategies and their related tools for multiple disciplines in concurrency for each of the phases in the design until construction, while maintaining the core values and characteristics of design.

6. Conclusions

The author has demonstrated several strategies for the application of parametric and associative design as a paradigm for delivering part of the Structural Design Tools approach to the practice of structural design and engineering. The author has proposed that parametric and associative technology can aid the designer as a tool to obtain insight, adaptability and control in the design process and to generate design data to be delivered to Building Information Modelling systems. Furthermore, the author has indicated where Building Information Modelling and parametric and associative design technology are applicable for design.

References

[1] Aish R., *Introduction to GenerativeComponents, a parametric and associative design system for architecture, building engineering and digital fabrication*, white paper, http://www.bentley.com, 2005.

[2] Autodesk, *White paper on Building Information Modelling*, white paper, http://www.autodesk.com, 2007.

[3] Breider J.W. and Coenders J.L., *StructuralComponents*, poster presentation at AAG2008, Vienna, Austria, 2008

[4] Coenders JL, Barriers in Computational Structural Design, *Journal of The International Association for Shell and Spatial Structures,* 2007; **48;** 51-62.

[5] Coenders J.L., Interfacing between parametric associative and structural software, in *Innovations in Structural Engineering and Construction*, Xie Y.M. and Patnaikuni, I. (ed), Taylor & Francis Group, 2008, 63-68.

[6] Coenders J.L., Rule-processing as a cross-cutting concern in parametric associative design systems, in IASS-SLTE 2008 simposium Acapulco Mexico, New materials and technologies, new design and innovations – a sustainable approach to architectural and structural design, Oliva Salinas J.G., Acapulco, Mexico, 2008, 87-88.

[7] Coyler A., Clement A., Harley G. and Webster M., *Eclipse AspectJ: Aspect-oriented Programming with AspectJ and the Eclipse AspectJ Development Tools*, Addison Wesley, 2005.

[8] Eastman C.M., *Building Product Models: Computer Environments Supporting Design and Construction*, CRC Press, 1999.

[9] Holzer D, Are you talking to me? Why BIM alone is not the answer, *Proceedings of the Fourth International Conference of the Association of Architecture Schools of Australasia*, Orr K. and Kaji – O'Grady, S. (ed), UTsePress, 2007, 108-114.

[10] Holzer D. & Tenogo Y. and Downing S., Developing a framework for linking design intelligence from multiple professions in the AEC industry, in *CAADFutures'07*, Dong A., Vande Moere A. and Gero J.S. (ed), Springer, 2007, 303-316.

[11] McNeel, *Grasshopper*, website, http://grasshopper.rhino3d.com, 2008.

[12] Steenbergen R. *Super Elements in High-Rise Buildings under Stochastic Wind Load*, Delft University of Technology, 2007.

[13] Wikipedia contributors, *Unit testing*, website, Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Unit_testing&oldid=214464104, 2008.

[14] Wikipedia contributors, *Object-oriented programming*, website, Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Object-oriented_programming&oldid=288090250, 2009.