Tolerating Inconsistencies
The Distributed Perspectives Model

Oliver Hoffmann, Markus Stumptner and Talik Chalabi
Technical University of Vienna

Key words: CAD, Microstation, artificial intelligence, creativity, urban design, typology, Java, JATLite, JATLiteBean, agent, JESS

Abstract: A new design model is presented. Information on the design is distributed over multiple self-contained design perspectives and translation functions between design perspectives. Inconsistencies between specifications in different design perspectives introduced by human designers are temporarily tolerated in order to support creative design processes. The implementation of a design support system currently under evaluation is outlined.

1. MOTIVATION

Tolerating inconsistencies during the design process can be seen as a necessary prerequisite for creativity (Hoffmann, Stumptner, et al, 2000). Architectural design in particular deals with both “measurable” criteria like economy, building codes, environmental issues, etc. and “unmeasurable” factors like cultural values, perception, psychology, *Zeitgeist*, esthetic canons, etc. Therefore, computer support for design should provide for heterogeneous representation of design problems and admit modifications of any assumptions at any stage of the design process. Instead of adopting a view of the design process as a linear sequence of design steps with limited revisions of the design goals, we advocate a view characterized by intertwining of design specifications and design requirement changes, which are possibly executed by different individuals.

2. AN URBAN DESIGN EXAMPLE

Urban design is a creative task that demands the juggling of issues from multiple domains such as architecture, zoning laws and traffic planning. The urban designer constantly alters his point of view and frequently re-evaluates his precepts to arrive at better or innovative solutions. As an example, the task to create an urban design for a rectangular residential area with a certain required built up density is investigated (density is defined as the ratio of built up space to plot area). The designer first chooses an existing building type to work with.

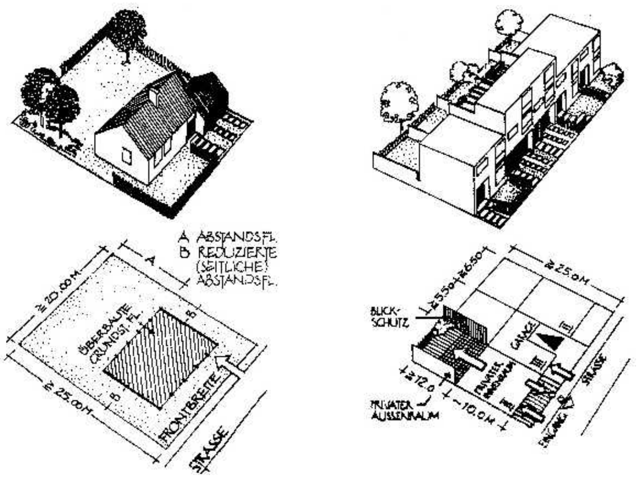

Figure 1. Residential building typology

In our example the designer uses two standard residential building types: single family house and row house (fig. 1) (Prinz, 1995). Both types have requirements related to building size and setback from neighboring plots and some key data associated with them, such as built-up density. In this example, the designer initially chooses to work with single-family houses (left side of fig. 1). This type allows a relatively high degree of flexibility in the two-dimensional (2D) layout. However, when the minimum setback distances for single family houses is taken into account, the required built-up density cannot be achieved. In the next step, the designer changes the selected building type to row house (right side of fig. 1) which helps to fulfil the density requirements but does not allow enough flexibility in the overall design. Therefore, the designer chooses to ignore the limitations of existing typologies for the time being and shifts individual floors by 90-degree, which results in the creation of a new building type based on the underlying row house. Fig. 2 shows an overall view of the resulting design. Fig. 3 shows the detailed facade of the first "row" exhibiting the variety of solutions attained.

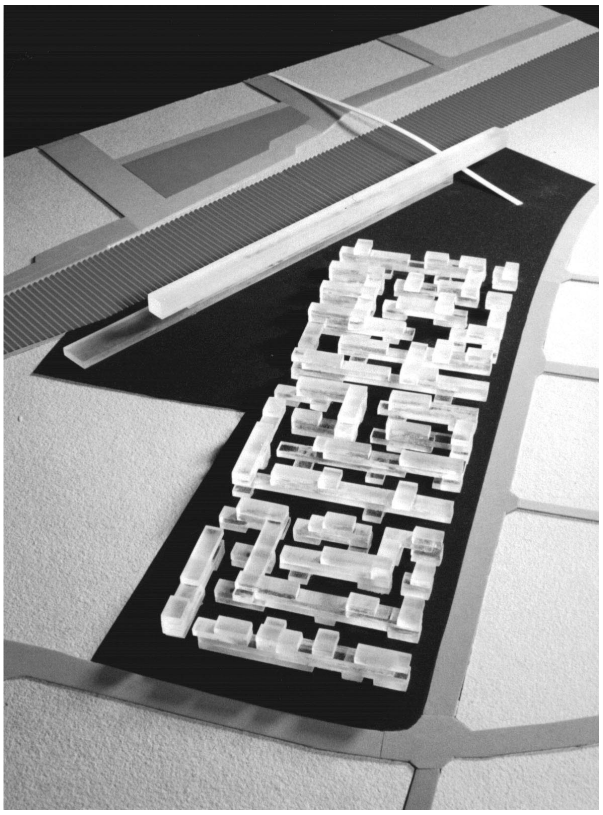
Figure 2. Result of creative transformation of row house typology

In effect, the designer has changed the rules for residential building types while using them in a specific design. Therefore, the designer has introduced a high degree of inconsistency between existing building types and the way the building types were used: From the computer support point of view, the 90° turn of 1st floors against ground floors is inconsistent with the row house type, since the row house is based on the linear addition of identical units. From the designer's point of view, the adaptation of existing typologies to site-specific design does not constitute inconsistency but an integral part of creative designing. The inconsistencies are resolved at the end of the design process, with the definition of a new building type. Computer support for this kind of creative design process has to tolerate inconsistencies at certain design stages.

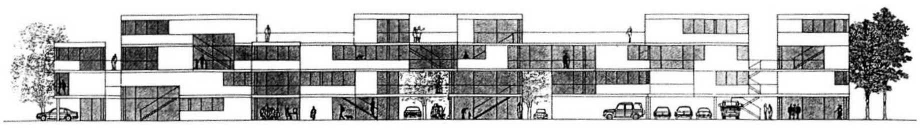
Figure 3. A new building type

3. DESIGN PERSPECTIVES

To allow designers to "change their perspective" on a given design, we introduce the concept of design perspectives. A design perspective is not a subset of a central data source, but a self-contained view of the design with a set of design requirements and specifications. The term "perspective" is used in the sense of "symbolic form" (Panofsky, 1924).

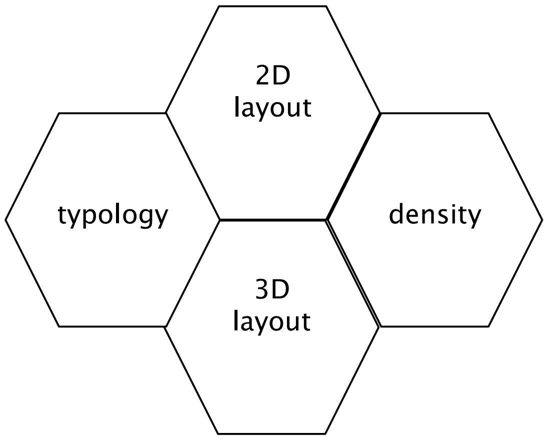

Figure 4. Design example with four perspectives

For the sake of simplicity, we restrict the design perspectives in our example to a small number of "measurables", all of them explicitly modeled in a design support system (fig. 4): typology (referring to the type of house and its relation to the ground it is built on), 2D layout (referring to the street grid), 3D layout (buildings as arranged on the site) and density (the urban density achieved). The term "layout" is used more in a computer science sense here, that is, some arrangement of shapes irrespective of their architectural interpretation.

At a given point in the design process, the current state of the design object, i.e., the specifications produced by the designer in a particular perspective have to be consistent with requirements in the same perspective. The current state in one perspective generally cannot be checked against requirements in other perspectives, since the symbols used in one perspective might have a different interpretation in another perspective. For example, the symbol "building" could at a given point be interpreted as "2-dimensional closed multi-line shape" in the 2D layout perspective and as "set of solid objects in direct contact with label 'building'" in the 3D layout perspective.

4. TRANSLATIONS

Perspectives are connected using translation functions. A translation function A => B is a mapping from specifications in perspective A, i.e., the current state of the design object in that perspective as specified by the designer, to requirements in perspective B. The fact that specifications are translated into requirements rather than specifications is the primary distinguishing factor to approaches like the DESIRE system (Brazier, Langen, et al, 1994). The design is globally consistent at a given point in the design process if specifications in design perspectives are consistent with requirements from within the perspectives (internal consistency) and requirements resulting from translations (external consistency). A translation is similar to a filter as defined in filter mediated design (Haymaker, Ackermann, et al, 2000), in that a translation, like a filter, transforms information on the design from one representation to another. But in contrast to filter mediated design, there is no central representation in the distributed perspectives model. There can also be bidirectional translations between perspectives. Moreover, translations, as well as perspectives, can be added to and removed from the system at any point in the design process. In particular, different versions of a translation from one perspective to another might be available, and these different versions can be included in or excluded from the system at any point in the design process. Fig. 5 shows a possible set of translations between perspectives for the presented urban design example. For instance, a selection of a specific grid in the 2D layout perspective can be translated into requirements for the 3D layout: If the 3D layout is to be externally consistent with the 2D grid selected, buildings in the 3D perspective have to be arranged according to the 2D grid. Similarly, specifications from the density perspective can be translated into requirements for the 3D layout: If 3D layout is to be externally consistent with the density perspective, 3D layout has to achieve a certain density. In this example, there are 2 translations between the typology perspective and the 3D layout perspective. Therefore, it is possible both to translate specifications for building types into requirements for buildings in the 3D

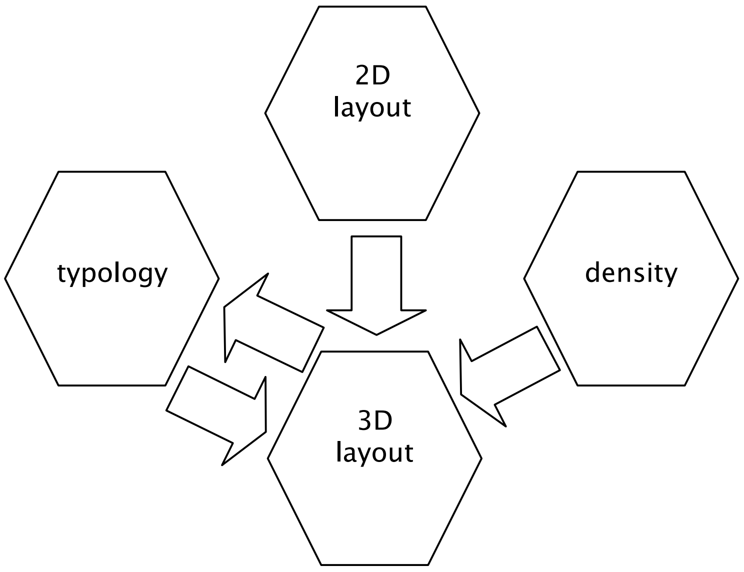

Figure 5. Translation functions between design perspectives

layout and to translate given buildings in the 3D layout into prototypes for new or updated building types.

5. IMPLEMENTATION

The distributed perspectives prototype has been implemented on the basis of a multi-agent system. Agent-based programming was applied more in the sense of software engineering (Petrie, 2000) than as an implementation of the belief-desire-intention theory (Rao and Georgeff, 1995). The human designer performs changes in some of the available perspectives and agents are used for translating between perspectives. Therefore, there is no control structure for designing (the human designer is free to choose among possible design steps).

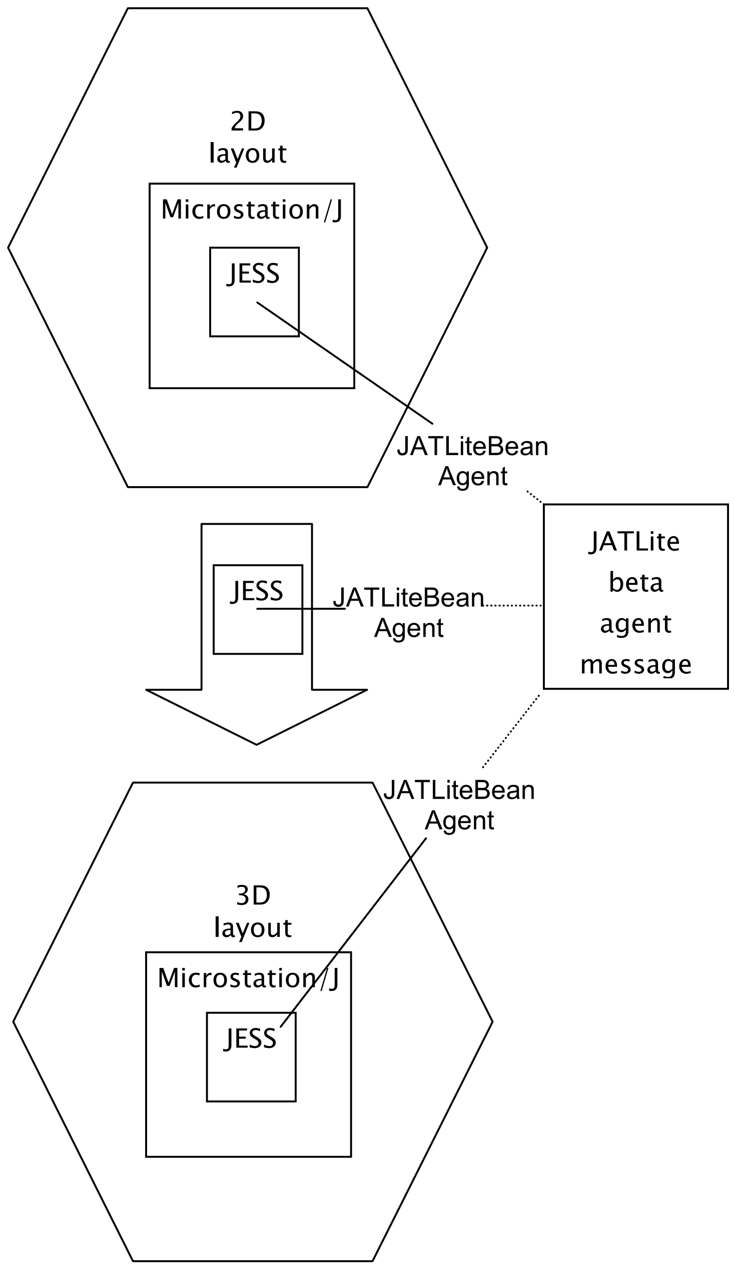

Figure 6. translation function implementation

Fig. 6 shows the software architecture chosen on the example of a translation function from the 2D layout perspective to the 3D layout perspective. The 2D layout perspective is implemented as an extension of the commercial CAD software Microstation/J (Bentley, 2000). Design support is provided through one or many design agents implemented in the Java expert system shell or JESS (Sandia, 2000). JESS is an implementation of the Rete algorithm (Forgy, 1982) in Java. Communication between perspectives and translation functions is implemented through one or many JATLiteBean agents. JATLiteBean (Moreale and Kinlay, 2000) is an extension of the Java agent template JATLite (Jeon, Petrie, et al, 2000).

Translation of the 2D layout specifications into 3D layout requirements is provided in the following fashion: If a new or updated grid is selected in the 2D layout perspective, the JATLiteBean agent sends a message to all relevant translation functions containing the updated specifications. The JESS system implementing the translation function from the 2D layout to 3D layout receives this message, translates the 2D grid specifications to 3D layout requirements and adds rules for achieving these requirements. Then the translation function JESS system sends a message to any relevant perspectives containing new rules for the 3D layout through a JATLiteBean agent. The 3D layout perspective receives this message through a JATLiteBean agent and a design support agent incorporates the new rules into its rule base. Depending on the kind of translation performed, the new rules can be used for design automation or design critique.

6. DESIGN PROCESS

The design process can go through different stages with different characteristics. These stages do not occur in a pre-determined sequence, instead transition from one stage to another is introduced by human designers interacting with the design support system. Some examples of design stages are described below.

6.1 Problem-Solving Stage

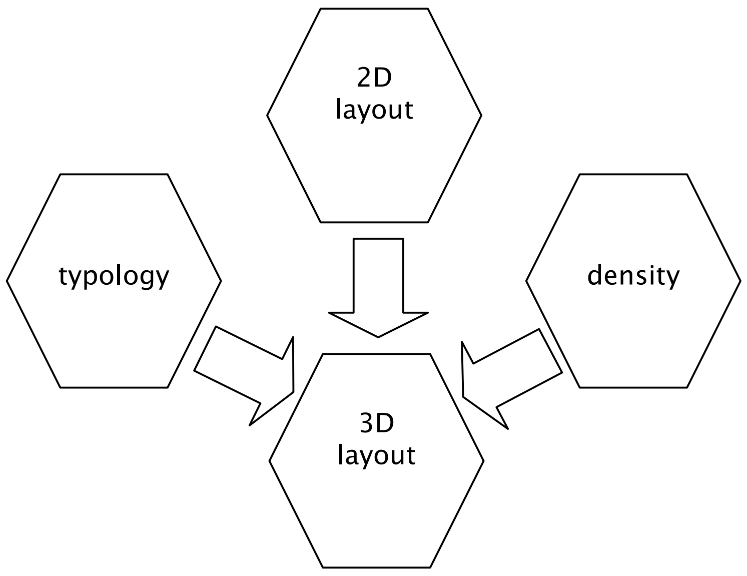

Figure 7. Problem-Solving Stage

Design as problem solving can be seen as the classical application of artificial intelligence in design (Brown and Chandrasekaran, 1989) In the distributed perspectives model, design as a problem-solving activity occurs under the following conditions:

– Exactly one design perspective is dynamic, that is, changes in design specifications are only occurring in this one design perspective.
– All specifications in other design perspectives have been successfully translated into requirements for the dynamic perspective.

The human designer and/or design support agents in the dynamic perspective try to find specifications that are consistent with the requirements. In the urban design example, the design problem is placing buildings in the 3D layout in such a way that density, 2D layout (grid) and building type requirements are met.

6.2 Creative Transformation

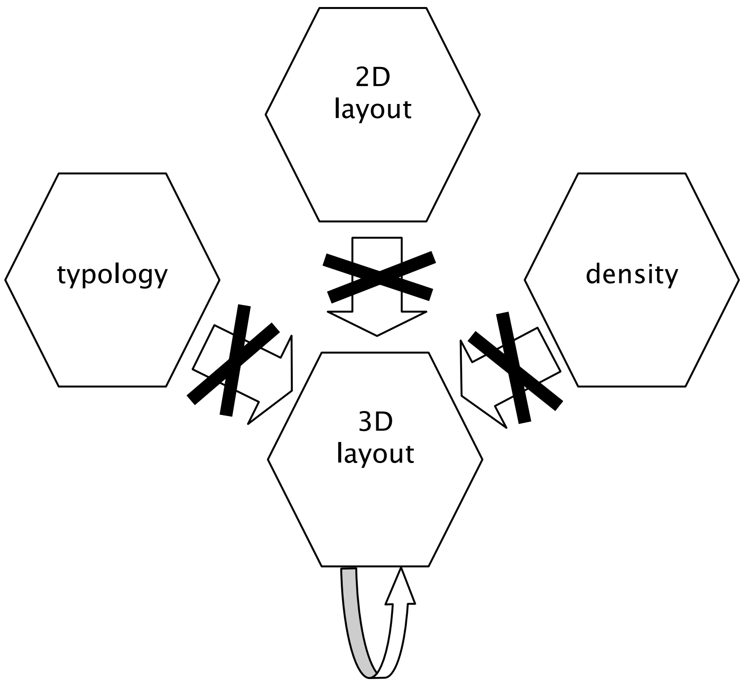

Figure 8. Creative Transformation

A creative design transformation can occur under the following conditions:

– All design perspectives contain design specifications.
– Translation functions are temporarily turned off.
– Specifications in one of the perspectives are modified in order to create an innovative design.

The creative transformation stage offers the opportunity to escape the "design as mapping" dilemma (Tomiyama, 1994). Fluctuating information, that is, temporary data that were produced on the basis of previous design knowledge (Oeser and Seitelberger, 1995); (Hoffmann and Kollingbaum, 1996), is modified to form the basis of a creative extension of the design problem space (McLaughlin, 1993). In the urban design example the row house building type is transformed *in situ* into a new building type.

6.3 Learning

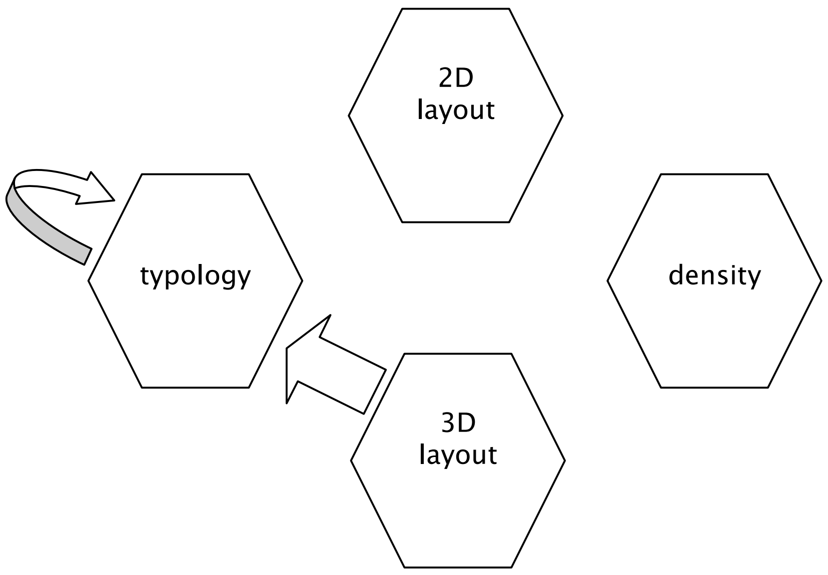

Figure 9. Learning

Learning as a result of a creative transformation occurs under the following conditions:

– A design perspective has received requirements resulting from specifications in a previous design stage.
– The flow of translation is reversed in order to adapt the set of possible specifications.

In the urban design example the new building type in the 3D perspective is translated into requirements for a new building type in the typology perspective. The human designer adapts the typology accordingly.

7. CONCLUSION AND OUTLOOK

In this paper, we have presented a new design model for creative design support, based on the notion of perspectives, different aspects of a design which provide separate and self-contained views of the design object. The interaction between perspectives happens in terms of transformations that map the current description in one perspective into requirements for the other perspectives. Contrary to classical views of design support, we show that tolerating inconsistencies between perspectives can in fact lead to better support of creative design processes. We have described the applicability of the design model to different kinds of design support and have outlined a prototype implementation. The perspective-oriented problem representation is naturally suited to an agent-based implementation approach. Usability of the prototype will be tested by urban design experiments involving designers with different levels of expertise.

8. REFERENCES

Bentley, 2000, http://www.bentley.com/products/mstation/j/

Brazier, F. T. M., P. H. G. Langen, Zs. Ruttkay and J. Treur, 1994 "On Formal Specification of Design Tasks", J.Gero, F. Sudweeks (eds.), *Artificial Intelligence in Design '94,* Kluwer, Dortrecht, The Netherlands, p. 535-552.

Brown, D. C. and B. Chandrasekaran, 1989 *Design Problem Solving: Knowledge Structures and Control Strategies*, Morgan Kaufmann, San Mateo, CA, USA.

Forgy, C. L., 1982 "Rete: A Fast Algorithm for the Many Pattern/ Many Object Pattern Match Problem", *Artificial Intelligence* 19 (1982), pp. 17-37

Haymaker, J., E. Ackermann and M. Fischer, 2000 "Meaning Mediated Mechanism, Prototype for constructing and negotiating meaning in collaborative design", J. Gero (ed.), *Artificial Intelligence in Design '00*, , Kluwer, Dortrecht, The Netherlands, p. 691-715.

Hoffmann, O. and M. Kollingbaum, 1996 "Creativity as Transformation: Multi-Agent Systems and Human Cognition", *Creativity and Cognition 2*, Loughborough University, Loughborough, England, UK, p. 19-26.

Hoffmann, O., M. Stumptner and T. Chalabi, 2000 "Consistency, Creativity and Perspectives", *Artificial Intelligence in Design '00*, workshop notes on Developing Intelligent Support for Collaboration in Distributed Design, Worcester, MA, USA.

Jeon, H., Petrie C. and M. R. Cutkosky, 2000 "JAT*Lite*: A Java Agent Infrastructure with Message Routing", *IEEE Internet Computing*, 4(2), (March/April 2000), p.87-96.

McLaughlin S., 1993 "Emergent Value in Creative Products*:* Some Implications for Creative Processes*". Modeling Creativity and Knowledge-Based Creative Design*, Erlbaum Associates, Hillsdale, NJ, USA.

Moreale, E. and B. Kinlay, 2000, JATLiteBean, http://waitaki.otago.ac.nz/JATLiteBean/

Oeser E. and F. Seitelberger, 1995 "*Gehirn, Bewußtsein und Erkenntnis*", Wissenschaftliche Buchgesellschaft, Darmstadt, Deutschland.

Panofsky, E., 1924, "Die Perspektive als symbolische Form," *Vorträge der Bibliothek Warburg,* Leipzig, Deutsches Reich.

Petrie C., 2000, "Agent-Based Software Engineering", Invited talk, *Fifth International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*, Manchester, England, U.K.

Prinz, D., 1995 *Städtebau* volume 1, Köhlhammer, Stuttgart, Deutschland.

Rao, A. S. and M. P. Georgeff, 1995 "BDI agents: From theory to practice," Technical Report 56, Australian Artificial Intelligence Institute, Melbourne, Australia.

Sandia, 2000, http://herzberg.ca.sandia.gov/jess

Tomiyama T., 1994 "From General Design Theory to Knowledge-intensive Engineering", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 8(4) (1994), pp.319-333.