Decision support for inner-city development

An interactive customizable environment for decision-making processes in urban planning.

Nils Seifert[1], Michael Mühlhaus[2], Gerhard Schubert[3], Dietrich Fink[4], Frank Petzold[5]
[1,2,3,4,5] Technische Universität München
[1] www.lib.ar.tum.de [2] www.ai.ar.tum.de
[1,2] {nils.seifert|michael.muehlhaus}@tum.de

Re-densification of inner-city areas is a highly topical socio-political issue. In order to meet the increasing demand for centrally located living space, the cities have to amend building laws. At present, there are no analytical means of directly comparing and assessing different strategies and measures aimed at identifying the potential for internal development in specific urban quarters and learning about the consequences for the citizens. In this publication, we describe an interactive digital tool that monitors the key building codes and visualizes their effects on the urban structure in real-time, so that this can serve as an informed basis for debate and argumentation in the political decision-making and planning process, consequently supporting the development of re-densification strategies that are well-suited to their urban context.

Keywords: *urban planning, redensification, decision support, visual programming, versioning*

INTRODUCTION

Urbanization has increased dramatically since the second half of the 20th century (Heilig, 2012). This is due to several factors - mainly because of demographic change and higher energy prices, but also because of tax procedures (e.g. "Gesetz zur Abschaffung der Eigenheimzulage" [1]). With it the demand for centrally located living space continues to rise strongly, especially in larger cities (Adam and Sturm 2011). Many cities are already exploiting their current building laws to the full and are now under pressure to amend building legislation to meet the increasing demand for affordable living space. To avoid uncontrolled growth, city-wide planning strategies - supported by a broad population - are essential (Fink et al. 2011). But what effect will specific changes to legislation have both on the target density and the appearance of the built environment? To answer this, it is necessary to develop case-by-case strategies that consider the existing building stock and its potential for infill development.

In the political discussion that precedes any amendments to building legislation, both the benefits and consequences of different measures can be difficult to assess. Public authorities have different means to control the building development in certain areas. Examples are regulations of minimum distances between buildings, clearance areas, limitations of building heights and amount of levels, but also in determining the zoning plan or development plan. At this point the question arises, what effect certain changes to the laws have, both on the target density and the appearance of the built environment. The redensification of inner city areas is a particularly sensitive subject because such decisions have a direct impact on the habitat of the citizens. For a democratic decision-making process, it is essential that the results and outcomes of alternative strategies are communicated and made available for public debate (Fink and Fischer 2012). At present, however, there are no analytical means for directly comparing and assessing different strategies and measures, identifying the potential for internal development in specific urban quarters and learning about the consequences for the citizens. This would provide the necessary basis for an informed debate.

REQUIREMENTS OF THE TOOL

In this publication, we describe an interactive digital tool that monitors the key building codes and visualizes their effects on the urban structure in real-time so that this can serve as an informed basis for debate and argumentation in the political decision-making and planning process and in turn support the development of redensification strategies that are well-suited to their urban context. As a planning support system (PSS), it enables planners to check various strategies and their execution variants, to compare several alternative approaches and to use their newly acquired knowledge to further substantiate their decisions (Klostermann 1997). In addition individual profiles and evaluation keys can be created and then applied to various planning areas. The direct visualization of results provides a quick overview of the qualities of different scenarios to citizens and authorities alike and provides information on the feasibility of the scheduled targets.

On the one hand, the tool serves as a basis for argumentation in the political decision-making and planning process, for example to verify the appropriateness of proposed changes to legislation vis-à-vis the existing building codes. In this context, the software assists in determining the effects of changes to building legislation. On the other hand, the software is able to support decision-making in the development of strategies for particular urban planning areas. An important aspect in this regard is to determine whether project design proposals are compatible with the applicable building codes and which advantages and disadvantages they may have over one another. The software makes it possible to consider planning strategies by automating key computation steps while continuously monitoring their fulfillment of the relevant building codes.

To meet these demands, basic requirements of the tool have been defined:

- Monitoring of building codes: the tool should take into account the key building codes and be able to visualize their effects on the urban structure.

- Comparability: there should be a way to compare different planning strategies and different states of work (Kwartler and Bernard 2001).

- Versioning: different approaches and the correlations and relationships between them should be plainly illustrated. There must be a way to reasonably archive stored data.

- Automatic analysis results: the key parameters and indicators - e.g. for urban density - should be calculated automatically, making it possible to directly compare different approaches (Gratzl et al. 2013). Results of planning measures should be visualized in real-time.

- Usability: the way of human information processing in general and particularly in planning and decision-making should be considered (Elgendy 2003). Therefore the operational concept should be intuitive and clearly structured.

- Extensibility: when new requirements arise or needs are identified in the process of working with the tool, it must always be possible to integrate the desired functionality into the system at a later stage. This should be possible without the need of any special programming skills.

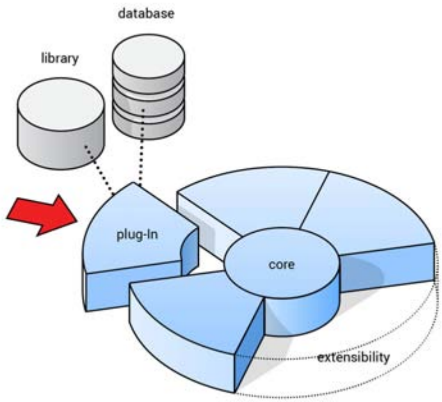


Figure 1
the modular design of the system

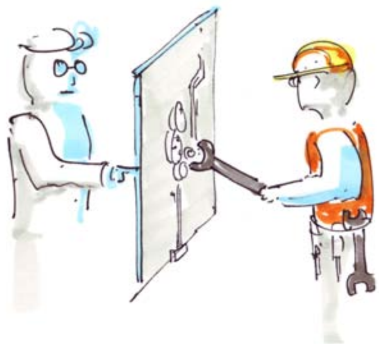


Figure 2
concept of the two modes: planning mode (left) and programming mode (right)

SETUP OF THE TOOL

As a decision support system for inner-city planning, a software environment was designed and implemented as a prototype (project video [1]). The developed software prototype is part of an interactive expandable IT concept (see figure 1). It was implemented in Java and evaluated by means of an exemplary inner-city planning scenario. The system is based on a software kernel that provides basic functions regarding user interaction, data management and program interfaces. It can in turn be expanded by plug-ins due to the modular software architecture of the program. This furthermore allows to integrate existing software data bases into the system (Degelmann and Miranda 1987). As a basic form of interaction, the kernel provides a visual programming interface. It enables the user to develop and customize desired functionality in real-time and thus implement user defined analysis and calculation methods.

The application is therefore divided into two modes: the planning mode and the programming mode (see figure 2). The planning mode shows a three-dimensional model of the selected building block, its urban context and also the control elements along with the results of the analyses. It helps users to quickly examine and assess different hypotheses. The programming mode, on the other hand, allows users to develop their own analytical methods and to adjust the functionality of the application using visual programming.

Planning mode

The planning mode (see figure 3) can be described as an interactive, graphically enhanced summary, where the results of planning measures can be visualized in real time. The working area consists of three main parts. In the input area on the left-hand side of the screen, parameters can be changed in real time with the help of controls such as sliders and buttons. The central part of the screen shows an axonometric view of the interactive parametric city model with the current state of the examined building block. The

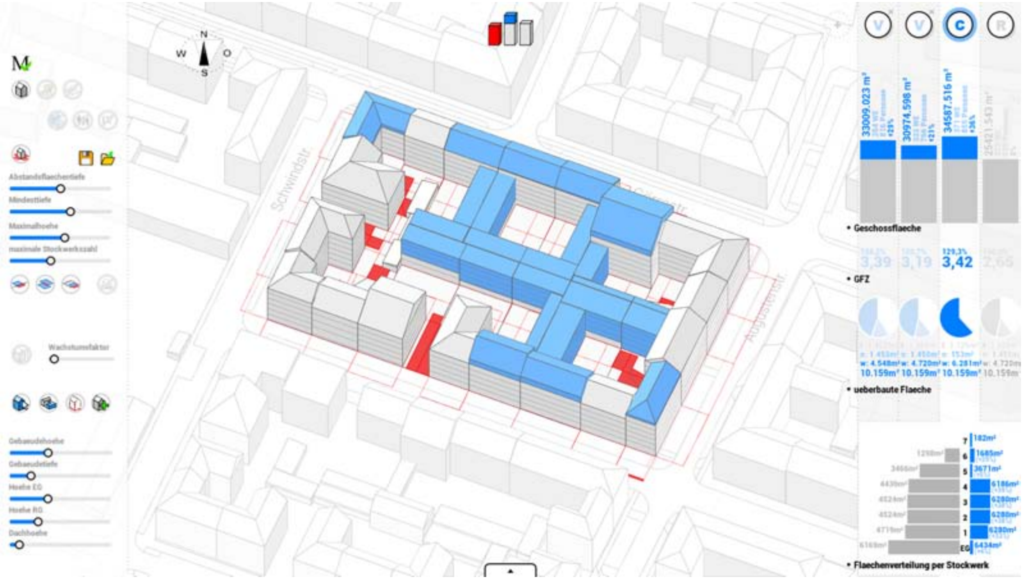


Figure 3
the working area in planning mode (screenshot)

user can see how building volumes have changed compared to the initial situation and can also directly manipulate the geometry of the buildings and check whether they still fulfil the applicable building codes. The output area on the right-hand side of the screen shows the results of the analysis. With the help of dynamic graphs and diagrams, the user is provided with direct feedback on whether targets can be achieved and how different parameters have changed in comparison to the initial state of the building block (Few 2012). In this manner, important parameters can be queried, allowing comparisons between different revisions and planning results.

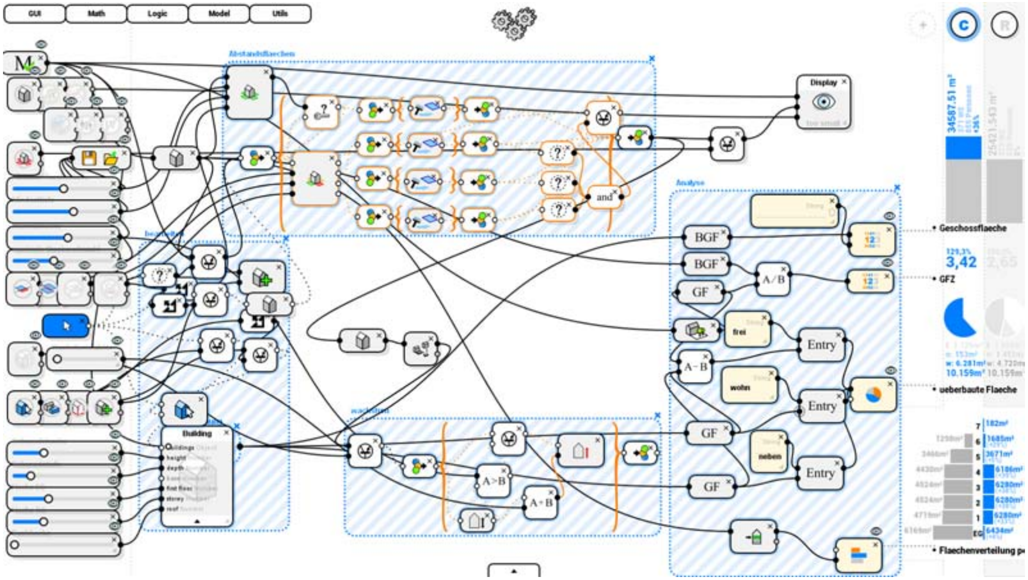
Programming mode

The functionality of the tool is defined using programming mode (see figure 4). The thus assembled definitions allow the user to access the implemented analysis and calculation methods in planning mode. Elements of the input and output area, which are also visible in planning mode can be integrated into the definitions for data input and output.

Visual programming has several advantages, especially for users with limited programming experience. Data flow and dependency graphs permit the user to directly manipulate the represented object-orientated structure (Hils, 1992). They can visualize individual objects, their condition as well as the interdependencies between them (Boshernitsan and Downes 2004). Consequently, users with little or no programming knowledge can work with the system without having to know textual programming languages and their syntactic details.

Different stakeholders are involved in the planning process - each with different backgrounds and perspectives (Achten 2002). Therefore it is necessary to provide different ways interacting with the functionality of the software. While on the one hand, for decision-makers in the municipal administration, the results of various planning considerations and their visual significance are more likely important, it must on the other hand, be possible for urban plan-

Figure 4
the working area in programming mode (screenshot)



ning professionals to quickly implement own calculation and analysis methods. To fulfill these requirements and with respect to extensibility, the application workflow has three levels of complexity:

- Using the application in planning mode: rapid results can be obtained by loading a pre-prepared workspace with a predefined set of functions. By using the control elements in the input area on the left-hand side of the screen, the user can change parameters and see the effects of these changes on the digital city model and the calculation results. In addition, planning variants can be exported as data sheets and 3D models intended for further planning and documentation.
- Development of functionality: individual calculation and analysis methods can be developed using library components and the graphical programming interface for use in planning mode. The results of the calculations can be visualized in the output area on the right-hand side of the screen using different kinds of customizable diagrams and display elements. This way, the user can implement own calculation methods based on the information contained in the city model.
- Development of individual components: the functionality of the system can be more fundamentally extended, for example to support complex simulation methods, by programming individual components and integrating them into the workspace. The graphical programming interface makes it possible to give these modules customizable graphical representations, and to link them to other components.

As a consequence, when new requirements arise or needs are identified in the process of working with the tool, it is always possible to integrate the desired functionality into the system at a later stage. We determined the essential requirements for the Visual Programming Language (VPL) as follows:

- Easy to use interface
- Exception handling
- Implementation of essential functionality like copy and paste or saving and loading
- Basic components to handle Strings, Numbers, Lists and Maps.
- Support for loops
- Appealing graphics that can be customized

IMPLEMENTATION OF THE TOOL

Due to the modular architecture of the IT-concept, different aspects of the program were developed as independent libraries in JAVA. The main parts of the software are described below.

3D city model

The data basis of the tool is a semantic three-dimensional city model (Kolbe 2009). It is an information model for the representation of 3D urban objects like blocks, land parcels, buildings and trees. It defines the classes and relations regarding geometric and semantic properties and the relations between them. Besides geometrical information such as area sizes, volumes, distances and heights, ownership structures are of crucial importance in the field of building legislation. Therefore the data-model has a hierarchical structure that consists of nested classes representing the logic structure of urban objects (building ⊠ plot ⊠ block ⊠ model, see figure 5). The parametrical building class makes it possible to change the geometry of buildings in accordance with properties such as depth, height, number of floors or roof shape (see figure 6). The objects degree of abstraction is adjusted to the information needed for working with the building codes. While oriels and small setbacks can be neglected, roof shapes and

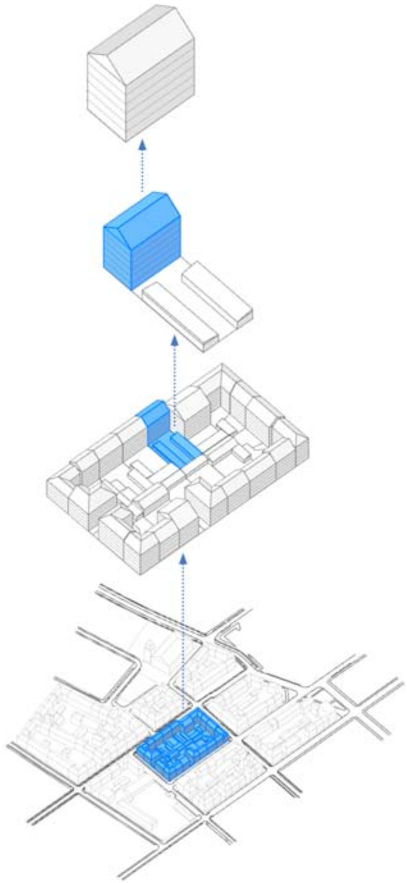


Figure 5
hierarchical structure of the city model

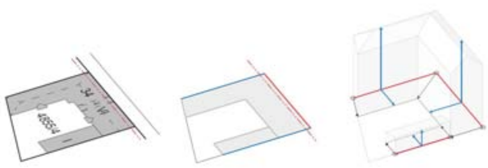


Figure 6
the parametric building model

floor heights are vital in building legislation. The geometrical, topological and semantical data that are required for the 3D city model of the tool can be obtained using interfaces to different data sources such as cityGML, ArcGIS or digital real estate cadasters.

Visual programming language (VPL)

The visual programing language is developed as a standalone library for Java, containing all the basic components to handle the common data types. New functionality can be added by using the provided protocol (Java: interface). The elementary components of the VPL are nodes, connectors and connections. The functionality is represented by the nodes. Each node can have one or more inputs and/or outputs. Inputs and outputs of the same data-type(s) can be connected. These connections define the data flow (Shu 1988).

One key aspect in the development of the VPL was to provide the possibility to build patterns out of individual functions, ie parts of the definition (Alexander et al. 1977, Gamma et al. 1994). This provides a distinct advantage: once a certain functionality has been defined via graphical programming, it can be saved and thus be reused. On this way partial aspects of a project such as analyses or calculations can be included into other projects.

To allow working with the visual programming system, a range of components have been prototypically implemented in the application. They can be selected from a component library which appears in the upper left-hand corner of the workspace in programming mode. The implemented components range from general logic and mathematical functions to more project-specific components for working with the urban model, along with graphic elements for data visualization and control elements. The library is designed much like an extensible toolbox so that components can be integrated into the system at a later date.

Versioning

Planning is a complex process consisting of a sequence of decisions, in which falling back to earlier versions or ideas or exploring alternative avenues is an inherent part (Rittel and Weber, 1973). It is therefore not a linear process. The process of developing ideas is better expressed as a tree structure. Branches occur at key points in the design process, where the idea is developed in different directions. At other points, separate strands may also converge again to merge different considerations into a new result. Besides a simple, but yet effective visualization of the interdependencies between different revisions as a graph structure, the possibility to directly return to any previous state of work and thus consider ideas and elements of different revisions, brings in a significant advantage: correlations and relationships between different versions can be clearly illustrated. Therefore the user can keep them in mind, but still keep his focus on the desired aim. This in return, reduces his effort of documenting and storing data, especially in extensive projects.

To meet these requirements, a versioning system was developed to manage all states of work. It can be understood both as a data model and graphical representation of the dependencies between different work states and results. For data backup and data exchange it is possible to save and load the whole repository. In addition individual branches can also be extracted, for example to integrate them into other projects and to make single partial results of different approaches comparable. The graphical representation of the repository is placed at the bottom of the working area. Here the user can see the dependencies between different revisions to follow the decent and the line of development of every state of work. To directly compare different revisions based on analysis and calculation results, they can be placed in the output area on the right-hand side of the working area via drag and drop.

Export and presentation of results

Planners and architects use digital data on the one hand and haptic models and printouts on the other hand. Regarding communication, discussion, archiving and submission to authorities, communicating data is profoundly crucial (Geertmann and Stillwell 2004). Various professionals and non-professionals have diverse views on topics that need to be respected. Large organisations such as administration authorities have a high demand for information sharing (Huber 1984). It is managed not only by digital data, but also via printed media. Therefore the tool provides the possibility to create templates, describing the needed characteristics that are assembled into a layout.

Figure 7
exported data sheets, showing key parameters and indicators of different planning strategies

Figure 8
3d printed model, showing planning results

The results of the studies can be exported directly in a way that no further post-processing is necessary. Data sheets showing selected properties (e.g. floor area, number of residential units, floor area ratio) are instantly available and can be used to communicate the results and act as a basis for further discussion (see figure 7). To allow the results to be reviewed using a real, haptic model, the tool has a three-dimensional geometric data exporter which outputs data that can be used directly - i.e. without the need for further preparatory steps - to produce real models of the result using systems for digital fabrication (CNC-Milling, Rapid Prototyping, see figure 8).

EVALUATION

To evaluate the software prototype, we selected a sample inner-city block consisting primarily of five to six storey buildings plus roof level and existing garages, one and two storey workshops and residential buildings in the courtyards within the block. A series of studies were created showing various stages of densification on the selected area (see figure 9).

CONCLUSION

In this publication, we described an interactive customizable digital tool that monitors the key building codes and visualizes their effects on the urban structure in real-time, for supporting the development of redensification strategies. After the requirements for such a tool were defined, a software environment has been designed and implemented as a prototype. It was then evaluated by means of an exemplary inner-city block.

The current implementation of the tool is already usable in a productive workflow. A range of basic components have been implemented and successfully evaluated using a specific urban block as an example. The facility to work directly in the parametric urban model and the ability to extend the range of functions at any time offer a suitable basis for a productive workflow.

In future research, we intend to develop and evaluate additional components in order to expand the collection of analysis and simulation methods. For example, sunlight simulation and shading analyses are useful in dense inner-city situations, as are distance determination and wayfinding analyses. These can provide further useful information to support the decision-making process. In further development of the tool new planning fields will be accessed by

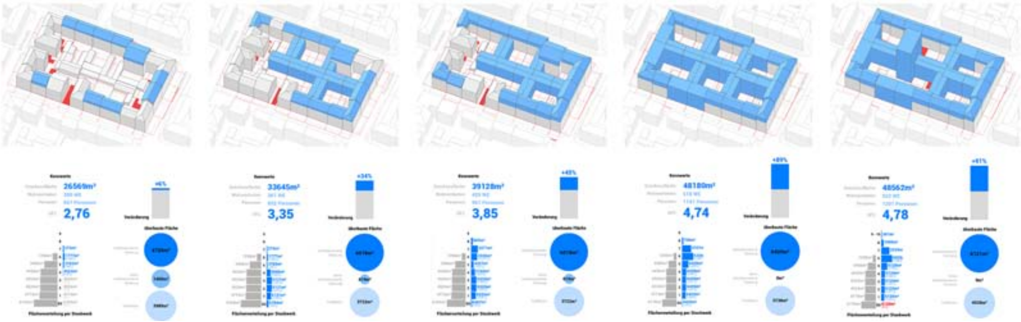


Figure 9
series of studies, showing various stages of densification of the inner-city block

adding the needed functionality to the toolbox. Real estate development for example is one of the major drivers of city densification. Detailed and reliable data regarding building volumes, facade surface area and other relevant factors should directly be taken into account for calculations concerning costs, finances and risk management. In this manner the feasibility of different planning approaches could also be monitored in real time.

REFERENCES

Achten, H 2002 'Requirements for Collaborative Design in Architecture', *DDSS 2002*, pp. 1-13

Adam, B and Sturm, G 2011, *Zurück in die Stadt: Gibt es eine neue Attraktivität der Städte?*, Bundesinstitut für Bau-, Stadt- und Raumforschung (BBSR)

Alexander, C, Ishikawa, S and Silverstein, M 1977, *A Pattern Language: Towns, Buildings, Construction*, Oxford University Press, New York

Boshernitsan, M and Downes, MS 2004, *Visual Programming Languages: A Survey*, University of California, Berkeley

Dengelmann, LO and Miranda, V 1987 'Development of Interfaces for CAD Processing in Architecture', *ACADIA Conference Proceedings 1987*, North Carolina, pp. 95-104

Elgendy, H 2003, *Development and Implementation of Planning Information Systems in collaborative spatial planning processes*, Ph.D. Thesis, Universität Fridericiana zu Karlsruhe

Few, S 2012, *Show Me the Numbers: Designing Tables and Graphs to Enlighten*, Analytics Pr

Fink, D and Fischer, F 2012, 'Neue Werkzeuge für die dichte Stadt', *Bauwelt*, 195, pp. 46-49

Fink, D, Fischer, F, Spengler, J, Gebhard, A, Schnetzer, B, Schinabeck, J, Wagner, T and Häusler, R 2011, *Langfristige Siedlungsentwicklung München: Qualifizierte Verdichtung*, TUM

Gamma, E, Helm, R, Johnson, RE and Vlissides, J 1994, *Design Patterns. Elements of Reusable Object-Oriented Software*, Prentice Hall

Geertman, S and Stillwell, J 2004, 'Planning support systems. an inventory of current practice', *Computer, Environments and Urban Systems*, 28, pp. 291-310

Gratzl, S, Lex, A, Gehlenborg, N, Pfister, HP and Streit, M 2013, 'LineUp: Visual Analysis of Multi-Attribute Rankings', *IEEE Transactions on Visualization and Computer Graphics*, 19, pp. 2277-2286

Heilig, GH 2012, *World Urbanization Prospects: The 2011 Revision*, United Nations publications, New York

Hils, DD 1992, 'Visual Languages and Computing Survey: Data Flow Visual Programming Languages', *Journal of Visual Languages and Computing*, 3, pp. 69-101

Huber, GP 1984, 'Issues in the Design of Group Decision Support Systems', *MIS Quarterly*, 8, pp. 195-204

Klostermann, RE 1997, 'Planning Support Systems: A New Perspective on Computer-Aided Planning', *Journal of Planning Education and Research*, 17 (1), p. 45–54

Kolbe, TH 2009, 'Representing and Exchanging 3D City Models with CityGML', in Zlatanova, L (eds) 2009, *3D geo-information sciences*, Springer, Berlin, p. 15–31

Kwartler, M and Bernard, RN 2001, 'CommunityViz: An Integrated Planning Support System', in Brail, RK and Klostermann, RE (eds) 2001, *Planning support systems. Integrating geographic information systems, models, and visualization tools*, ESRI Press, Redlands, CA, pp. 285-308

Rittel, H and Webber, M 1973, 'Dilemmas in a general theory of planning', *Policy Sciences*, 4(2), pp. 155-169
Shu, NC 1988, *Visual Programming*, Van Nostrand Reinhold Company Inc
[1] https://vimeo.com/79028980