Architects & Algorithms

Developing Interactive Visualizations for Architectural Communication

Nils Seifert¹, Frank Petzold²
¹,² Technical University of Munich
¹,² {nils.seifert|petzold}@tum.de

The paper presents the concept and results of a seminar that addresses the intersecting fields of architecture and urbanism, data and information visualization as well as information technology. In the first part of the paper, an introduction to the seminar topic and relevance in the context of architectural education and practice is given. Subsequently, the course concept, the learning contents and the corresponding learning objectives are presented. In the second part, selected student projects are shown as exemplary course results. In the conclusion, the results of the seminar for students, teachers and research implications are discussed. The overall aim of this publication is to draw on the experience gained in this field of education to offer starting points for others in developing similar teaching concepts and support for their implementation.

Keywords: Urban Planning, Programming, Information Design, Data Visualization, Smart City, Processing

INTRODUCTION

Architecture and urban planning have become data-rich professions. The paradigm shift towards the information society and the use of computers as a primary working medium have heavily influenced the work of professionals and students. This applies to both work processes and the profession's fields of expertise.

Nowadays architectural information and planning data are created, stored and communicated increasingly in digital form. On an urban and regional planning scale, this is primarily done using Geographic Information Systems (GIS), 3D city models or Urban Information Modeling on a building design scale the key method is Building Information Modelling (BIM). The potentials of these digital methods have been extensively discussed. What became clear is that in addition to their conventional fields of activity, the availability of information and access to an increasing supply of data creates new possibilities, challenges and fields of work for architects and urban planners.

Challenges

The development of our cities cannot be considered isolated from technological developments. Technology interweaves into urban systems in the same way as social, political or economic systems do. The term Smart City and its diverse interpretations can at least be reduced to this common point.

Looking to an ever-increasing amount of previously inaccessible and newly generated data, pro-
duced among other things by the growing use of mobile devices, the Internet of Things (IoT) or satellite imagery this poses urban planners to major challenges. In order to address these challenges and not leave them to third-party stakeholders with unknown objectives - especially in the light of recent data scandals, it is important to enable and foster future urbanists to develop a profound position to participate in the discourse.

Data and algorithms do not reflect an objective picture of the world. Instead, they rather create the rules of the digital realm. To participate in the discussions around the effects of digital technologies on society and our cities, it is necessary to understand how these rules are formed (Lessig 2006). For today's architectural education this doesn't mean to train architecture students as data specialists or programmers, but to encourage and enable them to develop their own technological proficiency as well as creative and visionary skills to be able to participate in the discussions, which affect not only the work processes of architects and urban planners in the proclaimed age of the Smart City but ultimately our everyday lives.

**Potentials**

Intelligibly communicating ideas and relationships is a fundamental part of the work routine of urban and regional planners. In an urban planning scale, where political decisions and planning measures have decisive influences on the lives of the people, the communication of their implications for the different aspects of citizens’ lives is an important prerequisite: “communication is not everything, but planning without communication is nothing” (Quote translated by the author; Selle 2005). Selle states that communication becomes a critical point whenever people plan for other people (ibid.) - in urban planning that's the nature of the game.

Digital data on spatial structures and processes are nowadays available in abundance. The free access to information and availability of an ever-increasing amount of open data has grown exponen-

tially since the beginnings of the internet. “Access to data or information translates into empowerment; power to make informed decisions, to solve problems, to generate economic activity, to improve living standards and, in the case of humanitarian emergencies, to protect and save lives.” (Stauffacher et al. 2012, p. 2)

In addition to unstructured data, which is e.g. the result of the growing use of mobile devices or the Internet of Things, there is also a growing number of web services that provide structured linked open data. On a spatial level, open geo databases allow access to structured geographical data of different levels of detail and at different scales and thematic references. Besides that, there are various other thematic data sources, for example from the fields of culture, science, finance, statistics, weather and environment [1]. From official sources e.g. for governments, “open data has been established as a fundamental cornerstone of official transparency and accountability initiatives” (Gray 2014). In a world where a massive amount of data and information - both previously inaccessible and newly generated - are freely available, the challenge lies in putting this information to use effectively. This means selection and organization of the underlying data, putting them into context in order to gain useful information and presentation of this information as a means of communication to foster understanding. The fastest and most effective way to both understand relationships and make them comprehensible is to visualize them.

**A First Résumé**

*Code literacy* is essential for both, facing the challenges just presented and the utilization of potentials. It is not necessarily directly connected to learning a certain programming language, but implies the understanding of algorithms and code and its intentions in a certain context (Dufva & Dufva 2016). This empowers to explore new means and solutions - to shape own tools and not just use them as an end user.

“The ability to ‘read’ a medium means you can access materials and tools created by others. The ability
to ‘write’ in a medium means you can generate materials and tools for others. You must have both to be literate. In print writing, the tools you generate are rhetorical; they demonstrate and convince. In computer writing, the tools you generate are processes; they simulate and decide” (Kay 1989, p. 125).

**COURSE CONCEPT**

The aim of the seminar was to explore the potential of interactive visualizations for the communication of architectural information in an urban and regional scale. The course, which has meanwhile been offered four times, addresses master level students from the fields of architecture, urbanism and landscape architecture. The students acquired theoretical knowledge about information design and data visualization, as well as the technical foundations for accessing, collating and using open data sources. While conveying practical programming skills, the semester task for the students was to identify specific research questions out of their experience on urbanism and urban planning topics and apply their newly acquired knowledge in a project work. Important milestones in the progress of the student projects during the semester were developing a concept idea, research on data sources, developing an implementation concept and the implementation of a prototype. These milestones were reviewed in the form of interim presentations.

The challenge for the students during the semester was on the one hand to acquire the theoretical knowledge to be able to develop a well-founded concept from an initial idea and on the other hand to gain the practical skills to prototypically implement it. Therefore, in the first half of the semester the focus was primarily on a theoretical exploration of the topics data and information science, (geo) information systems, information design and data visualisation as well as the acquirement of the necessary programming skills, while in the second half it shifted to the individual project works.

In weekly workshop sessions, the students, who were mostly novices in the field of programming, acquired basic programming skills using the open source software *Processing*. The programming language and integrated IDE *Processing* was developed by the MIT and focuses on the user group of designers, artists and programming beginners [2].

The theoretical exploration of the seminars complex of topics was done by lectures and student presentations as well as talks by invited experts, for example from the professions of data science or media architecture - and subsequent discussion between students, teachers and experts. The knowledge that the students acquired in this discourse to approach the semester task can be divided into two parts: design aspects and technical aspects.

**Design Aspects**

The first part addresses the creative and graphical aspects of creating visual representations. The key question is: What means of design can be used to convey visual information in an understandable way? To come closer to answering this question, the students approached this topic from two directions:

- On a theoretical level, they presented and discussed basic models and theories of information design, such as Jacques Bertin’s Visual Variables (Bertin 1983), Itten’s color theory (Itten 1973) or Edward Tufte’s writing about the visual communication of information (Tufte 1990, 2006). By researching these theoretical foundations, the students learned how to structure the information they want to present and to identify suitable visualization strategies.

- To apply and consolidate this theoretical knowledge on a practical level, the students investigated examples of visualizations ranging from historical cartographic representations to state-of-the-art interactive data visualizations (see e.g. Figure 1).

This investigation of theoretical foundations of information design and data visualization as well as the evaluation of various examples had two objectives:
on the one hand, for the students to identify the design tools they have at hand, e.g. for presenting qualitative and quantitative differences or for displaying different kinds of information like comparisons, proportions, range, hierarchy, distribution, relationships, location or flow. On the other hand, to develop a sensibility to the fact that for human perception, each choice of visual representation has a decisive influence on the usability of the underlying information. In the sense of this usability, there is no better or worse, it rather depends on for what the information is to be used and for whom it is intended. Since there is no visual representation that highlights all aspects of the underlying data at the same time, trade-offs are a fundamental part in designing them (Gelman & Unwin 2013). The greatest challenge for the students in this regards was to make obscured but relevant information more visible and explorable.

**LEARNING PROGRAMMING**

The focus of interest for us as teachers was particularly on the question of how the architecture students, who had little or no programming experience prior to the seminar, could acquire the necessary skills to be able to implement their own ideas within one semester. According to their own assessment at the beginning of the seminar, about 3/5 of the students stated that they had no experience with textual programming at all, while about 1/5 already had gained experience but considered their knowledge to be insufficient. The remaining 1/5 already had basic programming skills, for example by learning a scripting language within another seminar.

Learning a programming language means overcoming learning barriers. Brad A. Myers discussed this issue in *Past, Present and Future of User Interface Software Tools* (Myers et al. 2000): The more sophisticated the problem to be developed, the deeper the understanding of the underlying programming mechanisms must be. Ideally, the understanding should build up linearly: at increasingly complex challenges, the knowledge of what the learner of the

Figure 1
Graphical railway plan, original from E. J. Marey “La méthode graphique”, 1885 (figure from Tufte 1983)
programming language already has gradually leads to more understanding. During the learning process, however, learning barriers occur whenever the students have to accumulate a great deal of additional knowledge at once in order to gain a greater understanding of the programming language, for example when learning new programming concepts such as loops, conditional statements, methods, classes, and so on. In textual programming, learners are confronted with a certain learning barrier right from the beginning, since before they can start, they have to acquire basic knowledge about the syntactical language elements.

Based on Myers' findings, Andrew J. Ko specified six learning barriers that relate to different phases of problem solving in programming: design, selection, coordination, use, understanding, and information (Ko et al. 2006). These models gave us valuable indications to identify barriers in the students’ learning process and to develop target-oriented strategies for executing the programming workshops. With increasing sophistication of the students project works, accordingly different teaching strategies were applied: After learning the basic programming language elements such as variables, loops, lists, trees, methods, classes, inheritance and recursion, as the students’ knowledge progressed, we more and more introduced prepared modules to the students that offered more complex functionality (see Figure 2).

Developing own solutions
In order for the students to develop their own solutions, a fundamental understanding of syntactical language details, basic programming concepts and the principles of basic algorithms is necessary. Throughout the Processing workshops, real-world metaphors with references to the students’ professional domain and animated illustrations of algorithms were helpful tools to foster the students’ understanding. To evaluate the teaching process, during the semester, the students were asked to answer questionnaires that were based on Ko’s considerations. Their review has indicated that coordination and use barriers were the major challenges. This means that on the one hand finding out, what programming concepts they need in order to implement their own ideas posed a challenge to the students. On the other hand, even when the students knew what concepts they need, there still was the difficulty of finding out how to put them together. This has been shown prior by Spohrer and Soloway. They demonstrated by studies, that plan composition problems, i.e. the question of how individual components can be structured to match the mental model of a programming concern are a major cognitive challenge in programming (Spohrer & Soloway 1989).

Adapting examples
Especially for beginners in object-oriented programming, a helpful approach to overcome this cognitive challenge is to apply existing solutions to one’s own problem in order to be guided by examples of problem solving (Robins et al. 2003). Therefore, the approach to convey an understanding of basic programming mechanisms was to design and implement exemplary (geo) information visualization components such as cartograms and different kinds of diagrams or basic components of user interaction together with the students. These partial solutions could then further guide them to develop their own ideas.
Using prepared modules
At an advanced point in the semester schedule, we introduced prepared modules addressing more sophisticated topics. These modules provided the students with functionality that would require a great deal of technical effort and additional knowledge to implement on their own, e.g. of establishing HTTP connections or about data structures of certain web services. Other than the examples of information visualizations, where developing an understanding of how the code works, also provides a positive learning effect, the focus in this category was to develop a task-specific understanding of how to use the modules. This means, learning what the modules do and what the interfaces require and provide, but without developing a full understanding of the modules’ internal mechanisms.

The Processing environment already offers a large variety of useful libraries, that are either distributed with the Processing IDE itself, or by third-party developers. In addition, we prepared a number of modules that are tailored to the task of the seminar. Examples are:

Web Services SDK. The collection of libraries provides helper classes and methods to establish an HTTP connection to a web service and process the returned data. The predefined modules provide a usable way for students to retrieve data from (open) web services such as the APIs of GeoNames, DBpedia, Flickr, Twitter, Google Maps or Wolfram Alpha. In addition, external libraries are linked to parse the result in XML or JSON/GeoJson format.

OpenStreetMap SDK. Based on the Web Services SDK, the libraries facilitate access to the open geo-database of OpenStreetMap [3], which has proven to be a useful source of geographical data for many student projects. It offers several prepared modules to retrieve and process OpenStreetMap data via the Overpass API. The libraries include functionality

- to build up a search query using criteria such as search terms, object type filters or a geographical bounding box.
- to establish an HTTP connection to the Overpass API, send the query and receive the result.
- to process the result and translate the topological OSM data structure into JAVA classes. For this purpose, a class structure is provided that relates to the corresponding OSM data primitives. Furthermore, customizable methods are provided to draw the OSM objects in Processing using the UnfoldingMaps [4] library.

SVG & CSV I/O Library. In addition to the resources for working with web services, basic functions for importing and processing local data sources such as Excel tables or plans in SVG format are provided. Based on the built-in processing libraries, additional functions are offered here, for example for interaction with vector graphics or to find specific objects in an SVG file or Excel table by name.

Paintbox. This library provides subroutines to work with one- or multi-dimensional color gradients. In addition to prepared color codes, the collection offers interpolation methods for translating numerical values into colors.

COURSE RESULTS
The student’s projects have explored a wide range of topics on different scales from building scale to global scale. In addition to the projects presented below, some of the topics covered were urban infrastructure networks, migration, trees and green spaced, public transport, the influence and relationships of the city’s architectural schools, analysis of housing quality or twin towns and city friendships.

To give insights to some of the results of the course in more detail, three exemplary student projects with different thematic orientations, scales and data sources are presented:

Indicators of Gentrification
Students: Besjan Rahmani, Daria Rath

Gentrification is a process that often follows typ-
Gentrification is a complex matter that requires a number of different indicators to be considered. In order to obtain a meaningful analysis of the topic, the students had to collate various initial data. The selected indicators were

- the official representative list of rents,
- the internal migration within the city,
- the average age of the local population,
- the unemployment rate and
- the average household income.

For these topics, initial data were collected from several official sources like municipal authorities and statistical offices. The prerequisite for later visualisation of the data was that they were available at district level for the past five years. On this basis, the students developed an interactive map, which allows the different indicators to be explored and compared on a spatial and temporal level (see Figure 3). Various details can be displayed on demand for detailed comparisons.

The result showed a significant point of how data is published from official sources: Even when the initial data on a topic is fully publicly available, it is often not yet made possible to gain understanding or draw conclusions out of it. It takes a lot of additional effort and the political will, to put the data into a certain context and present it in a comprehensible way.

**Age of Amsterdam**

Students: Roman Freistätter, Victoria Rusina

The idea behind the student project *Age of Amsterdam* (see Figure 4) was to make the history of the city and its development explorable according to typology and historical period of the buildings. An interesting picture was generated by the possibility to filter the city’s buildings according to their year of construction, thus revealing urban patterns and a patchwork of different phases of urban development. The students chose Amsterdam for a proof of concept prototype since on the one hand the city shows an interesting mixture of different buildings from the Middle Ages to the modern age and on the other hand sufficient data is available in the OpenStreetMap database regarding typology and the year of construction or renovation.
The result is an interactive prototype, that allows the heritage of the city to be explored. The underlying data is retrieved in real-time via the Overpass API according to the specified filters (e.g. type of building, range of construction years). This means that the visualization can also be applied to any other city, provided the required attributes are available in sufficient quality. The user can evaluate the result by different kinds of diagrams.

**Flight Route Connectivity**  
Student: Eva Kukurite

The project *Flight Route Connectivity* (see Figure 5) visualizes the interconnectedness of cities worldwide by means of flight connections. The foundation is a comprehensive static data set containing all flight connections (about 70,000) connecting the 6977 airports worldwide. The data set was collated by the three big airline umbrella organisations. Besides the geographical location, name and continent of the airports, also the flight connections are assigned attributes like aircraft type, aircraft size, frequency or departure/arrival in local time.

The complex data set can be explored via an interactive world map. Different colours signal the five continents. The flight connections are displayed as geodetic lines. By mouse interaction (*zoom, pan, details-on-demand*) the data set can be explored ranging from regional up to intercontinental flight connections. In addition, an animation shows the progress over the day.

**CONCLUSION**

**For the students**

The seminar helped the students learn to think strategically and how to structure their work process towards the final result. The students were particularly enthusiastic about the variety of possibilities. While most of them have considered themselves as mere software users before, the experience of being able to generate software tools themselves was a major discovery, which motivated many to put great commitment into the work on their projects. The direct connection of programming, visualization and interaction that is possible through *Processing* has made a major contribution here.

In addition to the practical programming skills, the students acquired an understanding of how to exploit potentials of information technology for the communication of architectural information. Some interesting discussions within the seminar have shown that they developed personal interest and a healthy attitude towards the potentials, challenges and possible risks of the utilization of (open) data.

**For the teaching staff**

We especially focused on the question which learning barriers the students were confronted with when developing a basic understanding of programming mechanisms and applying this knowledge to particular set of own questions. We have learned a lot about which teaching methods make it possible to efficiently convey complex technological contents in a relatively short period of time. Through the prepared modules and examples, ranging from code snippets to advanced SDKs, the students were able to implement their original ideas within the semester. In the recent iterations of the course we were able to further improve this method.
Besides that, in times of ongoing digitalisation, the accessibility of digital tools, in particular the introduction of domain-specific user groups to programming will continue to be an important topic. In connection with ongoing research projects, we were able to draw useful conclusions from the evaluation of the students’ learning processes.

**Implications for research**

The combination of information visualization, information technology and specific questions from an architecture and urban planning context has revealed potential for architectural communication. The variety of student projects have resulted in a number of connections to different fields of research. Especially the interaction with data and information can offer valuable tools for analysis and communication. The students have shown creative ways to make political and planning processes more transparent and to develop efficient public information and participation processes supported by digital tools.

**REFERENCES**


Dufva, T and Dufva, M 2016, ‘Metaphors of code - Structuring and broadening the discussion on teaching children to code’, *Thinking Skills and Creativity*, 22, pp. 97-110


Spohrer, JC and Soloway, E 1989, Studying the Novice Programmer, L. Erlbaum Associates, Mahwah
Tufte, E 2006, Beautiful Evidence, CT: Graphics Press, Cheshire