A Design Methodological Framework for Interactive Architecture

Henri Achten[1], *Miloš Kopřiva*[2]
[1,2]*Czech Technical University in Prague, Czech Republic*
[1]*http://www.fa.cvut.cz/cs/o-nas/lide-na-fa/hhenri-achten,* [2]*http://www.fa.cvut.cz/cs/o-nas/lide-na-fa/milos-kopriva*
[1]*achten@fa.cvut.cz,* [2]*kopriva@fa.cvut.cz*

Abstract. *Interactive architecture is a fairly recent phenomenon enabled through new materials and technologies. Through experimentation architects are coping with questions of changeability, adaptability, and interaction. However, there are no comprehensive design methods to support this type of architecture. In this paper we aim to bring together methods that can support the design of interactive architecture. The methods are ordered in a methodological framework that provides an overview of possible approaches.*
Keywords. *Design methods; interactive architecture.*

Introduction

Throughout their lifecycle, buildings have to meet changing demands. Traditionally, architects have addressed this question by separating between the static part of a building which can remain more or less unchanged, and a dynamic part which can be left to the occupants as they see fit (Habraken et al. 1986, Leupen et al. 2005). In recent years we can see two important developments that may necessitate a different approach to flexibility: advancements in interactive technology (LED's, sensors, panel types, printing techniques, and so on), and a more adaptive and responsive approach to architecture (performative design, rapid prototyping, generative design, and so on). These developments point towards an architecture that can become more interactive.

Given that we are the start of such an interactive architecture, it is not surprising that there is no comprehensive framework to support design for interactive architecture. As opposed to the number of publications on computer-generated architecture, there are only a handful of publications concerning interactive architecture (Beesley et al. 2006, Fox and Kemp 2009, Schumacher et al. 2010). Nevertheless, it is also not the case that there is nothing available in terms of design methods. Several architectural offices such as UN Studio (Berkel and Bos 2006), OMA (Koolhaas 2004), and Greg Lynn Form (Lynn and Rappolt 2008) are taking into account dynamics to generate the logic of a design.

In this research we propose a methodological framework for the design of interactive architecture in which many different techniques may be used. The framework is divided into four informative steps for interactive design: A. Analysis, B. Concept generation, C. Simulation, and D. Assessment. We are fully aware that the design process has more steps (Roozenburg and Eekels 1994), but we focus on the above mentioned steps. From the field of interaction design we take two notions that can be applied in architectural design: the user experience brief and the persona.

User experience brief and persona

In interaction design, the concept of the design revolves around the user experience: how someone is actually involved when using a product, and how much satisfaction (in a good design) or frustration (in a bad design) he or she experiences. What matters is that a good design steers the user in an almost natural way how to use a tool. A good interaction design allows the user to fulfill his or her needs, without forcing the user to think too much about how to do it. Such an unobstructed sequence of actions by a user is also known as flow – therefore a good interaction design promotes flow in the user.

Is there an equivalent of the user experience in architecture? On face value, one would argue "yes." Obviously architects design for people. When architects talk about their design, they often argue for a design aspect through the perspective of the user. For example, an architect may state that the entrance of a building is obvious and inviting because it lies on the visual axis of the street and the entrance façade is elevated; the purpose of an atrium is for people to orientate themselves in a building, or for them to assess the complexity of a building.

Although this way of reasoning is much alike to the user experience, it is not quite the same. In interaction design expected user experiences are formulated in the design process and tested with people; in architecture they are not. Because architecture has a long tradition, architects tend to take for granted certain behavior of people in the built environment. They assume that given a certain configuration of spaces or circulation systems, people will act in a particular way. Almost no architect actually tests whether such assumptions are actually fulfilled after a building has been constructed and is left to the users. So-called POE – Post Occupancy Evaluation – is done mostly by facility managers. The knowledge gained from such studies hardly ever feeds back to architects.

For interactive architecture we cannot allow ourselves to be ignorant of this aspect: we simply do not know what people actually experience when confronted with new environments that engage with people in an interactive way. Therefore, architects should start learning again how architecture influences people, and how people influence architecture.

In a similar way the notion of the persona can be discussed. Interaction design usually concerns the design of mass-production products (remote controls, mobile phones, touch screens, and so on). There are therefore many different users for which the interaction designer has to create an optimal design. This raises two problems: for which person to design and how to understand the person's needs when not knowing the person him- herself? Architects face a similar problem, usually when concerned with mass-housing, office buildings, or shopping centers. The persona is a structured way to describe the target audience for which a product is intended. We apply this concept in architectural design.

Method structure

Each method that is identified in the framework below is presented in the same format. First there is a general *description* of the issues, reasons, and work approach of the method. Second, a concise *definition* is provided. Third, the method *how to do it* is outlined stepwise. Fourth, a worked out *example* is given of the method. Fifth, additional notes *what to look out for* are provided.

Analysis (A)

In the analysis phase, the design problem is investigated. It is necessary to identify the needs of the client, and assess the requirements that the building or urban design should meet. As the understanding of the design problem increases during the design project, the first analysis results are very likely to change throughout the design process. For interactivity, it is necessary to establish what aspect of the design should be interactive, who the involved actors (systems, environment, and people) are, and what the interactivity should amount to. Because of the novelty of the notion of interactivity, this analysis tends to be more explorative than in regular design.

Analysis techniques

In the following Table 1, analysis techniques are summarized.

Example: REMASC technological analysis checklist

Description: Technological developments are continually taking place. For application in interactive systems, building, or environments, developments occur in three major areas: technology, materials, and systems. Each of these groups has specific characteristics of development. Technologies such as the computer, internet, mobile phones, and so on tend to be rather comprehensive. This means that they usually are complex, consist of many components, and allow many kinds of activities. Architecture very often has a passive role in technology development. For materials the perspective is slightly different. In the past two decades there has been a lot of development (for example ETFE foil, translucent concrete, membrane surfaces, and so on), there are different material treatments (glass printing, finishes, gluing techniques, and so on), and new preparation techniques for materials (fabrication, CAD-CAM, mass-customisation, and so on). The characteristics of these new materials in terms of load-bearing capacity, thermal behaviour, the way they transmit light, and so on is quite different from the traditional materials. We cannot rely therefore on traditional knowledge or patterns of material use but have to investigate and find new applications.

Contrary to technologies, systems are more specific technological compounds that have a specific goal. HVAC systems for example, are comprehensive solutions to service air conditioning, heating, cooling, and so on. Variable ETFE cushions with different top and bottom prints form a system through which the amount of incoming daylight can be regulated.

It is very rare that for a given technology, material, or system it is immediately clear what its exact application is, let alone for interactive applications. Therefore, it is necessary to have a good general understanding what is currently used in interactive systems, how it is applied, how it enables interactivity, what is required to make it work in that way, and so on. Only then is it possible to become aware of potential use of other or new technologies, materials, and systems that may be applied in interactive applications as well.

There is no definite working method for scanning and investigating technological developments.

Table 1
Analysis techniques

Analysis techniques (A)	
A1. User-based analysis	User needs (A1.1), user-system interaction (A1.2), user activity schedules (A1.3), user change patterns (A1.4), user-response schemas (A1.5), single versus group users (A1.6), user group dynamics (A1.7), user contingency planning (A1.8), user comfort requirements (A1.9), user experience brief (A1.10), and the persona (A1.11).
A2. Time-based analysis	Cyclic analysis (A2.1), time functional depiction (A2.2), time cumulative depiction (A2.3), spider diagrams (A2.4), circle diagrams (A2.5), speed of change (A2.6), frequency analysis (A2.7)
A3. Function-based analysis	Function change analysis (A3.1), fitness for function change (A3.2), emergent functions analysis (A3.3), UML use case diagrams (A3.4).
A4. Object-based analysis	REMASC technological analysis checklist (A4.1), sensors and actuators (A4.2), change logic and controllers (A4.3), SPA (A4.4), obedience versus conflict (A4.5).
A5. Cost-based analysis	Cost of change (A5.1), profitability of interactivity (A5.2), assessment boundary (A5.3).

REMASC technology analysis checklist	Example question: "Can I use this technology to…"
Respond	...respond to changes, demands, factors, people, events, and so on in the environment or context of a system?
Enable	...make possible or enable that change, adaptation, movement, and so on can occur?
Materialise	...physically create components of an interactive system?
Actuate	...actuate or realize movement, rotation, scale, temperature, transparency, and so on?
Support	...support the realisation of an interactive system, not the interactivity itself, but the conditions to make it possible?
Control	...control the behaviour or dynamic aspects of an interactive system?

Table 2
REMASC technology analysis checklist

One way to do it is to keep in mind a checklist with relevant questions to ask when coming across something new. Here we provide a simple checklist of six questions to ask that are based on a general interaction cycle: respond, enable, materialise, actuate, support, and control – in short REMASC (see Table 2). For a given technology, simply ask according to the checklist, "can I use this technology to respond to something interactive... to enable something interactive... to materialise..." and so on. The checklist is inspired by the SCAMPER checklist for creative thinking (see Roozenburg and Eekels (1996), pp. 186-187)

Definition: Technological developments analysis is the ongoing activity of monitoring technological developments. For any given technology, it is possible to use the REMASC question checklist and the follow up questions to get a first insight in its potential.

How to do it: When reviewing a particular technology, use the REMASC checklist to investigate potential applications for that technology.

If the answer to any of the questions of the REMASC checklist is yes, then it pays to follow up with questions that ask in more detail how a technology may be useful (Table 3).

What to look out for: Because of their general nature, checklists never completely cover all possible aspects of an inquiry. Repeated use of a checklist also tends to introduce habituation, which lowers the critical view by which something new may be identified. Therefore, do not rely on the checklist alone, but also consult various other sources about the technology, see how it is applied, and so on.

Concept Generation (B)

In the concept generation phase a comprehensive principle solution for the design problem is formulated. The solution should be comprehensive in the sense that it tries to bring together various major aspects of the design problem – without necessarily being very detailed. The solution may involve a small part of the building or urban design, but it has to be explained what the impact on the building or urban design is. Just the generation of an isolated technical solution does not constitute a concept (even though

Follow up questions	Example question
How?	How can I use this technology, how does it work?
What else?	What else do I need for this technology to work?
Sufficient?	Is this technology sufficient to achieve what I need?
Combine?	How can this technology be combined with other parts of the system or technologies?
When?	When is this technology used in an interactive system, building or environment?

Table 3
Follow up questions after REMASC technology analysis checklist

it may lead to one). For interactivity, the concept should show the principles of interactivity of the solution – what we propose to call the "interactivity loop."

Concept generation techniques

In the following Table 4, concept generation techniques are summarized.

Example: Frame design drivers

Description: The design driver is a major parameter that informs the performance of an interactive system. Examples of design drivers are facade and window orientation with respect to the sun, material allocation in a structural system according to the load, shape of spaces according to the movement of people, and so on. As can be seen from these examples, a design driver combines a feature of the design with some aspect of the environment of that design. These are so-called performative design drivers and are closely related to performative design. Design drivers that concern only the feature of a design are called isolated design drivers. Isolated design drivers are for example the types of corridor systems for office buildings, organisation of a building layout by means of courtyards, and composing the building mass according to sight lines in an urban environment.

A design driver can only be a generative force in the design process if it is embedded in a design question. For example, façade and window orientation with respect to the sun can only assist in design when there is an issue: should the façade profit maximally from incoming daylight or should it be shielded from it; should the window orientation follow the sun path or not; and so on. Putting a design driver in such a design question is called "framing" as has been described by Schön (1983).

Definition: Framing a design driver means formulating the design driver in a design question. The question should be formulated in such a way that it is possible to assess whether or not the question has been solved.

How to do it: When dealing with a design driver, make sure that the following aspects are clear (for example by listing them somewhere): (a) characteristics of the feature of the design; (b) characteristics of the environment to which the feature is possibly related; (c) purpose that the feature fulfills in the buildings design; and (d) criteria which have to be achieved by the design driver. This list will be helpful to keep the goal of the design driver clear, and to avoid being side-tracked by irrelevant issues.

What to look out for: It is almost never the case

Table 4
Concept generation techniques

Concept generation techniques (B)	
B1. Identify design drivers	Identify major themes that can structure important parts of the design.
B2. Frame design drivers	Set a design question in which the design driver can operate.
B3. System view	Describe the whole of the building as a system.
B4. Design for emergence	Design a system bottom-up, not top-down. Allow for components to react to each other, so that they take profit from each other.
B5. Design by component	Focus on a key technological component from which to develop part of the design.
B6. Technology framing	Given a particular technology, set the technology in a particular context in which it will be used.
B7. Performative design	Set parameters of the building design, and link these parameters to one or more performance criteria.
B8. Design principles	Check how the design meets Norman's design principles: visibility, feedback, constraints, consistency, and affordances.
B9. User experience	Take the persona and related user experiences as a guideline for designing concepts.

that a design is based on a single design driver. Usually multiple design drivers can be at play. Therefore it is important to keep a good balance between these drivers and make sure that not a single driver becomes over important without good reason. Also the focus on a single design driver disregards interactions between design drivers that may lead to other issues and possibilities.

Simulation (C)

The generation of a concept does not conclude the creative process. It is necessary to determine whether the behavior of the design actually meets the requirements and needs of the user(s). As interactive systems have a very strong temporal component (something moves or reacts to actors), the evaluation of the performance of the interactive system during the design process has to be based on simulation. Simulation can vary from a digital simulation in a computer program up to a full-scale realization of a physical prototype.

Simulation techniques

In the following Table 5, simulation techniques are summarized.

Example: Wizard of Oz

Description: Wizard of Oz experiments (the term is attributed to Kelley 1983) are experiments in which a crucial part of the technology is not performed by the technology but by a human (the so-called "wizard"). This is done because either the technology is not available, too expensive, or does not meet the requested standard. Instead a person or some kind of prop stands in for this technology. Originally, Wizard of Oz experiments where mainly done in the area of Human-Computer Interface, in particular spoken or written human language recognition. Because human language recognition is very hard to achieve, this is often done by people behind the screens while test subjects would have the impression that the computer was doing the job (see for example Dahlbäck, Jönsson and Ahrenberg 1993). For interactive architecture, the Wizard of Oz experiment basically means that some part of a prototype is realized by something else than the piece of technology that should achieve that function.

Definition: A Wizard of Oz experiment is a working prototype of an interactive system, building or environment in which part of the technology is performed by a human or some kind of prop.

How to do it: In research, a Wizard of Oz experiment can only be done when the technology that is "faked" is not crucial to the nature of the prototype. When the goal of the prototype is not research oriented but more "proof of concept" like, then it is less important what part of the prototype is done by the wizard. The reasons for doing this can be various: technical complications, time pressure, cost, and so on.

Whenever a prototype is a Wizard of Oz experiment, the wizard part has to be well documented and reasoned. Otherwise, it can be easy to forget (especially for the non-informed audience or readers) that some part of the prototype is not properly functioning. When the wizard is a person then great care has to be taken that he or she does not perform outside his or her role. This is because of the tendency of people to react to the environment (they will help others when they perceive there may be a problem or uncertainty) or to interpret ambiguities by the

Simulation techniques (C)	
C1. Animation	Animation and interactive structure animation
C2. Physics-based simulation	Lighting and energy simulation. Environmental simulation.
C3. Performance simulation	Performance of the building; effectiveness.
C4. Prototype-based simulation	Wizard of Oz, wireframing.

Table 5
Concept generation techniques

users (thus outperforming a technology).

What to look out for: Wizard of Oz experiments are a fast way to get a working prototype even when some kind of technology is not available. It is important to keep a good check on the wizard, to see whether he or she is not doing things that cannot be reasonable expected from the faked technology.

Assessment (D)

A simulation only produces a kind of behavior of a design, but makes no statements whether such behavior is good or bad. Such a test is done in a different step, and is called assessment. In the assessment step, the resulting behavior of the simulation is compared with the identified needs and requirements of the analysis step, and the resulting degree of compliance is obtained.

Assessment techniques

In the following table, assessment techniques are summarized.

Example: Persona-based evaluation

Description: In the analysis phase, creating a number of relevant personas is important to gain insight in the user of a system, building or environment. When a design is completed or in prototyping stage, then it is time to check whether the design actually meets the demands of the personas that are involved with the system.

Because the persona is a fictitious character, interviews or similar tests with people are not possible. Therefore the persona profile forms the basis for checking the performance of the design. Two basic strategies can be used to make this assessment: (a) profile checklist, and (b) role-playing. In the profile checklist approach the listed desires, demands, and habits of the persona are taken and these are compared with the system performance. In the role-playing approach, the architect imagines him- or herself to be the persona and then interacts with this background with the design. Each of these approaches has advantages and disadvantages. The profile checklist is fairly objective but depends on the comprehensiveness of the checklist, which may not be a feasible goal. Also, a checklist can be limiting in the sense that aspects outside the checklist are not considered. The role-playing approach allows for a wider interpretation of the persona (and thus a wider checking for various people), but it is much more ambiguous and dependent on the imagination of the architect. Also, role-playing is more biased because it is done by the architect who will have difficulty to reach enough distance from his or her own preconceptions.

Definition: In persona-based evaluation the desires, demands, and habits of the persona are compared with the relevant features of the design. The assessment aims to identify to which degree the design meets the requirements of the persona. This can be done either through a checklist or by role-playing.

How to do it: In the assessment by means of checklist, first write down the desires, demands, and habits of the persona. For each aspect, list relevant features of the design that can be compared. Try to assess for each aspect the degree of compliance of the design. In the role-playing approach, carefully study the persona profile. Try to imagine that you are that particular persona, and that you are engaged with the interactive system, building or environment. Ask yourself whether the system meets your expectations and

Table 6
Assessment techniques

Assessment techniques (D)	
D1. Requirements comparison	Comparison of building performance with requirements
D2. User-client assessment	Determine whether there is convergence in the assessment for the user and assessment for the client.
D3. Persona-based evaluation	Determine to which degree the performance meets the requirements of the relevant personas.

needs, and whether it supports your routines and habits. Note possible deviations and try to understand the cause of these deviations.

To avoid bias by the architect, have at least two people perform the assessment and compare the results afterwards. In this way at least some intersubjectivity will be achieved. Go stepwise through the analysis and have each person explain his or her reasons for the results. This will identify different viewpoints that may be important to the understanding of the assessment.

What to look out for: The persona allows the architect to step out of his or her preconceptions of the design and to have a different look at the design. Nevertheless, there is always a risk of bias and post-rationalisation in this kind of assessment. Use at least two reviewers to make explicit the judgments in the assessment.

Methodological Framework

As stated above, we identify four critical steps that influence the design of interactive architecture: A. Analysis, B. Concept generation, C. Simulation, and D. Assessment. For each of these steps, we have identified design methods that can assist the architect. Obviously, in any concrete design project, it is not necessary that all methods should be used. In fact, the architect has to avoid "death by method," meaning that he or she should not become completely absorbed by the method, only to forget that the goal of using a method is to create a design. In all cases the method is the path or tool, but not the goal of the design work.

Therefore, at the start and during the design process, it is necessary to keep on asking which methods would be most beneficial to the design process. If the architect has the feeling that the most relevant issues have been solved, or that there is a strong and well-founded sense of direction in the project, then there is no reason anymore to stick to a method. However, when the process still seems open, undirected, and unclear, then it helps to step back a while and think about which method could be helpful.

Conclusion

The methodological framework brings together numerous techniques that are relevant to design for interactive structures in architecture. Many techniques already exist in architecture today, but they have not yet been brought together in a comprehensive framework. The framework presented here is a first step towards this goal. With the notion of user experience brief and the persona, a rigorous description of goals and users may guide the architect to – hopefully – better interactive designs.

References

Beesley, P, Hirosue, S, Ruxton, J, Trankle, M and Turner, C (eds.) 2006, *Responsive Architectures: Subtle Technologies*, Riverside Architectural Press, Toronto.

Berkel, B van and Bos, C 2006, *UN Studio Design Models: Architecture, Urbanism, Infrastructure*, Thames & Hudson, London.

Dahlbäck, N, Jönsson, A, and Ahrenberg, L 1993, Wizard of Oz Studies - Why and How, Natural Language Processing Laboratory, Linköping.

Fox, M and Kemp, M 2009, *Interactive Architecture*, Princeton Architectural Press, New York.

Habraken, NJ, Boekholt, JTh, Dinjens, PJM, Wiewel, W, and Gibbons, S 1986, *Variations: The Systematic Design of Supports*, Laboratory of Architecture and Planning at MIT, Cambridge, Massachusetts.

Kelley, JF 1983, Natural Language and Computers: Six Empirical Steps for Writing an Easy-To-Use Computer Application, unpublished doctoral dissertation, The Johns Hopkins University (Can be obtained from University Microfilms International; 300 North Zeeb Road; Ann Arbor, Michigan 48106).

Koolhaas, R (ed.) 2004, *Content*, Taschen, Köln.

Leupen, B, Heijne, R and Zwol, Jv 2005, *Time-Based Architecture*, 010 Publishers, Rotterdam.

Lynn, G and Rappolt, M 2008, *Greg Lynn Form*, Rizzoli, New York.

Norman, D 1988, *The Design of Everyday Things*, Basic Books, New York.
Roozenburg, NFM and Eekels, J 1996, *Product Design: Fundamentals and Methods*, John Wiley & Sons, Chichester.
Schön, DA 1983, *The Reflective Practitioner: How Professionals Think in Action*, Temple Smith, London.
Schumacher, M, Schaeffer, O and Vogt, MM 2010, *Move: Architecture in Motion – Dynamic Components and Elements*, Birkhäuser, Basel.