

1 **PLACING USER NEEDS AT THE CENTRE OF BUILDING PERFORMANCE**
2 **SIMULATION: TRANSFERING KNOWLEDGE FROM HUMAN COMPUTER**
3 **INTERACTION**

4
5 Simon Tucker¹, Clarice Bleil de Souza²,

6 ¹School of Art and Design, Liverpool John Moores University

7 s.s.tucker@ljmu.ac.uk

8 ²Welsh School of Architecture, Cardiff University, Cardiff, Wales - UK

9 bleildesouzac@cardiff.ac.uk
10

ABSTRACT

This paper reviews and explores some principles and theories of Human-Computer Interaction (HCI) and the related field of Interaction Design in relation to Building Performance Simulation (BPS). HCI seeks to make computer systems and software more useable and more attractive to its users. The main focus of the paper is on the interaction between user and computer system and how interaction could facilitate the knowledge transfer of BPS procedures and processes from experts to non-experts.

The paper discusses users and their tasks, designing for interaction, and the level of control different users might have over BPS. Design patterns are proposed as a means of interaction between user and computer system. The aim of the paper is to provide a platform for a future discussion on the extent to which BPS has engaged with HCI practices and principles, and the possibilities HCI holds for the further development of BPS. A number of research directions are identified.

INTRODUCTION

The aim of this paper is to review and explore selected aspects of HCI in relation to BPS particularly in relation to knowledge transfer of BPS procedures and processes from experts to non-experts through use of design patterns. Enabling knowledge transfer of BPS through greater use of HCI techniques could increase the acceptance of BPS and lead to greater use of it in designing low energy and low carbon buildings, whereas at present its use is limited (Hensen and Lamberts, 2011). A premise of the paper is that simulation processes and outputs could be seen as 'products' and be perceived by the user as being useful in helping to complete tasks and achieve goals. Design of such digital products is a concern of HCI, Interaction Design (ID) and User Experience Design.

The aspects of HCI considered here focus on interaction set in the context of user-centered design. Interaction models and theories are discussed because they might be useful in understanding and addressing the *lack* of interaction that many building design professionals have with BPS. The 'user-centered' product design methods used in HCI are referred to because they illustrate some practical aspects of a user-centered focus towards software development. By exploring these aspects, the authors wish to

generate a discussion as to the degree to which BPS software is (or could be) 'user-centered' and what this means. This is worth examining particularly as BPS continues to be integrated into CAD and BIM systems and can potentially be used to design buildings with higher levels of environmental performance.

We do not provide a survey here of how BPS has made use of HCI in the past. Many initiatives have been made in BPS as to how it can be used, how results are generated, and the development of interfaces of various sorts (see Bleil De Souza (2009) for a review of how BPS has been integrated throughout the design process). Neither do we consider other aspects of HCI more often addressed in BPS related research such as information visualization. We want to focus on user interaction with BPS, and discuss HCI interaction models which could potentially substantiate it.

The authors have previously proposed a theoretical system of ordering and formulating BPS procedures and outputs to aid design decision making using BPS which they aim to develop further (Bleil de Souza and Tucker, 2014: 2015: Tucker and Bleil de Souza, 2015). Design patterns are used in this system to 'package' simulation procedures, model settings and outputs such that the user can select them to meet design goals. They enable the user to interact with BPS in a different way than is currently possible.

The paper gives a short introduction to HCI and ID explaining what these disciplines are and what they aim for. It describes the importance of designing for the users and their tasks, i.e. adopting a user-centered approach in software/interface design. Knowledge transfer between HCI and BPS is initially discussed at the level of the BPS user and the tasks for BPS, and then further developed by discussing specific aspects related to BPS user interface and interaction: ideal models of the user interface, simplicity and complexity of interfaces, interface design models and interface automation versus control. The core of the discussion of knowledge transfer between these two disciplines lies in seeing BPS interfaces as learning systems. A case study of a potential BPS learning system is presented by summarizing the work of Tucker and Bleil de Souza (2015) which explores the concept of design patterns in design decision-making.

This paper is not intended for HCI experts as we draw on selected material from only a small number of HCI

and ID textbooks. Its intention is to spark a discussion on widening the use of BPS and on theoretical aspects of BPS and its users.

HUMAN-COMPUTER INTERACTION AND INTERACTION DESIGN

The discipline of HCI examines the relationships between humans and the computer systems they use to perform various tasks (Faulkner, 1998). HCI is an interdisciplinary design science founded originally on application of experimental psychology methods and cognitive science to the systems and tools of computer science (Sutcliffe, 1995; Shneiderman and Plaisant 2010), subsequently growing into an extensive field that can be related to computer science, artificial intelligence (AI), anthropology, ergonomics, linguistics, philosophy, art, sociology, design, psychology, engineering and physiology (Faulkner, 1998). It can draw from these various disciplines at any stage of the HCI design process from conceptual design of the overall system, to using research methods from those fields. As a 'design science' HCI is supported and guided by well understood procedures, protocols, and a body of theory, although design of HCI systems is also open to intuition and 'what feels right' (Dix et al., 2004). There is no general and unified theory of HCI but the underlying principle is that people use computers to accomplish work, meaning that the main areas of HCI concern people, computers and tasks (Dix et al., 2004).

HCI is a dynamic field, ever changing as systems and technologies are developed. Current HCI topics include interface and interaction device design, data visualisation, interactive systems, development of models, paradigms, theories, and principles, social aspects of computers and computing, user modelling, and design and evaluation. HCI has many applications including medicine, education and training, computer games, robotics, science, engineering, arts and humanities. Because the use of the computer is ubiquitous, so is the reach of HCI.

Whereas traditional HCI tended to focus on experimental psychology to study human-machine interrelations the scope of interest has since widened to embrace the diversity of design, focussing on understanding what people need technology to do for them, and how to design the technology to meet these desires and requirements. There is now more emphasis on the design of products that support interaction and on the nature of interaction itself. This shift of interest has given rise to the fields of Interaction Design (ID) and User-Experience Design.

Introductions to HCI are provided by several authors including Faulkner (1998) and Dix et al. (2004). An introduction to Interface design is given by Shneiderman and Plaisant (2010), and to ID by Rogers et al. (2015) and Cooper et al. (2007).

Rogers et al. (2015) describe ID as '*designing interactive products to support the way people*

communicate and interact in their everyday and working lives'. Such products (designed with the user in mind) contrast with engineered systems that will deliver specified functions effectively but could be difficult for 'real people' to use. Crampton-Smith (2007) emphasizes further the interactive focus stating that ID is about '*shaping our everyday life through digital artifacts – for work, for play, and for entertainment*'. ID seeks to integrate such artifacts as fully as possible into daily work and leisure activities. Cooper et al. (2007) describe ID as 'the practice of designing interactive digital products, environments, systems and services.' Therefore ID can be seen as being interested not only with the products that support users but also in environments and systems, a cue perhaps for thinking of BPS as part of a system that includes the BPS user and the buildings designed using BPS.

DESIGNING FOR USERS AND THEIR TASKS

"Know the user" (Hansen, 1971) is a key principle which according to Shneiderman and Plaisant (2010) is often ignored or undervalued, and which the current authors believe has been insufficiently addressed by the BPS community. While a wide range of BPS software and interfaces have been produced (see DOE website list of BPS software) these are typically based on the software developer's idea or assumption of who the user is. Similarly, they are often based on the developer's idea of what the task is rather than what users themselves see as their tasks. It is difficult at present to gauge the level of use and identify the users of existing BPS software (whether 'standalone' or whether accessed through CAD and BIM software). This difficulty in itself points to a lack of focus in the BPS community on the user.

A user-centered approach

A 'user-centered approach' will require the HCI designer to let users and their goals drive product development (Rogers et al., 2015). Three principles of Lewis and Gould (in Rogers et al., 2015) illustrate the design and evaluation process typically adopted in HCI:

1. An early focus on users and their tasks through studying them.
2. Empirical measurement of user's reactions to proposed scenarios, prototypes, simulations (i.e. assessment of how the product is used).
3. Iterative design in response to step 2 (i.e. design – test – measure – redesign as required).

The processes of HCI design and ID are extensively described in the literature. In outline, the following activities take place although these might not always be carried out sequentially in the order shown:

- Establishing requirements for user experiences.

- Designing alternatives to meet those requirements.
- Prototyping alternatives for communication and assessment.
- Evaluating at all stages for user experiences.

From these aforementioned activities one can see that users and the tasks they undertake are interrelated. This is why some researchers prefer to approach users and tasks together, seeing both as equally important (e.g. Johnson 1992; Dix et al., 2004). This distinction highlights how HCI practice and theory not only seeks to understand the users of a system but also to analyse and understand the tasks that users might undertake using the system. This approach is seen as the starting point for the BPS community to provide tools which better respond to the different stakeholders involved in the design of low energy buildings.

TRANSFERRING KNOWLEDGE: FROM HCI TO BPS

HCI is concerned with understanding and optimising the relationship between two complex systems: the user and the computer system. It seeks to understand the user, the tasks performed by the user, and how a computer system might be structured such that the user can carry out those tasks efficiently and easily (Faulkner, 1998). The user's knowledge of the task should be enough to complete the task without having to gain knowledge of the computer systems (ibid). An ideal HCI system would allow the user to carry out their tasks without even noticing the computer system.

This is almost the antithesis of what currently happens in BPS: the user is expected to have knowledge and fundamentals of the three aspects highlighted in Table 1 to be able to confidently operate the tools with the ultimate aim of designing low energy buildings. However, the fact that the knowledge and fundamentals listed in Table 1 are rather complex do not mean one should not aim for ideal HCI systems in this case. It means simply that more imagination is necessary to define what/how this 'ideal' system should be and how it could cater for the different types of BPS users.

Table 1 – Knowledge and fundamentals currently required to use BPS to design low energy buildings

Knowledge	Fundamentals (examples)
How to operate BPS tools	Modelling, analytical methods, generation and interpretation of data
How low energy buildings work	Building physics, building services and energy systems, influence of occupants, passive design principles
Using BPS to design low energy buildings	Meeting targets, comparing design options, identifying causal relationships, integrating RE systems, minimising heat losses

The BPS user

All HCI and ID researchers place considerable emphasis on understanding the user and there are two main aspects to this: (i) who is the user, what does s/he do, what are her goals? (ii) what knowledge, skills and capabilities does this user have (for example, in relation to using computers). Research on actual and potential users is usually carried out to answer these questions. This paper proposes two different user groups who may be expected to have a wide range of skills and abilities in terms of the fundamentals behind running BPS software: (i) building designers with various competencies in relation to building physics etc. and (ii) an educational audience of building designers under training.

There is considerable advice in the literature on designing software for different skill levels, with the following categories often proposed;

- Experts, who are usually frequent users with high levels of domain knowledge.
- Intermediates, who may be intermittent but knowledgeable users.
- Beginners, who are typically novices and/or first time users.
- Others who do not use the software but can be affected by its use (e.g. managers).

Each of these categories could be applied to the knowledge domains of Table 1 to form a matrix which can be used to guide the construction of more detailed understanding of the user's needs. Through considering these two groups and their different skills levels we believe that the appropriate systems and interfaces can be developed that will cater for a wider range of users than currently exists. Guidelines are provided throughout the HCI literature on providing for the various users' skills levels (e.g. menus, toolbars, help facilities, and use of restricted or constrained interfaces, etc.).

In addition, software developers should consider the underlying reasons and motivations that influence the user's approach to completing tasks, which could include satisfaction at designing well, seeking personal or organisational success, fear of failing to achieve design targets and so on. HCI and in particular ID have developed a range of methods to understand and address these more nebulous drivers including development of 'personas', fictional characters based on archetypal users (Cooper et al., 2007), and construction of narrative driven scenarios, which might involve the developed personas (Rogers et al., 2015). The aim of these methods is to guide product design and ensure attention remains focussed on the user (see accompanying paper for further discussion on this topic).

Tasks for BPS

Task analysis involves identifying the tasks the user needs to complete and the subtasks within them, in the

context of a particular domain and in relation to user goals. Dix et al. (2004) refers to all these together as forming the 'problem space', and task analysis involves identification of this space

The user's *goals* may be seen as higher-level aims connected to their motives, such as the motive of wanting to design an environmentally responsive building resulting in the goal of designing a 'zero energy' building. In this study, *tasks* are ways to achieve those goals, for example using BPS to achieve zero fossil-fuel energy use. However, the design of buildings takes place in many different contexts which often change over time, for example as designers gain experience, different personnel join the design team, or the building designer gets a new desk next to a building services engineer. These contexts can affect how software is used because they can influence how the tasks are perceived by the user and carried out. HCI has become very aware of contexts, and its methods seek to take into account the user's physical and social environments and how these influence interactions with technology (Shneiderman and Plaisant, 2010).

It follows from this that successful task analysis will usually depend on extended observation and interviews with users (ibid). Some potential users will not currently be making use of BPS and in this case the study would be of their overall motives and goals in design and the processes followed, such that uses or tasks for BPS within the design process can be proposed. Similar studies would be made of designers who were already using BPS, with the additional analysis of how and when they used it. In order to understand how BPS is used to support low energy design, expert BPS users such as consultants would also be studied.

A series of interviews with building designers identified a large number of problems that designers had with obtaining and using building performance information (Tucker and Bleil de Souza. 2015). For example one practice often used in their designs a shading system incorporating a light shelf that they had developed, and just needed to modify it for each new orientation and window size. Obtaining data on the performance benefits and costs for each design was cumbersome and slow, and what they needed was a system where the CAD model could be quickly analysed by the building designer to produce this information. In this case simply asking the building designer what problems he had as regards to performance analysis resulted in a very clear idea of possible solutions.

BPS USER INTERFACE AND INTERACTION

The user interface is where the user and the system interact and is a key concern of HCI and ID. Some researchers argue for development of 'universal design' systems that caters for a wide diversity of users

(Dix et al., 2004). Current BPS software tends to follow this approach and all users access the same interface (Sefaira is an exception with two modes of operation). Others argue for a range of interfaces to cater for different users. Whichever of these approaches is taken it is important that the interface is designed in relation to the users and their goals and tasks, which are established before the design takes place.

As BPS systems are not generally used by building designers one could ask what sort of interface and interaction *is* desirable. An interactive system should help the user to achieve her goals, identified as part of the task analysis. The interaction can be seen as being between the language of the computer system and the language of the user (or task language) (Dix et al., 2004). If the interface is too oriented toward the computer language then the user has to learn the computer language or parts of it, which may distract from the actual tasks and interfere with the users design process.

Ideal models of the user interface

The users relation to the interface and hence the system behind it is critical. A theoretical aspect of interaction is the user's 'mental model' or conceptual model of the interface. This is what allows the user to predict what will happen when actions are taken. The actual operation of the system can be described by a 'system model' or implementation model. Interface designers can also provide a 'designers model' or represented model, which does not necessarily have to reflect how the system works but is the designer's explanation to the user of how it can be taken as working. The developers of BPS have tended to be physicists, engineers, and occasionally architects with a particular interest in this area. According to Cooper et al. (2007) engineer designed interfaces tend to afford represented models that are closely related to implementation models, meaning that users must adjust to the system operation.

What represented model(s) might work for a range of BPS users? The answer would seem to partly depend on how much the user is expected to learn of the various domains of Table 1 through using the interface and underlying system. Having an idea of what is supposed to be learnt through use of the interface can guide the formulation of the represented model. The questions as to *how* someone learns is certainly a topic in HCI but is beyond the scope of this paper. However there is an established literature that addresses how professionals working in different domains learn (e.g. Schon, 1983) which would most likely influence this aspect of interface design.

Research into the needs of the user should therefore keep in mind this possibility of learning through use of the interface, and should seek to identify just what could or should be learnt by various users. It is important to appreciate these possibilities offered by interface systems at the design stage to avoid falling

into the trap of designing an interface to suit the developer rather than the user.

Simplicity and complexity in the interface

Catering for a range of user skill levels would tend to require ways of simplifying the potential complexity of simulation and have the interface represent that simplicity. In addition the full complexity and flexibility of simulation should be retained for its current advantages. Current BPS interfaces cover a fairly wide spectrum in this regard although each one tends to offer one general interface that all users must adapt to, with relatively minor adjustments possible. A key question can be formulated: *how can the complexity of BPS be simplified without losing its useful qualities?*

The authors work to date indicates that a promising direction is in the identification of specific design scenarios and situations where BPS can be used. For example, the user might have a goal of eliminating overheating in a passively operating building. The interface could offer the user the choice (amongst others) of varying thermal mass levels while introducing night cooling, or only using night cooling. Design patterns (see case study below) for each of these options could be made available. Here the potential and complexity of the full BPS system has been reduced to a number of options, in a similar way to a constrained interface. Information can be communicated to the user that explains what each pattern offers.

Interface Design models

How a user might operate an interactive interface is also described by HCI models that show how tasks may be broken down and structured to help the user solve problems. Required objects and processes can be decomposed into smaller objects and into coherent steps and actions. User tasks can be described by series of actions. The GOMS model (goals – operators – methods – selection rules) decomposes goals into a number of actions and then into methods (Card et al., 1983). However, different software users use different methods and processes to solve problems and achieve goals, and the interface designer might not be able to anticipate all of these. The GOMS approach is well suited to expert users (Shneiderman and Plaisant, 2010) as its success is dependent on the knowledge of the user, in that it is assumed that the user learns how the interface designer has understood the problem and can use the provided steps and sub-steps to achieve goals. Forming goals, intentions and specifying actions and methods also requires prior knowledge in the different domains of Table 1.

Norman's (1988) 'cycles of action and evaluation' (form goal – form intention – specify action – execute action – perceive system state – interpret system state – evaluate outcome) offers a more dynamic model of how a user interacts with an interactive product. This model describes how the user needs to know how an

intention can be translated into an action on the system (the 'gulf of execution'), and the interface should inform/advise the user about how to do this. The model also describes how the user needs to be able to evaluate what has happened when the system responds to an action (the 'gulf of evaluation'), and the interface should make clear the new state of the system (Sutcliffe, 1995). So for example, when the user instructs the system to run a simulation, the system informs the user on the progress of the simulation.

This simple and influential model is routinely used to guide interface design but it could arguably be used more expansively in BPS, for example to indicate to the user what s/he could do to achieve a goal or even suggest what the goal might be, based on previous simulation results or actions of the user. For example, if results of a simulation showed that significant levels of overheating were occurring AND solar gains are contributing to this, the system could suggest that tests for the effects of shading or increased ventilation are carried out. A BPS interface could actively help the user to evaluate what has happened following a simulation. Again, prior degrees of knowledge are required for forming goals etc. and so if the interface can do some of these things for the user then it could be seen as potentially transferring BPS knowledge.

Models of this type are widely applied, for example in analysing how users seek for information or order goods online (Shneiderman and Plaisant, 2010). This might suggest that BPS software offers more help and direction in enabling users to seek information or search for knowledge related to their current task or goal, activities that could widen the scope of BPS and allow it to become more integrated into design processes (for example, linking to appropriate benchmarks, precedents, case-studies, etc.).

Interaction: Automation versus control

The final aspect we touch on is the degree to which the system is automated and how much control the user has of the system. Automation could afford simplification of BPS complexity.

Unless every possible event can be predicted and the system designed to deal with each event then human judgement and supervisory control in a system is needed (Shneiderman and Plaisant, 2010). However, there is the possibility of assisting the user through intelligent agents. Intelligent agents carry out operations on the system on behalf of the user, based on the agent's knowledge of the user's goals. Autonomous agents do the same, but independently of the user to a certain degree. There is an ongoing debate in HCI over whether to develop better interfaces that offer the user full control or to focus on agent type interfaces. The latter might have anthropomorphic qualities, or use avatars that represent the system and/or the user. An implication could be that the user need not understand the underlying system functions or structure, as the agent carries out interpretation between machine and user. An intelligent agent for

example could assess the results of simulations and recommend strategies for improving performance. Agent driven selections (e.g. of suitable strategies, of appropriate help information) might make use of recommender systems (Ricci et al., 2011). These are currently used on commercial websites (e.g. Amazon) and recommend items of potential interest to a customer based on previous purchases and ratings. In the BPS environment, a recommender system might act on information held on the user and her goals as well as on building performance. For example the system could keep track of building performance and external environment, the stage of the design process (e.g. outline or detailed design), the goals of the user (e.g. meeting a performance target), building type and so on, and use this information to infer and recommend appropriate strategies and design changes.

An alternative to the AI approach is full user control of consistent and predictable interfaces, which can offer the user a feeling of accomplishment in mastering the system. An implication is that the user would (eventually) understand the underlying system, and the how's and whys of manipulating it through the interface. Traditionally a high level of user control is provided by BPS systems because users tend to be experts. There have been attempts to provide for less expert users through provision of CAD type front ends (IES, DesignBuilder, OpenStudio) and more recently through simplifying the whole process of running simulations and obtaining performance results (e.g. Sefaira).

Shneiderman and Plaisant (2010) believe that successful interfaces of the future will be based on 'user-centered scenarios' rather than on machine oriented AI, agents, expert systems etc. This implies that the user will be actively learning in some way. A system that includes both approaches (automation *and* user control) could be one where the user carries out a sequence of actions that are recorded (i.e. the system is instructed or controlled) and which can then become generic procedures perhaps called automatically under specified conditions. These procedures are also then available for less expert users. An early example of this approach are OpenStudio 'measures' that allow automated tests on building models to be created and reused as required, although these need to be manually called. While such technology for automation is being introduced to BPS, the community does not really understand how a range of users can use these technologies consistently and productively.

BPS INTERFACE AS A LEARNING SYSTEM

The considerations above not only highlight what the users might want from BPS and how it could fit within the tasks they undertake, but also question what the user needs to learn. HCI informs the development of educational systems such as intelligent tutors and collaborative learning systems. These and others such as expert systems are used in several science and

engineering fields such as biology, chemistry, medicine and mathematics. HCI also informs the field of 'serious games' and the use of gaming techniques as part of a learning strategy (e.g. use of scenarios, narratives, 'scaffolded learning' or increasing levels of difficulty). Each of these are examples of interactive systems or strategies and could be considered as means of delivering the learning topics of Table 1. However, here we look more closely at design patterns as a means of enabling interaction between user and system. We argue that design patterns are uniquely suited to act as the device by which user language can interact with the system language in the BPS context.

Case study: Patterns for design decision making

In previous work (Tucker and Bleil de Souza, 2015) the authors explored the delivery of knowledge highlighted in Table 1 to building designers, by developing a learning structure which matches a design structure: that of design patterns. The authors refer to this learning structure in this paper in the context of transferring knowledge from HCI to BPS, believing the concept of design patterns can be seen as an example of how building designers can potentially interact with BPS.

Design patterns are a means of representing knowledge in a particular field through description of generic and abstract problems encountered in that field together with proven successful solutions to those problems. Solutions are modified for each instance of a problem depending on specific context and details of the problem. Patterns usually consist of a description of the problem and the solution, illustrated with examples and supporting information. They were introduced in the field of architectural design by Alexander and colleagues (Alexander et al., 1977; Alexander, 1979) and have also been used in the fields of software engineering (e.g. Gamma et al., 1995), education (e.g. Laurillard, 2012), and in other fields.

Design patterns have long been recognised as enabling users to better understand complex systems (Gamma et al., 1995). They are used to teach complex concepts such as the use of Information Technology in learning, how to construct tools that enable users to deal with the complexity of the internet, and indeed how to apply the knowledge built up in HCI and ID when designing interfaces (Tidwell, 2011). In software engineering they are used for teaching expert knowledge (e.g. on object oriented programming) and also encourage re-use of code that has been shown to solve successfully a recurring problem. 'Patterns can be seen as helping the user to make sense of complex and changing systems, and would seem to be useful where procedures and functions that solve regularly occurring problems can be modularized for re-use perhaps with small modifications.' (Tucker and Bleil de Souza 2015). Therefore patterns address user needs for learning as well as holding expert knowledge for transmission and reuse. The dialectic nature of design patterns (problem – solution) is essentially easy to

understand and their proven use in other fields have shown their suitability as a learning tool.

The design patterns proposed in Tucker and Bleil de Souza 2015 “focus on connecting design aims with simulation outputs that are tailored to respond to these aims’. Satisfying these aims will in general imply the construction of a model(s) and running of simulations, together with the structuring of analysis processes that when applied to these models will allow meaningful outputs to be retrieved” (Tucker and Bleil de Souza 2015). Patterns have some similarities with 'performance assessment methods' (Clarke et al., 1996) and 'analysis functions' (Augenbroe et al., 2004) although are structured explicitly to transfer simulation knowledge from experts to non-experts. Therefore they potentially support knowledge transfer and learning through interaction between user and system. To use the terminology of HCI, patterns could act as the device by which user language can interact with the system language (Table 2).

Table 2 - Design pattern template: Information presented to the user (from Tucker and De Souza, 2015)

Pattern name:	Name should clearly reflect the abstract problem and solution, and can refer to building typology, specific design actions, goals addressed, analysis processes, and outputs.
Introduction:	Situates the pattern in context to larger patterns.
Problem:	A brief outline of the problem addressed by the pattern.
Context and examples:	Situates the use of the pattern in relation to simulation and design practice and provides examples of these. Research, pedagogy and/or practice is cited that justify the advice given by the pattern.
Modelling Details	Instructs the user on what must be modelled and what is provided automatically
Interpretation and Quality Assurance	Instructs the user on how to interpret results, what to expect from results and why, and which QA patterns to use.
Further patterns	Information on which smaller patterns to move on to, in light of the aims of the user and results given by the current pattern.

A pattern for example could instruct the user how to explore the effects of adding shading, as a solution to the problem of providing daylight and views while preventing overheating through solar gains. A pattern dealing with window sizing and glazing type might address the problem of admitting useful solar gains while minimising heat losses in the heating season. The pattern would give details of the modelling and simulations required, the results to be presented, and

the options for interacting with the results etc. Patterns are highly communicative and ‘talk’ users through complex procedures using a narrative style, using the language of the user.

Design patterns are well suited to a user-centered approach as they can be constructed to address specific user goals and specific types of user such as building designers, reflecting for example their levels of knowledge and learning needs. They also fit very well the HCI design process as their structure is intended to facilitate modification and refinement following user evaluation.

Patterns could also contain elements or attributes that together define a profile for the user. These elements could refer to building type, preference for passive or HVAC operation, climate, level of user expertise, etc. The element states could then influence which patterns are presented for use in solving the problems, and/or how patterns are modified to suit the user (e.g. more help provided, or less user input required). Patterns could also be constructed at a number of levels corresponding loosely to stages of design processes (Table 3). These levels would provide an additional means of selecting appropriate patterns for presentation to the user.

Table 3 – Pattern levels (modified from Tucker and De Souza, 2015)

Level	Type / purpose
High-level, planning related	Site analysis, guidance on climatic strategies, passive and low energy strategies, renewable-energy systems potential
Mid-level, building related	Exploring building form, glazing ratios, insulation of building elements, Renewable energy systems integration, site specific ‘rules-of-thumb’
Low-level, detailed modelling	Effect on performance of building parameters, plant efficiencies, effect of occupants

In summary, design patterns could guide the formulation of an interface between BPS and the user. They can be constructed for different users and different tasks, communicating the knowledge they contain and stimulating interactions between user and computer system. This combination of communication and interaction is at the heart of what a learning interface based on patterns can bring to BPS, allowing the user to ‘make sense of complex systems’.

CONCLUSIONS

The interaction of user and system has been highlighted through exploration of a user-centered approach to BPS with reference to the principles and methods of HCI and ID. A question has emerged: What should or could the user learn about low energy buildings, BPS techniques and methods, and how to use BPS in low energy design?

Within a user-centred approach, this question suggests that one area of BPS that seems underdeveloped is that of 'learning systems', in which the interface could potentially enable users to learn how and when to undertake simulations. Through focusing on learning systems the work explored knowledge transfer from HCI to BPS by discussing interface models, how simple or complex an interface should be, and the desired balance of user control versus intelligent agent control. A case study illustrating a learning system using the format of design patterns is presented as a hypothesis of a suitable BPS interface for building designers.

Patterns for design decision making are a learning structure, which matches a design structure. They are therefore an example of knowledge transfer from experts to non-experts in a format compliant with the non-expert way of thinking. This structure could only be unfolded and expressed once a deep understanding of designer's goals and tasks was achieved (see Tucker and Bleil de Souza 2015 for details).

Although this discussion has been somewhat speculative, it does illustrate how consideration of HCI techniques might provoke the development of user-centered BPS systems. Moreover, the case study shows that by starting with the user and understanding what s/he needs and wants there is the opportunity to find different models of BPS interface, and new uses for BPS.

There is a lack of knowledge in BPS on user profiling, and on understanding user experience, user goals and associated task analysis. When more is known about the user, opportunities will arise for the construction of methods and systems to enable their tasks to be supported by BPS systems. New types of interactive user interface may be required, possibly employing intelligent and adaptive agents that drive multi-user and collaborative systems.

REFERENCES

- Alexander, C. 1979. *The Timeless Way of Building*, New York: Oxford University Press, 1979.
- Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S. 1977. *A Pattern Language*, New York: Oxford University Press.
- Augenbroe, G., de Wilde, P., Moon, Hyeun Jun., Malkawi, A. 2004. An interoperability workbench for design analysis integration, *Energy and Buildings*, 36, pp.737-748.
- Bleil De Souza, C. 2009. A critical and theoretical analysis of current proposals for integrating building thermal simulation tools into the building design process, *Journal of Building Performance Simulation*.
- Bleil de Souza, C., Tucker, S. 2014. Thermal Simulation Software Outputs: A Framework to produce Meaningful Information for Design Decision Making. *Journal of Building Performance Simulation* 8 (2): 57-78.
- Bleil De Souza, C. Tucker, S. 2015. Thermal simulation outputs: a conceptual data-model to present meaningful information for design decision making, *Journal of Building Performance Simulation* 9 (3): 227-254.
- Card, S.K., Moran, T.P., Newell, A. 1983. *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, Hillsdale, NJ.
- Cooper, A., Reimann, R., Cronin, D. 2007. *About Face 3: The Essentials of Interaction Design*, John Wiley & Sons.
- Dix, A., Finlay, J., Abowd, G.D., Beale, R. 2004. *Human-Computer Interaction*, Pearson, Harlow.
- Faulkner, C. 1998. *The Essence of Human-Computer Interaction*, Prentice Hall, London.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, MA.
- Hensen, J.L.M., Lamberts, R. 2011. Introduction to building performance simulation, In *Building Performance Simulation for Design and Operation*, eds. Hensen and Lamberts, Spon Press.
- Johnson, P. 1992. *Human-Computer Interaction*, McGraw-Hill, London.
- Laurillard, D. 2012. *Teaching as a Design Science: Building Pedagogical Patterns for Learning and Technology*, Routledge.
- Norman, D.A. 2002. *The design of everyday things*. New York: First Basic Paperback. (Orig. pub. 1988 as *Psychology of everyday things*. New York: Basic Books.)
- Ricci, F., Rokach, L, Shapira, B. 2011. Introduction to Recommender Systems Handbook, *Recommender Systems Handbook*, Springer, pp.1-35.
- Rogers, Y., Sharp, H., Preece, J. 2015. *Interaction Design: Beyond Human-Computer Interaction*, 4th Edition, John Wiley & Sons, Chichester.
- Shneiderman, B., Plaisant, C. 2010. *Designing the User Interface*, 4th ed. Pearson, Boston.
- Sutcliffe, A.G. 1995. *Human-Computer Interface Design*, Macmillan Press Ltd, London
- Tidwell, J. 2011. *Designing Interfaces*, O'Reilly, Canada.
- Tucker, S., Bleil De Souza, C. 2015. Thermal Simulation Outputs: Exploring the Concept of Patterns in Design Decision Making, *Journal of Building Performance Simulation*, 9 (1): 30-49.