Conference papers   School of Electrical and Electronic Engineering

2018-09-07

Design patterns as a collaborative entity within the smart environment

Aitor Arribas Velasco
d16126348@mytudublin.ie

John McGrory
Technological University Dublin

Damon Berry
Technological University Dublin, damon.berry@tudublin.ie

SEEDS International Conference

Design patterns as a collaborative entity within the smart environment

Aitor Arribas Velasco[1], John McGrory[2], Damon Berry[3]

[1,2,3]Dublin Institute of Technology, School of Electrical and Electronic engineering, DIT, Kevin Street, Dublin 8, D08 NF82.

Abstract

Modern technology is increasingly being employed to create a "smart" living experience. These "smart" technology entities are producing copious of amounts data, which in turn rely on increased storage, distribution and computation needs to manage the data. Depending on the scenario, the diversity of piecemeal solutions almost mirrors the numbers of problems posed. Some successful solutions touted as being "smart" for example: save energy, or to support in assisted living, have been created, but the true underlying pattern of interactivity has not been identified.

In the field of computing, patterns can provide a general, reusable solution to commonly repeatable occurring problems within a given context through software design. Similarly, can a technology-independent design pattern format and an open software framework be developed to capture, share and redeploy existing successful and reusable strategies for commonly encountered smart environment use cases in areas such as assistive technology, energy management and environmental monitoring? The underpinning notion of this paper is to introduce "how, where and why" a rule set based in "design pattern" format could contribute to describe a general "understanding" of given cases in the smart environment domain, as well as allow different processes to collaborate with each other.

At this point, our project performs a preliminary research on how different communities use popular logic encoding paradigms to essentially represent the same idea; a set of conditional statements. With a view to determining a framework that could be used to define the interconnection among each process. This paper extends previous research by exploring different uses of patterns in the domain of software architecture and design. Ultimately, our study aims to link the principle of "rule of thumb" to the concept of design patterns, by making this accessible enough to allow successful "smart space" solutions to be shared widely between outstanding solution providers.

1. INTRODUCTION

A pattern is a regular and comprehensible "form" or sequence discernible in the way in which something happens, is prepared or is completed and contributes to guide or solve common reusable design issues (The free dictionary, 2017). However, patterns are not solely a human invention, patterns occur naturally in various contexts, such as, nature, art and architecture, computer science, process optimisation and so on, allowing humans to mathematically model/generalise processes. Patterns can also be used to gauge the past, present, and future: archaeologists use the layers of earth to date their findings. Many others have been able to translate them into maths, geometric shapes and building equations, which are used by computers to simulate a wide range of biological processes (Wikipedia, 2018). In software design, patterns can provide a general, reusable solution to a

commonly occurring problem within given context parameters. The *term design patterns* relates to the way in which a recurring design issue is identified, labelled, and coded in order to provide a general solution (Wikipedia, 2018).

This paper attempts to describe a preliminary design-pattern approach for smart processes.

2. Literature review

Design patterns emerged in different problem domains, and have being widely applied to a multitude of design techniques. However, computer design has become a major torchbearer for pattern research. In 1977, Christopher Alexander introduced the concept of a pattern language, composed out of 253 of these elements (Alexander, C.,1977). Alexander's work provides, not only to professionals but also to non-technical people, a tool with which to improve a town or neighbourhood, design a house or work with others to design different spaces. Pattern based design employs a catalogue of notions to be considered without needing to resort to mathematical or algebraic expression to describe the patterns, or their application. Nonetheless, his work has had a considerable impact on computer engineering. In this field, Alexander's contributions are applied to Object Oriented Programming. These theoretical structures enable the linking together of objects in programs in a co-operative and sequential way. The significance of this approach has motivated the appearance of conferences such as Pattern Languages of Programs (PLoP) (Conference on Pattern Languages of Programs, 2018).

At the time that patterns were being developed in computer science, a shift occurred in the field of human-computer interaction (HCI) research, from focusing on how people interacted with programs towards a communications oriented approach. This happened mainly because of the growth of the Internet and the web. As a result, the number of research fields grew under the umbrella of HCI (Dearden, A., 2006). By 2005, the research focused on collaboration, connection and communication (Lazar, J., 2010). This coincided with the period in which the Internet of Things concept emerged. The reason for this shift in emphasis, was the opportunity that Internet gave to technologists for communicating wirelessly. It made it possible to digitise data transfer by transitioning from analogue to digital formats. The next phase was digitalization, which focuses on business rules and systems that synthesize and manipulate digitised information. An example of this is the intelligence behind the ever changing and adapting Google search engine. It involves far more than simply accessing digitised data.

Based on the two terms mentioned above, *design patterns* and *HCI*, researchers have focused their efforts towards new approaches to improve the experience of the users within *smart spaces* (Vega-barbas, M., Pau, I., Augusto, J., 2017). In the same way, different studies have tried to define a mathematical model to describe context within smart spaces (Yang S., Kabir, M., Hoque, M., 2016).

The concept of *context* is needed when we discuss and evaluate meaning. Somewhat ironically, this term has often been used in different senses. In the field of logic, the philosopher C. S. Peirce introduced for the first time a representation of context as a formal object. From 1980, three main theories highlighted this concept of context: Kamp´s discourse representation theory; Barwise and Perry´s situation semantics; and Sowa´s conceptual graphs (Sowa, John F., 1995). In essence, popular identification techniques and frameworks to efficiently capture and manage context are funnelled into discipline specific

solutions. In the Artificial Intelligence literature, there is no single authoritative definition of the concept of context in the form of a universally accepted identification or framework, however, it is widely employed in different approaches. The difficulty is when two or more systems (or approaches) need to interact to provide a more comprehensive solution.

3. Methodology

This first phase of our work involved an initial investigation of popular diagram based pattern formats used by a variety of expert communities. The objective was to uncover commonality of identification and structure and possible links to algebra hidden within diagrams. Table 1 shows the layout used to describe the different logical elements.

Category	Symbol	Description
Name	Image	• Year created. • Attributes. • Purpose. • Intended use.

Table 1, Logic description properties.

This analysis is motivated by the necessity to identify common structures used in different fields, by different communities, applying and using different design-methods, symbols and descriptions to build logical sequences. All of them based on the same fundamental principle.

These diagram nomenclatures represent broadly the same idea, which is, the development of a common framework which could unify and simplify the use of different instructions under the same umbrella of a pattern format. By doing so, we aim to set the basis for a collaborative pattern-based model which will enable cooperation between different processes within the smart-living domain.

In addition, our search has led to an analysis of different processes aiming to identify the main elements of a generalised pattern. A whole process, overall, can be decomposed into smaller parts, in the same way that an image is made up of pixels. Each of these parts carries its own portion of meaning. The combination of these parts, among other things, add meaning, and context to the process. Context, in our approach, means we can consider collected data as information that can be reused and applied in different solutions. In Table 2 we list the different sub-elements, in which we have split each process.

| Process Name |
|:---:|
| **Triggering Events**
(Conditions that must be met at the beginning of the process) |
| **Terminating Events**
(These are the conditions that must be met at the end of the process) |
| **Inputs**
(The resources needed to execute the process) |
| **Outputs**
(The items created as a result of the process) |
| **Sequence**
(These describes the steps in the process along with the actors responsible for executing each step) |
| **Conditions** |
| **Metrics (Attribute)**
(What aspects of the process are measured) |
| **Sample Images** |

Table 2, Decomposition of a process.

Each of these elements describe an essential and individual part of the overall process. Each individual piece can have its own meaning, but as a collaborative whole, they produce a process, with contextualised meanings. Interaction between different elements is a critical aspect to capture.

This paper will present a preliminary approach in which the concept of design patterns can be applied to identify patterns within the daily routine of a stakeholder within the smart living experience. This concept is described as follows, by common iconography.

### 3.1 Patterns within activities

As mentioned above, our research is focused on smart environments. Here, a great variety of scenarios such us a home, a hospital or an art gallery may be contemplated. Furthermore, in each environment, every user interacts with it in a particular and unique way. Figure 1 highlights three different user routines within three different smart environment.
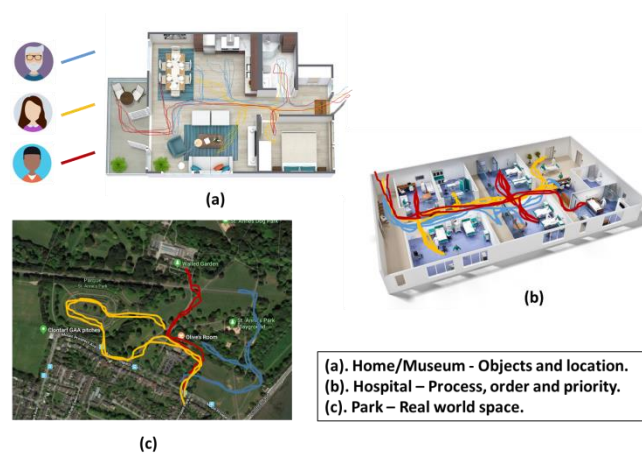
Figure 1. (a) Smart home routine. (b) Smart Hospital routine. (c) Real world routine.

In Figure 1-(a), which may correspond to a home or museum, we can find Objects, and Localisation. These general design elements can be used for example, to build a system to guide a user through an art gallery. While in Figure 1-(b), at a Hospital scenario, Objects may have different priorities within the whole process followed by the user, in which case a defined sequence must be followed in a particular order. Other case, Figure 1-(c), of interaction within the environment is the real world scenario, for instance a park, where each person may interact in a different way, following personalised routines and preferences.

Each of the paths drawn can be described as a process or sequence of transitions between different nodes or states. Each of these nodes may represent a position, which can be defined by its localisation, context, or metadata. Furthermore, each node can refers either to a zone Figure 2-(a) or an activity Figure 2-(b).
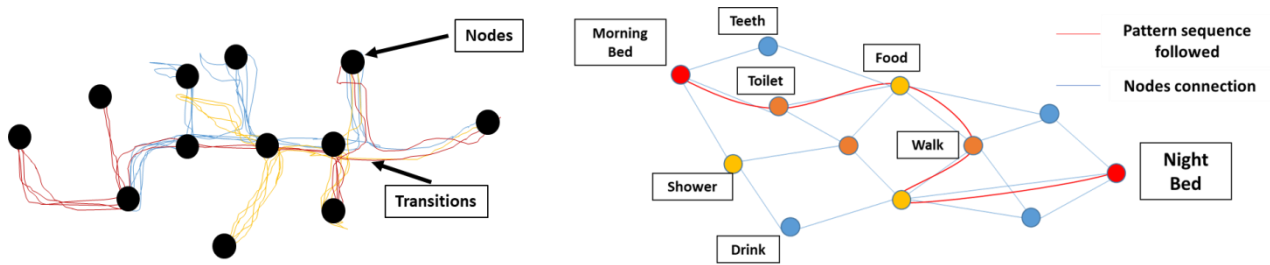
Figure 2. (a) Nodes as positions. (b) Nodes as activities.

On the other hand, the transitions or connections among these nodes or states illustrate the order or priority of events along the processes in which the different activities are carried out. These shapes described how people interact in different ways with their environment, allowing to define a *pattern routine* of each user within smart environments.

3.2 Reshaping different Smart Objects into a pattern-format

Figure 3-(a) symbolises nodes or states from different smart environments. Then, there is a need to develop a common structure which shapes nodes from different processes or scenarios. At this stage, an analysis of design-pattern techniques aims to identify and perform the most accurate pattern-form, which will enable the re-building of data into a common format.
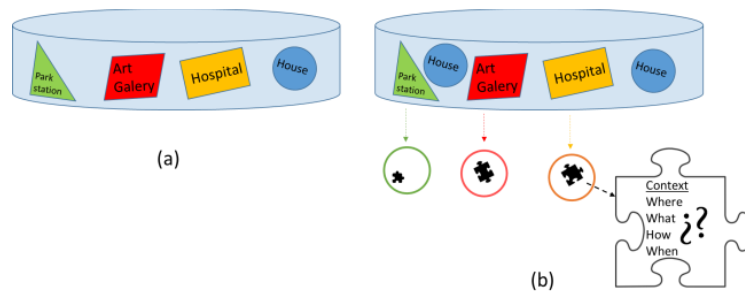
Figure 3. (a) Nodes from different processes and scenarios. (b) Nodes structured into a pattern based format.

A study published by Alexander proposed a game card structure for this purpose (Kumo Brian, 2018). In line with Alexander's proposal, nodes from different scenarios and processes will be stored and re-structured into a uniform format, as shown in Figure 2-(b).

This pattern-based model aims to act as an enabler for the main purpose of the tool as a whole, which is for collaboration among discrete smart solutions.

3.3 Describing understanding within processes

As a model that attempts to provide new generic solutions from the activities carried out in different scenarios of the smart environment domain, understanding the possible connections between nodes, and so capturing context becomes a requirement.

Following the discussion of some of the problems relating to context from section 2, our study performs a theoretical investigation of the properties shared by natural languages. This analysis is intended to reveal the underpinning principles behind the natural languages to provide understanding that can be applied in our approach to capturing a notion of

5

context in logic processes. A programing or artificial language is defined as a formal language that specifies a set of instructions that can be used to produce different outputs. Furthermore, a natural language cannot be easily understood and interpreted by computers. In order to achieve this we need to be very specific about giving commands or asking for information. Therefore, can a set of rules be described to capture context within processes?

Hence, our approach studies the rules that apply to natural languages, and provides a preliminary graphic which seeks to gather the elements required to build sentences that are meaningful in the sense of computability. Our aim is to describe the basis for capturing context within patterns. The nodes described and shaped into pattern-based structures needs to be analysed in order to provide a notion of context. This is needed because users will interact differently with the environment depending on their needs. For example, guiding to a user through an art gallery will vary the transition from one state to another in case it is required to move from one floor to another of the building and the user moves in wheelchair, thereby avoiding stairs or obstacles. Therefore, from a collaborative perspective, context enables different nodes to interconnect. In Figure 4, context is represented as a black puzzle piece within each block. Besides this, Figure 4 is intended to symbolise the concept of collaboration between different solutions gathered within smart spaces.
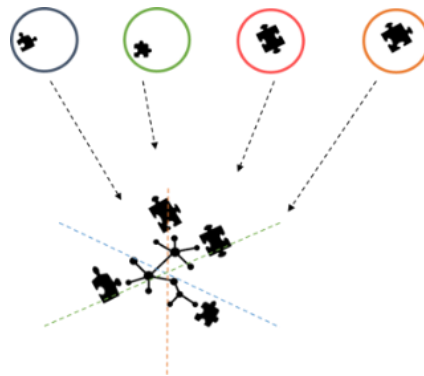
Figure 4. Context within pattern structures.

The study of languages is often segregated into syntactic, semantic and pragmatic components (Sowa, John F., 1995). Similar patterns of division are employed in other disciplines. For example, in computer science an 'if' or 'while' loop has a common syntax structure, semantic elements to describe the condition which applies to that loop and finally the pragmatic elements relating to the real-world effect of what the 'if' or 'while' loop is supposed to achieve. By illustrating the three components as separate axes in Figure 5 we begin to uncouple the components so they can be closely examined.
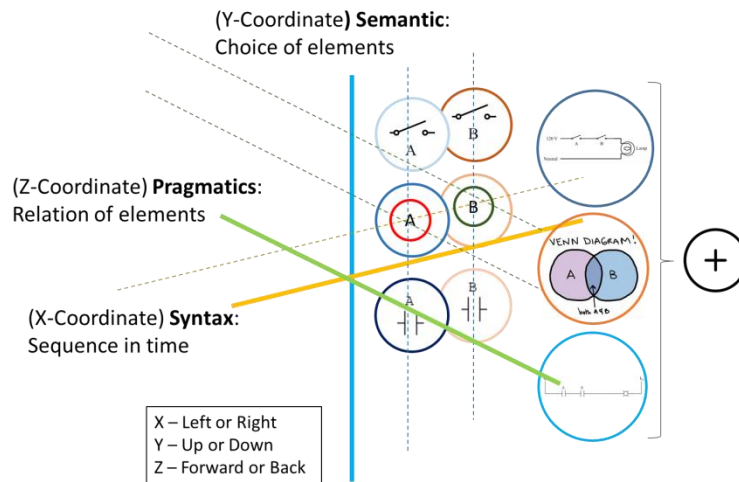
Figure 5. The rules of languages used to describe processes.

The syntax refers to the sequence in time in which the event occurs. Also, it represents the form of a valid structure. Though, it does not provide any information about the meaning of the sentence, program or results. The syntax of a process is represented by the natural sequence of steps.

The semantic or choice of elements, handles the meaning given to a combination of symbols. Yet, this first piece of information is not enough for a full understanding of why or what is described in the process. It is represented on the vertical axis because each block, which may be a physical entity, a mathematical construction, or some other expression in a natural or artificial language, can be associated with different meanings, altering the whole understanding of the linguistic sentence or process.

From a linguistic view, it can be sub-divided into 3 categories or levels namely Sentence, Phrase and Word Level.

Therefore, by following this division, the nodes can be categorised into different levels depending on their meaning.

Finally, the pragmatic will be drawn onto the Z-coordinate providing the purpose. The meaning in context of every process. Referring to how the user interacts with the piece of environment described within each node, and to the relation of elements and sequence of extra-linguistic information. Helping to understand how context aids the transmission of meaning in utterances.

However, these elements do not exist independently and can only be understood in terms of their interrelationships.

4. RESULTS

In order to analyse the different community nomenclatures, Table 3 provides an idea of different popular nomenclatures for representing patterns that is an extract of previous work by the authors (Arribas A., McGrory J., Berry D., 2017). In this case the pattern is a simple conditional statement.
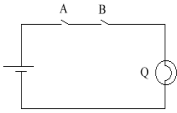
Category	Symbol	Description
Circuit/Wiring diagram		**Year created:** 1820´s (Circuit diagrams, 2018) **Attributes:** It combines addresses and datatype/value components. **Purpose:** illustrates actual wiring of the circuit. The logic is implicit to the user knowing the devices and the physical flow of electricity. Suitable for physical layouts and implementations for non-technical persons. **Intended use:** to guide workers who may be unskilled on the desired layout and placement of wires and devices.
Venn diagram		**Year created:** 1880 (Ruskey F.,Weston M., 2005) **Attributes:** Combines addresses and datatype/value components. **Purpose:** Technical or non-technical. **Intended use:** Graphical representation for non-technical persons and uses the pattern matching and grouping ability of humans.

Table 3, AND logic used in different communities.

All of the diagram nomenclatures, in Table 3, represent broadly the same idea, which is, the development of a pattern-based model which could unify and simplify the usage of different instructions under the same structure.

5. CONCLUSION

This paper has highlighted the phrase "design pattern" is an inclusive term referring to a repeating arrangement of an entity or combination of entities, such as, shape, influence, impression, form, function, fit, model, fashion, etc. For example, if we take a simple arithmetic expression, where operation such as '+' is performed, Figure 5, we can state '4+3', 'A+B', 'X AND Y', but we can also express '+' graphically as shown in Table 3. The AND operation is, therefore, in essence, a "design pattern" where, each of these expressions achieve the same goal.

However, what brands these AND expressions as 'different', 'not compatible with each other' and 'not cross compatible' is not the concept or notion of a repeating arrangement, but more specifically the incompatibility of the entity types (through lack of context).

Our work hints at an a technique, where if we can identify, record, describe and structure elements contained in the smart environment, we begin to open the door to cross-compatibility and then to collaboration. Areas such as assistive technology, energy management and environmental monitoring each use these design patterns extensively, but an obstacle issue is cross-compatibility and lack of context and situation awareness.

Future work, will test the designed pattern framework system and to show how the resulting patterns can be transferred to a "living laboratory" smart environment test bed being developed at the Greenway Hub / ESHI Building at DIT Grangegorman.

6. REFERENCES

The Free Dictionary: Pattern. https://tinyurl.com/ybwug9lc (Online; accessed 2-Feb-2018).

Wikipedia: Patterns in Nature. https://tinyurl.com/o4t6llg (Online; accessed 2-Feb-2018).

Wikipedia: Software design pattern. https://tinyurl.com/nkxzo4v (Online; accessed 2-Feb-2018).

Alexander, C., A Pattern Language: Towns, Buildings, Construction: Oxford Univ. Press, 1977.

25th Conference on Pattern Languages of Programs. http://www.hillside.net/plop/2018/

Dearden, A., (2006) Pattern languages in HCI: A Critical review L. Erlbaum Assoc, Vol 21, pp. 49-102.

Lazar, J., et al. (2010) Research Methods in human-computer interaction. J. Wiley & Sons, UK.

M. Vega-barbas, I. Pau, J. Augusto, (November, 2017). Interaction Patterns for Smart Spaces: A Confident Interaction Design Solution for Pervasive Sensitive IoT Services.IEEE Access (Volume: 6).

S. Yang, M. Kabir, M. Hoque, (2016). Mathematical Modeling of Smart Space for Context-Aware System: Linear Algebraic Representation of State-Space Method Based Approach. Mathematical Problems in Engineering, vol. 2016, Article ID 8325054, 8 pages, 2016.

Sowa, John F. (1995). Syntax, Semantics, and Pragmatics of Contexts. Int. Conf. on Conceptual Structures, pp. 1-15. vol 954. Springer, Berlin, Heidelberg.

Kumu. Brian, 2018. 64 Selected Patterns of Christopher Alexander. https://tinyurl.com/y9brvvsq

Arribas, A., McGrory, J., Berry, D..(2017, Dec.) Initial Investigation of popular diagramming used by different communities to inform development of a pattern language .8th Annual Graduate Research Symposium. DIT, Dublin. https://tinyurl.com/y9ll6qoo

Circuit diagrams and component layouts. https://tinyurl.com/yck48c74 (Online; accessed 17-Feb-2018)

Ruskey, F.; Weston, M. (June 2005). "A Survey of Venn Diagrams". Electronic Jnl of Combinatorics.