2018-07-07

Patterns within Patterns within the Smart Living Experience

Aitor Arribas Velasco
d16126348@mytudublin.ie

John McGrory
Technological University Dublin

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

Patterns within Patterns within the Smart Living Experience

Aitor Arribas Velasco
Dublin Institute of Technology
School of Electrical and Electronic Engineering, DIT, Kevin Street, Dublin 8, D08 NF82
d16126348@mydit.ie

John McGrory
Dublin Institute of Technology
School of Electrical and Electronic Engineering, DIT, Kevin Street, Dublin 8, D08 NF82
john.mcgrory@dit.ie

Damon Berry
Dublin Institute of Technology
School of Electrical and Electronic Engineering, DIT, Kevin Street, Dublin 8, D08 NF82
damon.berry@dit.ie

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 capacity to manage the data. Depending on the scenario, the diversity of piecemeal solutions almost reflects the diversity of problems they address. But some solutions can be reapplied. In the field of computing, design patterns can provide a general, reusable solution to commonly recurring problems within a given context through software design. This work seeks to determine the core elements of a technology-independent design pattern format and an open software framework can be developed to capture, share and redeploy existing successful and reusable strategies for commonly encountered smart environment use cases. Applying 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.

Design, Design Patterns, Smart Environments, Processes, Systematic approach.

1. INTRODUCTION

A pattern is an entity that when repeated in a sequence, contributes to guide or solve common reusable design issues [1]. However, patterns are not solely a human invention, these elements are found in the natural world. They exist in different contexts such as nature, art and architecture, computer science and so on, allowing humans to mathematically model natural processes for example; patterns, as observed in the natural environment, represent the optimization of a process. 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 [2].

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 [3].

This paper seeks to describe a preliminary design-pattern model 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 [4]. 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 these entities has motivated the appearance of conferences such as Pattern Languages of Programs (PLoP) [5].

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 [6]. By 2005, the research focused on collaboration,

connection and communication [7]. 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.

The concept of *context* is needed when we talk about 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 [8]. 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 [9]. The last work introduced Peirce's approach to the Artificial Intelligence community. However, an approach to capture the context in logic processes has not yet been determined. This work is motivated by the wide range of ideas around the notion of "context". In some cases, this leads to confusion and fragmentation depending on the field in which it is used. In the Artificial Intelligence literature, there is no single authoritative definition of the concept of context, whereas it is being widely used and applied to different approaches.

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 structure 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 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 describe the concept of a collective collaborative tool [10]. This concept is described as follows, by common iconography.

3.1 Data collection

As mentioned above, our research is focused on smart environments. Here, we can find Objects, Localisation and Meta data.

These general design elements can be used for example, to build a system to guide a service user through an art gallery. Hence, a piece of art could be described by elements mentioned above while other such elements could help to guide the visitors based on their preferences.

However, these same general design elements can be found in different settings such us a hospital or

nursing home where patients need to be monitored. Furthermore, in both scenarios, the paths followed by the service users are linked to attributes that describe the processes that interact with them.

This first step, Figure 1, aims to highlight the importance of capturing data from different processes of the smart environment [11].

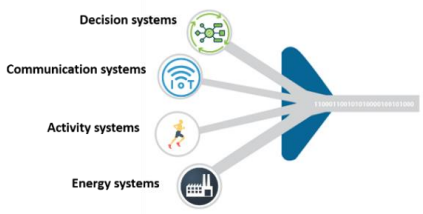


Figure 1 Capturing processes in smart environments.

3.2 Reshaping data into a pattern-format

Figure 2 (a) symbolises data from different smart environments. Then, as a part of the whole collaborative system, there is a need to develop a common structure which shapes data from different processes. As mentioned above, in an art gallery an object can be a piece of art, while in a hospital scenario, an object can be one of the steps followed within the process of a patient. 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 collected into a common format. In other words, the same structure attempts to define objects from different scenarios aiming to facilitate the cross-compatibility, and collaboration among different smart processes where a user can take part.

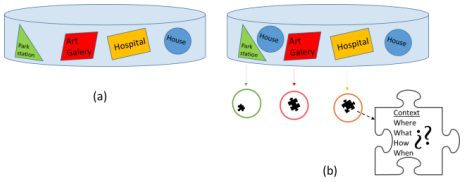


Figure 2 (a) Data collection from different processes and scenarios. (b) Data structured into a pattern based format.

A study published by Alexander proposed a game card structure for this purpose [12]. In line with Alexander's proposal, data 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 Understanding and building

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 data recorded, 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 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 information captured and shaped into pattern-based structures needs to be analysed in order to provide a notion of context. This is needed because different processes produce blocks of data from different scenarios. Therefore, from a collaborative perspective, context enables different blocks to interconnect.

The study of languages is often segregated into syntactic, semantic and pragmatic components [9]. 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 3 we begin to uncouple the components so they can be closely examined.

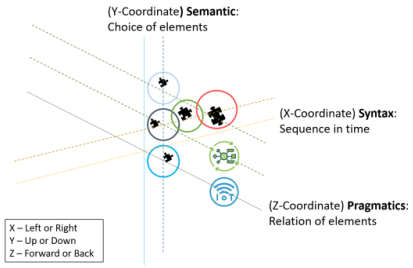


Figure 3 The rules of languages used to describe processes.

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 blocks 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 blocks are used, 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.

3.4 Integration

The last step is the interconnection among the different blocks, Figure 4, which would generate the collaborative solutions. In the context of this paper, design patterns together with the concept of collaboration allow us to introduce a way to describe solutions to common accessibility or usability problems in a specific context [13].

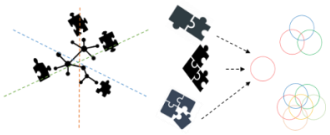


Figure 4 Blocks combination.

The separate processing components need to merge and collaborate to produce a greater solution than any one processing component can produce on its own.

4. OUR METHOD

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 [14]. In this case the pattern is a simple conditional statement.

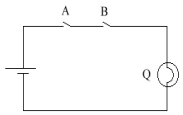
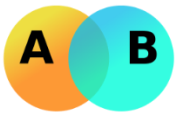
Category	Symbol	Description
Circuit/Wiring diagram		**Year created:** 1820´s [15] **Attributes:** It combines addresses and datatype/value components. **Purpose:** illustrates actual wiring of the circuit. The logic is implicit to the user knowing the devices and the physical flow of electricity. Suitable 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 [16] **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. CONCLUSIONS

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, 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, describe and structure elements in each of the three plane posed in Figure 3, through syntax, semantic and pragmatic lenses, 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

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

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

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

[4] Alexander, C., *A Pattern Language: Towns, Buildings, Construction*: Oxford Univ. Press, 1977.

[5] 25th Conference on Pattern Languages of Programs. http://www.hillside.net/plop/2018/

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

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

[8] Keeler, M., (1995) *The Philosophical Context of Peirce's Existential Graphs.* Third International Conference on Conceptual Structures. University of Washington, Seattle.

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

[10] Penciuc, D., Durupt, A., Belkadi, F., Eynard, B., and Rowson, H. (2014). *Towards a PLM interoperability for a collaborative design support system*. 8th International Conf. on Digital Enterprise Tech. Elsevier B. V. 25, pp. 369-376.

[11] Saponas, T., Prabaker, M., Abowd, G., Landay, J. (2006). *The impact of pre-patterns on the design of digital home applications*. Proc. Of 6th conference on Designing Interactive systems.pp. 189-198.

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

[13] J. O. Borchers, "A pattern approach to interaction design," Ai Soc., vol. 15, no. 4, pp. 359–376, 2001.

[14] 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

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

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