Future Directions for Research on Cyberpatterns

Clive Blackwell and Hong Zhu

Abstract As patterns in cyberspace, cyberpatterns shed light on research on the development of cyber systems from a new angle. They can help us move from an improvised craft to an engineering discipline because they help to transfer knowledge about proven solutions in an understandable and reusable format. They allow innovative applications in cloud, cyber-physical and mobile systems, and novel methods of use with data patterns for observation and analysis of 'big data' problems. The ultimate aim of research on cyberpatterns is an overall framework for cyberpatterns integrating all the cyber domains to help develop a better-understood and effective cyberspace. However, there are many research questions in cyberpatterns that remain unanswered regarding both their conceptual foundation and practical use. This chapter concludes the book by exploring some of the most critical and important problems needing to be addressed.

Keywords: Cyberpatterns, cyberspace, pattern language, pattern space, big data

Cyberpatterns are predictable regularities in cyberspace that may help us to understand, design, implement and operate successful and reliable cyber systems. In this chapter, we explore the possible future directions for research on cyberpatterns that we hope eventually leads to the development of a general framework. We group these research questions into two closely related sets of representation and engineering issues giving very brief summaries of the current state of the art.

1 Representation Issues

Patterns are reusable knowledge about recurring phenomena in certain subject domains, including repeatable successful solutions to certain engineering problems. How to represent such knowledge is of critical importance for all aspects of a pattern-oriented methodology. The following are among the most important research questions associated with pattern representation.

- ***Definition of the Semantics of Patterns***

Clive Blackwell and Hong Zhu

Department of Computing and Communication Technologies, Oxford Brookes University, Wheatley Campus, Wheatley, Oxford OX33 1HX. United Kingdom.

email: {CBlackwell, HZhu}@brookes.ac.uk

Ideally, the representation of patterns needs to package knowledge in a useful way to help both experienced and novice users alike. Moreover, it should also enable machine understanding and efficient processing. However, currently many patterns are inadequately documented with nature language, which are inevitably vague and incomplete. Thus, they typically lead to poorly implemented solutions and are hardly ever processed by computer. We need to design new pattern representation frameworks and languages to realise the full potential of cyberpatterns. A challenge in the design of pattern representation languages is that the real world phenomena and its digitalised presentation in the form of the data that a cyberpattern represents are often interpreted in the context of the environment and situation. Their semantics are not always clear, but heavily depend on human interpretation. Defining the semantics of patterns accurately enough for machine processing is a grave challenge.

- ***Uniformity of Pattern Representation for All Sub-domains***

In a complicated subject domain like cyberspace, there are a great number of patterns. In order to use them effectively, we must classify and categorise related patterns and understand their interactions within various cyberspace sub-domains as well as across sub-domain boundaries. Cyberpattern representation languages need to be expressive, extendable and flexible enough to apply to a range of potential problem contexts, but also needs standardisation, clarity and precision for knowledge acquisition and to aid automation. Another grave challenge in research on cyberpatterns is to meet these seemly conflicting requirements on pattern representation.

A good pattern representation language will allow effective and efficient investigation of the relationships and interactions between patterns so that they can be systematically classified, catalogued and organised for efficient retrieval and application through development of automated tools. An important research question is that, whilst patterns in different subject sub-domains demonstrate specific features of pattern interactions and relationships, is there a general framework or theory of pattern interaction applicable to all sub-domains and across the boundaries of sub-domains? Such a general framework or theory would help the development of tools for implementing patterns, detecting patterns in existing systems, and automated reasoning with patterns.

- ***Integration of Different Pattern Representations***

In the process of research in order to meet the above challenges, it seems most likely that different languages and frameworks will be proposed and advanced before a generally applicable framework or theory emerges. Different pattern knowledge bases will be, and in fact, some have already been, constructed by different vendors. If this does occur, the community will face another grave challenge, that is, how to investigate the relationships and interactions between patterns not only within one pattern representation language but also between different languages, not only within one pattern classification and cataloguing system but also across different systems.

2 Engineering Issues

Patterns help us move from an improvised craft to engineering discipline because they help to transfer knowledge about proven solutions in a comprehensible and reusable format. In order to advance such an engineering discipline, the following research problems must be addressed.

- ***Validation, Verification and Testing Patterns***

As a kind of knowledge representation, the correctness of patterns plays a critical role in their applicability. Validated cyberpatterns help to form a foundation for the study of cyberspace by encapsulating knowledge about the cyber world, analogous to scientific theories in the natural world. However, how to validate, verify and test the correctness of patterns is a grave challenge to research on cyberpatterns, on which little attention has been paid in the research community so far. We need to provide a solid theoretical foundation and to develop practical techniques to address these issues.

- ***Pattern-Oriented Knowledge Bases***

When patterns are formalised to give an adequate and unambiguous semantics, they can be stored and processed by computer systems for use as a knowledge base. A question then arises how to organise such a knowledge base. In particular, what kind of relationships between patterns should be included in the knowledge base? For example, the most important relationships between design patterns are sub-pattern, uses, composition and alternative choice [1]. It is noteworthy that inheritance thoroughly pervades the classification of attack patterns [2] in the CAPEC schema [3], and therefore a theory of inheritance would be useful for both attack patterns and possibly other pattern types.

Should redundancy between patterns be removed from the knowledge base? And, in what sense is a pattern redundant? Design pattern composition has been formally modelled by extending the *Pattern:Role* graphical notation with composition represented as overlaps between pattern instances [4]. The composition and instantiation of design patterns was studied in finer detail using a set of six operators for precise expression [5]. Decomposition of patterns into their fundamental components is also important. Smith [6] demonstrated that some design patterns can be composed from more elementary patterns. Should a composite pattern (i.e. a pattern that is a composition of other patterns) be regarded as redundant? Should we only store atomic patterns? How do we detect whether a pattern is a composite of others? These questions deserve a thorough investigation since they are fundamental to the development of a practical pattern-oriented knowledge base.

- ***Mechanisms for Pattern Interactions***

Moreover, given the fact that there are a large number of patterns in each complicated sub-domain of cyberspace, each subject sub-domain may develop its own pattern knowledge base(s). We need to extend the existing work on pattern languages to consider the relationships and interactions between patterns in different pattern spaces. We can compose cyberpatterns across different pattern spaces using both existing composition operators and novel ones, such as the ‘defeats’ relation between security and attack patterns, where the security pattern resolves the security forces caused by attack patterns in the particular problem context. This is closely related to the pattern representation problem that requires compatible representation to enable associations between patterns across different cyber subdomains to be meaningful and useful. There is some research to indicate that the existing formal modelling of design patterns may possibly be extended to cyberpatterns [2,7].

- ***Mechanisms for Efficient Application of Patterns***

Patterns can be represented in descriptive or prescriptive ways. The descriptive view characterises the predictable structure and behaviour of patterns in deployed systems. Many novel types of cyberpattern such as attack patterns, machine patterns, network traffic patterns and user behaviour patterns are usefully considered descriptively. In contrast, the prescriptive view provides recommendations of best practice to help develop general methods of providing solutions to recurring problems. Many patterns that have been developed, such as architectural and security patterns, are typically used prescriptively in building systems.

A special form of prescriptive representation of design patterns is as transformational rules. For example, Lano [8] proposed that a design pattern is represented as a set of transformation rules to refactor a system that contains design flaws. Refactoring alters the internal structure of an existing system to improve its quality without changing its externally observable behaviour [9]. Within the cyberpatterns domain, most of Hafiz's catalogue of security patterns could be seen as security-oriented transformations [10].

Each of the descriptive and prescriptive ways has its own advantages and disadvantages in the acquisition and application of pattern knowledge. The former helps to discover a pattern as repeated phenomena and makes it easier to detect occurrences of a pattern from observed data. Tools have been developed to support the uses of design patterns at the reverse engineering stage of software development to detect patterns in code and in design diagrams. The latter helps to generate instances of a pattern in a constructive way. Many tools have been developed to support uses of design patterns at the design stage and coding stage for instantiating patterns. A research question worthy of investigating is whether we can develop a general mechanism independent of application domain that enables us to instantiate patterns and detect patterns based on a pattern representation language.

Moreover, it will be redundant to store each pattern with two different representations: one descriptive and one prescriptive. This will not only consume more storage space and processing time, but also cause inconsistency between the two. More research is needed to reconcile these two approaches, for example, by developing a mechanism to enable transformation of one representation into another.

- ***Mechanisms for the Automatic Acquisition of Patterns***

With the advent of cloud computing, mobile computing and cyber-physical systems, a great quantity of data is now available as recorded observations of real world phenomena and human behaviour, as well as the interplay between them. There is also a great amount of work in the design and implementations of cyber systems. Such design knowledge is available in the form of program code, design diagrams, and operation manuals, etc. How to discover pattern knowledge in such big data or "big code" is the most challenging but also most important research question. The characteristics of cyberpatterns and their interrelationships suggest that we must go beyond the existing machine learning and data mining techniques. A new mechanism for automatic discovery of patterns from complicated combinations of structured and non-structured data is needed.

3 Conclusions

To conclude this book, let us quote Nobel Laureate Ernest Rutherford, who once asserted that "*all science is either physics or stamp collecting*". By physics, Rutherford meant clean, succinct principles that apply to diverse phenomena, whereas by stamp collecting he meant the cataloguing and organising of large sets and varieties of observations. Being reusable knowledge about recurring phenomena, we believe patterns form a bridge that possibly spans the gap from stamp collecting to physics, because pattern-oriented research methods do not only recognise, catalogue and organise observations, but also discover regularities as well as interrelationships and interactions between such regularities. This knowledge is the mother of clean, succinct principles.

Rutherford also once said, "*Scientists are not dependent on the ideas of a single man, but on the combined wisdom of thousands of men, all thinking of the same problem, and each doing his little bit to add to the great structure of knowledge which is gradually being erected*". The pattern-oriented research methodology offers such a platform for thousands of computer scientists to contribute to a structure of knowledge. We hope this book is only the start of the erection of a great structure of knowledge of cyberspace.

References

1. H Zhu. Cyberpatterns: A Pattern Oriented Research Methodology for Studying Cyberspace. In Unifying Design Patterns with Security and Attack Patterns. Springer. 2014.
2. I Bayley. Challenges for a Formal Framework for Patterns. In Cyberpatterns: Unifying Design Patterns with Security and Attack Patterns. Springer. 2014.
3. Mitre Corporation. Common Attack Pattern Enumeration and Classification (CAPEC). Mitre Corporation. http://capec.mitre.org. Accessed 11 Jan 2014.
4. T Taibi. Formalising design patterns composition. IEE Proceedings – Software. 2006; vol 153 no 3: pp 126–153.
5. H Zhu and I Bayley. An Algebra of Design Patterns. ACM Transactions on Software Engineering and Methodology. ACM Press. July 2013; Vol 22 no 3: Article 23.
6. J Smith. Elemental Design Patterns. Addison-Wesley. 2012.
7. H Zhu. Design Space-based Pattern Representation. In Cyberpatterns: Unifying Design Patterns with Security and Attack Patterns. Springer. 2014.
8. K Lano. Design Patterns: Applications and Open Issues. In Cyberpatterns: Unifying Design Patterns with Security and Attack Patterns. Springer. 2014.
9. M. Fowler. Refactoring: Improving the Design of Existing Code. Addison-Wesley. 1999.
10. M Hafiz. Security on demand. PhD dissertation. University of Illinois. 2011. http://munawarhafiz.com/research/phdthesis/Munawar-Dissertation.pdf.