

Pattern languages and genres for writing computer science discourse

Dai Fei Yang

Learning Skills Unit, Student Services (ODS)

University of Western Sydney

Peter Goodyear

CoCo Research Centre, Faculty of Education & Social Work

University of Sydney

How can a successful adjunct grammar program be modelled using pattern languages and genre theory? The authors will first discuss the genre grammar based design for an undergraduate computer science grammar program, using examples of writing a Use Case to illustrate this pedagogical approach. The paper will then introduce the pattern language approach and how this can be combined with genre theory to address students' academic literacy problems. Finally, the paper will discuss the potential use of pattern languages to design reusable programs for academic literacy support in higher education.

Keywords: pattern languages, genre theory, academic literacy, computer science

Introduction

The use of pattern languages, derived from Christopher Alexander's work in architecture and planning, has been extended in computer science to the development of software programs. In recent years researchers and educators have started to explore the use of pattern languages to the study of teaching and learning.

In this paper, pattern language is used to model a framework for a successful adjunct grammar program for an undergraduate computer science course. The aim is to help students develop academic writing skills, specifically in writing a Use Case.

In the development of the grammar program, well established genre theory derived from Martin and others (Martin & Rose, 2003; Martin, 1992a; Hasan, 1996; Halliday, 1994) in systemic functional linguistics (hereafter SFL), was used to identify the social context and purposes of a Use Case, its text type and distinctive language features. Students developed skills in writing a Use Case by practising the recurring patterns in discourse structure and grammar.

Using pattern languages with genre theory to provide a theoretical framework for solving computer science students' language problems in academic writing and communication is a new approach and needs further exploration. This paper aims to invoke discussion among the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE) community in order to take this pedagogical approach forward.

Background

The Object Oriented Software Development (OOSD) program, which is the setting for this work, is for first year university computer science students. An adjunct grammar program has been designed by the Learning Skills Unit, Student Services at University of Western Sydney to help students develop language skills in writing a Use Case. A Use Case is a collection of possible sequences of interactions between the system and its users (or actors). It consists of unambiguous descriptions and a step by step text format to allow a computerised system to translate the user's need to achieve a desired goal. (Box & Ferguson, 2002)

According to the comments made by lecturers, students tend to make grammatical errors in tenses and by writing long and redundant sentences in their descriptions. These errors often result in misinterpretation and confusion of the sequence of interactions.

A genre grammar based framework for course design

The system of genres is derived from SFL. According to Martin (2000), from a semantic perspective we interpret genres 'as patterns of meaning'. From a social perspective we characterise genres 'as staged goal oriented social processes' (p.4). Genre is a system of texts, consisting of patterns of meanings and schematic structures used in various social contexts (Martin & Rose, 2003; Martin, 1992a; Halliday, 1994). In university settings, *genres* are used to refer to text types with various schematic structures, for example an essay, a lab report or a case study. An essay, for instance, typically has an introduction, body and conclusion while a lab report typically has an abstract, an introduction, hypothesis, methodology, results, a discussion and a conclusion.

In a discourse community, genres are employed by community members to share their expertise and technical knowledge (Swales, 1990). In order to be recognised and accepted by a discourse community, students need to develop their knowledge about the explicit use of genres and language forms in the course of their study. Students gain apprenticeship in these language skills through classroom dialogues and writing academic essays, oral presentations and other forms of academic activities (Unsworth, 2001; Martin, 1993, 2000; Lemke, 1989; Halliday, 1993a).

This paper describes how the above concept is integrated into the pedagogical approach taken in designing a genre based grammar course to help undergraduate computer science students write a Use Case.

The example of the following Use Case is based on the real life context of the Moggan Bay Sailing Club Regatta (Box & Ferguson, 2002). It is used for the race nomination acceptance process and it is highly contextualised in performing the specific functions of a sailing race registration. The system must provide an ability to check the skipper, the crew and the boat for eligibility for the race, to accept/reject nomination and notify users of the registration outcome. In order to carry out these functions, a sequence of interactions between the actor (the skipper) and the system has to take place (see examples in the description below).

Example of the Moggan Use Case

Use case:	Accept nomination
Category:	Core
Actors:	Skipper
Description:	This Use Case begins when a skipper submits a nomination to participate in a regatta. The skipper, the crew and the boat listed in the nomination are checked for eligibility for the race. The boat is recorded as a nomination in the regatta. The Use Case ends when the skipper is advised of the acceptance of the nomination.
Purpose:	To register the nomination of a boat, the skipper and crew in a regatta.
Trace:	Business functions: 2.1, 2.2 and 2.3
Source document:	Regatta Entry Form.
Notes:	In regard to regattas that have passed the date by which nominations close yet have not occurred being available for late nominations: the user decided that these regattas will not be available. Late nominations are not accepted for any regatta. (Box & Ferguson, 2002, p.230)

Social context and register category: Before writing a Use Case the understanding of its social context helps students identify the circumstances in which culturally recognised activities are taken place. In SFL such circumstances and activities are discussed as *field*. (Halliday, 1994; Martin, 1992a) The concept of field in turn helps students develop the skills in the construction and selection of *lexical items*. In SFL it is referred to as *register category*. Register category establishes the expected indexical vocabulary which is likely to appear in the Use Case. According to Martin (2001), the ease of comprehension and construction of academic discourse depends on the degree of familiarity with issues and social context being discussed.

While social context deals with a higher level of metafunction, the introduction to a Use Case schematic structure helps students control the staging of sequence of activities. For example, the explicit linear schematic structure of the Moggan Use Case can be illustrated as below:

The third strata concerns with the Moggan Use Case is its distinctive language features as illustrated in Table 1.

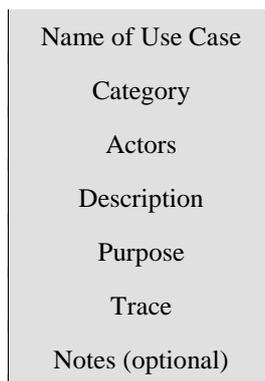


Figure 1: Schematic structure of the Moggan Use Case

Table 1: Language features in the Moggan Bay Sailing Club Regatta Nomination Use Case

Language features	Examples	Explanations
Present tense	The Use Case <u>begins</u> when a skipper <u>submits</u> a nomination.	We use present tense to describe an habitual or usual action, a procedure or a situation.
Passive voice	The skipper, crew and the boat of nomination <u>are checked</u> for eligibility.	We use passive voice to focus on what happens to the person or the object.
Present perfect	In regard to regattas that <u>have passed</u> the date by which nominations close yet <u>has not occurred</u> being available for late nomination.	We use present perfect tense to describe an action or event which is completed but is still connected in some way to the present.
Nominalisation* * Nominalisation refers to a verb or an adjective being changed into a noun, eg. accept – acceptance, eligible – eligibility	The Use Case ends when the skipper is advised of <u>the acceptance</u> of the <u>nomination</u> .	In academic writing, nominalisation represents a formal writing style, condensed meanings and a focus on ideas rather than action. Students need to learn the rules for changing a verb/adjective into a noun. For example: the verb <i>complete</i> is changed to <i>completion</i> by deleting the ‘e’ and adding ‘ion’. Non English Speaking Background (NESB) students sometimes may find mastering these rules challenging, although native English speaker students may do it automatically without knowing these rules of grammar.
Nominal group* * A nominal group is sometimes referred to as a noun group. In a nominal group there is always a headword with pre or post modifier/modifiers (sometimes with both).	<u>Source document</u> <u>Regatta Entry Form</u> <u>late nominations</u> <u>a/the boat</u> <u>the boat listed in the nomination</u>	Nominal groups allow us to construct very concise expressions. Students need to apprentice the skills in unpacking and constructing them. For example: ‘a completed regatta program’ is used instead of ‘a regatta program which is completed’.

The genre based theoretical approach provides a useful tool for deconstruction and scaffolding linguistic knowledge. The strata of such knowledge building are defined as follows:

Macro level: the understanding of social context, social purposes, activity staging and register category

Text level: schematic structure and stagings of discourse

Sentence level: dominating specific language features including sentence structures, grammatical forms and lexical grammar (eg. nominalisation and noun groups).

This genre based approach model has greatly enhanced students’ deep learning and skills in academic writing. However, students need to be exposed to a much wider range of sample texts and learn to write a

variety of use cases within similar social context and genre structures. This will develop their meta-functional skills (social functional concept) to use and apply various schematic structures and language features flexibly and creatively within the genre framework.

Pattern languages

Ideas about design patterns and pattern languages, now filtering into educational design, originated with the work of Christopher Alexander. The first book in Alexander's trilogy, *The Timeless Way of Building*, addresses fundamental principles and practice in architectural design. 'The quality without a name' (p. x) is the main theme and he uses many metaphors, such as 'self maintained fire', 'ocean waves or blades of grass', 'flower', 'seed' (p. x-13) to illustrate the social aspects of architectural design. Well designed buildings and towns support recurrent patterns of activity which give shape and meaning to people's lives.

Alexander's work shares a concern of SFL by addressing social functionality. In SFL, social goals and purposes are realised by the choice of language in meaning making. Genres are a system of patterns of meaning used in different social contexts and situations. So Alexander's pattern language is particularly relevant to language teaching and learning. His theory provides the SFL community with a new perspective on how language and interaction relate to inhabited space. SFL addresses the social function of language for communication while pattern languages focus on social functions of architectural design.

In the second book in the trilogy, *A Pattern Language: Towns, Buildings, Construction*, Alexander and his colleagues present a language of 253 patterns. Architects, communities and ordinary people can use these patterns to create and improve buildings and neighbourhoods (Alexander *et al.*, 1977).

Each Alexandrian pattern follows the same format: a picture which depicts the end product of the pattern; an introductory paragraph explaining the context of the pattern; a description of the problem which the pattern is designed to solve; a headline in bold type stating the essence of the problem; the body of the problem describing the empirical background of the pattern; the evidence for its validity and the different ways the pattern can be used; a paragraph on the solution appears in bold type with precise instructions. The closing part of the paragraph relates the pattern to other smaller patterns which are needed for its completion and embellishment.

The pattern structure resembles the schematic structure of text types, from the SFL genre perspective, in a number of ways. It has a linear sequence and every pattern should use the same sequence of textual composition to ensure consistency and easy application.

The pattern based approach has recently been extended into the area of software design (Gamma *et al.*, 1995), learning management systems (Avgeriou *et al.*, 2003) and educational design for networked (online) learning. (Goodyear *et al.*, 2004) Due to the scope and limit of length of this paper it is not possible to discuss these studies here in details. However, it is important to address the focus of using pattern languages. It is to encourage users to enhance their creativity and flexibility of combining existing patterns for design and not to be too caught up in the patterns. Designers should use patterns only when they are most relevant, appropriate and interesting.

Using pattern languages to design reusable computer science literacy programs

Designing academic literacy support programs is a very complex task. Some computer science students, in particular international students, have a range of language problems. While genre theory provides a valuable framework for developing students' knowledge about language, the authors believe that pattern languages can be used for designing a network of reusable patterns, capturing the pedagogical strategies, available resources and teachers' knowledge to support computer science literacy program design. The problem solution based framework used by pattern language is a valuable tool for addressing students' specific language problems for a particular subject or of a particular learner group.

In what follows, the authors present an example, using the Moggan Use Case to illustrate how a pattern language can be developed for a literacy program for writing a Use Case. (In the following examples we

use literacy skills instead of language skills in order not to confuse ‘language’ here with the ‘language’ as in pattern languages.)

Table 2: Steps in designing a pattern language for computer science literacy

Step 1	Give the literacy support pattern a name.
Step 2	Write a paragraph describing the context of this literacy support pattern.
Step 3	Insert three diamonds (◆◆◆) in a new line
Step 4	In bold type write one or two sentences describing the outcomes of the program
Step 5	Describe the students’ language problems
Step 6	In bold type, provide instructions on how to develop a program to solve students’ language problems.
Step 7	Insert another three diamonds (◆◆◆) in a new line
Step 8	Write a short paragraph describing how this pattern is related to other patterns, such as patterns of genres, patterns of different sentence constructions, patterns of tenses and patterns of noun groups.

Literacy skills for writing a Use Case for computer science students

This pattern is concerned with the development of students’ linguistic knowledge and language skills in writing a Use Case. Students learn to use the schematic structure and explicit forms of language which repeatedly appear in Use Case development.



Students learn to write a Use Case succinctly and unambiguously by using specific textual sequence and register category within its social context

Writing a Use Case is a common task in object oriented software design. Students need to be able to use precise language to describe a sequence of interactions between actor/user and the system in order to achieve a desired goal. A Use Case follows a specific pattern of text structure and language features. Grammatical features include the use of the present tense, the present perfect, the passive voice, nominalisation and nominal groups.

Due to the lack of language skills or being unable to identify such a restricted genre, students tend to write long and redundant sentences. This often results in a confused sequence of interactions and ambiguous descriptions.

Genre theory provides a useful framework for addressing students’ problems as described above. A grammar support program focused on modelling, deconstructing and scaffolding of a Use Case can help students improve their skills in writing a concise Use Case. Examples of Use Case can be found from relevant textbooks. Instructors need to first develop students’ awareness of the social context and purposes of a Use Case. Students then learn how the designed sequence of interactivities results in its restricted schematic structure and dominant language patterns used. The program should provide students with a variety of sample texts and tasks to practise these language skills and to avoid rigid application of patterns. Tasks and examples for student practice need to be relevant and interesting, preferably using real life situations within the computer science environment.



This pattern can be used in conjunction with other patterns, such as patterns of social context, patterns of social purposes, patterns of genres, patterns of sentence constructions, patterns of tenses and patterns of noun groups.

Implications and further development of pattern languages

Genre theory and pattern languages both address social functions. While genre theory focuses on the social context in which language is used to realise purposes, pattern languages address the functionality of regions, towns and buildings to enable active social interactions and activities among the community and its people. Genre based design provides a set of patterns of meanings and text types to help students develop writing competence in academic discourses while pattern languages capture a wealth of knowledge, experience and resources which can be easily shared within the discourse community.

From the examples illustrated in this paper, a genre based and a pattern based program design have different strengths. Genre theory provides a framework for describing detailed language knowledge needed for writing a Use Case while pattern languages provide a step by step description of how to develop this type of genre based language support program. In the meantime its problem solution principle helps lecturers establish the key relationship between the problem and the approach to address these problems. A combination of these two pedagogical approaches provides a more powerful and complimentary framework.

However, applying pattern languages is a new pedagogical approach in academic literacy design. Beyond the comfort zone of using genre theory this paper aims to explore a new territory by extending pattern languages. To explore this approach further, future research needs to focus on the following examinations:

1. Conduct empirical studies to gather data, critical analysis of learning results and evidence to evaluate the pattern language pedagogical approach.
2. Investigate ways of integrating such a framework into computer literacy program design and teaching.
3. Perception of using pattern languages by academic staff and students, for example, the advantages and disadvantages of using pattern designs and to what extent a pattern is used and under what situation a pattern may become rigid and rules should be challenged to create flexibility and creativity.

Conclusion

An academic community uses a system of genres to share its specialised disciplinary knowledge and purposes. Within a genre based framework, students develop the awareness of social context and purposes and the meta-knowledge of text types, schematic structures and explicit forms of languages used in discourse. While genre theory provides a model in deconstruction and scaffolding in the strata of writing a Use Case, students need to develop skills in using such schematic structures and language features in a flexible and creative manner. This can be achieved by apprentice students with a wide range of discourses and examples of how different Use Cases are used to achieve different social functions. Students should avoid rigid application of structures.

Pattern languages share a similar theoretical principle with SFL in addressing social function, activities and purposes. In this paper, we also explore the potential use of pattern languages to provide a problem solution based approach for academic literacy program design. The initial investigation in this paper aims to step out of the comfort zone by extending pattern languages in the combination of genre based approach. Further research on empirical experiment and trial in applying this new approach with the investigation of its integration into teaching program is much needed. This will enable researchers to further evaluate the effectiveness and produce evidence in support of such approach.

The authors invite the learning community, in particular, those working in computer science literacy support, to provide feedback and further explore this approach in order to take it forward.

References

- Alexander, C. (1979). *The Timeless Way of Building*. New York: Oxford University Press.
- Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. & Angel, S. (1977). *A Pattern Language: Towns, buildings, construction*. New York: Oxford University Press.
- Averious, P., Papasalouros, A., Retalis, S. & Skordalakis, M. (2003). Towards a pattern language for learning management systems. *Educational Technology & Society*, 6(2), 11-24.
http://www.ifets.info/journals/6_2/2.html
- Box, I. & Ferguson, G. (2002). *Object-oriented software development: Step by step*. Sydney: Pearson Education Australia.
- Goodyear, P., Avgeriou, P., Baggetun, R., Bartoluzzi, S., Retalis, S., Ronteltap, F. & Rusman, E. (2004). Towards a pattern language for networked learning. *Proceedings Networked Learning Conference 2004*, Lancaster University, England, UK. [verified 29 Oct 2004]
http://www.shef.ac.uk/nlc2004/Proceedings/Individual_Papers/Goodyear_et_al.htm
- Halliday, M. A. K. (1993a). Some grammatical problems in scientific English. In M. A. K. Halliday. & J. R. Martin (Eds), *Writing Science: Literacy and Discursive Power*. London: Falmer Press.
- Halliday, M. A. K. (1994). *An introduction to functional grammar*. London: Edward Arnold.
- Hasan, R. (1996). Speech genre, semiotic mediation and the development of higher mental functions. In D. Cloran, D. Butt & G. Williams (Eds), *Ways of saying: Ways of meaning; Selected papers of Ruqaiya Hasan*. London: Cassell.
- Lemke, J. (1989). *Talking Science: Language, Learning and Values*. New Jersey: Ablex.
- Martin, J. R. (1992a). *English text: System and structure*. Amsterdam: Benjamins.
- Martin, J. R. (1993). Genre and literacy: Modelling context in educational linguistics. *Annual Review of Applied Linguistics*, 13, 141-172.
- Martin, J. R. (2000). *Grammar meets genre: Reflections on the Sydney School*. Inaugural lecture, Sydney University Arts Association.
- Martin, J. R. (2001). Language, register and genre. In A. Burns & C. Coffin (Eds), *Analysing English in a Global Context*. London: Routledge.
- Martin, J. R., & Rose, D. (2003). *Working with discourse: Meaning beyond the clause*. London: Continuum.
- Swales, J. (1990). *Genre Analysis: English in academic and research settings*. Cambridge: Cambridge University Press.
- Unsworth, L. (2001). Evaluating the language of different types of explanations in junior high school science texts. *International Journal of Science Education*, 23(6), 585-609.

Acknowledgement

The authors would like to thank Ilona Box for giving her permission to use her textbook material in this paper. Also thanks to the project team members of the Use Case grammar program for their permission and encouragement for writing this paper.

Dai Fei Yang, Learning Skills Unit, Student Services (ODS), University of Western Sydney, Australia.
d.yang@uws.edu.au

Peter Goodyear, CoCo Research Centre, Faculty of Education & Social Work, University of Sydney, Australia.
P.Goodyear@edfac.usyd.edu.au

Please cite as: Yang, D.F. & Goodyear, P. (2004). Pattern languages and genres for writing computer science discourse. In R. Atkinson, C. McBeath, D. Jonas-Dwyer & R. Phillips (Eds), *Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference* (pp. 961-967). Perth, 5-8 December.
<http://www.ascilite.org.au/conferences/perth04/procs/yang.html>

Copyright © 2004 Dai Fei Yang & Peter Goodyear

The authors assign to ASCILITE and educational non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ASCILITE to publish this document on the ASCILITE web site (including any mirror or archival sites that may be developed) and in printed form within the ASCILITE 2004 Conference Proceedings. Any other usage is prohibited without the express permission of the authors.