Chapter 3
Relevant Canonical Genre Families

ABSTRACT

The basis for a functional communication perspective for elicitation strategies is a selection of generalised high-value genres referred to as canonical genres. There are a number of families of genre but those that are of particular relevance to interviewing and interview protocols are the so-called factual and narrative genre families. The purpose of the chapter is to describe each of these in turn. Following the software engineering pattern movement, each genre that belongs to these families will be described using a common format: a name (capitalized as is convention for genres in SFL), a description, a genre element inventory or a table that provides the list of the genre elements codes, names and function, the corresponding genre digraph, and an authentic transcript marked up with genre stages. The transcripts used to exemplify these canonical genres, reveal a range of unusual features unpacked in more detail in the next chapter.

INTRODUCTION

There are some apparent substantive similarities between the concept of canonical genres in SFL and work that has been undertaken in the realms of Artificial Intelligence (AI) and Software Engineering. Although theoretically unrelated to these, genres considered here have similarities to some of the approaches developed in AI that seek to address questions of the representation
Relevant Canonical Genre Families

of meaningful situations in discourse. For example, the concept of a **frame** was developed by Minsky (1975) as a way of representing in the form of a data structure, knowledge about a stereotypical situation as well as including expectations and presumptions about the situation. An alternative representation of knowledge is in the form of **scripts** and **plans**, developed by Schank and Abelson (1977). Like these AI concepts, canonical genres can also be used to represent knowledge but perhaps a closer analogy is the notion of a pattern.

The use of the concept of a **pattern** is now widespread, but its origins in “technical” analysis and design applications dates to the work of Christopher Alexander, Emeritus Professor in Architecture at University of California, Berkeley, and his colleagues, at the Centre for Environmental Structure. They developed an entirely new approach to architecture, building and planning based on two big ideas (Alexander, Ishikawa & Silverstein, 1977). The first “big idea” is that people always rely on certain “languages” that allow them to articulate and communicate an infinite variety of designs within a formal system that gives them coherence. In effect, Alexander et al (1977) proposed that the built environment could form a kind of language, the parts of which could be reused in many ways. This language would then enable people to build any kind of building or any aspect of the built environment. While they did not use any theory of communication as a foundation for this “language”, nonetheless their work foregrounds the significance of communication to discipline.

The second “big idea” is that the standard units of this architectural language are patterns. Specifically, patterns are reusable (that is, generic) answers to architectural design problems. Their book offers 250 such patterns, including answers to questions like the optimum height of a window or the amount of space a neighbourhood should have reserved for green spaces. In what was, no doubt, a practical solution to a vexing problem of describing these patterns, but actually became a defined aspect of this work, each pattern was described in the same way, employing a problem statement, a discussion of the problem, with an example, and a solution. The importance of this approach for architecture is that it was a systematic way of approaching design activities. Significantly, patterns could be corrected until they became as general as possible and reused within the architectural domain in many different ways.

In one of the most recent and interesting cross-fertilizations between technical disciplines, software engineers adopted these ideas about patterns
GOTOPS: Code of Technoethics Governance
www.igi-global.com/chapter/gotops-code-technoethics-governance/64098?camid=4v1a

Using Network Analysis to Improve Decision Making for Partner Selection in Inter-Organizational Networks
Laura Ponisio, Pascal van Eck and Lourens Riemens (2013). International Journal of Human Capital and Information Technology Professionals (pp. 26-39).
www.igi-global.com/article/using-network-analysis-to-improve-decision-making-for-partner-selection-in-inter-organizational-networks/96837?camid=4v1a