The Journal of Systems and Software 122 (2016) 215–238

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Usage-based chunking of Software Architecture information to assist information finding

Moon Ting Su[a,b,*], John Hosking[c], John Grundy[d,1], Ewan Tempero[a]

[a] Department of Computer Science, University of Auckland, Building 303 s, 38 Princes Street, Auckland 1010, New Zealand
[b] Faculty of Computer Science and Information Technology, University of Malaya, Lembah Pantai, 50603 Kuala Lumpur, Malaysia
[c] Faculty of Science, University of Auckland, UniServices House, Level G, 70 Symonds Street, Auckland 1010, New Zealand
[d] Faculty of Science, Engineering and Technology, Swinburne University of Technology, 1 Alfred Street, Hawthorn, Victoria 3122, Australia

ARTICLE INFO

Article history:
Received 20 February 2016
Revised 2 September 2016
Accepted 3 September 2016
Available online 13 September 2016

Keywords:
Usage-based chunking
Software architecture document
Information finding
Task

ABSTRACT

One of the key problems with Software Architecture Documents (ADs)[2] is the difficulty of finding information required from them. Most existing studies focus on the production of ADs or Architectural Knowledge (AK)[3], to allow them to support information finding. However, there has been little focus placed on the consumption of ADs. To address this, we postulate the existence of a concept of "usage-based chunks" of architectural information discoverable from consumers' usage of ADs when they engage in information-seeking tasks. In a set of user studies, we have found evidence that such usage-based chunks exist and that useful chunks can be identified from one type of usage data, namely, consumer's ratings of sections of ADs. This has implications for tool design to support the effective reuse of AK.

1. Introduction

Finding useful information in large amounts of software documentation is not easy. This is a key problem in addition to the perennial problems of out of date (but sometimes still useful), poorly written and untrustworthy documents that have a high creation cost (Lethbridge et al., 2003). The difficulty of finding information also applies more specifically to Software Architecture Documents (ADs) (Koning and van Vliet, 2006; Rost et al., 2013).

ADs hold many benefits for Architectural Knowledge (AK) sharing but as documentation increases with size and complexity of the software system, many challenges await current Software Architecture (SA) documentation approaches (Jansen et al., 2009). One of these challenges is locating relevant AK (Avgeriou et al., 2007; Jansen et al., 2009) either across multiple documents or within these documents (Jansen et al., 2009). Knowledge retrieval features in existing AK management tools are simple and reactive (Tang et al., 2009).

The problem of finding information from ADs is further worsened by the various stakeholders' having only partial interest in the total content of the documents. Many stakeholders' concerns are addressed by a small fraction (sometimes as little as 25%) of an AD (Koning and van Vliet, 2006). Consequently, the readers of ADs complain of having to wade through too much irrelevant information. Information needed to solve a specific task may be spread throughout the document and be organised in a linear fashion not matching user needs for a specific AK information-seeking task.

Thus, despite the wealth of AK that ADs contain, they may not be used, or not used most effectively, because of the difficulty of finding information in them. To support finding information in an AD, we argue that architectural information in it needs to be structured into or presented as *chunks* (Su, 2010; Su et al., 2011a, 2011b). A *chunk* is *a collection of related pieces of architectural information* (Su, 2010; Su et al., 2011a, 2011b). We posit that identifying and reusing chunks simplifies finding of information, by enabling related architectural information, which may be dispersed in a document, to be retrieved collectively as a unit. We propose to identify chunks by finding 'commonality' in consumers' usage of the information in ADs when engaged with certain information-seeking tasks.

We investigated this idea by carrying out studies that acquired AD usage data when consumers performed certain information-seeking tasks. We collected both explicit data, where consumers were asked to provide information about their AD usage, and implicit data, where the usage data was gathered by KaitoroCap

* Corresponding author.
E-mail addresses: smting@um.edu.my (M.T. Su), j.hosking@auckland.ac.nz (J. Hosking), j.grundy@deakin.edu.au (J. Grundy), e.tempero@auckland.ac.nz (E. Tempero).
[1] Present address: School of Information Technology, Deakin University, Burwood, Victoria 3125, Australia.
[2] Software Architecture Document (AD).
[3] Architectural Knowledge (AK).

http://dx.doi.org/10.1016/j.jss.2016.09.009
0164-1212/

(Su, 2014; Su et al., 2011b), the tool we developed to track consumers' interaction with ADs. We analysed the collected usage data to identify chunks for these tasks. Our work is a preliminary study of the concept of 'usage-based chunks' in ADs. Our work aims to show that usage-based chunks exist and that they are likely to vary across different information-seeking tasks in an AD. We chose three representative use cases (or information-seeking tasks) for SA documentation to illustrate this.

This paper is organised as below: Section 2 explains the concepts of chunking and information chunk. Section 3 presents the methodology. Section 4 discusses the chunking results. Section 5 details the threats to validity of our findings. Section 6 compares our work with existing work. Section 7 presents our key findings, conclusions and possible future work.

2. Chunks to support finding of information

In this section we present the concepts of "chunking" and "information chunk" introduced in other research areas and how we have adopted these concepts in our work. We also review existing work in the field of SA that supports different forms of chunking, and define the concept of a "chunk" as used in our work.

2.1. Chunking and information chunk

The idea of chunking in this research draws upon a number of areas that involve human processing of information. These include human cognition, human learning, perception, and the study of chess. In these areas, *chunking* generally refers to the grouping of related items into a single unit or *chunk*. In the field of human learning, a *chunk* is defined as "meaningful unit of information built from smaller pieces of information", and *chunking* is "the process of creating a new chunk" (Gobet and Lane, 2012). These notions of the terms are also used in the study of expertise, and acquisition of language and education, all of which are related to learning.

Miller, a cognitive psychologist suggested that our short-term or working memory can only hold 'seven plus-or-minus two' (i.e. between five to nine) items (Miller, 1956). While this finding may not be universally true, there is nevertheless some limitation on how much information we can process and recall. However, the capacity of the working memory can be increased through a *chunking* process, where items with similar or related attributes are bound conceptually to form a single unit or *chunk* (Curtis, 1984; Miller, 1956). Since Miller's work in 1956, work in cognitive science has established chunking as one of the key mechanisms of human cognition (Gobet et al., 2001).

Chunking can be goal-oriented, involving a deliberate conscious process (Gobet and Lane, 2012). An example is Miller's re-coding of specific information (Gobet and Lane, 2012) as fewer chunks with more bits per chunk (Miller, 1956). For example, the 9-digit binary number 111001110 can be re-coded as a 3-digit decimal number 716, which is easier to process and remember. Another type of chunking is perceptual chunking which is more of an automatic and continuous process that occurs during perception (Gobet and Lane, 2012). Perceptual chunking has been used to explain the ability of chess experts to recall briefly-presented positions with high precision.

We adopt similar notions for these terms in our work: *chunking* here refers to the *grouping of related pieces of information* and a *chunk* is *a collection of related pieces of architectural information*. We observe that the principle underlying all the above areas in human processing of information is: the users or consumers of information construct information chunks during their usage of the information, and use the chunks in later recall or retrieval of the information. Our work builds upon this principle of how humans process information and takes it further in two aspects. Firstly, by making the derivation of information chunks explicit. Secondly, the derivation of the information chunks is based on the 'commonality' found in the consumers' usage of information. The 'commonality' serves as possible means to group information into a chunk.

All the areas above focus on the consumption of information. *Chunking* also exists in structured writing (Horn, 1997), which focuses on the production of information. In structured writing, chunking refers to grouping of pieces of information into manageable units, called information blocks and information maps. An information block is the basic unit of subject matter. An information map is a collection of information blocks. The notion of information map in structured writing resembles the notion of a chunk in the areas that focus on the consumption of information mentioned earlier. We use the term *chunk* instead of *information map*, since our work focuses on the consumption instead of the production of information, and, the chunking principle used in structured writing originates from Miller's work in human cognition (Miller, 1956).

2.2. Chunking in the field of Software Architecture

In the field of SA, there is no general consensus on what the chunks of architectural information Software Architecture Documents or Architectural Descriptions should comprise (Greefhorst et al., 2006). In addition, the term *chunking* is not established in SA although the following forms of *chunking* seem to be supported:

a) *Chunking supported by architecture documentation constructs such as architecture framework, view, view packet, and template*. These constructs provide guidance on grouping of architectural information. Architecture frameworks such as Zachman's Framework (Zachman, 1987), provide guidance on what the chunks should be (Greefhorst et al., 2006). A view is a representation of a coherent set of architectural elements and the relations among them (Bass et al., 2003). View packets organise view information in digestible chunks (Clements et al., 2003). Documentation templates such as interface template (Bass et al., 2003) and architecture decision template (Tyree and Akerman, 2005) assist the documentation of interface and decision, respectively, by providing guidance on what should be documented for them and the organisation of their constituents. Using templates such as these place together pieces of information that are related, by following the standard groupings suggested by the templates.

b) *Chunking supported by searching facilities*. A search using the searching facilities of the documentation environment returns pieces of information that are related in certain ways. In keyword-based searching, items retrieved are related because they contain the same or similar terms as the searched terms. In query-initiated discovery of the semantic structure of documents based on words in the documents (de Boer, 2006; de Boer and van Vliet, 2008), the documents or the units of texts retrieved are related because of their semantic structures. In the retrieval of architectural information chained by underlying models (de Boer and van Vliet, 2011; de Graaf et al., 2012; Jansen et al., 2009; Su et al., 2009; Tang et al., 2011), architectural elements or knowledge instances retrieved are related because of the pre-defined relations in the underlying models.

c) *Chunking supported by automatic generation of stakeholder-specific ADs*. Sections or knowledge instances in the stakeholder-specific ADs are related because of the semantic information in the sections' profiles (Diaz-Pace et al., 2013;