

VU Research Portal

Ontology-based Software Architecture Documentation

de Graaf, K.A.

2015

document version

Publisher's PDF, also known as Version of record

[Link to publication in VU Research Portal](#)

citation for published version (APA)

de Graaf, K. A. (2015). *Ontology-based Software Architecture Documentation*. [PhD-Thesis - Research and graduation internal, Vrije Universiteit Amsterdam]. Proefschriftmaken.nl.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:

vuresearchportal.ub@vu.nl

3

Organising and Retrieving Architectural Knowledge in File-based Documentation

In this chapter we review literature to find out how AK is typically retrieved from file-based documentation (RQ1). We found that file-based documents have a linear organisation of AK whilst document users do not necessarily retrieve AK following the same organisation. Users may not easily find AK that is outside the document organisation, however, creating a document organisation that supports the AK retrieval needs of all users is difficult and introduces redundant and scattered AK descriptions. AK retrieval challenges reported in literature stem from limitations of the linear file-based document organisation.

The previous chapter showed that industry professionals experience AK retrieval challenges when the document organisation does not support their questions. In Section 3.1 of this chapter we study literature to find out how SA documents are typically organised, why they do not support the questions of all AK users, and to identify AK retrieval challenges reported in literature. We shortly discuss in Section 3.2 to which extent hypertext documentation can alleviate the identified AK retrieval challenges. The next chapter details how the challenges may be alleviated by ontology-based documentation.

3.1 File-Based Documentation and its Issues

In their highly influential paper on multi-dimensional software decomposition [105], Tarr *et al.* describe how traditional formalisms in software engineering can only provide a single “dominant” dimension when achieving separation of concerns. Use of a single dominant dimension, by software decomposition based on

CHAPTER 3. ORGANISING AND RETRIEVING ARCHITECTURAL KNOWLEDGE IN FILE-BASED DOCUMENTATION

e.g., quality, functions, development tasks, or data objects, causes problems with reuse, traceability, comprehension, evolution, and maintenance. These problems not only apply to the software itself, but also to its documentation.

Parnas and Clements argue [78] that documents should be designed and structured with separation of concerns in mind; each aspect of a system is described in one section. File-based documentation can achieve this separation by using, for example, a view-based structure [9, 21, 2].

Each view provides a ‘cross-section’ of AK. Views are useful to stakeholders who are interested in different cross-sections of AK. Cross-referencing of AK between views can help to make interrelated AK traceable and retrievable.

The separation of concerns achieved through a particular set of architecture views makes the retrieval of certain knowledge – knowledge contained in one view – relatively easy, but at the same time it makes the retrieval of knowledge scattered across views difficult. This is a wicked problem: choosing a different set of views does not solve the issue, but simply moves it elsewhere. This problem is recognized by Rozanski and Woods in [81], where the notion of perspectives is introduced next to that of views. Perspectives serve to organise specific types of knowledge across views.

As the number of different stakeholders and their unique needs for AK increase in large and complex projects, there is also an increase in misalignment between the AK needed by stakeholders and how they may retrieve this AK from file-based documentation. In practice, most stakeholder concerns are addressed by a small documentation subset that is different for each concern [57]. Moreover, existing approaches for documenting decisions only frame part of the stakeholders concerns related to decisions [112]. Extensive use of cross-references between scattered AK or (alternatively) redundant recording of AK in file-based documents makes AK retrieval and maintenance impractical and error-prone.

The questions about AK that documentation users may ask based on their concerns are illustrated in the right-hand side of Figure 3.1. The questions are about certain types of AK, e.g., decisions and requirements, and relationships between AK, e.g., ‘impacts’ and ‘realized by’. Relationships between AK show how an architectural element is connected to or associated with the rest of the architectural design. For example, a developer may want to find all requirements realized by a component s/he has to build, whilst an architect may want to find all decisions that impact the same component during impact analysis.

The left-hand side of Figure 3.1 illustrates how a linear organisation of AK in a table of contents supports file-based document users in finding a limited set of relationships between AK. As a result, only one out of three questions asked by

3.1. FILE-BASED DOCUMENTATION AND ITS ISSUES

document organisation	supported AK relationship	match?	required AK relationship	questions of document users
Table of Contents 1. Functional requirements 1.1 Requirement 1 - <i>Login</i>	Requirement → subsystem			
1.1.1 Subsystem <i>FrontEnd</i> [...] 1.1.1.5 Decision D5	subsystem → decisions	No	decisions → design alternatives decisions → (related) decisions	"I need to find all alternatives and decisions that are related to decision D5"
2. Performance 2.5 Decisions	quality attribute → decisions	Yes	quality attribute → decisions	"I need to find all decisions that impact performance"
3. Architecture design 3.5 Subsystem <i>FrontEnd</i> 3.5.2 Component <i>GUI</i>	subsystem → component	No	component → requirements	"I need to find all requirements that are realized by component GUI"
3.8 Maintenance 3.8.3 decision [...]	quality attribute → decisions			

Figure 3.1: Mismatch between supported and required AK relationships in file-based document organisation

the document users is supported in this AK organisation.

For instance, the organisation does not detail where every type of AK can be found, e.g., design alternatives. Moreover, the relationship between components and requirements, necessary to answer the question about component *GUI*, is missing in this organisation. Extending the AK organisation to support this question would introduce redundant and scattered descriptions of either requirements or components.

Indexing additional relationships (or 'cross-sections') between AK in a file-based document organisation introduces redundant and scattered AK descriptions. Relationships between AK that are not indexed by the document organisation have to be searched inside document contents. It is however difficult to make document content unambiguous [77] and organise the AK therein such that it is successfully communicated to users with different backgrounds [83].

Explicitly describing relationships between AK makes the AK traceable. Empirical evidence is given by Shahin *et al.* in [85] and Javed and Zdun in [53] that improved traceability leads to better architectural understanding. Lack of traceability in SA documentation is considered a major problem in industry practice [80].

CHAPTER 3. ORGANISING AND RETRIEVING ARCHITECTURAL KNOWLEDGE IN FILE-BASED DOCUMENTATION

In [51] Jansen *et al.* identify AK retrieval challenges that partially stem from above issues with organising AK. We describe in Section 4.2 how the challenges can be alleviated by the ontology-based approach.

1. *Architecture documentation understanding*
Document understandability becomes more challenging when documentation size increases in large and complex systems [21]. The original intention of the authors is often lost.
2. *Locating relevant architectural knowledge*
Knowledge is often spread over multiple documents [23] which makes it hard to locate AK, especially if documents lack finer details.
3. *Support for traceability between different entities*
Providing traceability between documentation sources is difficult [49]. Text and tables are limited in communicating different relationships.
4. *Support for change impact analysis*
Because decisions, requirements, and their relationships are usually not explicit, it is often very hard to reliably analyze and predict the impact of changes to the architecture.
5. *Assessment of design maturity*
Architecture design is difficult to evaluate if there is no status overview of the conceptual integrity, correctness, completeness, and buildability of the architecture [111, 9].
6. *Credibility of information*
AK often changes in large and complex systems and the cost to update is sometimes prohibitive [51]. Documentation is quickly outdated and its users lose confidence in its credibility [66].

Problems related to the above issues and challenges are reported by Rost *et al.* in a recent survey [80] on SA documentation among practitioners working in 33 companies around the world. The top three reported problems with the representation of AK in the documentation that 109 of these practitioners work with are 1) *inconsistent and missing structure*, 2) *scattered information*, and 3) *missing traceability*.

3.2 Hypertext Documentation and Its Issues

Hypertext and wiki systems have been used in some software organisations for SA documentation. The use of tags and categories can help to organise knowl-

edge. However tags quickly lose meaning when used arbitrarily. Hypertext is also known as nonlinear text [22], yet its organisation remains linear with the use of categories.

Hyperlinks provide cross-referencing by pointing to information, however, the pointers do not specify the meaning of relationships. Without explicit semantics, not all AK user will be able to understand an organisation of AK by means of hyperlinks.

Several researchers [32, 22, 39] report that users of hypertext documents feel 'lost' and have difficulty gaining an overview of the material being read and how this material is interrelated [106]. Likewise, users of wiki-systems may also experience a lack of structure when navigating and finding relevant information [14].

Hypertext systems address AK retrieval challenges 2 and 3 (in Section 3.1) to some extent using hyperlinks and challenge 6 using version control, e.g., in wikis. However, because hyperlinks do not have specific semantics, they are not practical for filtering and querying AK based on the properties of relationships between AK. This is necessary for effectively addressing AK retrieval challenges 1, 4, and 5.

Semantics can be conveyed by named hyperlinks in hypermedia systems [118], hyperlinks in knowledge-based hypertext [70], and labelled links in spatial hypertext systems. Solis *et al.* describe a spatial hypertext systems for AK retrieval in [90] and its qualitative evaluation in [89], which is the only study on using spatial hypertext for AK retrieval that we know of.

3.3 Conclusion

We studied literature to identify how AK is typically retrieved from file-based documentation. Many AK retrieval challenges reported in practice stem from limitations of linear file-based documentation organisation. This AK organisation makes it hard to separate concerns, and comprehensively organise interrelated AK to support the AK needs of all document users. Hypertext-based documentation used in practice lacks the semantics to fully address the limitations and challenges of file-based documentation. This suggests that there is room for improving AK retrieval practice.