Semantic Component Selection – SemaCS.

Maxym Sjachyn
Ljerka Beus-Dukic

School of Informatics

Semantic Component Selection – SemaCS

Maxym Sjachyn
Cavendish School of Computer Science
University of Westminster
sjachym@wmin.ac.uk

Ljerka Beus-Dukic
Cavendish School of Computer Science
University of Westminster
l.beus-dukic@wmin.ac.uk

Abstract

In component based software development, project success or failure largely depends on correct software component evaluation. All available evaluation methods require time to analyse components. Due to the black box nature of components, preliminary judgments are made based on vendor descriptions. As there is no standard way of describing components, descriptions have to be interpreted using semantics and domain knowledge. This paper presents a semi-automated generic method for component identification and classification based on generic domain taxonomy and user generated semantic input. Every query is semantically tailored to what is being looked for, arriving at better results then it is currently possible using available automated categorisation systems.

Keywords: Component Selection, Ontology, Semantic Web.

1. Introduction

Component based software development introduced fundamental changes to the way systems are acquired, integrated, evolved and deployed [1]. The most significant of these is linked to design and implementation stages. Fundamentally, an application framework has to be conceived before components fitting that framework are identified. As a result, requirements stage of a typical software development process is extended to include component selection and assessment.

Due to its importance, component assessment has been addressed relatively well (e.g. PORE [2], DesCots [3], CMMI [4] OTSO [5], STACE [6], BASIS [7], etc.). However, given a large number of components, it is unlikely that all available alternatives can be considered. Thus, it is crucial to select most relevant components for assessment. This strategy results in a better suited solution being found faster. But, this is not easy to attain, as components can originate from previous projects, open source, or can be purchased commercially. Furthermore, components are diverse in terms of size, functionality, and areas of use (e.g. desktop applications, math libraries, operating systems, etc) [8]. Moreover, there is no component description standard; hence preliminary judgments are typically based on non-uniform top-level descriptions. All these shortcomings make categorisation and selection a difficult issue to resolve. To give an idea of the type of information that may be available, Figure 1 demonstrates WinSCP description obtained from [9]. WinSCP is a file transfer client. However, unless one is familiar with Internet based file transfer issues, the way WinSCP functions may not be as obvious simply by looking at the description.

Project	: WinSCP
Description	: WinSCP is a SFTP and SCP client for Windows using SSH. Its main function is secure copying of files between a local and a remote computer. Beyond this basic function, WinSCP manages some other actions with files. Plugin to FAR manager is available too.
Development Status	: 5 – Production/Stable
Operating System	: Win 32
Intended Audience	: Developers, End Users
Programming Language	: C++
License	: GNU (GPL)
Topic	: Communications

Figure 1. WinSCP component description [9]

SemaCS (Semantic Component Selection) attempts to analyse component descriptions, like that in Figure 1, offering a better, more precise and faster evaluation. SemaCS is inspired by the Semantic Web [10] community and its approaches because the nature of current component portals introduces similar issues. In both cases, there is abundant non-uniform information available and a need to analyse it efficiently. Consequently, there is a need for an automated generic method of component identification and classification capable of addressing current problems like accuracy, heterogeneity, and non-standard descriptions.

Remainder of this paper is structured in the following way: Section 2 provides general background and looks at ontologies and domain hierarchies, Section 3 describes SemaCS approach, and Section 4 presents a small selection of related work, followed by conclusion and future work in Section 5.

2. Ontology and domain hierarchies

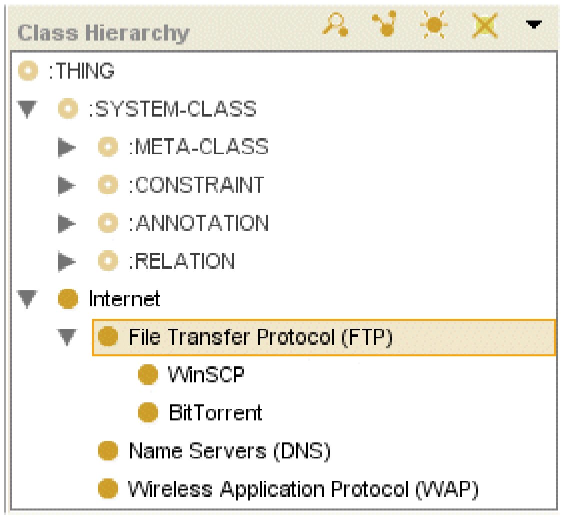
When selecting a component, many solutions have to be considered. This is easier to achieve if all needed information is available in one place. Web-based component portals like eCots [11], SourceForge [9], ComponentSource [12], Flashline [13] attempt to provide this functionality. Component catalogues, supplied by portals, contain information about a range of vendor solutions described in a relatively uniform way (in some circumstances open source options as well). These catalogues normally rely on ontologies and domain hierarchies to function.

An ontology is a reusable machine and human readable definition of domain knowledge containing domain terms, their meaning and logical relationships [14, 15]. It is through those definitions that reasoning and query interpretations are performed. For Web-based systems W3C's Web Ontology Language [16] is currently the most popular choice. Additionally, graphical ontology editing tools like Protégé [17] can aid with the creation process. Overall, there are three main approaches taken to ontology generation: manual, automatic, and bridging.

Manually generated ontologies are normally very accurate. However, they are usually quite expensive and inflexible because they are created using human judgment [18].
Systems that employ purely automatic means of generation are inexpensive but, thus far, not as accurate. As a compromise, automatic generation is often combined with manual to improve reliability while keeping costs relatively low (e.g. Yahoo search engine).

Finally, there is ontology bridging whereby multiple ontologies can be employed by defining a translation metric (bridge) between them [18, 19].

Domain hierarchies are somewhat similar to domain ontologies. However, hierarchies contain little domain semantic definition. Instead, they provide a general description of the domain itself (this is sometimes referred to as taxonomy). Hierarchies start with a top level class and get more specific with depth. For example 'Internet' class is general and contains more explicit subclasses Transfer Protocol (FTP), Name Servers (DNS), File and Wireless Application Protocol (WAP) [9]. In turn those classes have their own subclasses that are more concrete e.g. WinSCP is of type FTP. This example is shown in Figure 2 and is created using Protégé [17].



```
:THING
▼ :SYSTEM-CLASS
    ▶ :META-CLASS
    ▶ :CONSTRAINT
    ▶ :ANNOTATION
    ▶ :RELATION
▼ Internet
    ▼ File Transfer Protocol (FTP)
        WinSCP
        BitTorrent
      Name Servers (DNS)
      Wireless Application Protocol (WAP)
```

Figure 2. FTP domain example

3. SemaCS

Regardless of underlying technologies, when software components are being selected, be it through a portal like eCots [20] or any other repository, suitable candidates have to be chosen. Normal keyword matching techniques [21] are inefficient as the meaning of what is being searched for is simply ignored. Furthermore, keyword matching has a tendency to return unrelated results. Domain knowledge stored in ontology can be used to interpret component descriptions and user queries in order to arrive at relevant results. However, domain knowledge is constantly changing. New terms appear and meaning of old terms can alter. Because of this, domain ontologies are very expensive and hard to keep up-to-date.

3.1. SemaCS specification.

SemaCS is a generic method of component identification and classification. Due to its dynamic nature, this method stays up-to-date and does not suffer from drawbacks normally associated with manually or automatically created taxonomies (see Section 2). SemaCS consists of four modules (Figure 3):

- WWW Crawler locates components on the web
- Generic Taxonomy is used to classify components
- Query Interpreter is used to interpret queries and extract semantic user input
- Cache is used to store recently evaluated components and user taxonomies

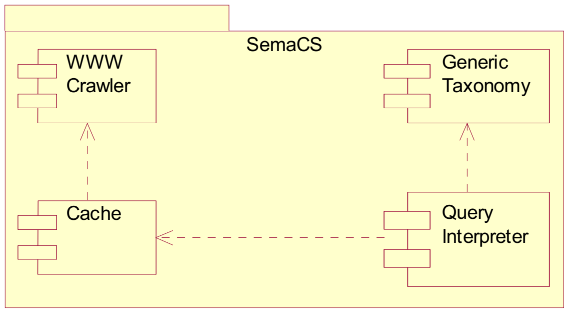


Figure 3. SemaCS component View

WWW Crawler

WWW (World Wide Web) Crawler scans the web to extract component descriptions. Because this process does not involve processing the extracted information, it is not implemented in the current prototype. At the moment, manually extracted data for component descriptions are obtained from Dedicated Systems Encyclopaedia [22]. However, when implemented, Crawler module will take advantage of currently available search engines (e.g. Google). Eventually, Crawler would be expanded to aid with the categorisation process.

Generic Taxonomy

As stated in Section 2, ontology is a reusable machine and human readable definition of domain knowledge containing domain terms, their meaning and relationships [14, 15]. This domain knowledge is represented as classes (e.g. class of FTP clients, class of Web browsers, etc.). Further, ontology is structured using taxonomy of classes and subclasses [15]. In simple terms, taxonomy is a domain hierarchy.

SemaCS is not designed for a specific domain. Therefore, to be generic, its taxonomy is only concerned with descriptions that every component has, such as vendor, version, implementation language etc. However, each user query results in a tailored taxonomy being created, using generic taxonomy as the starting point. Thus, accuracy expected from a complex taxonomy can be achieved but without drawbacks associated with similar large structures.

Query Interpreter

Search process starts with the taxonomy not being tailored to any specific domain. This generic taxonomy is then updated and expanded with keywords, definitions, and descriptions obtained from the user, Wikipedia [23], Swoogle ontology dictionaries [24], and Automatic Meaning Discovery Using Google [25].

Information used in the process can be divided into distinct types:

- Component Infrastructure
- Operating System
- Component Type
- Custom Taxonomy coupled with knowledge gathered over time

Component Infrastructure and Operating System

Component Infrastructure and Operating System form part of the generic taxonomy. This information is used to minimise the amount of information to be searched by applying top level category filter. For example, given that there are six hundred components in the repository, but only one hundred are created for Windows Operating System, a large number of components that are not suited to the task can eliminated. At the next pass, given the user is searching for components implemented using .NET Infrastructure, the search can be further refined.

Component Type

Component Type refers to specific types (e.g. WAP, FTP, etc) rather then general types one expects to find in manually created taxonomies (e.g. transfer protocols). The accuracy of 'type' definitions is dependant on data gathered over time, comprising of successful queries and resulting user selections as well as knowledge about the domain acquired in the process. In effect the more SemaCS is used, the more accurate it gets. Eventually, knowledge about domains would evolve to create higher level descriptions and hence the ability to perform searches more efficiently (in terms of processing time).

Nevertheless, some components can not be categorised into a single atomic 'type' parameter. In such a case, previous stages aid in eliminating mismatches. For example, if user is looking for an FTP client, results that can be classified into 'FTP' category are returned, even if they can perform some other function (e.g. encryption). However, this process can be reversed as well (e.g. in a situation where complex functionality is required) arriving at the results just the same. In other words, to provide the needed flexibility, components may have more then one keyword and more then one type associated.

Custom Taxonomy

A custom taxonomy tailored specifically to the search is created as the last SemaCS's stage. This taxonomy is based on a combination of related and user defined keywords/definitions. Additionally, search taxonomy can be expanded and refined at any time using additional input from the user (e.g. additional keywords, selecting additional or different or related technologies, etc).

Cache

Cache is used to store categorised components as well as recent queries. In effect, all system data is stored in the Cache component. However, due to dynamic nature of SemaCS, this data is regularly changed and updated. Furthermore, even at this point, user taxonomies can be edited and expanded, therefore improving accuracy of the system as it is being used.

4. Related work

This section presents a relevant sample of related work. Approaches outlined here are of two types: search engine based and semantic or domain model based. Discovery based on a search engine is not as accurate as results obtained using semantic methods mainly because searching is performed using search engine's implementation and algorithms. However, search engine approaches are fast, inexpensive, and grant access to a large part of the WWW.

Google by reformulation

Google by reformulation [18] allows to select the domain being searched for by extending a query on the go through hyperlink selections based on already located information. Even though there is no automated domain interpretation or reasoning employed, this add-on to Google can limit a number of unrelated results significantly using a simple user friendly approach. An adaptation of this method is used in SemaCS.

Automatic meaning discovery using Google

Automatic Meaning Discovery Using Google [25] is a simple novel approach to semantic meaning and relationship extraction. It is based on the fact that most of the knowledge is available on the Internet. Moreover, being able to use a count of occurrence for words or phrases can provide semantic distance or relationship. Initial results published in the case study [25] have provided positive evidence. An adaptation of the method is used in SemaCS to obtain preliminary relationship results.

SemRank

SemRank [26] attempts to detect the possibility of a certain type of relationship to be more likely (conventional) or less likely (unconventional) in order to decide what to search for. As the decision is left to the system, it is unlikely that accuracy could equal that of manual ontology driven methods. However, if adapted to utilize user input, accuracy can be improved. This adapted variation is used in SemaCS.

Semantic Web

Semantic Web is a framework created to make web pages easily searchable because there was no description standard; it is based on description tags (Meta data) defined in Extensible Markup Language (XML) and Resource Description Framework (RDF). Today, almost every page contains Meta data and most search engines can use it. Meta descriptions can be analysed and searched automatically because their semantic meaning is understood. This is achieved through the use of a dictionary of domain terms containing domain concept definitions and their meaning – a domain ontology [16]. Semantic Web had a large impact on SemaCS design as it provided an insight into the use of ontology and taxonomy.

Swoogle

Swoogle [24] is a crawler-based indexing and retrieval system similar to Google search engine. However, Swoogle is aimed at Semantic Web and works with RDF and OWL (Web Ontology Language) documents. Swoogle extracts metadata from discovered documents and provides ontology rank, a measure of importance. One of the features Swoogle provides is the ontology dictionary. This is compiled based on ontologies that have been discovered on the Web and is used by SemaCS as a reference to locate definitions for new terms as well as to provide interpretations for those that are already known.

ARTEquAKT

ArtEquAKT [27] uses an ontology that was created using sections from the CIDOC Conceptual Reference Model ontology [28] coupled with WordNet [29], a general-purpose lexical database, and GATE6 (General Architecture for Text Engineering) - an entity recogniser [30]. A synergy of those projects has allowed ArtEquAKT to identify knowledge fragments consisting of not just entities but also the relations between them. For example, the fact that 25/10/1881 is Pablo Picasso's date of birth can be recognised. Nonetheless, domain of Painters that was studied as part of the project is much less complex then that of software components. It is planned for similar functionality to be added to SemaCS.

eCots

eCots [20, 31] provides COTS component identification, characterisation, experience feedback, discussions and rating services. In the future, filtering functionalities would also be provided, based on the rating service. While eCots does use a domain hierarchy, it does not provide semantic-based search and filter facilities. This component portal was used as part of the initial case study.

COTS product characterization

COTS products characterisation [32] is based on a description framework consisting of two parts: application domain characteristics (e.g. for database components response time is important) as well as a set of generic quality characteristics (resource utilisation, stability, maintainability, etc.) obtained from the ISO/IEC 9126 standard. Nevertheless, information about those characteristics still has to be gathered manually. COTS products characterisation provided an insight into the way components could be categorized.

Web-based component evaluation

Web-based component evaluation [33] allows for components to be self testing. It relies on components having a standard set of interfaces that can be used to query them in order to obtain information like manufacturer, performance, security constraints, required interfaces etc. However, this approach requires manufacturers to provide a testing interface for every component. At present, there are no components complying with this approach apart from those used in a case study [33]. Nonetheless, it is feasible that such an approach may eventually be used with open source components.

5. Conclusion and future work

SemaCS addresses the issues of accuracy and cost as it is a crossbreed between manually generated taxonomy system and query reformulation. However, instead of using in-house experts to create taxonomies, expert input is provided by system users.

At the moment the first prototype is nearing completion. Initial testing would be carried out using component descriptions obtained from [22]. After addressing any identified issues, SemaCS would be tested against component portals (e.g. eCots [11], ComponentSource [12], etc.) using four different unrelated domains and sets of data. In the next stage, SemaCS prototype will be made available to select professionals from industry and academia to validate the method. Having addressed any deficiencies and improvements identified at this stage, SemaCS would then be made available to open source community. The system will be further improved over time to include, for example, knowledge elicitation (to extract implied meaning from component descriptions) and natural language interpreter (allowing the use of 'natural' English to perform searches).

References

[1] Brown, A.W. and K.C. Wallnau, "*Engineering of component-based systems", Second IEEE International Conference on Engineering of Complex Computer Systems*, Montreal, Que. Canada, 1996, pp. 414-422.

[2] Maiden, N. and C. Ncube, "*Acquiring COTS Software Selection Requirements",* IEEE Software, Volume 15, Issue 2, 1998, pp. 45-46.

[3] Grau, G., et al., "*DesCOTS: A Software System for Selecting COTS Components", 30th EUROMICRO Conference*, Rennes France, 2004, pp. 118-126.

[4] Larsson, S., I. Crnkovic, and F.E. ABB, "*On the Expected Synergies between Component-Based Software Engineering and Best Practices in Product*

Integration", 30th EUROMICRO Conference, Rennes France, 2004, pp. 430-436.

[5] Kontino, J., G. Caldiera, and V.R. Basili, "*Defining factors, goals and criteria for reusable component evaluation", 1996 conference of the Centre for Advanced Studies on Collaborative research*, Toronto, Ontario Canada, 1996, pp. 21-33.

[6] Kunda, D. and L. Brooks, "*Identifying and Classifying Processes (traditional and soft factors) that Support COTS Component Selection: A Case Study", European Conference on Information Systems 2000*, Vienna University of Economics and Business Administration, 2000, pp. 173-180.

[7] Ballurio, K., B. Scalzo, and L. Rose, "*Risk Reduction in COTS Software Selection with BASIS", 2002 International Conference on COTS-Based Software Systems*, Orlando, FL, USA, 2002, pp. 31-43.

[8] Clark, B. and M. Torchiano, "*COTS Terminology and Categories: Can We Reach a Consensus?" 2004 International Conference on COTS-Based Software Systems*, Redondo Beach, CA, USA, 2004, pp. 4-5.

[9] *SourceForge, Website, WinSCP client*, 2005, http://SourceForge.net.

[10] Miller, E. and W.W.W. Consortium, "*The W3C's Semantic Web Activity: An Update",* IEEE Intelligent Systems, Volume 19, Issue 3, 2004, pp. 95-96.

[11] *eCots, Website*, 2005, http://www.ecots.org.

[12] *ComponentSource, Website*, 2005, http://www.componentsource.com.

[13] *Flashline, Website*, 2005, http://www.flashline.com.

[14] Edgington, T., T.S. Raghu, and A. Vinze, "*Knowledge Ontology: A Method for Empirical Identification of 'As-Is' Contextual Knowledge", 38th Hawaii International Conference on System Sciences*, 2005, pp. 13-23.

[15] Uschold, M. and M. Gruninger, "*Ontologies and semantics for seamless connectivity",* ACM SIGMOD Record, Volume 33, Issue 4, 2004, pp. 58-64.

[16] McGuinness, D.L. and F.v. Harmelen, *OWL Web Ontology Language Overview*, 2004, W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1.2.

[17] *The Protégé Ontology Editor and Knowledge Acquisition System*, 2005, http://protege.stanford.edu/.

[18] Aberer, K., et al., "*Emergent Semantics Systems", First International IFIP Conference*, Paris France, 2004, pp. 14-43.

[19] Wang, P., et al., "*A novel Approach to Semantic Annotation Based on Multi-Ontologies", 3rd International Confernce on Machine Learning and Cybernetics*, Shanghai, 2004, pp. 1452-1457.

[20] Mielnik, J.-C., et al., "*Using eCots portal for sharing information about software products and the Internet and in corporate intranets", 2004 International Conference on COTS-Based Software Systems*, Redondo Beach, CA, USA, 2004, pp. 1-4.

[21] Mili, A., R. Mili, and R. Mittermeir, "*Storing and retrieving software components: a refinement based system", 16th International Conference on Software Engineering*, Sorrento, Italy, 1994, pp. 91-100.

[22] *Dedicated Systems Encyclopaedia, Website*, 2005, http://www.omimo.be/encyc/.

[23] *WikipediA, Website*, 2005, http://www.wikipedia.org/.

[24] *An Introduction to Swoogle, Website*, 2005, http://swoogle.umbc.edu/.

[25] Cilibrasi, R. and P. Vitanyi, *Automatic Meaning Discovery Using Google*, (preprint, arxiv.org/abs/cs.CL/0412098, 2004), 2004.

[26] Anyanwu, K., A. Maduko, and A. Sheth, "*SemRank: Ranking Complex Relationship Search Results on the Semantic Web", 14th international conference on World Wide Web*, Chiba Japan, 2005, pp. 117-127.

[27] Alani, H., et al., "*Automatic Ontology-Based Knowledge Extraction from Web Documents",* IEEE Intelligent Systems, Volume 18, Ussue 1, 2003, pp. 14-21.

[28] *CIDOC Conceptual Reference Model ontology, Website*, 2005, http://cidoc.ics.forth.gr/.

[29] *WordNet, Website*, 2005, http://wordnet.princeton.edu/.

[30] *GATE - General Architecture for Text Engineering, Website*, 2005, http://gate.ac.uk/.

[31] Mielnik, J.-C., et al., "*eCots platform: an inter-industrial initiative COTS related information sharing", 2003 International Conference on COTS-Based Software Systems*, 2003, pp. 157.

[32] Torchiano, M., et al., "*COTS Products Characterization", 14th international conference on Software engineering and knowledge engineering*, Ischia, Italy, 2002, pp. 335-338.

[33] Barbier, F., "*Web-Based COTS Component Evaluation", 2004 International Conference on COTS-Based Software Systems*, Redondo Beach, CA, USA, 2004, pp. 104-116.