Deriving competencies from the SWEBOK ontology

Elena Mena, María Pareja, Miguel-Angel Sicilia, Elena García-Barriocanal

Information Engineering Research Unit
Computer Science Dept., University of Alcalá
Ctra. Barcelona km. 33.6 – 28871 Alcalá de Henares (Madrid), Spain
{emg04019, mpb18336}@alu.uah.es, {msicilia, elena.garciab}@uah.es
http://www.cc.uah.es/ie

Abstract. Competencies can be used as a modeling paradigm for observable professional behavior. Competencies relate different kinds of human knowledge to practical work situations, and thus the knowledge captured in representations as ontologies can be used as a source for deriving competencies. However, this requires some specific support and the definition of knowledge representations for the different kinds of human capabilities. In consequence, guides to bodies of knowledge for some disciplines can be given an alternative formulation in terms of the competencies that are considered to be required and specific to the professionals in that concrete area. This paper describes how ontologies can be used as a vehicle for modeling disciplines, using as a case study the discipline of Software Engineering.

Keywords: Software Engineering, SWEBOK, ontologies, competencies, knowledge.

1 Introduction

The notion of competency is linked to the concept of *human performance*, which according to the model of Rummel (Rothwell & Kazanas, 1992) encompasses several elements: (1) the work situation is the origin of the requirement for action that puts the competency into play, (2) the individual's required attributes (knowledge, skills, attitudes) in order to be able to act in the work situation, (3) the response which is the action itself, and (4) the consequences or outcomes, which are the results of the action, and which determine if the standard performance has been met. Ontologies of competencies provide a formal model for that notion of competency (Sicilia, 2005).
As competencies are related to professional practice, they can be used to define the scope and requirements of established and mature disciplines. The requirements for a given discipline can be extracted from curricular guidelines, books and specialized journals, or even from the design of educational or training programs. However, to come up with a comprehensive set of guidelines for a discipline, the sources need to have some process of evaluation in order to obtain a proper definition of the field. The concept of "body of knowledge guide" fulfills that requirement, especially when the guide has been elaborated from a process of iterative refinement and broad,

worldwide evaluation. The SWEBOK Guide[1] for Software Engineering is an example of that kind of artifacts.

Ontologies of competencies can then be used as the formal schema for crafting competency-based formal definitions from bodies of knowledge. This is the focus of the research outcomes reported here.

The `<Onto-SWEBOK>` project is an attempt to engineer a formal ontology that covers the results of the elaboration of the SWEBOK Guide (Abran et al., 2006). In doing so, the resulting ontology will encompass the consensus achieved in the SWEBOK project, but providing formal semantics that are useful to build knowledge-based applications. Ontologies have been used in addition for several applications in the domain of Software Engineering (Calero et al., 2006). Since the SWEBOK aims at compiling the knowledge and competence required for the *professional* activity of Software Engineers, a convenient approach to give propositional content to the guide a practical form would be that of expressing it in terms of the professional competencies that are supposed to be required on Software Engineers – the technical ones, since the SWEBOK guide focus on specific knowledge.

This paper reports on the method used to derive competencies from the results of the `<Onto-SWEBOK>` project. The method used can be applied to the domain of other professional disciplines, and it also may serve as a way to link bodies of knowledge to learning processes and resources based on competencies (Naeve and Sicilia, 2006).

The rest of this paper is structured as follows. Section 2 describes the main aspects of the <Onto-SWEBOK> ontology with an emphasis on activities and processes. Then, Section 3 reports on the method and techniques used to derive formal competency definitions. Finally, Section 5 provides conclusions and future research directions.

2 General structure and faceted approach of <Onto-SWEBOK>

The method to extract the ontological elements from the SWEBOK was first described in (Abran et al., 2006). That method was first used for a fragment of the guide as a test for the overall procedure. The complete translation into ontological terms additionally included a pre-arranged way of referencing the source. Concretely, the references were in the form:

`#swebokref:`*`predicate`*`:`*`parameter(s)`*`:`*`text-pointer`*`#`

For example, the following reference associated to the concept `SoftwareEngineer`:

`#swebokref:superconcept-rationale:oc_Professional:1.pre#`

Means that the rationale for declaring `SoftwareEngineer` as a subconcept (subsumed by) the concept `oc_Professional` is based the *previous* text of Chapter 1 of the SWEBOK Guide[2] (referred to as "the *source*"). This is a way of approaching literature-based annotation (Sicilia, García-Barriocanal, Díaz and Aedo,

[1] http://www.swebok.org/

[2] The version used is the 2004 version.

2003). This referencing is useful as a mechanism to assess the decisions taken in identifying competencies, since it serves to trace back to the original text.
Despite the straightforward structure of the ontology that follows the Knowledge Areas (KA) defined in the SWEBOK Guide, the form of the ontology can be structured around the *OpenCyc* basic theories reused. Concretely, Table 1 summarizes the main elements of that structure.

Core elements (oc)	**Description**	**Examples**
`IntelligentAgent`	Workers and teams that reflect the organization of engineering work.	`SoftwareEngineer` `ProjectTeam`
`PurposefulAction`	The different engineering activities	`UnitTestActivity` `ChangeManagementActivity`
`Artifact`	The different tangible outcomes of the activities	`RequirementsDocument` `SourceCodeModule`
`Specification`	Models and specifications in general	`RequirementSpecification` `PerformanceRequirement`
`Process`	Chains of activities that serve a higher-level aim.	`ChangeManagmentProcess`

Table 1. General structure for the SWEBOK ontology

Activities are the main elements to be considered for identifying competencies, since they capture the dynamics of workers acting to obtain concrete tangible artifacts.

3 Identifying competencies from <Onto-SWEBOK>

The GCO ontology

The `Competency` concept represents a discrete competency of an individual generally portrayed as processors. The elements influencing competencies are of a various kinds, including knowledge, skills, abilities, and also attitudes. By using these concepts a clear separation about three types of traits that represent different aspects of competency is clearly achieved.

For example, an employee may have the knowledge about the different phases of a given internal process, since he or she has attended trainings about it. This is different than having the skill of implementing the process correctly. In fact, the knowledge about the internals of the process may not be necessary for its proper usage, and on the contrary, knowing the internals does not guarantee that the employee is able to use the process efficiently. In addition to that, attitudes represent elements that are not necessarily connected to specific knowledge or skills. For example, having good influencing skills does not always entail that an employee would have the attitude to make his/her opinion prevail. The part of the current version of the ontology that models competencies and competencies definitions is depicted in figure 1. For the sake of clarity, not all the ontology properties are shown.

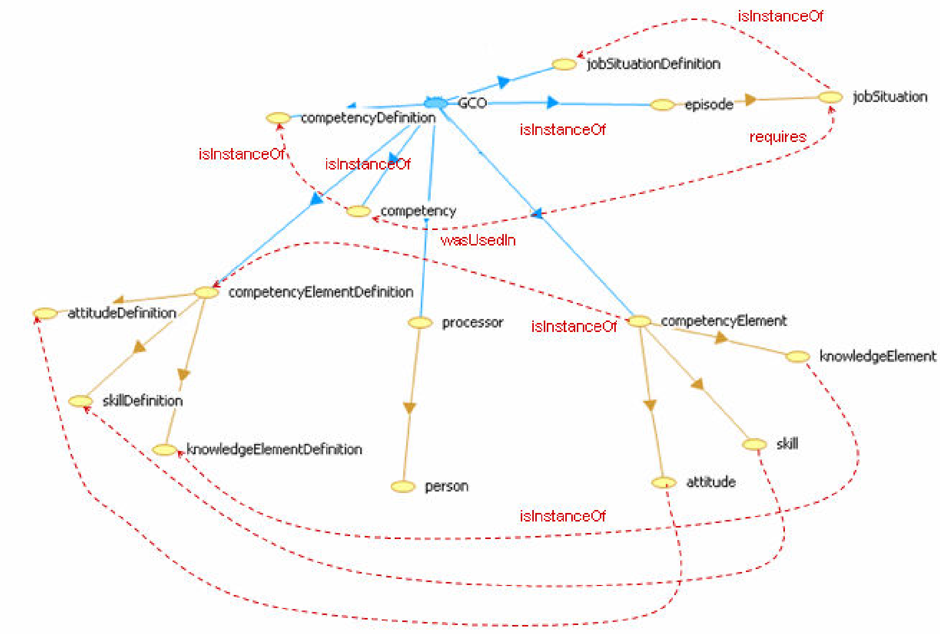


Figure 1. Partial graphical view of the ontology: Competencies and Competencies definition

The following criteria were used as the general mapping guidelines for the competencies:

- Competencies are observable, measurable SE activities that produce or change at least one artifact, adding some value to the previous state of things.
- The breakdown in finer granularity competencies should follow the same principle.
- Competency elements are either specification of chunks of declarative knowledge (KnowledgeItemDefinition) or tasks considered procedural knowledge (SkillDefinition). Attitudes were not used in this version.

Figure 2 summarizes the mapping of project LUISA general competency ontology (GCO) to elements in <Onto-SWEBOK>. The dependencies represent the mapping used. In the case of procedural knowledge, the mapping is often direct to some kind of SE activities, and the main decision to be taken is the difference between a skill and a competency. In the case of declarative knowledge, there are two options. In some cases, the knowledge items represent the internal, cognitive structure of workers for some given instances, for example, knowing on the different `FunctionalSoftwareEstimation` methods. However, in some cases these items should refer to concepts instead of instances, and a sort of reification is required, with a similar technique as that described in (Sicilia, 2003). This does not represent simply a workaround for representational purposes, but the fact that in describing competencies it is required to represent some form of cognitive structures that are ontologically different from the real-world objects they convey.

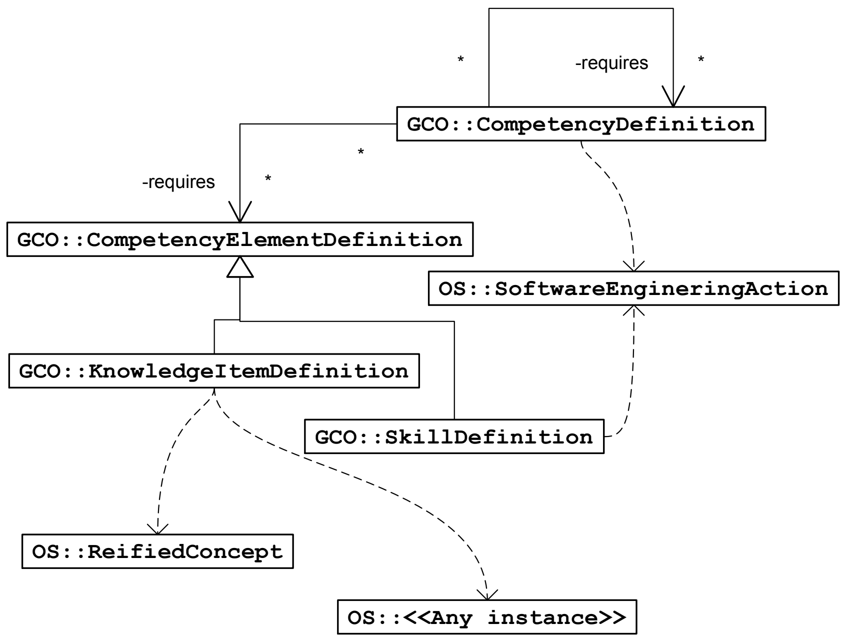
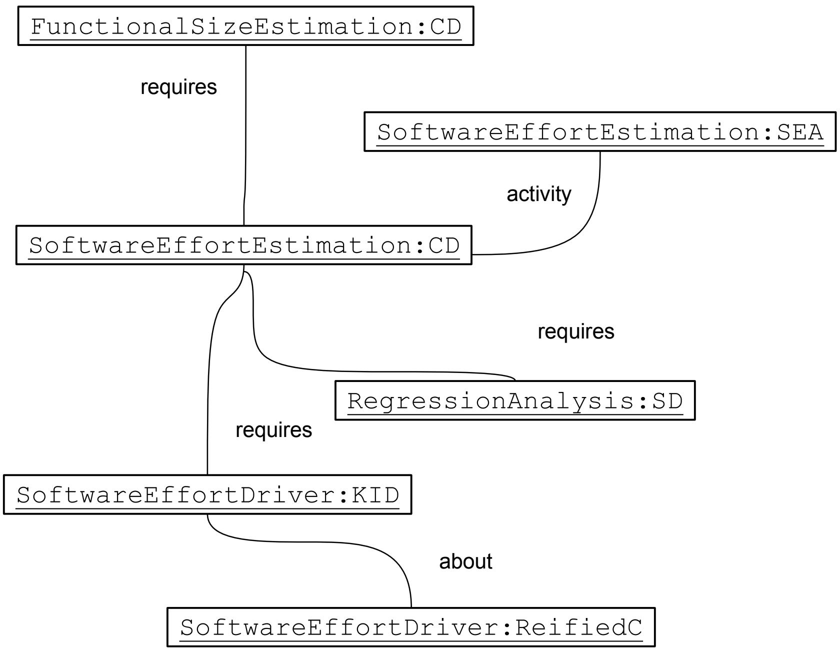


Figure 2. Mapping the GCO and <Onto-SWEBOK>

The following is an example of a given concrete competency[3] modeling `SoftwareEffortEstimation` in which estimation is done by regression on a historical database of projects. Other related descriptions could be provided for concrete software effort estimation methods, which may serve to capture the way that task is done in a given organization.

---

[3] The level of detail in competency description might vary depending on the purpose of the ontological engineering effort, the example here reflects a concrete position intended to be roughly coherent with the level of description that is provided in the SWEBOK Guide.



The above approach delineates the principal aspects of the mapping. However, not any reified concept or instance is subject to become a knowledge item, and something similar happens with classes of software engineering actions (in this latter case, also there is no automated way to decide on the difference between *skills* and *competencies*. In consequence, there is a need for annotation.

The following pseudocode (the notation is informal) sketches a possible direct mapping algorithm for the conversion. The pseudocode maps instances in an ontology `o` that is supposed to contain descriptions of `PurposefulAction` categories (in our example, of `SoftwareEngineeringActions`). The idea is that there will be annotations in the ontology that specify the kind of GCO entity that could be used for the conversion. This is denoted in the pseudocode as properties as in `item.ki` which means that the concept or instance is annotated to be a potential definition of a knolwedge item. This can be done in OWL in a non-intrusive way by using annotations outside the language. For practical purposes, we have used the `rdfs:seeAlso` property that is available directly in the user interface of Protégé editor[4]. However, this mechanism is not compliant with the original purpose of the `seeAlso` property, and a better solution would be that of developing a schema for competency-specific meta-annotations that could be combined with the OWL markup.

```
procedure convert(o:Ontology, gco:GCOOntology)
  for each (concept|instance) item in o with item.ki do
```

[4] http://protege.stanford.edu/

```
begin
    gco.newInstance(item, KnowledgeItemDef);
    if (item.hasPartOf())
        gco.createNewPartOfRelations(item);
end;

for each (concept subsumedBy
                 PurposefulAction) in o
                          item with item.sk do
begin
   gco.newInstance(item, SkillDef);
end;

for each (concept subsumedBy
                 SoftwareEngineeringAction) in o
                          item with item.comp do
begin
   gco.newInstance(item, CompetencyDef);
   if (item.hasPartOf())
        gco.createNewPartOfRelations(item);
   if (item.hasComponents())
        gco.createNewComponentRelations(item);
end;
```

The above pseudocode can be easily translated to a conversion program using libraries as HP-Jena[5]. Obviously, the algorithm is only a first attempt since more detailed mappings could be devised. For example, the notion of "process" could be used as another source for competency descriptions.

4 Conclusions and outlook

The current 0.1 version of the <Onto-SWEBOK> ontology represents a milestone to provide the formal semantics in the form of axioms and rules that are not explicit in the text. It also provides the first approach to deriving a competency framework from the SWEBOK. The competency view reported here complements the original concept-based approach with an emphasis on observable professional behavior, which is especially useful for applications as project team building, human resource assessment or for the programming of learning activities inside organizations using a "competency gap" approach. The approach for deriving competencies fosters reuse of conceptualizations and allows for the synchronization of domain ontologies with corpus of competency definitions.

[5] http://jena.sourceforge.net/

Future work will include the continuous revision of the ontology engineered, and it will also attempt to integrate other potential sources of information for professional competencies as the ACM/IEEE curricula[6].

5 Acknowledgements

This work has been supported by project LUISA (Learning Content Management System Using Innovative Semantic Web Services Architecture), code FP6−2004−IST−4 027149 and by project SERE-OC (Resource selection in e-learning based on competency ontologies), code CCG06-UAH/TIC-0606 funded by UAH and the CAM (Comunidad Autónoma de Madrid).

References

1. Abran, A., Cuadrado-Gallego, J.J., García-Barriocanal, E., Mendes, O., Sánchez-Alonso, S. and Sicilia, M.A. (2006). Engineering the ontology for the Swebok: Issues and techniques. In Calero, C., Ruiz, F. and Piattini, M. (Eds.), *Ontologies for software engineering and software technology* (pp. 103-122). New York: Springer.
2. Calero, C., Ruiz, F. and Piattini, M. (Eds.), *Ontologies for software engineering and software technology*. New York: Springer.
3. Naeve, A., Sicilia, M-A., (2006), Learning processes and processing learning: From Organizational Needs to Learning Designs, Presented at the ADALE workshop, 20 June, 2006, in conjunction with the Adaptive Hypermedia conference (AH'06) in Dublin, 20-23 June 2006.
4. Rothwell, W., & Kazanas, H. (1992). *Mastering the instructional design process*. San Francisco, CA: Jossey-Bass.
5. Sicilia, M.A. & García-Barriocanal, E. Aedo, I. and Díaz, P. (2003). A literature-based approach to annotation and browsing of Web resources . Information Research 8(2) 4(2): 1-10.
6. Sicilia, M. A. (2005), Ontology-based competency management: Infrastructures for the Knowledge Intensive Learning Organization, In Lytras, M. D. and Naeve, A. (Eds.), Intelligent Learning Infrastructure for Knowledge Intensive Organizations: A semantic web perspective (pp. 302-324), Information Science Publishing, Idea Group Inc.

.

[6] http://www.acm.org/education/curricula.html