

An Evolutionary Ontology Approach for Community-Based Competency Management

Peter De Baer, Robert Meersman, and Gang Zhao

STARLab, Vrije Universiteit Brussel, Pleinlaan 2,
1050 Brussels, Belgium
{pdebaer, meersman}@vub.ac.be, Gang.Zhao@intelartes.com

Abstract. In this article we describe an evolutionary ontology approach that distinguishes between major ontology changes and minor ontology changes. We divide the community in three (possibly overlapping) groups, i.e. facilitators, contributors, and users. Facilitators are a selected group of domain experts who represent the intended community. These facilitators define the intended goals of the ontology and will be responsible for major ontology and ontology platform changes. A larger group of contributors consists of all participating domain experts. The contributors will carry out minor ontology changes, like instantiation of concepts and description of concept instances. Users of the ontology may explore the ontology content via the ontology platform and/or make use of the published ontology content in XML or HTML format. The approach makes use of goal and group specific user interfaces to guide the ontology evolution process. For the minor ontology changes, the approach relies on the wisdom of crowds.

Keywords: Evolutionary ontology, competence ontology, competency management, human resource management.

1 Introduction

In this article we describe an evolutionary ontology approach to support community-based competency management across domains and organizations. During past projects, i.e. PoCeHRMOM¹ and CODRIVE², VUB STARLab developed competence ontologies [2] and community-driven ontology engineering tools to support competency-based human resource management (HRM) [9]. During the PROLIX project³ (Process Oriented Learning and Information Exchange), we further investigate the requirements of an evolutionary ontology platform for competency description and competency-based qualification description. The main goal of the intended platform is to enable the broad community of training and certification providers to publish the qualifications they offer, and describe these qualifications in function of required initial competencies and expected end competencies. The main

¹ PoCeHRMOM info: <http://starlab.vub.ac.be/website/PoCehrMOM>

² CODRIVE info: <http://starlab.vub.ac.be/website/codrive>

³ PROLIX website: <http://www.prolixproject.org/>

difference with the methodology [1] and platform [6] that was used during the CODRIVE project is that we now broaden the community across domains. The new methodology and platform should also support more dynamic changes of the concept definitions over longer periods of time.

In the next sections we will describe the communities involved (2), the structure of the evolutionary ontology (3), the user roles and use case diagrams for the evolutionary ontology platform (4), the publication and use of the validated ontology content (5), and the meta-data to support tracking of ontology changes and the publication of validated content (6).

2 Community

We divide the community involved in the development and management of the evolutionary ontology into three groups: facilitators, contributors, and users. The facilitators are the partners in the PROLIX project who represent organizations that:

- a) Provide training materials and/or training software e.g. the Social Care Institute for Excellence, Ernst Klett Verlag, Giunti Labs, imaginary srl, and IMC AG.
- b) Use competency-based HRM in their organizations, for example British Telecom, or (want to) support competency-based HRM via their business process management software, for example IDS Scheer and QPR Software Plc.
- c) Research and develop approaches and software for information, knowledge, and ontology management e.g. Forschungszentrum L3S and VUB STARLab.

The facilitators analyzed the needs of the community for competency information related to qualifications. Taking the different perspectives into consideration the evolutionary ontology as described in section 3 and the evolutionary ontology platform as described in section 4 could be distilled. Note: This group is comparable to the central board in the DILIGENT methodology [11].

All training and certification providers are potential contributors to the ontology content. To make it easy for such organizations to contribute to the ontology we believe it is important to provide a set of user interfaces that guide the user when describing competences and qualifications. An example of such a user interface for semantic qualification description is illustrated in Fig. 1. Using specific ontology engineering interfaces distinguishes the proposed solution from other collaborative ontology engineering approaches, e.g. Collaborative Protégé [7] and HCOME-30 [10]. These user interfaces could be discussed, developed and tested by the facilitators before opening them to the entire community.

The users of the ontology content may be organizations and individuals in search for suitable training, i.e. qualifications, based on competency information. This will be further discussed in section 5.

3 Evolutionary Ontology

The goal of the ontology is that organizations should be able to describe qualifications in function of required initial competencies and expected end competencies. This

scope is more specific than related work in the CommOn framework [8], however, it allows broadening the community to different domains. The structure for the ontology now includes the following concepts: competence, competence level, competency, context, qualification, and qualification level.

- A *competence* is a generic competence that may be used across organizations and domains.
- A *competence level* indicates the proficiency level of a person for competences. Such competence level may be used across organizations and domains. Examples are the reference levels in the Common European Framework of Reference for Languages⁴. Each competence level has a minimum and maximum value between 0 and 1 to enable inter-organizational comparison. If an organization uses four competence levels A, B, C, and D, competence level A might, for example, have the minimum value 0 and the maximum value 0.25, i.e. $1/4$.
- A *competency* is a specific competence that includes a certain competence level and optionally a context. For an in-depth discussion of the difference between a competence and a competency we refer to [3].
- A *context* is a category that may help to contextualize competences. For example the context BT would indicate that the competence should be interpreted within the context of that organization. For an in-depth discussion of context dependency management we refer to [4, 5].
- A *qualification* is a training certificate issued by an organization that certifies a set of competencies for a person. The qualification should correspond to a set of initial competencies required to start the training, and a set of expected end competencies after the training.
- A *qualification level* is a level that indicates the difficulty level of a qualification. Examples are the European Qualification Levels⁵. Each qualification level has a minimum and maximum value between 0 and 1 to enable inter-organizational comparison.

Instances of the above described concepts may be added, described, and deleted by the training and certification community. English could be used as the Interlingua language on the platform; however, the platform should also allow adding multilingual terminology to describe concepts and concept instances. The addition of multilingual terminology could facilitate the adoption of the ontology content in an international setting.

In the CODRIVE project, ontology templates were developed by core domain experts and further specified by domain experts [6]. The design and management of a goal-oriented core ontology and ontology platform, by the group of facilitators, is an extension to that approach. The new approach further relies on the wisdom of crowds to create, update and verify the ontology contents. This new approach to evolutionary ontology engineering will be described in the next section.

⁴ <http://tinyurl.com/2k7zko>

⁵ <http://tinyurl.com/agw657>

4 Evolutionary Ontology Platform

We identified five different types of actors that may interact with the ontology platform: (1) a potential user, (2) a user that only consults the platform, (3) a contributor of content on the platform, (4) a validator of content on the platform, and (5) an administrator of the platform. Each administrator is also a validator, a validator is also a contributor, and a contributor is also a user.

We also identified fifteen use cases of how the actors might use the ontology platform. For these use cases we designed the following use case diagrams:

Table 1. Register new user

Actor	Action
Potential user	A potential user registers on the platform. He provides his e-mail address, username, password, and optionally a comment to explain why he wants to use the platform.
Ontology platform	The list with 'New user' requests is available to all validators.
Validator	A validator approves the new user (a) or not. a. The new user is added. The 'New user' request is deleted.
Ontology platform	The platform sends the potential user an e-mail that states whether the request is approved or not. If approved, the e-mail contains the username and security code to login on the platform.
User	The user uses his username, security code, and password to login.
Ontology platform	The platform verifies username, security code, and password. If valid, the user may browse the ontology and the security code protection is removed.

Table 2. Change password or e-mail address

Actor	Action
User	The user logs in on the platform with his username and password.
Ontology platform	The platform verifies username and password. If valid, the user is logged in.
User	The user may change his password and/or e-mail address.

Table 3. Ask password or username

Actor	Action
User	The user provides his username and/or e-mail address, and asks to send his password or username.
Ontology platform	The platform verifies username or e-mail address. If valid the password or username is sent to the e-mail address of the user.

Table 4. Read content

Actor	Action
User	The user logs in on the platform with his username and password.
Ontology platform	The platform verifies username and password. If valid, the user may browse the ontology.

Table 5. New contributor

Actor	Action
User	A user requests to become a contributor.
Ontology platform	The list of ‘New contributor’ requests is displayed to all validators.
Validator	A validator approves the new contributor (a) or not. a. The new contributor is added. The ‘New contributor’ request is deleted.
Ontology platform	A message is sent to the user with the result of the request.

Table 6. Create content

Actor	Action
Contributor 1	Contributor 1 creates new content.
Ontology platform	A list of unvalidated new content is displayed to all contributors.
Contributor 2	Contributor 2 approves the new content (a) or not (b). a. The new content is validated. b. The new content is removed.

Table 7. Update validated content

Actor	Action
Contributor	A contributor updates validated content.
Ontology platform	A list of unvalidated updates is displayed to all validators.
Validator	A validator approves the update (a) or not (b). a. The update is validated. b. The update is removed.

Table 8. Delete validated content

Actor	Action
Contributor	A contributor deletes validated content.
Ontology platform	A list of unvalidated deletes is displayed to all validators.
Validator	A validator approves the delete (a) or not (b). a. The delete is validated. b. The delete is undone.

Table 9. New validator

Actor	Action
Contributor	A contributor requests to become a validator.
Ontology platform	A list of ‘New validator’ requests is displayed to all administrators.
Administrator	An administrator approves the request (a) or denies the request. a. The new validator is added. The ‘New validator’ request is deleted.
Ontology platform	A message is sent to the contributor with the result of the request.

Table 10. New admin

Actor	Action
Validator	A validator requests to become an administrator.
Ontology platform	A list of ‘New administrator’ requests is displayed to all administrators.
Administrator	An administrator approves the request or denies the request.
Ontology platform	Counters keep track of the number of approvals and denials. A flag will indicate whether the logged-in administrator already voted. If the number of approvals minus denials is higher than half the number of administrators, the new administrator is added and the ‘New administrator’ request is deleted. Send result. If the number of denials reaches a quarter of the number of administrators, the ‘New administrator’ request is deleted. Send result. <i>Send result:</i> The result of the request is sent to the validator.

Table 11. Revoke admin

Actor	Action
Administrator	An administrator may request to revoke an administrator.
Ontology platform	A list of ‘Revoke administrator’ requests is displayed to all administrators.
Administrator	An administrator approves the request or denies the request.
Ontology platform	Counters keep track of the number of approvals and denials. A flag will indicate whether the logged-in administrator already voted. If the number of approvals minus denials is higher than half the number of administrators, the administrator is revoked and the ‘Revoke administrator’ request is deleted. Send result. If the number of denials reaches a quarter of the number of administrators, the ‘New administrator’ request is deleted. Send result. <i>Send result:</i> The result of the request is sent to both the requesting administrator and to the administrator that was the subject of the request.

Table 12. Revoke validator

Actor	Action
Administrator	An administrator may revoke a validator that is no administrator.
Ontology platform	A message is sent to the validator that he is now only a contributor.

Table 13. Revoke contributor

Actor	Action
Validator	A validator may revoke a contributor that is no validator.
Ontology platform	A message is sent to the contributor that he is now only a user.

Table 14. Delete user

Actor	Action
User	A user may delete his user account.
Administrator	An administrator may delete a user that is no contributor.
Ontology platform	A message is sent to the user that his account was deleted.

Table 15. Publish validated ontology content

Actor	Action
Administrator	An administrator may publish the validated ontology content in XML format. He has to specify a valid location.
Ontology platform	The validated ontology content is published in the specified location. The ontology meta-data will be updated (see section 6).

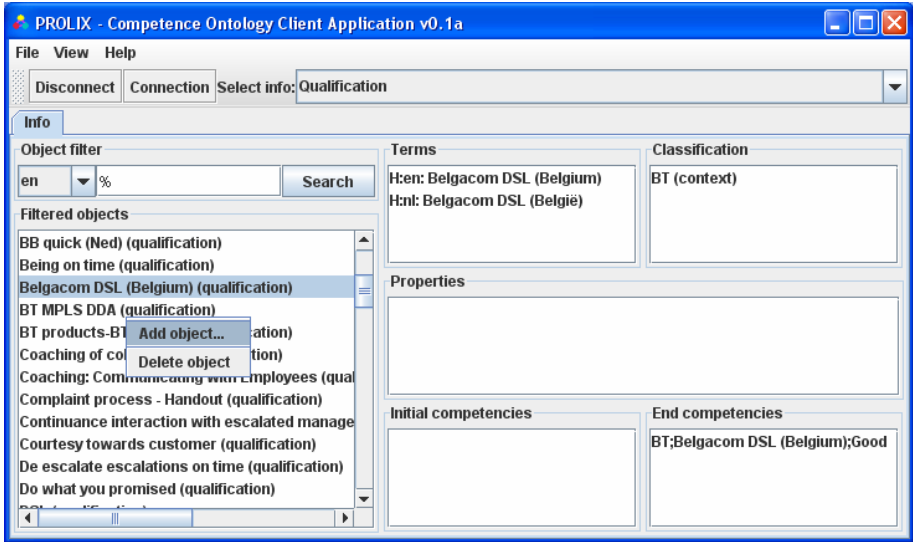


Fig. 1. A contributor may use a client application to add and describe qualifications with terminology, classification information, and properties. A *qualification* may have a set of initial competencies and a set of end competencies.

The goal of the different user roles and use case diagrams is to make the ontology platform scalable and sustainable over time. This could be achieved by a separation of concerns so that users are shielded from complexity and are able to specialize themselves in their roles. A set of activity and role specific user interfaces also guides the user to manage the ontology. For an example of how the platform could be used to describe a qualification see Fig. 1. Depending on the user role and user activity, the platform should limit the possible user actions. The platform should also check the consistency of the ontology changes. For example, an expected end competency of a qualification should not be lower than the required initial competencies to start the training. Notable differences between our approach and other evolutionary ontology engineering approaches, for example [7] and [10], are that we mainly rely on natural language to capture the knowledge and use specific applications to guide the user during the conceptualization.

5 Publication and Use of the Ontology Content

Although the ontology platform allows a broad community to explore and maintain the ontology, we would also include the option of publishing the validated ontology content as XML pages on the web. The advantage of such static XML pages is that software systems could easily make use of the information. For the PROLIX project, organizations could, for example, extract the list of competences for internal use, or filter the qualifications that provide the required competences for their organization.

The use of static XML pages is also less demanding of the ontology platform, since organizations may download the information for (offline) custom querying. We would also provide XSL Transformations⁶ to transform the XML pages into HTML pages. This would allow end users, for example, HRM managers and job candidates, to browse the ontology content in their web browser without even registering on the ontology platform. For an example of how the ontology content might be presented as HTML pages, see Fig. 2.

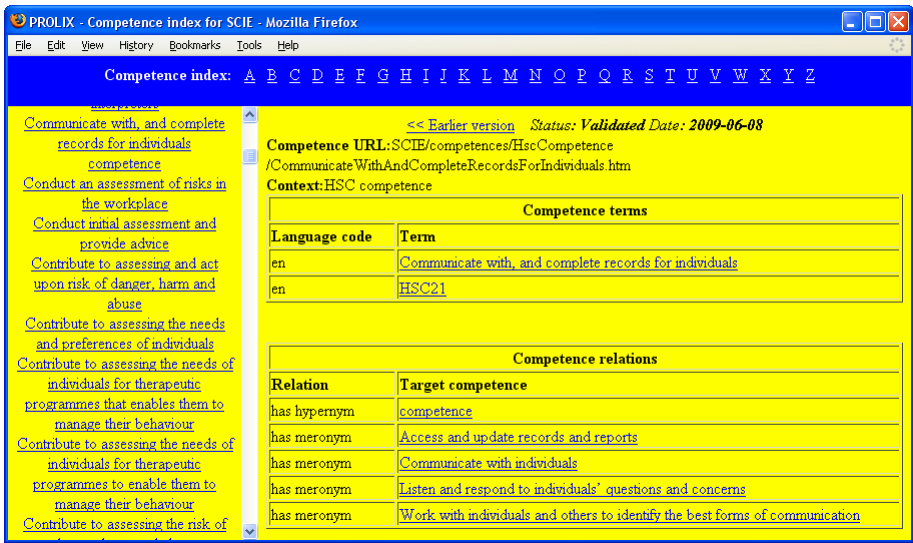


Fig. 2. Validated ontology content could be presented as static HTML pages

Publishing the ontology as static XML or HTML pages makes it easier to keep track of ontology changes (see also section 6). If a concept, that was already published, is modified, validated, and again published, a new concept version would be created. The concept list would refer to the latest concept version and the status of the latest concept version would have a link to the previous concept version. Moreover, the status of the previous concept version would change to *Updated* and have a link to the new concept version.

Each concept version in the published ontology has a unique URL which makes it possible for end users to refer to the concept. For example, a job candidate could refer

⁶ <http://www.w3.org/TR/xslt>

(in a portfolio) to the URLs that describe his list of qualifications and competences. An HRM manager (possibly a software system) could then better evaluate this information.

6 Meta-data in the Evolutionary Ontology

Content in the ontology may be divided into two major categories, i.e. ontological and terminological information. Ontological information includes the concepts, concept properties, and concept relations. Terminological information includes the terms, term properties, and term relations. Terms are the language specific character strings that denote a certain concept.

To enable the above described use case diagrams and publication of the ontology it is necessary to add meta-data to the ontology so that content changes may be tracked, validated, and published. For each data element, i.e. concept, concept property, concept relation, term, term property, and term relation, the following meta-data should be recorded:

- Date and timestamp of creation, modification, or deletion.
- User ID that created, modified, deleted, or validated the data.
- Status that indicates the validity of the data. An overview of the different states and the possible transitions between two states is illustrated in Fig. 3.
 - Created: indicates that the data was newly created.
 - Updated: indicates that the data was updated.
 - Deleted: indicates that the data was marked for deletion.
 - Create validated: indicates that the new data was validated.
 - Update validated: indicates that the update was validated.
 - Published: indicates that the validated data was published.
 - Delete validated: indicates that the delete was validated.
 - Delete published: indicates that the validated delete was published.

For the entire ontology the following meta-data should be recorded:

- Date and timestamp of publication
- User ID that published the validated ontology content.
- Location of the latest published ontology content.

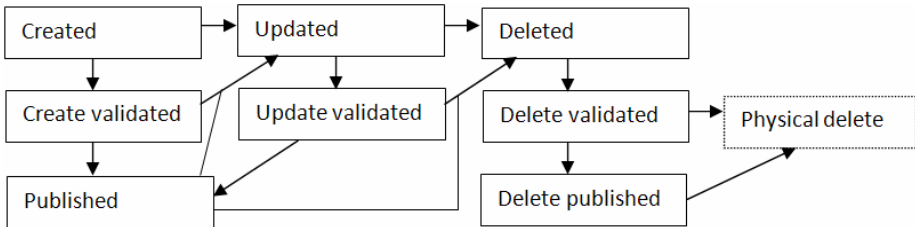


Fig. 3. The different meta-data states of ontology content and the possible transitions between two states

7 Conclusion and Future Work

We described an evolutionary ontology approach that distinguishes between major ontology (platform) changes and minor ontology changes. We divide the community into facilitators, contributors, and users. Facilitators are responsible for major ontology (platform) changes. Contributors are responsible for minor ontology changes. The wisdom of crowds approach is followed to manage minor ontology changes. Users may explore the ontology via the ontology platform and/or make use of the published ontology content in XML or HTML format.

Currently, this approach is being tested for community-based competency and qualification description. In the future we would like to further develop and evaluate the approach in other ontology engineering projects.

Acknowledgments. The research described in this paper was sponsored in part by the EU IP 027905 Prolix project.

References

1. Christiaens, S., De Leenheer, P., de Moor, A., Meersman, R.: Business Use Case: Ontologising Competencies in an Interorganisational Setting. In: *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*, from *Semantic Web and Beyond: Computing for Human Experience* (2008)
2. Christiaens, S., De Bo, J., Verlinden, R.: Competencies in a Semantic Context: Meaningful Competencies. In: *Proceedings of the OntoContent workshop*, Montpellier, France (November 2006)
3. De Coi, J.L., Herder, E., Koesling, A., Lofi, C., Olmedilla, D., Papapetrou, O., Siberski, W.: A Model for Competence Gap Analysis. In: *Proceedings of the First European Conference on Technology Enhanced Learning, EC-TEL* (2006)
4. De Leenheer, P., de Moor, A., Meersman, R.: Context Dependency Management In *Ontology Engineering: a Formal Approach*. In: Spaccapietra, S., Atzeni, P., Fages, F., Hacid, M.-S., Kifer, M., Mylopoulos, J., Pernici, B., Shvaiko, P., Trujillo, J., Zaihrayeu, I. (eds.) *Journal on Data Semantics VIII. LNCS*, vol. 4380, pp. 26–56. Springer, Heidelberg (2007)
5. De Leenheer, P., de Moor, A.: Context-driven Disambiguation in *Ontology Elicitation*. In: Shvaiko, P., Euzenat, J. (eds.) *Context and Ontologies: Theory, Practice and Applications*, AAAI Technical Report WS-05-01, pp. 17–24. AAAI Press, Menlo Park (2005)
6. de Moor, A., De Leenheer, P., Meersman, R.: DOGMA-MESS: A Meaning Evolution Support System for Interorganizational Ontology Engineering. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) *ICCS 2006. LNCS (LNAI)*, vol. 4068, pp. 189–202. Springer, Heidelberg (2006)
7. Noy, N., Chugh, A., Liu, W., Musen, M.: A Framework for *Ontology Evolution in Collaborative Environments*. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) *ISWC 2006. LNCS*, vol. 4273, pp. 544–558. Springer, Heidelberg (2006)
8. Radevski, V., Trichet, F.: *Ontology-Based Systems dedicated to Human Resources Management: an application in e-recruitment*. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops. LNCS*, vol. 4278, pp. 1068–1077. Springer, Heidelberg (2006)
9. Tang, Y., Christiaens, S., Kerremans, K., Meersman, R.: PROFILE COMPILER: *Ontology-Based, Community-Grounded, Multilingual Online Services to Support Collaborative Decision Making*. In: *Proc. of RCIS 2008* (2008); IEEE catalog: CFP0840D-PRT
10. Vouros, G.A., Kotis, K., Chalkiopoulos, C., Lelli, N.: The HCOME-3O Framework for Supporting the Collaborative Engineering of Evolving Ontologies. In: *Proceedings of the First International Workshop on Emergent Semantics and Ontology Evolution 2007*, vol. 292. CEUR-WS.org/ (2007)
11. Vrandečić, D., Pinto, S., Tempich, C., Sure, Y.: The DILIGENT knowledge processes. *Journal of Knowledge Management* 9(5), 85–96 (2005)