Ontology's Crossed Life Cycles

Mariano Fernández López, Asunción Gómez Pérez, and María Dolores Rojas Amaya

Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo s/n
Boadilla del Monte, 28660. Madrid. Spain.
Tel: +34 91 3367439, Fax: +34 91 3367412

email: {mfernand, mrojas}@delicias.dia.fi.upm.es,
asun@fi.upm.es

Abstract. This paper presents the idea that the life cycle of an ontology is highly impacted as a result of the process of reusing it for building another ontology. One of the more important results of the experiment presented is how the different activities to be carried out during the development of a specific ontology may involve performing other types of activities on other ontologies already built or under construction. We identify in that paper new intra-dependencies between activities carried out inside the same otology and inter-dependencies between activities carried out in different ontologies. The interrelation between life cycles of several ontologies provokes that integration has to be approached globally rather than as a mere integration of out implementation.

1 Introduction

A lot of ontologies have been developed in recent years in different domains and for all sorts of applications. Many of these ontologies were built from scratch, that is, without reusing definitions from other existing ontologies. Furthermore, the reuse of other ontologies has been confined to the implementation phase, and the most commonly used software environments today [2],[3], [13] admit integration of code. Additionally, there is hardly any information or documentation describing the development *process* followed to build the ontologies. In other words, the community was more interested in the end product than providing information about the Ontology Development Process (ODP), that is, the activities carried out when building ontologies in a given domain. As consequence of this situation, not many ontology development methodologies have been developed to date.

Uschold and King's methodology [14] and Grüninger and Fox [10] methodologies start by setting out the need and purpose for building the ontology. Having acquired a significant amount of domain knowledge, they propose direct codification in special-purpose ontology implementation languages. The main drawbacks of this approach are: a) conceptual models are implicit in the implementation codes; b) Domain experts and human end users have no undesrtanding of formal ontologies codified in ontology languages; c) direct coding of the knowledge acquisition result is too abrupt

R. Dieng and O. Corby (Eds.): EKAW 2000, LNAI 1937, pp. 65-79, 2000.

a step, especially for complex ontologies. So, ontologists should think simoultaneously on the analysis of the knowledge and the technology to implement such knowledge; and d) ontology-developer preferences in a given languages condition the notation and implementation of the acquired knowledge.

Methontology [4], [5], however, provides guidelines for specifying ontologies at the knowledge level as a conceptualisation that is independent of the ontology implementation languages. Methontology includes the definition of the ODP, which is based on the IEEE Standard 1074-1995 [11] for software development process; a life cycle based on evolving prototypes [5]; and the techniques that encompasses the activities identified in the ODP. Three kind of activities are identified in Methontology are [5]: *Project Management Activities* include: Planning, Control and Quality Assurance; *Development-Oriented Activities* include: Specification, Conceptualization, Formalization, Implementation and Maintenance; and *Support Activities* include: Knowledge Acquisition, Integration, Evaluation [8], Documentation and Configuration Management. The ontology life cycle identifies the set of stages through which the ontology moves during its life time, describes what activities are to be performed in each stage and how the activities are related (relation of precedence, return, etc.).

ODE [4] is an environment that gives technical support to Methontology. Ontologies can be conceptualised in ODE using tables and graphs. Ontologies are also evaluated at the conceptual level, and the translators generate the computable code. Thus, domain experts can use this approach to conceptualise new ontologies and validate domain ontologies, leading to cuts in the time spent on and resources invested in the knowledge acquisition and evaluation activities.

With the purpose of identifying new activities and obtaining new techniques, we have developed ontologies in different domains and with different representation needs. One of the more interesting ontologies, from the viewpoint of the methodological results, is the Monatomic Ions (MI) ontology. During its development, not only *intra-dependencies* inside the activities of its ODP were found, but also *inter-dependencies* between the MI ontology and activities in some of the ontologies reused by the MI ontology. By intra-dependences we refer to the relationships between activities carried out inside the same ontology, for example, the relationship between knowledge acquisition and conceptualisation in Monatomic Ions. That is, intra-dependences define the ontology life cycle. By inter-dependences we refer to the relationships between activities carried out in different ontologies. For example, the MI development involved the performance of reengineering, merge, evaluation and configuration management on other ontologies, concretely, standard-units (SU) and chemical-elements (CE) (which were existing ontologies). In this paper, we describe some results about intra-dependencies, but our attention is mainly focused in inter-dependencies.

2 Need of Environmental Ontologies

Experts in several domains, including biology, geology, chemistry, law, computing, etc., work in the field of the environmental sciences. Each expert uses a vocabulary related to one of the areas of this science, and there is neither a common terminology nor any standard to support the accurate use of each term.

There are many potential ontologies in that domain, but we have centred on environmental pollutants. An ontology of this kind must include the methods for detecting all the pollutant components in several media (water, soil, air, etc.) and the maximum permitted concentration of these components, taking into account the legislation (European and Spanish) in force. The components of pollutants are ionic. Therefore, ions are the primary entities to be taken into account, as they are indicators of environmental pollution, deterioration, etc. Background knowledge of the elements in their pure state and their properties, as well as the units of measure of some properties, are needed to represent knowledge on ionic concentrations. The ontology of pollutants aims to output a unified, complete and consistent terminology that can be used precisely, non-ambiguously and consistently in environmental applications that employ the maximum permitted concentration to detect alterations in the above-mentioned media.

3 Monatomic Ions Ontology Development Process

The MI ontology was developed within the Methontology framework and using ODE. Methontology proposes that the ontology be specified after having started knowledge acquisition. Fig. 1 presents in continuous line the *intradependencies* between the activities of the MI ontology. While knowledge is being acquired, the ontology developer builds the conceptual model, integrates the selected ontologies, evaluate the ontology under development, as well as generate the associated documentation. Note that many of these activities take place at the same time. Having completed the conceptualisation, the system would be formalised and implemented. In our framework ontologies are not formalised, as ODE has a module that maps the conceptual model to a computable model using its translators to Ontolingua, OCML and Flogic.

Related to the *interdependencies* between the activities of different ontologies (see discontinuous lines at Fig. 1), they emerged at the specification and integration activities of the monatomic ion ontology. First, at the specification of the MI ontology, we looked for candidate ontologies at the Ontolingua Server, at the Cyc server and ODE. Then, we made a preliminary evaluation the content and suitability of the candidate ontologies, and we selected the Ontolingua ontologies, which are more suitable for our purposes.

The second interdependency emerged during the MI ontology integration activity. The selected ontologies (Standard Units and Chemical Elements) (SU and CE) were reviewed in depth to assure their correctness and completeness before their integration in the MI ontology. We performed reengineering of SU, and merge, evaluation and configuration management on CE. The aim behind this was to assure that these ontologies provided a solid basis on which to incrementally develop new ontologies.

3.1 Requirements Specification

The purpose of the specification phase is to output a document that includes the purpose, level of formality and scope of the ontology, including other significant information. The starting point of the MI ontology are the ions, both anionic and

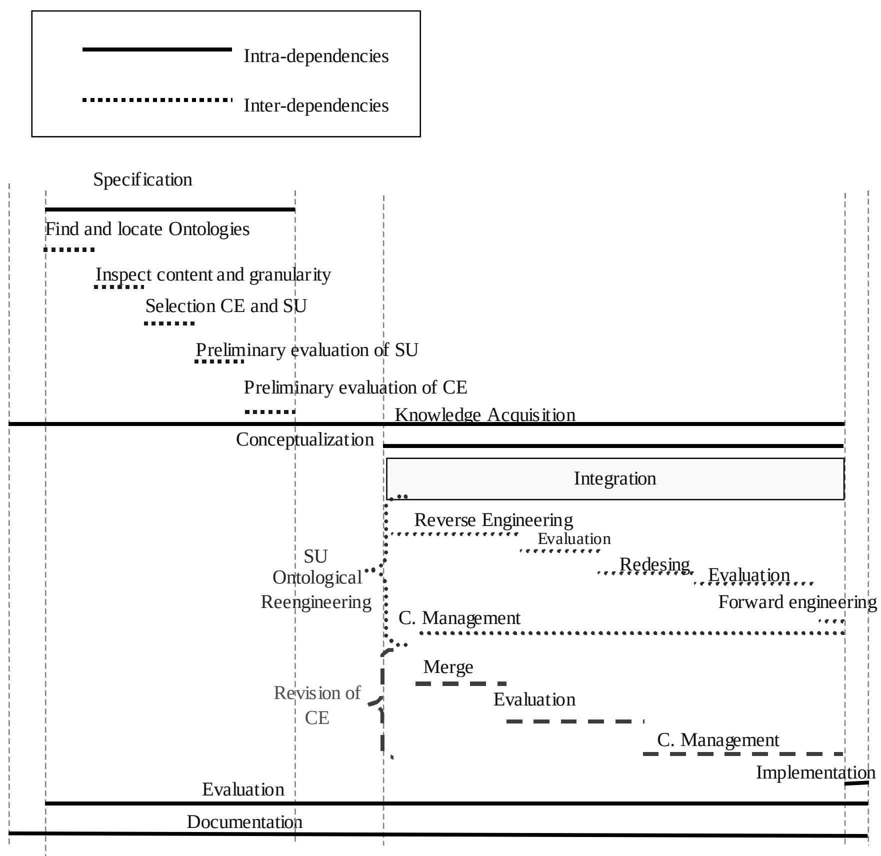

Fig. 1. Inter and intra dependencies between activities of the Monatomic Ions ODP

cationic, addressed from the viewpoint of inorganic chemistry and, also, analyzed with a view of standardization in the soil and waterfields within the physical environment and in terms of human health. From the environmental viewpoint, the monatonic ions detected in the physical variables –water, soil and air- are defined, specifying the methods of detection and maximum permitted concentrations.

It is important to mention here that at the requirements specification phase of the MI ontology we started the integration process with other ontologies. The initial activities we performed were:

- To find and locate candidate ontologies to be reused. We located the SU ontology [9], which defines the basic units of measure at the Ontolingua Server, and CE [4], which defines the chemical elements of the periodic system in their pure state, at ODE and its Ontolingua code at the Ontolingua Server. At the Cyc server, we found some units of measure and chemical entities (atom, ion, molecule and radical).
- To inspect the content and granularity of the candidate ontologies. The SU ontology at the Ontolingua Server includes for each unit: a natural language definition, its physical quantity and factors of conversion to other units of the same quantity, whereas the Cyc ontology included only a natural language definition.
- We selected the ODE and Ontolingua Server ontologies, and we used Cyc ontologies for reference purposes. Fig. 2 shows how all these ontologies, and the ontologies included by the SU and CE ontologies are related at the Ontolingua Server. Ontologies at the top include the ontologies at the bottom.
- Ontologits did a preliminar evaluation of SU from the knowledge representation point of view. As described in [7], several problems were found at the SU ontology

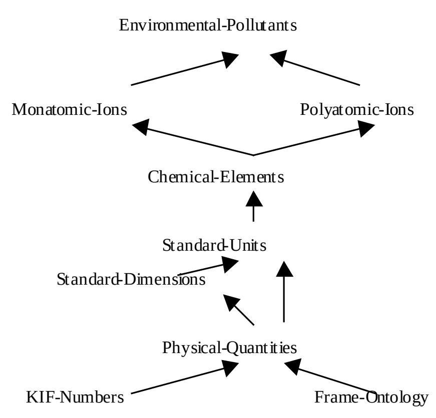

Fig. 2. Relationships between the ontologies

and CE. The most important problem in SU was the lack of taxonomic organization since all the instances were of the root class. The review process in CE showed that different versions of the ontology needed to be merged to output a new unified and corrected ontology with which could be extended before being included in the MI ontology.

- Simoultaneously with the previous evaluation, domain experts also did a preliminar evaluation of CE ontology since its conceptual model was available in ODE and was understandable for the experts. However, we have postponed SU domain experts evaluation since domain experts were unable to understand Ontolingua code.

Thus, the need of the SU preliminar evaluation forces the SU reverse engineering for obtaining its conceptual model, and the presence of several versions on CE forces its merge, evaluation and configuration management.

3.2 Knowledge Acquisition

Knowledge acquisition was performed using techniques recommended in Knowledge Engineering for developing Knowledge-Based Systems. So, two open interviews (which output a preliminary classification of ions) and six structured interviews (to get the final classification of ions, concepts, attributes, etc.) were held and informal text analysis and table analysis was conducted.

3.3 Conceptualisation

As we have already said, the conceptualisation was performed, following Methontology and using ODE. The Methontology intermediate representations used were: glossary of terms, concept classification trees, relationship diagram, table of relationships, concept dictionary, class attribute tables, logical axiom tables, and constants table. Other intermediate representations, like instance attribute tables, formula tables and instance tables, have not been used, because this ontology does not have any instance. Of all the representations, the Concept Classification Tree (CCT) deserves a special mention. Only one CCT was built from the concepts identified in the GT, which means that we have only one ontology.

Four criteria were applied when building the CCT. First, the chosen model must be easily understandable and must accurately reflect the knowledge specified by the experts. Second, the ontology must be easily extendible. Third, the ontology must be easily integrated with other ontologies. Fourth, it must be possible to select only part of the ontology for use in other ontologies or applications.

According to these objectives, ions can evidently be studied from more than one viewpoint: from the general viewpoint, which is concerned with specifying the name and symbol of the ion among other properties; from the chemical viewpoint, which is concerned with defining chemical properties; from the viewpoint of the physical environment. Any taxonomy built should enable an ion defined from the chemical viewpoint to inherit the name and the symbol that are defined for that ion from the general viewpoint. The CCT designed takes into account all these considerations. As shown in Fig. 3, the CCT is actually a graph. The benefits of this classification are:

- As the ions are defined from more than one viewpoint, it is possible to *reuse part of the knowledge gathered by the ontology*. Thus, if an application is defined strictly in the domain of chemistry, it is possible to reuse only the knowledge present in the Chemical Ion subhierarchy.
- If the water pollutant ions are required, the water ions will be selected (e.g., Cadmium (+II) in Water), as well as all the classes that link each ion to the

hierarchy root class. Note that apart from classes related to environmental ions, classes such as Cadmium (+II) and General Ion will also be included, thus making it possible to access properties which are associated with the ion in its pure state, irrespective of the viewpoint used.

- All the knowledge can be reused if it is *integrated into a higher level ontology*.
- The *ontology can be easily extended to other media and variables*, such as the human or social environment.
- *New definitions of ions can be entered from any viewpoint*. For example, if a future directive were to include any additional ion as a possible pollutant of the physical medium water, it would be easy to define this new ion and enter it as new knowledge into the ontology.

  It is just as straightforward to enter new ions from the chemical viewpoint, where all you have to do is to correctly select the group to which the above ion belongs and get the attribute values identified for the other ions from any of the referenced knowledge sources.
- By using *inheritance*, an ion in water has the properties defined for the ion and will inherit the properties defined for the above ion in the physical medium and from the general viewpoint.
- *New properties can be easily included* if required by any individual application.

The concept classification tree was verified to assure that: (a) No concepts are repeated and there are no synonyms, which rules out redundancies in the conceptual model, (b) There are no cycles [6] among concepts, (c) There are no isolated subtrees for concepts that should be related, and (d) All the concepts represented in the tree are in the glossary of terms and vice versa.

3.4 Formalisation and Implementation

No formalisation was performed, as ODE has a module that maps the conceptual model to a computable model automatically using translators.

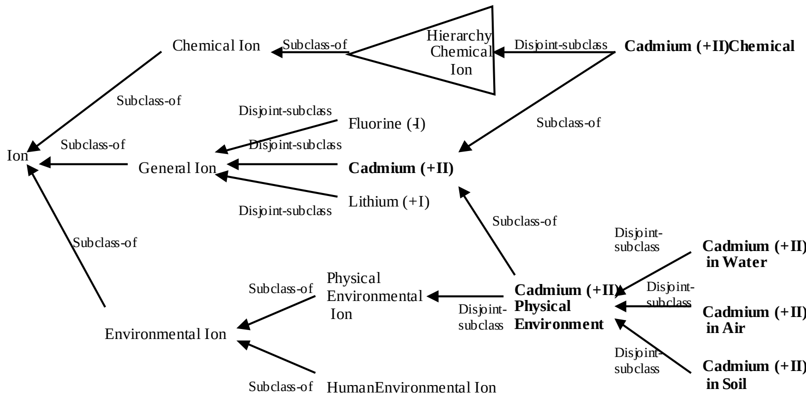

Fig. 3. CCT of thonatomic ion ontology

3.5 Integration

The objective of integration is to build an ontology by reusing definitions of knowledge present in other ontologies. Before they were reused, the following activities were carried out: reengineering on SU, and merge, evaluation and configuration management on CE. It is mainly in this integration activity where the main and stronger interdependencies appear. These *interdependencies* are shown in Fig. 1. As we said before, the main reason for reengineering the SU ontology were: (a) domain experts and human end users have no understanding of formal ontologies codified in ontology languages. So, they can not validate the content of these ontologies; and (b) the lack of taxonomic organisation and conversion factor between some units of the SU ontology. Here, we also present the process for reviewing CE to assure their suitability.

3.5.1 Ontological Reengineering on Standard Units

Ontological reengineering [7] is the process of retrieving and mapping a conceptual model of an implemented ontology to another, more suitable conceptual model, which

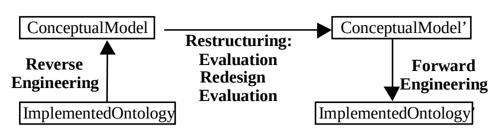

Fig. 4. Ontological reenginering process

is re-implemented. The method for reengineering ontologies is presented in Fig. 4 and adapts Chikofsky's software reengineering schema [1] to the ontology domain. Three main activities were identified: reverse engineering, restructuring and forward engineering. Fig. 5 pictures in detail an organizational chart showing the activities performed during the reengineering process and the documents generated in each step. The goal of the processes described at Fig. 4 are:

Step 1. **Reverse Engineering**. Its objetive is to output a possible conceptual model on the basis of the code in which the ontology is implemented. SU was analyzed on the basis of its Ontolingua implementation.

Step 2. **Restructuring**. The goal of restructuring is to reorganized this initial conceptual model into a new conceptual model which is built bearing in mind the use of the restructured ontology by the ontology/application that reuses it. As presented in [7], the restructuring activity contains two phases: analysis and synthesis. The analysis phase includes evaluation (steps 2 to 5 of Fig. 5), whose general aim is to evaluate the ontology, that is, to check that the hierarchy of the ontology and its classes, instances, relations and functions are complete, consistent (there are no contradictions), concise (there are no explicit and implicit redundancies) and syntactically correct. The synthesis phase (step 6 of Fig. 5) seeks to correct the ontology after the analysis phase and document any changes made. So, activities related with **configuration management** arise in that context, which goal is to keep a record of ontology evolution and strict change control.

SU was restructured bearing in mind its future use by the CE and MI ontologies. Since the reverse engineering phase provided a possible conceptual model, domain experts can now validate the SU ontology. The evaluation they performed at the conceptual model were if all the units to be used at the MI ontology were already defined at the SU ontology, as well as the factor conversions needed between the

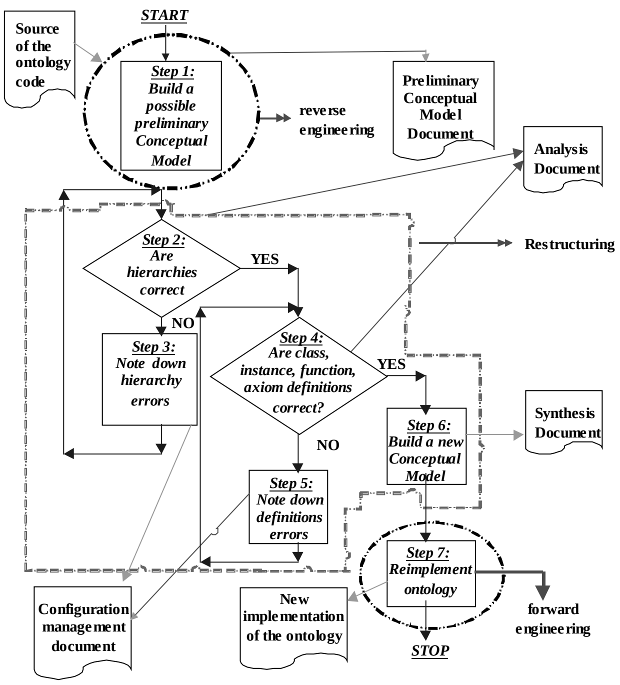

Fig. 5. Ontological reenginering activities

units. It is important to stress here that this restructuring is guided by the ontology that is to reuse the knowledge, which means that there is no way of assuring that the restructured ontology will be a hundred per cent valid for ontologies that reuse the restructured knowledge.

Step 3. **Forward Engineering**. The objective of this step is to output a new implementation of the ontology on the basis of the new conceptual model. The implementation of the new conceptual model of SU was carried out using ODE translators. Note that after doing reengineering on SU, there exist two versions of that ontology, the old version at the Ontolingua Server, and the new version in ODE.

Finally, to keep control of the changes made, we performed **configuration management** during the whole process. It is defined in software engineering as [12]: „Activity that is applied throughout the software engineering process for the following purposes: establish and maintain the integrity of the products generated, evaluate and control the system changes and provide product visibility". Adapting the idea of configuration management from Software Engineering into the ontological engineering field, on the Methontology framework, we recommend the following activities:

CM.1. **Identification of the elements to be controlled**. These elements include not only the documents related with the development of the ontology (requirement specification document, the sources used in Knowledge acquisition, conceptual models, implementation, integration documents, ontologies reused by this ontology, etc.) but also management activities (plan, quality assurance and control) and the software used to develop the ontology. To identify which are the elements to be controlled you can think what are your needs if the project stops and you re-start the project sometime later. We controlled the previous elements in CE and the mechanisms used to identify them was the concatenation of: the name of the ontology, the name of the element within the ontology, the version identifier, etc.

CM.2. **Control of changes**. Adapting Software Engineering steps to Ontological Engineering, and for each of the changes request, we propose that the Change control starts with a petition for change, followed by the classification and registration of request; approval or rejection of the change petition; report on how the change is to be made and what implications it has; order to make the change; making of change; performance of the change and certification that the change was made correctly. It ends when the result is reported to the person who proposed the change.

CM.3. **Generation of status reports**. We distinguish in that section the daily documentation generated about each configuration element and also general reports by demand (i.e., a report requested on the latest changes made to the ontology).

For the purpose of assuring information about the evolution of the Standard-Units ontology, a rigorous change control has been performed throughout the restructuring phase. The goal is to have all the changes documented, detailing the changes made, their causes and effects. It is important to perform proficient change control of both definitions and taxonomies. In this manner, any ontologist who needs to use part of or the entire ontology can easily understand its evolution. Even if an ontology has not been fully developed, provided it is well documented, it could be finished off by another developer using the existing documentation. The configuration management documents can rule out incorrect decision making, if they state the courses of action to be taken at any time, and justify the choice of one rather than another. Change control also helps end users to determine which version of the ontology they require for their system or for the new ontology they are to develop.

Consequently, although the reengineering activity and configuration management of the SU ontology would belong to the life cycle of that ontology, the MI integration activity forces the realisation of that activity on a "stable" ontology. Besides, although the SU reengineering process provokes the SU evaluation and configuration management, we stress that these activities make sense *per se*. For more information on the re-engineering process, see [7].

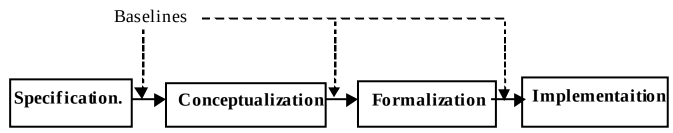

Fig. 6. Points where configuration elements are approve

3.5.2 Review of Chemical-Elements (CE)

CE is also a stable ontology whose conceptual model is available in ODE. It is also available at the Ontolingua Server. Again, the experts wanted to review the ontology before reusing it. The review process revealed that there were different versions of this ontology, which needed to be merged prior to any extension. The review process was divided into the three following activities (see Fig. 1):

- **Merging.** Chemicals started to be developed in June 1995, and the first stable version was built in December 1996. New versions of this ontology have been created since then to be used in different ontologies. Therefore, it was necessary to group all the conceptual models into one, which includes all the improvements made to the ontology.
- **Evaluation**. The knowledge present in the resulting conceptual model after merging was evaluated with the experts in order to assure that the knowledge was correct and complete, and to detect omissions.
- **Configuration Management.** Configuration management was carried out according to the guidelines described previously to make this new version of CE easier to understand for users and also to keep records of all the versions of the CE ontology.

  Configuration management activities had a strong relationship with evaluation activities. There are two reasons. First, evaluation has to be run at least after each of the phases of the ontology development process, since configuration elements can be used as a basis in the following phases of the ontology development process. Second the changes performed as a consequence of the evaluation activity need to be controlled. Fig. 6 presents the baselines on the ontology development process.

Again, although the merge, evaluation and configuration management activities belong to the life cycle of CE ontology, the MI integration activity forces the realisation of those activities on a "stable" ontology.

3.6 Evaluation

The MI ontology was evaluated by the experts throughout the entire life cycle and, especially, in the conceptualisation using ODE evaluation module.

3.7 Documentation

Documentation has been carried out throughout the whole ODP. The previous activities output: a requirements specification document; a knowledge acquisition document; the conceptual model, composed of a set of intermediate representations; an integration document; the configuration management reports, and the evaluation document.

4 Crossed Life Cycles

In the previous section we have presented the main activities carried out during the development of the MI ontology, the order of execution of such activities as well as the interdependencies with other activities that were performed in other ontologies prior to their integration on the MI ontology.

In that section we present the idea that when an ontology reuses definition of other ontologies, the ontology life cycle of the first crosses with the life cycle of the second and provokes some changes on its life cycle.

The main intersections between the life cycles of the SU, CE and MI ontologies are shown in Fig. 7. Note that the SU ontology was built at the beginning of the last decade and, probably, several applications have already used its definitions. The SU ontology life cycle was "latent" or "hibernate". That is, since the SU ontology was built, nobody has changed its definitions at the Ontolingua Server and ontologies and applications reuse that ontology as it is. When we developed CE [4], we identified some units of measures that did not appear at SU, and we added them to the SU ontology at the Ontolingua Server. We updated that ontology with the new units but we did not performed big changes in its structure and on its content. Consequently, these updates could be seen as *maintenance* activities on the SU ontology. We can really say that the life cycle of the SU "wakes up" when SU is going to be reused on the MI ontology and the reengineering process over the SU ontology starts. At this point, the SU life cycle is alive since we modified its structure and its content.

Another interesting observation in Fig. 7 is that the life cycle of the SU branches in two. So, two SU ontologies -the Ontology Server SU and the reengineered SU- were available after running a reengineering process on SU. The opposite occurs with CE, where several ontologies exist, each one with its life cycle, and they meet with the new life cycle of the merged CE ontology after the merging process.

These confluences and forking of life cycles call for a global management of ontologies. We claim that, the configuration management of each ontology must not be carried out separately from the others in which are integrated. Configuration management must be global and simultaneously affect all the ontologies handled by the group.

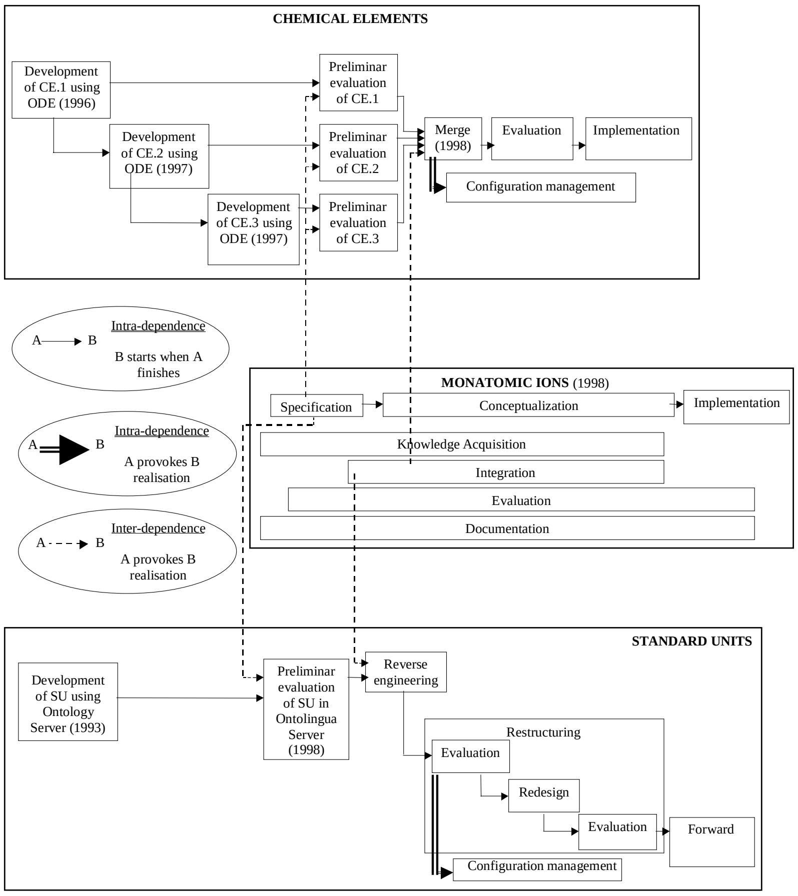

Fig. 7. Crossed life cycles of Standard Units, Chemical and Monatomic Ions

5 Conclusions

In this paper we present how the different activities to be carried out during the development of a specific ontology may involve performing other types of activities on other ontologies already built or under construction. Such activities include: reengineering, merge, technical evaluation and configuration management. So, neither integration at the implementation level nor at the knowledge level is sufficient. There is also a need to unify ontology development management policies and to integrate products output throughout the development of ontologies whose development processes are interrelated. Therefore, the life cycle of an ontology should always be documented and accessible.

We have presented the intradependencies between the activities (specification, knowledge acquisition, conceptualization, formalization, integration, implementation, evaluation, documentation) of the MI ODP. We have also presented how the different activities to be performed during the development of a concrete ontology (i.e., specification and integration activities in MI) may involve performing other activities (reengineering, evaluation, configuration managemnent) on other ontologies already built (SU and CE ontologies). The idea to consider the activities on each ontology as separate and dependant life cycles helps to understand clearly the complementarity between knowledge reuse and knowledge modelling of domain specific knowledge. Below, an assessment is given of each activity performed, specifying the contributions made from the methodological viewpoint. Also, the results obtained in each of the following areas are evaluated:

- **Knowledge reuse.** This paper presents a clear example of an ontology that reuses knowledge from other ontologies. In this case, the reuse and integration of ontologies led to the reengineering of SU and the merge, evaluation and configuration management of CE.
- **Ontology evaluation.** Three ontologies were evaluated, each by means of a different process:
  a. **Standard Units**. It is done during the restructuring phase of the ontological reengineering process.
  b. **Chemicals**. This ontology passed three evaluations. The first was a technical judgement throughout its ODP, that is, when it was built [4]. The second was the evaluation after the merging process. The third is the assessment by the experts to determine the compliance with the new MI ontology.
  c. **Monatomic Ions**. The ontology was evaluated throughout the ODP to assure that: the requirements specification was met, the knowledge represented was comparable with reality, the content of the ontology was consistent, complete and concise, etc.
- **Configuration management.** Configuration management was conducted on SU and CE as a supplementary activity to reengineering, merge and evaluation. It is important to control de changes because an ontology developer who needs to reuse that ontology (in full or in part) can easily understand its evolution.
- **Development of the MI ontology.** The conceptual model finally developed, in which MI is studied from several viewpoints, assures that the definitions are independent of the end use of the ontology. For proving this with the experience, ontologies reusing MI are been developed at present. Additionally, this ontology can be totally or partially reused in other ontologies or applications. In fact, an important purpose that we have with the development of MI and other environmental ontologies is to prove that it is possible to build reusable and usable ontologies incrementaly.

To refine the reengineering process showed in this paper, it should be desirable to apply this process to more complex ontologies than SU. However, MI is an ontology that has need several month of work, and its ODP has required not only development activities, but also management, control and support activities. Besides, for some activities (for example, evaluation), we had former experience based on the development of others ontologies. So, the more likely is that the most techniques and processes showed in this paper will need a minor refinement for being applicable to other developments.

We are scaling up the method making the necessary changes for adapting it to the development of other ontologies in the environmental field. Normally, when a new ontology is built, there are less modifications on the method than the necessary modifications for former ontologies, so there should be a moment in which the method is general enough to be applied to the development of a large quantity of ontologies.

Acknowledgements. This research is funded by the Polytechnic University of Madrid, under the multidisciplinary research and development project grants programme, reference no. AM-9819. Thanks to the experts, Almudena Galán and Rosario García. Thanks to Oscar Corcho for their technical comments.

References

1. Chikofsky, E.J.; Cross II, J.H. „Reverse Engineering and design recovery: A taxonomy." Software Magazine. January 1990. PP:13-17.
2. Domingue, J. „Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web". Knowledge Acquisition Workshop (KAW). Banff (Canadá). 1998.
3. Farquhar A., Fikes R., Rice J., The Ontolingua Server: A Tool for Collaborative Ontology Construction, KAW'96. PP. 44.1-44.19, 1996.
4. Fernández, M.; Gómez-Pérez, A.; Pazos, J.; Pazos, A. Building a Chemical Ontology using methontology and the ontology design environment. IEEE Intelligent Systems and their applications. Vol.4 (1):37-45. 1999.
5. Fernández, M., Gómez-Pérez, A. Juristo, N. METHONTOLOGY: From Ontological Art Toward Ontological Engineering. Spring Symposium Series on Ontological Engineering. AAAI97.
6. Gómez Pérez, A. „Evaluation of Taxonomic Knowledge in Ontologies and Problem Solving Methods". 12th Banff Conference on Knowledge Acquisition for Knowledge Based Systems. 1999.
7. Gómez-Pérez, A.; Rojas-Amaya, M.D. Ontological Reengineering for Reuse. Knowledge Acquisition Modeling and Management. EKAW'99. pp. 139-156.
8. Gómez-Pérez, A. A Framework to Verify Knowledge Sharing Technology. Expert Systems with Application. Vol. 11, N. 4. 1996. PP: 519-529.
9. Gruber, T.;Olsen, G. „An ontology for Engineering Mathematics". Fourth International Conference on Principles of Knowledge Representation and Reasoning. Doyle, Torasso y Sandewall (eds.) Morgan Kaufmann. 1994.
10 .Gruninger M., Fox M., Methodology for the Design and Evaluation of Ontologies, Proceedings of IJCAI95's Workshop on Basic Ontological Issues in Knowledge Sharing, 1995.
11. „IEEE Standard for Developing Software Life Cycle Processes". IEEE Computer Society. New York (USA). April 26, 1996.
12. Pressman, R.S. „Ingeniería del Software. Un enfoque práctico." Mac-Graw Hill. 1993.
13. Swartout, B.; Ramesh P.; Knight, K.; Russ, T. „Toward distributed use of large-scale ontologies". Symposium on Ontological Engineering. American Association for Artificial Intelligence (AAAI). Stanford (California). Marzo 1997.
14. Uschold M., Grüninger M., ONTOLOGIES: Principles, Methods and Applications, Knowledge Engineering Review**,** Vol. 11, N. 2, June 1996.