A Methodology for Ontology Integration

Helena Sofia Pinto & **João P. Martins**
Grupo de Inteligência Artificial
Departamento de Eng. Informática
Instituto Superior Técnico
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
sofia,jpm@gia.ist.utl.pt

Abstract
Although ontology reuse is an important research issue only one of its subprocesses (merge) is fairly well understood. The time has come to change the current state of affairs with the other reuse subprocess: integration. In this paper we describe the activities that compose this process and describe a methodology to perform the ontology integration process.

INTRODUCTION AND MOTIVATION

Ontologies aim at capturing static domain knowledge in a generic way and provide a commonly agreed upon understanding of that domain, which may be reused and shared across applications and groups [4]. Therefore, one can define an ontology as a shared specification of a conceptualization. Ontology reuse is now one of the important research issues in the ontology field. There are two different reuse processes [18]: *merge* and *integration*. Merge is the process of building an ontology in one subject reusing two or more different ontologies on that subject [18]. In a merge process source ontologies are unified into a single one, so it usually is difficult to identify regions in the resulting ontology that were taken from the merged ontologies and that were left more or less unchanged.[1] It should be stressed that in a merge process source ontologies are truly different ontologies and not simple revisions, improvements or variations of the same ontology. Integration is the process of building an ontology in one subject reusing one or more ontologies in different subjects[2] [18]. In an integration process source ontologies are aggregated, combined, assembled together, to form the resulting ontology, possibly after reused ontologies have suffered some changes, such as, extension, specialization or adaptation. In an integration process one can identify in the resulting ontology regions that were taken from the integrated ontologies. Knowledge in those regions was left more or less unchanged. It should be noted that both reuse processes are included in the overall process of ontology building.

A lot of research work has been conducted under the merge area. There is a clear definition of the process [21], operations to perform merge have been proposed [16, 25], a methodology is available [8] and several ontologies have been built by merge [22, 8]. The first tools to help in the merge process are now available [16, 14]. In the integration area a similar effort is now beginning. The most representative ontology building methodologies [24, 13, 7] recognize integration as part of the ontology development process, but none really addresses integration. Integration is only recognized as a difficult problem to be solved. They don't even agree on what integration is: for some it is an activity, for others a step. We have been involved in two integration experiences where publicly available ontologies were reused: we built the Reference ontology [1, 19, 17] and we were involved in building some of the subontologies needed to build an Environmental Pollutants ontology (EPO), namely the Monoatomic Ions ontology [19, 17, 10].

We have found that integration is far more complex than previously hinted. It is a process of its own [17, 19]. Other important conclusions are that integration takes place along the entire life cycle and should begin as early as possible in the ontology building life cycle so that the overall ontology building process is simplified [17, 19]. In both our experiences, integration began as early as the conceptualization phase.

In this article we begin by describing our assumptions and some terminology. Then we discuss and analyze the integration process in relation to the overall ontology building process. Finally, we present our methodology, namely we describe each ontology integration activity and the methods, guidelines and procedures developed to perform them. As far as we know, this is the first integration methodology proposed in the area.

[1] In some cases, knowledge from merged ontologies is homogenized and altered through the influence of one source ontology on another (is spite of the fact that source ontologies do influence knowledge represented in the resulting ontology). In other cases, knowledge from one particular source ontology is scattered and mingled with knowledge that comes from the other sources.

[2] The subjects of the different ontologies may be related.

K-CAP'01, October 22-23, 2001, Victoria, British Columbia, Canada.

ASSUMPTIONS AND TERMINOLOGY

Ontology building is a process composed of several activities. Some are performed at particular stages: specification, conceptualization, formalization, implementation and maintenance. Others take place along the entire life cycle: knowledge acquisition, documentation and evaluation. The development of an ontology follows an evolving prototyping life cycle [6]. Since integration is a process that takes place along the entire life cycle, integration activities can take place for one ontology in any stage of the ontology building process.

The aim of the conceptualization phase is to describe in a conceptual model the ontology that should be built. We assume that, in this phase of *any* ontology building process questions like the following are answered: (1) what should be represented in the ontology? (2) how should it be represented (as a class, relation, etc.)? (3) which relation should be used to structure knowledge in the ontology? (4) which structure is the ontology going to have (graph, tree, etc.)? (5) which ontological commitments and assumptions should the ontology comply to? (6) which knowledge representation ontology should be used? (7) should the ontology be divided in modules? (8) in which modules should the ontology be divided?

At conceptualization, one uses knowledge level [15] representations of ontologies. Usually, only implemented ontologies are publicly available at ontology libraries. If the knowledge level representation of an ontology is not available, then an ontological reengineering process [10] can be applied. This process returns one possible[3] conceptual model of an implemented ontology. When one begins integration as early as conceptualization, one needs the ontologies that are going to be considered for integration represented in an adequate form. Any conceptual model representation is adequate. In our case, we had access to knowledge level representations of most reused ontologies as proposed by METHONTOLOGY [6]: $(KA)^2$ [2] to build the Reference ontology and Chemicals [7] to build the Monoatomic Ions subontology of EPO. In the case of $(KA)^2$ and Chemicals we had access to the actual conceptual models that produced their Ontolingua versions, but in the case of EPO, a reengineering process was applied [10] to produce one conceptual model of Standard Units [12] (which is reused by Chemicals). However, any knowledge level representation would be appropriate. Moreover, due to the particular framework that was used, ODE [7], all of our work was done at the knowledge level. This simplified the overall process of integration a lot. Since in conceptualization much of the design of the ontology is specified, it is considerably more difficult to try to integrate an ontology at the implementation phase because, unless one has prior knowledge of the ontologies available for reuse, available ontologies will rarely match the needs and the conceptual model found for the resulting ontology. One of the consequences of this conclusion is that more integration effort should be made at the earliest stages, specially in conceptualization and formalization, than at final ones, implementation or maintenance [19]. We would like to point out that in both our experiences there was no need to translate ontologies between different knowledge representation languages. Translation of ontologies is a very important and difficult problem to be solved in order to allow more generalized reuse of ontologies.

[3] This process may not produce the actual conceptual model that originated the final ontology. Moreover, if the conceptual model found for the ontology after the reverse engineering step shows some deficiencies, it may be improved through a restructuring step.

For us, an ontology consists of: classes, instances, relations, functions and axioms. Each one of the components of an ontology is generically referred to as a *knowledge piece*. Each knowledge piece is associated with a name, a documentation and a definition.

A METHODOLOGY

As any process, integration is composed of several activities. We have identified the activities that should take place along the ontology building life cycle to perform integration. All integration activities assume that ontology building activities are also performed, that is, the integration process does not substitute the ontology building process, it rather is a part of it. We now describe each activity and the methods, guidelines and procedures developed to perform them. Examples from case studies are partially described in [17, 19]

Identify integration possibility

The framework being used to build the ontology should allow some kind of knowledge reuse. For instance, the Ontolingua Server maintains an ontology library and allows integration operations, such as inclusion or restriction. More general systems, such as KACTUS, do not allow such kind of operations, but allow pre-existent ontologies to be imported and edited. In other cases, integration (or any kind of reuse) may involve rebuilding an ontology in a framework different from the one where the ontology is available. In some cases, this may be cost-effective, in others it may be more cost-effective to build a new ontology from scratch that perfectly meets present needs and purposes than to try to rebuild and adapt a pre-existent ontology.

Identify modules

The modules (building blocks) needed to build the future ontology are identified, that is, the subontologies in which the future ontology should be divided (in integration, the modules are obviously related to ontologies). In integration upper-level modules and domain modules are identified. Representation ontologies are chosen in any ontology building process, therefore they are not specifically addressed in integration.

Identify assumptions and ontological commitments

One needs to identify the assumptions and ontological commitments [11] that *each* module should comply to. They are described in the conceptual model and in the specification requirements document of the future ontology. This is one of the activities where documentation of an ontology can be crucial to allow better, faster and easier reuse. The assumptions and ontological commitments of the building blocks should be compatible among themselves and should be compatible with the assumptions and ontological commitments found for the resulting ontology.

Identify knowledge to be represented in each module

One needs to identify what knowledge should be represented in each building block. At this stage, one is only trying to have an idea of what the modules that will compose the future ontology should "look like" in order to recognize whether available ontologies are adequate to be reused. At this stage one only identifies a list of essential concepts. The conceptual model of the ontology and abstraction capabilities are used to produce this list.[4]

Identify candidate ontologies

This is subdivided into: (1) *finding available ontologies*, and (2) *choosing from the available ontologies which ones are possible candidates to be integrated*. To find possible ontologies one uses ontology sources. Since available ontologies are mainly implemented ones one should look for them in ontology libraries, as for instance, in the Ontolingua Server (`http://WWW-KSL-SVC.stanford.edu:5915`) for ontologies written in Ontolingua, in Ontosaurus (`http://www.isi.edu/isd/ontosaurus.html`) for ontologies implemented in Loom or in the Cyc Server (`http://www.cyc.com`) for Cyc's upper-level ontology. Conceptualized or formalized ontologies are more difficult to find. Sometimes they are available in the literature or can be obtained by contacting ontology builders. However, not every ontology in a given subject will be appropriate to be reused (some may lack some important concepts, etc.).

To choose candidate ontologies one analyzes all available ontologies according to a series of features. At this stage of the ontology integration process one does not want to leave out any possible candidate. Therefore, only a very general analysis is made. Some of the features are *strict requirements*: (1) domain, (2) is the ontology available? (3) formalism paradigms in which the ontology is available, (4) main assumptions and ontological commitments, (5) main concepts represented. If the ontology does not have adequate values for these properties they cannot be considered for integration. Therefore, these properties are used to eliminate ontologies. Some of these features can only be analyzed at a qualitative level (main concepts represented, main assumptions and ontological commitments). Other features are *desirable requirements* or desirable information: (1) where is the ontology available? (2) at what level is the ontology available? (3) what kind of documentation is available (technical reports, articles, etc.)? (4) where is that documentation available? If some of the properties have certain values, the ontology is a better candidate: if the knowledge level representation of an ontology is available, then this ontology is a better candidate since the reengineering process would not have to be performed, if the internal and external documentation is available, then the most relevant information about the construction and choices made during the construction of the ontology is available, but if only articles are available about the ontology, then it is likely that some of the choices are not explained. If all of the values of these properties are unknown, that is, if one cannot find where the ontology and the documentation is available, then one cannot reuse it, therefore, the ontology is not a candidate. However, if there is enough documentation available, then it may be possible to reconstruct the ontology, and if the ontology is available, then it may be possible to understand it, provided that the domain is common enough and the ontology is simple and not very large (and possibly after some knowledge acquisition). One can use a very simple *metric* to combine these features. If strict requirements do not have adequate values, the ontology is eliminated. If desirable requirements have appropriate values, then the ontology is a better candidate. If none of the desirable requirements have appropriate values, then the ontology is not a candidate. One does not want to eliminate any possible candidate at this stage of the process, only those that are of no use at all. If, in a particular integration process, other features should be considered while choosing candidate ontologies, the metric can be easily updated. One only has to decide whether the features are strict or desirable requirements. For instance, one can impose the condition that only already evaluated ontologies should be considered as candidates. In that case, one should add this feature as a strict requirement. If one only wishes to prefer already evaluated ontologies, then this feature should be added as a desirable requirement. The advantage of the flexibility of this metric is the fact that it can be adapted to integration processes that should take into account particular features during the choice of one ontology. In particular, this kind of changes can narrow down the possible ontologies to choose from, if one introduces more strict requirements.

Get candidate ontologies

Getting candidate ontologies in an appropriate form includes, not only, their *representations*, but also, all available *documentation*. As already discussed, one should prefer to work with the knowledge level representation of an ontology. In some cases, this representation can be found in the literature (technical reports, books, thesis, etc.), or at least parts of it. Another possibility is contact ontology developers. However, in most cases, only the implementation level representation of an ontology is available. Therefore, the *reengineering process* may be applied using the particular framework that was adopted to design the resulting ontology. If the ontology is not available (either at the implementation or knowledge

[4] At later stages one will need to know to what level of detail should that knowledge be represented, which relations should organize (structure) the ontology, and it would be helpful to know how it should be represented (concept, relation, etc.).

level), one can still try to reconstruct it, or, at least, parts of it, using available documentation. While getting the implementation level representation of an ontology, if the ontology is not written in the adequate language (the language chosen to represent the resulting ontology) a knowledge *translation process* must take place. There are only a few translation attempts. Translation is far from being a fully automatic process in the near future [23, 20]. In general, there are not many translators available, their technology is still immature and improving existing translators is a rather difficult task. If translators are available they should be used to produce initial versions. Then, these initial versions should be improved by hand. Translators between different knowledge level representation languages are currently not available. The translation process is, in general, complex. It is important that, if the ontology includes other ontologies, one should also get the included ontologies. When reusing/using one ontology one must understand it fully, which includes every definition of every knowledge piece represented in the ontology. Included ontologies are a part of the ontology. Knowledge pieces from included ontologies can be used in the definitions of the ontology, therefore, in order to understand the ontology and know what one knowledge piece defined in an included ontology means one must have access to its definition or its technical documentation.

Study and analysis of candidate ontologies

To study and analyze candidate ontologies we must perform two activities: (1) *technical evaluation* of candidate ontologies by *domain experts* through specialized criteria *oriented to integration* and (2) *user assessment* of candidate ontologies by *ontologists* through specialized criteria *oriented to integration*. Both domain experts and ontologists should evaluate and assess whole and all candidate ontologies. To *technically evaluate candidate ontologies* domain experts should pay special attention to [17, 19]: (1) what knowledge is missing (concepts, relations, etc), (2) what knowledge should be removed, (3) which knowledge should be relocated, (4) which knowledge sources changes should be performed, (5) which documentation changes should be performed, (6) which terminology changes should be performed, (7) which definition changes should be made, (8) which practices changes should be made, Since domain experts usually find the languages used to implement ontologies difficult to understand [7], they should preferably be given a knowledge level representation of the ontology. To *user assess candidate ontologies* ontologists should pay special attention to [17, 19]: (1) the overall structure of the ontology to assess whether the ontology has an adequate (and preferably well-balanced) structure, adequate and enough modules, adequate and enough specialization of concepts, adequate and enough diversity, similar concepts are represented closer whereas less similar concepts are represented further apart, knowledge is correctly "placed" in the structure so that inheritance mechanisms can infer appropriate knowledge from the ontology, etc.; (2) the distinctions (classification criteria made of the concepts described in the ontology) upon which the ontology is built to assess whether they are relevant and exactly the ones (quantity and quality) required; (3) the relation used to structure knowledge[5] in the ontology to assess whether it is the required one; (4) the naming convention rules used to assess whether they ease and promote reuse; (5) the quality of the definitions (do they follow unified patterns, are simple, clear, concise, consistent, complete, correct —lexically and syntactically—, precise and accurate); (6) the quality of the documentation of the ontology; (7) the knowledge pieces represented (or included) are the ones that should be represented and all appropriate knowledge pieces are represented.

Choosing source ontologies

At this stage, and given the study and analysis of candidate ontologies performed by domain experts and ontologists, the final choices must be made. Among the candidate ontologies that passed strict requirements and among those that scored best in integration-oriented technical evaluation and user assessment one has to choose the source ontology (or set of source ontologies) that best suit our needs and purpose. Once again, the ontology(ies) chosen to be reused may lack knowledge, may require that some knowledge is removed, etc., that is, it may not exactly be what is needed. The best candidate ontology is the one that can better (more closely) or more easily (using less operations) be adapted to become the needed ontology. This choice also depends to some extent on the other ontologies that are going to be reused since in an integration process one can reuse more than one ontology. It is important that reused ontologies are compatible among themselves, namely in what concerns overall coherence. Sometimes, one can choose more than one ontology in a given domain if each one focuses different points of view of that domain. This is a rather complex multi-criteria choice where a lot of different aspects are involved. Since the choice of source ontologies is much more complex than choosing candidate ontologies, we propose that it should be divided into *two stages*.

In the **first stage** one tries to find which candidate ontologies are best suited to be integrated. Domain expert and ontologist analyses are crucial in this process. We propose that candidate ontologies should be analyzed according to a taxonomy of features, Figure 1.

General features give general information about the ontology. *Development status* gives information about the degree of readiness of an ontology to be reused (intended, on-going, toy example, implemented, mature). A toy example only contains representative knowledge pieces. An implemented ontology can be a good candidate if it has been carefully built or it has been evaluated. A mature ontology used in applications is a good candidate. The ontology should be a more or less stable ontology (provided that the domain does not evolve very rapidly).

[5] An ontology can be thought of as structured or organized according to one privileged relation, for example, ISA, part-of, etc.

- general
  - type (general, domain)
  - formality
  - development status
- development
  - knowledge acquisition
    - quality of knowledge sources
    - adequacy of knowledge acquisition practices
  - maintenance
    - is it maintained?
    - who does maintenance?
    - how is maintenance done?
  - documentation
    - quality of the documentation available
    - is the available documentation complete?
  - implementation
    - language issues
      - language(s) in which it is available
      - translators: are there translators? for which languages? quality of those translators
      - properties needed of the KR system in which it is built
- content
  - level of detail
  - modularity
  - adequacy from the domain expert point of view
  - adequacy from the ontologist point of view

Figure 1: Choosing source ontologies, first stage

Development features are related to how the ontology was built. The *quality of knowledge sources* and *adequacy of knowledge acquisition practices* are analyzed during the domain expert integration-driven technical evaluation. The ontology should be *maintained*. One interesting finding about ontologies is the fact that they evolve, are "living", since their domains also evolve. Therefore, if they are maintained, it is most likely that they are updated. Maintenance policies differ in *who* changes the ontology (can anybody change the ontology, or only authorized personnel?) and *how* those changes are performed (is the ontology changed regardless of people that built it, use it or reuse it? are the suggestions of change previously discussed among those groups? is there any attempt to reach a consensus between groups? is there a special board that decides upon suggestions for changes?). The *documentation* should have enough *quality* (it is clear, it describes the domain, the ontology, the alternative representations and the preferred alternatives) and is *complete* (the ontology is completely described). If the ontology is available in the required *language* the task is greatly simplified (translation is avoided). Otherwise, it is important to know whether *translators* from those languages are available, *for which languages* and their *quality*. One needs to know which *reasoning capabilities* are required by the ontology from the knowledge representation system where it is implemented, in order to know whether it can be represented under a different knowledge representation system. Full translation between different knowledge representation systems may not be possible. For instance, while translating an ontology represented in first order logic into a pure frame system, if axioms are represented, they are lost. Therefore, one needs to know, among other issues: (1) *formalism paradigm* (frames, semantic networks, description logics, etc.), (2) needed *inference mechanisms* (general purpose, automated concept classifier, inheritance,[6] monotonic vs modal vs nonmonotonic), 3) are *contexts* required?

Content features give information about what is represented in the ontology and how that knowledge is represented. One needs to know whether the ontology has an adequate *level of detail* (enough intermediate concepts are represented between two arbitrary concepts) and *which concepts are represented in which modules*. Under the feature *adequacy from the domain expert point of view* several analyses are made: does the content of the ontology include most of the relevant knowledge pieces of the domain? is the terminology adequate? are the definitions adopted correct and widely accepted? is the ontology complete in relation to present needs (at least, one needs to know what important knowledge pieces are missing)? is there superfluous knowledge that should be removed from the ontology while integrating it? Under the feature *adequacy from the ontologist point of view* several analyses are made: are the basic distinctions represented in the ontology appropriate? does the ontology have an adequate structure? is the ontology structured according to appropriate relations? are needed knowledge pieces represented (including the appropriate relations, and certain key concepts)? are those knowledge pieces adequately represented (this covers issues like fidelity, minimal encoding bias, correction, coherence, granularity, conciseness, efficiency in terms of time and space[7])? do they follow adequate naming convention rules? can missing knowledge pieces be added to the ontology without sacrificing coherence and clarity (extendible)? is the ontology clear?

The preponderant parts in this choice are played by the adequacy analyses that domain experts and ontologists have made of candidate ontologies. Since this choice is rather complex, simple metrics as the ones proposed to choose candidate ontologies are rather limited. The development of more accurate metrics is an open research area in the OE field. After the first stage, one has chosen one possible set of ontologies to be integrated. It may be possible to have more than one ontology about one particular domain in that set. Those different ontologies represent knowledge about the domain from different perspectives. Those different perspectives should have been found important to be present in the resulting ontology (there should not be duplicated knowledge represented in the resulting ontology). However, chosen ontologies may not be compatible among themselves.

In the **second stage** one tackles compatibility and completeness of possibly chosen ontologies in relation to the desired resulting ontology, Figure 2. If the ontologies which are possibly going to be chosen are not coherent in what concerns the terminology and the definitions of the concepts that are common to more than one ontology, then they are not *compatible* and, therefore, cannot be assembled. Sometimes the

[6] Which kind?: defeasible, strict, mixed; credulous vs skeptical; on-path vs off-path; bottom-up vs top-down.

[7] One needs to know if we are reusing an ontology that is not going to meet our needs and the means that we currently have at our disposal.

- compatibility
  - terminology of common concepts
  - definitions of common concepts
- completeness

Figure 2: Choosing source ontologies, second stage

same concept is named differently in different ontologies. In the resulting ontology one concept only has one denomination, therefore one must be adopted. If one concept has the same definition in all chosen ontologies but different denominations, then a change in terminology can solve the problem. All definitions involving the renamed concept have to be checked and revised accordingly. Sometimes different ontologies adopt different definitions for the same concept. One cannot have this kind of inconsistencies in the resulting ontology. One definition should be chosen and adopted all over. It is more difficult to ensure that the same definition can be adopted by all integrated ontologies. A thorough analysis of all ontologies where one particular concept has a different definition from the adopted one has to be made. It is obvious that only a coherent set of ontologies should be considered for integration purposes. If chosen ontologies are not compatible among themselves, then this may imply choosing another possible set of ontologies by combining candidate ontologies into a different set, or it may imply building ontologies from scratch (if none of the candidate ontologies adopts the adequate terminology and definitions, or profound changes have to be made to them in order to integrate them). If chosen ontologies are not *complete*, that is, they do not comprehend all the ontology that has to be built, then this must be known so that missing knowledge pieces are built from scratch and added or another compatible ontology that contains those knowledge pieces is integrated. Since one of the issues involved in the domain expert analysis is missing knowledge, one can check whether it is not represented in another ontology about the same domain that is also (or can also be) integrated.

The problem of choosing the appropriate set of source ontologies is also rather complex. From the set of candidate ontologies, a coherent and adequate subset must be found that is as close as possible to the resulting ontology. Once again, the ontologies in that set may not be perfect candidates. As long as the changes to be made are not very extensive it is more cost effective to reuse the ontologies. This analysis has to be performed on a case by case basis. If it is more cost effective to build the ontology from scratch, then existing ontology building methodologies can be used to build an ontology that perfectly suits our needs. If not, ontologies should be reused and integration operations applied so that adequate changes transform the ontologies into perfect candidates. The result of this activity is a set of ontologies that can and should be assembled together, a description of lacking knowledge that is going to be built from scratch and included in the resulting ontology (since none of the chosen ontologies has it and that knowledge has been identified as essential knowledge that must exist in the resulting ontology) and a description of the changes that should be performed to the integrated ontologies so that they can be perfect candidates and successfully reused (which is the starting point for the application of the integration operations).

Apply integration operations

All activities described so far precede integration of knowledge from source ontologies into the resulting ontology. They help the ontologist to analyze, compare, and choose the ontologies that are going to be reused. When this part of the process ends, that is the appropriate ontologies to be reused in one particular integration process are found, we must integrate the knowledge of those ontologies. For that, one needs *integration operations* and *integration oriented design criteria*. Integration operations specify how knowledge from an integrated ontology is going to be included and combined with knowledge in the resulting ontology, or modified before its inclusion. These can be viewed as composing, combining, modifying or assembling operations. Knowledge from integrated ontologies can be, among other things, (1) used as it is, (2) adapted (or modified), (3) specialized (leading to a more specific ontology on the same domain) or (4) augmented (either by more general knowledge or by knowledge at the same level). Sometimes the adaptation of ontologies may require restructuring activities similar to those that are performed in reengineering processes. Moreover, it may require introduction/removal of knowledge pieces, correction and improvement of the definitions, terminology and documentation of the knowledge pieces represented in the ontology, etc. These adaptations transform the chosen ontology into the needed ontology. In [5, 3, 19, 17] initial sets of integration operations are proposed. *Integration operations* can be divided into two groups: basic and non-basic. While the former can be algebraically specified the latter can be defined from the former but are custom-tailored operations to be defined in a case by case basis. We have developed an algebraic specification of 39 basic integration operations and specified how 12 non-basic operations can be defined from the previous ones. Design criteria guide the application of integration operations so that the resulting ontology has an adequate design and is of quality. We identified a set of *criteria* to guide integration of knowledge [1]: modularize, specialize, diversify each hierarchy, minimize the semantic distance between sibling concepts, maximize relationships between taxonomies and standardize names of relations.

Analyze resulting ontology

After integration of knowledge one should evaluate and analyze the resulting ontology. Besides having an adequate design [11] and compliance with evaluation criteria [9] the ontology should have a *regular level of detail all over*. By regular level of detail we mean that there are no ”islands” of exaggerated level of detail and other parts with an adequate one. None of the parts should have less level of detail than

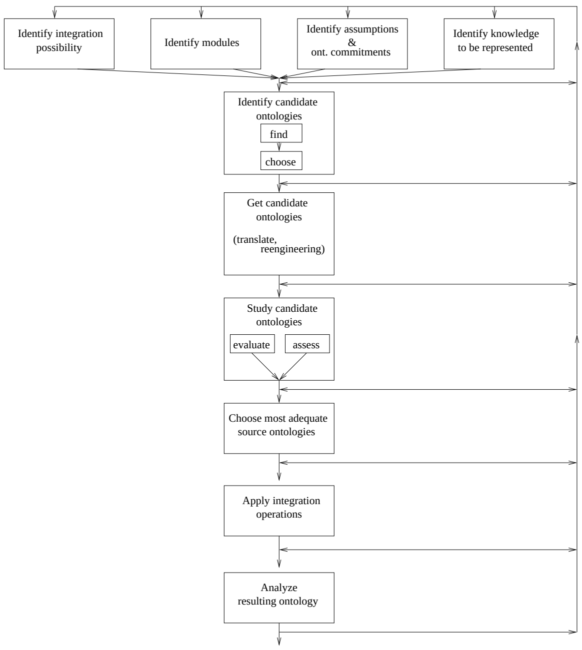

Figure 3: The integration process

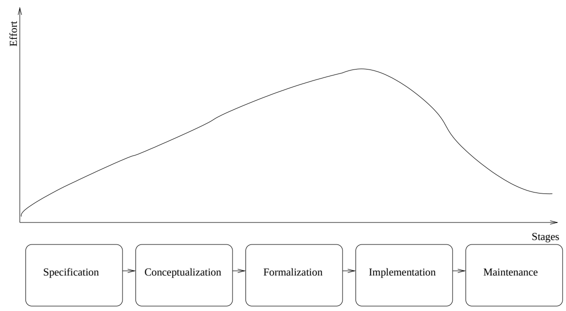

Figure 4: Integration effort along the ontology building process

the required one or else the ontology would be useless, since it would not have sufficient knowledge represented. It should be noted that the features involved in evaluation and design criteria are analyzed in relation to the resulting ontology, for instance, the resulting ontology should be consistent and coherent all over (although composed by knowledge from different ontologies).

DISCUSSION

In Figure 3 we present the activities that compose the ontology integration process. Although ontology building and consequently ontology integration follows an evolving prototyping life cycle, some order must be followed. In general, the activities that compose the integration process tend to be performed following the order by which they were presented. However, some of the activities (and subactivities) to be performed before applying integration operations are interchangeable and some may be even performed in parallel. For instance, integration-oriented technical evaluation and user assessment of candidate ontologies. Moreover, the auxiliary subprocesses, reengineering and translation, may not occur in a particular integration process. If we find an ontology that matches the whole ontology that one needs to build, then one does not need to apply integration operations or analyze the resulting ontology. However, finding candidate ontologies, getting them, their evaluation and assessment for integration purposes, and the choice of the most adequate one remain essential activities to be performed. Finally, one can go back from any stage in the process to any other stage as entailed by the kind of life cycle. The important issue is that these activities are present in any integration process, although sometimes not explicitly or with different levels of importance and effort. All activities, in particular those that precede application of integration operations, should be performed preferably in conceptualization or in formalization stages, that is, before implementation. However, if integration begins later in the ontology development life cycle, they still have to be performed. In both our integration experiences the framework that we used, ODE, automatically generated the implemented versions of the resulting ontologies. Therefore, we performed all integration activities during conceptualization and formalization stages. Using other frameworks may extend the process a bit. If the framework being used does not generate the implementation of the resulting ontology from the conceptual representations, after performing all activities at the knowledge level, the implemented versions of the chosen ontologies must be obtained and then one must apply the already determined sequence of integration operations in order to build the implemented version of the resulting ontology. In this case, only two activities (get ontologies and apply integration operations) had to be performed at the implementation level. This particular process falls into a typical evolving prototyping life cycle. One important aspect of integration is the fact that this process is included in the overall ontology building process. The relation between the integration process and the overall ontology building process is shown in Figure 4. The integration effort is not null during maintenance since integrated ontologies may themselves change due to maintenance activities making it necessary (or desirable) to reapply the integration process.

CONCLUSIONS

In this article we describe the activities that compose the ontology integration process and present a methodology that provides support and guidance to perform those activities. The advantages of the proposed integration methodology are a direct consequence of its generality. One of the advantages of our integration methodology is the fact that it *can be used with different methodologies to build ontologies from scratch*. The only assumption made by this methodology is that knowledge should be represented at the knowledge level.

Special emphasis is given to the *quality* of the ontologies involved in a particular integration process. Our methodology proposes that all reused ontologies should be evaluated by domain experts from a technical point of view and assessed by ontologists from a user point of view. This assures that reused ontologies have enough technical quality to be used in the process. The analysis of the resulting ontology assures that it has enough quality to be made available and (re)used.

REFERENCES

1. J. Arpirez-Vega, A. Gómez-Pérez, A. Lozano-Tello, H. Sofia Pinto. Reference Ontology and $(ONTO)^2$ Agent: the Ontology Yellow Pages. *Knowledge and Information Systems*, 2(4):387-412, 2000.

2. V.R. Benjamins, D. Fensel. The Ontological Engineering Initiative $(KA)^2$. In N. Guarino (ed.), *Formal Ontology in Information Systems*, pages 287-301. IOS Press, 1998.

3. P. Borst, H. Akkermans, J. Top. Engineering Ontologies. *International Journal of Human Computer Studies*, 46(2/3):365-406, 1997.

4. B. Chandrasekaran, J. Josepheson, V.R. Benjamins. Ontologies: What are they? Why do we need them? *IEEE Intelligent Systems*, 14(1):20-26, 1999.

5. A. Farquhar, R. Fikes, J. Rice. Tools for Assembling Modular Ontologies in Ontolingua. In *Proc. AAAI97*, pages 436-441. AAAI Press, 1997.

6. M. Fernández, A. Gómez-Pérez, N. Juristo. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In *Proc. of AAAI97 Spring Symposium Series, Workshop on Ontological Engineering*, pages 33–40. AAAI Press, 1997.

7. M. Fernández, A. Gómez-Pérez, A. Sierra, J. Sierra. Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment. *IEEE Intelligent Systems*, 14(1):37-46, 1999.

8. A. Gangemi, D. Pisanelli, G. Steve. Ontology Integration: Experiences with Medical Terminologies. In N. Guarino (ed.), *Formal Ontology in Information Systems*, pages 163-178. IOS Press, 1998.

9. A. Gómez-Pérez, N. Juristo, J. Pazos. Evaluation and Assessment of the Knowledge Sharing Technology. In N. Mars (ed.), *Towards Very Large Knowledge Bases*, pages 289-296. IOS Press, 1995.

10. A. Gómez-Pérez, D. Rojas-Amaya. Ontological Reengineering for Reuse. In D. Fensel, R. Studer (eds.), *Proc. of EKAW99*, pages 139-156. Springer Verlag, 1999.

11. T. Gruber. Towards Principles for the Design of Ontologies for Knowledge Sharing. *International Journal of Human Computer Studies*, 43(5/6):907-928, 1995.

12. T. Gruber, G. Olsen. An Ontology for Engineering Mathematics. In J. Doyle, E. Sandewall, P. Torasso (eds.), *Proc. KR94*, pages 258-269. Morgan Kaufmann, 1994.

13. M. Gruninger. Designing and Evaluating Generic Ontologies. In *Proc. of ECAI96's Workshop on Ontological Engineering*, pages 53-64, 1996.

14. D. McGuiness, R. Fikes, J. Rice, S. Wilder. An Environment for Merging and Testing Large Ontologies. In A. Cohn, F. Giunchiglia, B. Selman (eds.), *Proc. KR2000* , pages 483-493. Morgan Kaufmann, 2000.

15. A. Newell. The Knowledge Level. *Artificial Intelligence*, 18(1):87-127, 1982.

16. N. Noy, M. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proc. AAAI2000*, pages 450-455. AAAI Press, 2000.

17. H. Sofia Pinto. Towards Ontology Reuse. In *Proc. of AAAI99's Workshop on Ontology Management*, pages 67-73. AAAI Press, 1999.

18. H. Sofia Pinto, A. Gómez-Pérez, J. P. Martins. Some Issues on Ontology Integration. In *Proc. of IJCAI99's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends*, 1999.

19. H. Sofia Pinto, J.P. Martins. Reusing Ontologies. In *Proc. of AAAI2000 Spring Symposium Series, Workshop on Bringing Knowledge to Business Processes*, pages 77-84. AAAI Press, 2000.

20. T. Russ, A. Valente, R. MacGregor, W. Swartout. Practical Experiences in Trading Off Ontology Usability and Reusability. In *Proc. of KAW99*, 1999.

21. J. Sowa. *Knowledge Representation: logical, philosophical and computational foundations*. Brooks/Cole, 2000.

22. B. Swartout, R. Patil, K. Knight, T. Russ. Toward Distributed Use of Large-Scale Ontologies. In *Proc. of AAAI97 Spring Symposium Series, Workshop on Ontological Engineering*, pages 138-148. AAAI Press, 1997.

23. M. Uschold, M. Healy, K. Williamson, P. Clark, S. Woods. Ontology Reuse and Application. In N. Guarino (ed.), *Formal Ontology in Information Systems*, pages 179-192. IOS Press, 1998.

24. M. Uschold, M. King. Towards a Methodology for Building Ontologies. In *Proc. of IJCAI95's Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.

25. G. Wiederhold. Interoperation, Mediation and Ontologies. In *Proc. of the Intern. Symposium on the 5th Generation Computer Systems, Workshop on Heterogeneous Cooperative Knowledge-Bases*, 1994.