A3-D11

Testbeds

Project title:	Reasoning on the Web with Rules and Semantics
Project acronym:	REWERSE
Project number:	IST-2004-506779
Project instrument:	EU FP6 Network of Excellence (NoE)
Project thematic priority:	Priority 2: Information Society Technologies (IST)
Document type:	D (deliverable)
Nature of document:	R (report)
Dissemination level:	PU (public)
Document number:	IST506779/Turin/A3-D11/D/PU/b1
Responsible editors:	M. Baldoni, C. Baroglio, V. Patti
Reviewers:	Daniel Krause (internal)
Contributing participants:	Hannover, Teckniker, Turin, Warsaw
Contributing workpackages:	all partners
Contractual date of deliverable:	February 29, 2008
Actual submission date:	14 February 2008

Abstract

This deliverable reports the most recent advancements of research activity carried on within the A3 working group, particularly concerning the testbeds. More specifically, the report focusses on the BEATCA application and on curricula modeling, planning and validation. We also briefly describe the DYNAWeb platform for component integration and reuse.

Keyword List

semantic web, personalization, testbeds, clustering, elearning, web services

Project co-funded by the European Commission and the Swiss Federal Office for Education and Science within the Sixth Framework Programme.

Testbeds

Fabian Abel[1], Aitor Arnaiz Irigaray[2], Matteo Baldoni[3], Cristina Baroglio[3], Ingo Brunkhort[1], Nicola Henze[1], Mieczyslaw A. Klopotek[4], Vivana Patti[3], Slawomir Wierzchon[4]

[1] L3S Research Center, University of Hannover, Germany
Email: `henze@l3s.de`

[2] Fundación Tekniker, Spain
Email: `aarnaiz@tekniker.es`

[3] Dipartimento di Informatica, Università degli Studi di Torino Italy
Email: {`baldoni,baroglio,patti`}`@di.unito.it`

[4] Instytut Podstaw Infornmatyki Polskiej Akademii Nauk, Warszawa
Poland
Email: `klopotek@ipipan.waw.pl,stw@ipipan.waw.pl`

14 February 2008

Abstract
This deliverable reports the most recent advancements of research activity carried on within the A3 working group, particularly concerning the testbeds. More specifically, the report focusses on the BEATCA application and on curricula modeling, planning and validation. We also briefly describe the DYNAWeb platform for component integration and reuse.

Keyword List
semantic web, personalization, testbeds, clustering, elearning, web services

Contents

1 Introduction

This deliverable reports about the most recent achievements on the issue of testbeds, carried on within the working group A3, with a particular attention to the transfer of the results of the I-groups of the network into the testing environments developed within A3. This deliverable witnesses the advancement of the works in the following way:

- The next sections report brief descriptions of the main research directions that have been investigated, reporting both a summary of the theoretical background and a description of the implementations, including a reference bibliography.
- Further details are, then, supplied in the form of scientific articles, that were included to this deliverable as an Appendix. Such articles either appeared in the proceedings of international conferences/workshops or in international journals.

The deliverable is organized as follows:

- Section 2 reports the achievements concerning the the BEATCA application for document clustering. BEATCA stands for Bayesian and Evolutionary Approach to Text Connectivity Analysis. It represents a new approach to document collection analysis and search in the main stream of so-called "landscape metaphor". BEATCA applies new concepts for document maps generation, including map stabilization and navigation based on large scale Bayesian networks techniques called ETC and new concepts of map cell filling based on Growing Neural Gas and Artificial Immune Paradigms combined with WebSOM approach. BEATCA may be considered as a search engine with document map type user interface, dedicated for small collections of documents (a couple of millions at present). The core of this application can also be accessed by Personalization Services of the Personal Reader framework to search for data in an individualized way.
- Section 3 concerns the achievements in the curricula modeling framework. With the Personalized Curriculum Planner we present a service-oriented personalization system, set in an educational framework, based on a semantic annotation of courses, given at a knowledge level. The system supports reasoning-based curriculum sequencing and validation:
  - Curriculum Planning: building personalized curricula, formalized by means of an action theory. Classical planning techniques are adopted, which take into account both the student's initial knowledge (context) and her learning goal.
  - Curriculum Validation: verifying the compliance of curricula w.r.t. the course design goals. Course design goals are specified in a curricula model, where the design goals formalized as a set of LTL temporal formulas expressing constraints at the knowledge level.

  Learning Objects are modeled as Actions: what the course teaches, and what is requested to be known for attending it in a profitable way, is described by means of preconditions (prerequisites) and effects (learning objectives).
- Section 4 concludes the report by briefly introducing the DYNAWeb platform for service reuse and integration (still on-going work). Overall conclusions and the mentioned collection of scientific articles end the technical report.

2 The BEATCA application

Clustering of Web documents, especially in large and heterogeneous collections, is a challenging task, both in terms of processing time complexity and clustering quality. But the most challenging part is the way how the clustering information is conveyed to the end user and how it meets his expectations (personalization).

Note that a text document is usually a very complex information structure from the point of view of human beings dealing with this document. But actually a computer system fully understanding the document contents is beyond technological possibilities. Therefore some kinds of *approximation* to the content are done. Documents are frequently treated as a *bag of words* in computer models, and the documents are viewed as points in term-document vector space, but this approach turns out to be insufficient, hence more complex representations, both of the content of a single document and their collections are investigated.

In the recent years, various projects (WebSOM, [11, 12, 13, 7], Themescape etc.) were aimed at developing a new form of cluster description – a visual document map representation. In a two-dimensional space, consisting of quadratic or hexagonal cells, the split into clusters is represented as an assignment of documents to cells in such a way, that documents assigned to cells are as homogenous as possible and cells (clusters) containing similar documents are placed close to one another on the map, and map regions are labeled with best-fitting terms from the documents. An inversion of the clustering (that is clustering of terms instead of documents) is also possible.

It is generally believed that individual information needs of users may differ and there is a general feeling that therefore also the data processing results should accommodate to the profile of the user. Countless methods and ways of user profile representation and acquisition have been designed so far. The problem with map-like representation of document collections, however, lies in the pretty expensive processing / high complexity (in terms of time and space) so that a personalized ad-hoc representation is virtually impossible. With the BEATCA project we have made essential steps to contest this view and claim that personalization of map representation of large scale document collections is possible by a careful separation of the concept of individual needs from the concept of common knowledge. This leads to the possibility of separation of computationally intense tasks of identification of the structure of clustering space from the relatively less resource consuming pure presentation part.

2.1 Clustering Space

It is usually assumed that personalization is needed because of cultural, ethnical etc. differences that influence the *world of values*, the *attitudes* and the *views*. So, in the particular case of clustering, the *distances* (or more strictly speaking dissimilarities) between the objects may change from person to person. So a personalization under this assumption would be reduced to a total re-clustering of the objects.

We disagree with this assumption. While avoiding here a deeper philosophical discussion of the issue, let us point at some issues significant for the text processing task. Human beings possess to a large extent an objective world which they describe with a common set of concepts (a vocabulary) that is intended for them to communicate to other human beings. So the vast majority of concepts is shared and their meaning not determined by *values*, *attitudes* etc. What differs the human beings is the current task they are interested in. So if discussing e.g. an issue in biology, one does not care about concepts important for chemical engineering. Hence it is not the personal attitude, but rather the context in which an issue is discussed that impacts the feeling of dissimilarity of *opinions* (in this case documents).

Therefore, in our opinion, the proper approach to personalization has to be well founded on the proper representation of document information.

It has been generally agreed that the processing of textual information, especially at large scale, requires a simplification of the document representation. So a document is treated as a bag of words (without bothering about sentence structure). It is represented in the space of documents (with dimensions being spread by words and phrases, that is terms) as a point with coordinates being a function of the frequency of each term in the document. The similarity between two documents is measured as a cosine of the angle between the vectors drawn from the original of the coordinate system to these points. Furthermore, dimensionality reduction may be of high importance [6]

So the weight of a term in the document is calculated as the so-called $tfidf$ (term frequency times the inverse document frequency) computed according to the formula:

$$w_{t,d} = f_{t,d} \times \log \frac{|D|}{f_D^{(t)}} \tag{1}$$

where $f_{t,d}$ is the number of occurrences of term t in document d, $|D|$ is the cardinality of the set of documents, and $f_D^{(t)}$ is the number of documents in collection D containing at least one occurrence of term t.

A document d is described by a vector $d = (w_{t_1}, \ldots, w_{t_{|T|}})$, where T is the set of terms. Usually, the document is turned into a normalized form $di = (wi_{t_1}, \ldots, wi_{t_{|T|}})$ where di has a unit length.

We notice immediately that the weights of terms in a document are influenced by their distribution in entire document collection. If we look, however, at any process of clustering, we immediately notice that weights calculated by the very same method, but within any reasonable (homogenous) cluster, not being a random subsample of the whole collection, would exhibit considerable differences to the global weighing scheme. However, this fact is actually ignored by most researchers.

So our first methodological step is to resign from the rigid term weighing scheme for the sake of local differentiation of term weighing taking into account the local context (of the identified cluster). Though this vision of document similarity works quite well in practice, agreeing with human view of text similarity, it has been early recognized, that some terms should be weighed more than other. One should reject the common words (stop-words) appearing frequently in all documents, as well as those which appear quite seldom.

But we do not want to replace global term weighing scheme with a global weighing scheme. Rather than this, we reconsider the impact of the documents that are far away from cluster core, on the term weighing. Another important point is that terms specific for a given cluster should weigh more than terms not specific for any cluster. Last not least, let us relax the notion of a cluster to the concept of clustering space. Let us speak of a point p in the document space (a point of an unit hyper-sphere) as a vector $p = (wi_{t_1}, O, wi_{t_{|T|}})$ where $||p|| = 1$ (unit length). Each such a point can be treated as cluster center in a (*continuous*) clustering space. We can then define, for each document d, a membership function $m_{d,C(p)}$ in the style of fuzzy set membership function [2], for example as

$$m_{d,C(p)} = \sum_{t \in T} wi_t(di) \cdot wi_t(p) \tag{2}$$

that is the dot product of vectors di and p.

Given this concept, we can define the specificity $s_{t,C}$ of a term t in a cluster $C(p)$ as

$$s_{t,C(p)} = |C(p)| \cdot \frac{\sum_{d \in D} \left(f_{t,d} \cdot m_{d,C(p)} \right)}{f_{t,D} \cdot \sum_{d \in D} m_{d,C(p)}} \tag{3}$$

where $f_{t,d}$ is (as earlier) the number of occurrences of term t in document d, $f_{t,D}$ is the number of occurrences of term t in document collection D, and $|C(p)|$ is the *fuzzy cardinality* of documents at point p, defined as

$$|C(p)| = \sum_{d \in D} \mu_{d,C(p)} \tag{4}$$

where $\mu_{d,C}$ is the normalized membership:

$$\mu_{d,C(p)} = \frac{m_{d,C(p)}}{\int_{p \in HS} m_{d,C(p)}}$$

and HS is the unit hyper-sphere.

In this way we arrive at a new (contextual [5]) term weighing formula for term t in the document d from the point of view of the

$$w_{t,d,C(p)} = s_{t,C(p)} \times f_{t,d} \times \log \frac{|C(p)|}{f^{(t)}_{C(p)}} \tag{5}$$

where $f^{(t)}_{C(p)}$ is the fuzzy count of documents in collection $C(p)$ containing at least one occurrence of term t,

$$f^{(t)}_{C(p)} = \sum_{\{d : f_{t,d} > 0\}} m_{d,C(p)} \tag{6}$$

For consistency, if $f^{(t)}_{C(p)} = 0$ we define $w_{t,d,C(p)} = 0$.

The universal weight $tfidf$ given by equation (1) will be replaced by the concept of an *averaged* local weight

$$w_{t,d} = \frac{\int_{p \in HS} m_{d,C(p)} \cdot w_{t,d,C(p)}}{\int_{p \in HS} m_{d,C(p)}} \tag{7}$$

where HS is the unit hyper-sphere.

Note that the definition of term weights $w_{t,d}$ becomes recursive in this way ($m_{d,C(p)}$ is used here, which is computed in equation (2) based on $w_{t,d}$ itself) and the fixpoint of this recursion is the intended meaning of term weight.

Our further concern is the way how typical hierarchical (or other multistage) algorithms handle lower level clusters. The cluster is viewed as a kind of *averaged* document, eventually annotated with standard deviation of term frequencies and/or term weights. In our opinion, the distribution (approximated in our approach by a discrete histogram) of the term weight (treated as a random variable) reflects much better the linguistic nature of data. The clusters should be formed not as hyperspheres around some center, but rather as collections of documents with terms used in a similar way. This was confirmed by our reclassification experiments [4], showing higher stability of histogram-based cluster description versus centroid-based representation[1].

[1] reclassification measure evaluates consistency of the model-derived clustering with the histogram-based clustering space description (cf. [4])

So for any point p in the clustering space and any term t we define a term-weight distribution as one approximated by the histogram in the following manner: Let $\Delta(wi, t)$ be a discretization of the normalized weights for the term t, assigning a weight for a term the integer identifier of the interval it belongs to (higher interval identifiers denote higher weights). Let $\chi(d, t, q, p)$ be the characteristic function of the term t in the document d and the discretization interval identifier q at point p, equal to $m_{d,C(p)}$ if $q = \Delta(wi_{t,d,C(p)}, t)$, and equal zero otherwise. Then the histogram $h(t, p, q)$ is defined as

$$h(t, p, q) = \sum_{d \in D} \chi(d, t, q, p) \tag{8}$$

With hi we denote a histogram normalized in such a way that the sum over all intervals q for a given t and p is equal 1:

$$h'(t, p, q) = \frac{h(t, p, q)}{\sum_q h(t, p, q)} \tag{9}$$

We can easily come to the conclusion, when looking at typical term histograms that terms significant for a cluster would be ones that do not occur too frequently nor too rarely, have diversified range of values and have many non-zero intervals, especially with high indices.

Hence the significance of a term t for the clustering point p may be defined as

$$m_{t,C} = \frac{\sum_q \left[q \cdot log\left(hi(t, p, q)\right)\right]}{Q_t} \tag{10}$$

where Q_t is the number of intervals for the term t under discretization.

Let us denote with $H(t, p, q)$ the *right cumulative* histograms, that is $H(t, p, q) = \sum_{k \geq q} h(t, p, k)$. The right cumulative *histograms* are deemed to reflect the idea, that terms with more weight should be more visible. For technical reasons Hi is a histogram normalized in the same way as hi.

Let us measure the divergence between clustering points p_i, p_j with respect to term t as (Hellinger divergence, called also Hellinger-Matsushita-Bhattacharya divergence, [1])

$$Hell_k(p_i, p_j, t) = \sqrt{\sum_q \left(Hi(t, p_i, q)^{(1/k)} n Hi(t, p_j, q)^{(1/k)}\right)^k} \tag{11}$$

Finally let us measure the divergence between clustering points p_i, p_j as such as

$$dst(p_i, p_j) = \frac{\sum_{t \in T} m_{t,C(p_i),C(p_j)} \cdot Hell_k(p_i, p_j, t)}{\sum_{t \in T} m_{t,C(p_i),C(p_j)}} \tag{12}$$

where

$$m_{t,C(p_i),C(p_j)} = \sqrt{(m_{t,C(p_i} + 1) \cdot (m_{t,,C(p_j)} + 1)} - 1$$

With this definition, we can speak of a general notion of a cluster as *islands* in the clustering space such that the divergence within them differs not significantly, and there exist at least n documents belonging predominantly to such an island. Thus, it can be treated as a dissimilarity measure.

It may be easily deduced that equation (10) gives also interesting possibilities of labeling of cluster space with meaningful sets of terms (concepts).

User-related sources of information Let us now turn to the user related information. Some documents may be pre-labeled by the user (with category, liking, etc.), there may be past queries available etc.

Note that the contextual document space, as described in the previous section, may be viewed as a *pure space* with some *material objects* causing a kind of curvature of this space.

The user-related sources can be viewed as consisting of two types of *documents*: *material objects* (all the positively perceived, relevant information) and the *anti-material objects* (all the negatively perceived information).

The user-related documents may be also represented in a clustering space, in at least two different ways:

- in separate user-material, user-anti-material and proper document clustering spaces - in this case a *superposition* of these spaces would serve as an additional labeling of the proper document space, beside the original labels derived from document collection content.
- in a joint space $\top n$ in this case user-related information will transform the document space of the document collection.

While the second approach may be considered as a stronger personalization, it will be more resource consuming and raises the issue of pondering the impact of user related documents against the entire collection, and also that of the relation between positive and negative user information. The first approach will be for sure much less resource consuming, because the processing of the big entire document collection has to be done only once, and the user related information is usually of marginal size and can be processed in a speedy way.

2.2 Clustering Space Approximation

If we look at the clustering space as a continuum, it is obvious, that we cannot consider clusters in isolation, but we want to take relationships between them into account. It is also obvious that need to provide a finite, discrete approximation of this continuum. To achieve it, the usually wide areas of clustering space of next to zero proximity to the documents will be ignored, as well as those terms that within the given subspace are of marginal significance. This is caused by the fact that each document modifies the space close to it having marginal impact of the rest. So the space may be greedy subdivided into non-empty subspaces that are deemed to be linked if they adhere to one another, and not, if they are of next to zero similarity.

The process, that we apply to approximate the clustering space [10], which we call Adaptive Clustering Algorithm, starts with splitting of the document collection into a set of roughly equally sized sub-collections using the expression (1) as an approximation of term weights for document similarity computation in a traditional clustering algorithm. We work in a hierarchical divisive mode, using the algorithm to split the collection in a small number of subcollections and apply further splitting to sub-collections of too big size. At the end too small clusters are merged with most similar ones. As a next iteration for each sub-collection, being now treated as a context (as it is now feasible), an iterative recomputation of term weights according to equation (7) with respect to cluster center, making the simplifying assumption that documents from other contexts have no impact. Within each context, the dictionaries of terms are reduced removing insignificant terms in a given context (different terms may be *zeroed* in different contexts). Subsequently the inter-document structure is formed. For this purpose one of the known networking clustering algorithms is used, either the growing neural gas [8] or idiotypic (artificial immune) network [14, 3]. Finally we turn back to the global set of contexts and apply a networking clustering algorithm to representatives of each context. This time, the histograms of contexts are applied to compute a measure of similarity between contexts $\top n$ see equation (12). While applying the networking clustering, we additionally compute so-called *major topics*, that is a split of the (sub)collection into up to 6 sub-clusters, the representatives of which are deemed to be major topics of the collection.

In this way, an approximation of the clustering space is obtained. In case of visualization, the WebSOM algorithm is applied to context representatives, in case one wants to view the global map, and to neural gas cells, or immune network cells in case of detailed view of a context. The computation of the map given the clustering space model is drastically simplified because e.g. with a collection of 12,000,000 documents we need to cluster only 400 representatives. So given such a cluster network, its projection onto a flat rigid document map structure, with treating each whole cluster as a single *document*, is a dramatically simpler task than the map creation process for individual documents.

Our implementation of WebSOM differs from the original one in a number of ways, accelerating the processing significantly. One of the features is the topic-sensitive initialization. While WebSOM assigns random initial cluster centers for map cells, we distribute evenly the vectors of major topics over the map and initialize the remaining cells with in-between values (with slight noise). In this way the maps are learned usually quicker and are more stable (no drastic changes from projection to projection).

We have demonstrated in our earlier work [3, 4, 5, 10] that such an approach to document space modeling is stable, scalable and can be run in an incremental manner.

Exploiting user-related sources With this background we can explain our approach to personalization. We treat the document collection as a piece of knowledge that is esteemed by any user in the same way. So the identified clusters and the identified interrelationships between them are objective, independent of the user. The user at a given moment may be, however, interested to view the collection from a different direction. So the personalization may be reduced to the act of projection of the cluster network onto the flat map, that is, contrary to projection of document collection, a speedy process, to be managed within seconds. In this process, we can proceed in two distinct ways:

- instead of using the topical vectors of a context / global collection, the user profile topical vector is applied, or
- the user related *documents* are attached to the collection clusters prior to projection (and may or may not influence the projection process) and serve as a source of additional labeling.

Another view of the Adaptive Clustering Algorithm Our incremental textual data clustering algorithm relies on merging two known paradigms of clustering: the fuzzy clustering and the subspace clustering. The method differs essentially from Fuzzy C-Means in that it is designed solely for text data and is based on contextual vector representation and histogram-based description of vector subspaces.

Like Fuzzy-C-Means, we start with an initial split into subgroups, represented by a matrix $U(\tau_0)$, rows of which represent documents, and columns representing groups, they are assigned to. Iteratively, we adapt (a) the document representation, (b) the histogram description of contextual groups, (c) membership degree of documents and term significance in the individual groups.

These modifications can be viewed as a recursive relationship leading to a precise description of a contextual subspace in terms of the membership degree of documents and significance of terms in a context and on the other hand improving the understanding of document similarity.

So we can start without any knowledge of document similarity, via a random assignment of documents to a number of groups and global term weighing. But through the iterative process some terms specific for a group would be strengthened, so that class membership of documents would be modified, hence also their vector representation and indirectly similarity definition.

So we can view the algorithm as a kind of reinforcement learning. The usage of histogram approach makes this method incremental.

2.3 Architecture

Our research convcepts have been validated by creating a full-fledged search engine (with a working name BEATCA) for collections of up to million documents, capable of representing on-line replies to queries in a graphical form on a document map. We followed the general architecture for search engines, where the preparation of documents for retrieval is carried out by an indexer, which turns the HTML etc. representation of a document into a vector-space model representation. After that the map creator is applied, turning the vector-space representation into a form appropriate for on-the-fly generation of the map, which is then used by the query processor responding to user's queries.

Modular Structure The architecture of our system has been designed to allow for experimental analysis of various approaches to document map creation. The software consists of essentially five types of modules, cooperating via common data structures. The types of modules are as follows:

1. robot (spider, crawler), collecting documents for further processing,
2. indexer, transforming documents into a vector space representation,
3. optimizer, transforming the document space dictionary into more concise form,
4. document clustering, identifying compact groups of documents sharing similar topics,
5. mapper, transforming the vector space representation into a map form
6. search engine, responding to user queries, displaying the document maps in response to such queries.

Additionally, we have an experiment management module, that can be instructed to configure the search engine process out of selected modules, to repeat the execution of some parts of the process, and to collect various statistics about the execution of the other modules and on the quality of the final and the intermediate results.

2.4 Data Structures

The data structures interfacing the modules are of the following types:

1. HT Base [hypertext documents],

2. Vector Base [vector space representations],
3. DocGR Base [thematical document groups]
4. Map Base [repository of various maps],
5. CellGR Base [map areas (groups of cells)]
6. Base Registry [registry of all databases, parameters and evaluation results].

A HT Base is the result of a robot activity. We have currently two types of robots, one collecting documents from the local disk space, and another from the Web. A robot collects the hypertext files walking through links connecting them and stores them in a local directory and registers them in an SQL (actually MySQL) database. Standard information like download (update) date and time, original URL, summary (if extractable) , document language and the list of links (together with information if already visited) is maintained by the robot.

A HT Base can be processed subsequently by an indexer and possibly an optimizer to form a Vector Base for the document collection. A Vector Base is a representation of a document space $\top n$ the space spanned by the words (terms) from the dictionary where the points in space represent documents.

A Vector Base is then transformed to a document map by a mapper process. A map is essentially a two-level clustering of documents: there are clusters of documents (stored in DocGR Base) and clusters of document clusters (stored in Map Base). Document clusters are assigned a graphical representation in terms of elementary "pixels" (labeled by appropriate phrases) in a visual representation, whereas clusters of document clusters are assigned "areas" consisting of "pixels". Note that in our approach we use a kind of multilevel maps, where higher levels "pixels" are "expanded" into maps/map fragments at a detailed level.

Note that the same HT Base may be processed by various indexers and optimizers so that out of a single HT Base many Vector bases may arise. Similarly one single Vector base may be processed by diverse mappers to form distinct maps. To keep track of the various descendants of the same HT Base, the Base Registry has been designed. The search engine makes use of all the maps representing the same HT Base choosing the one most appropriate for a given user query.

The search engine has been explicitly designed as a test-bed for various algorithmic solutions to constituent search engine components. Hence an important additional feature is a database keeping track of results of experiments (constituting of selections of process components and data sets as well as quality evaluation procedures). The database of experiments is filled (and used in case of continued experiments) by the special experiment management module.

2.5 Personalization and Recommendation

The outlined approach to document map oriented clustering enables personalization among others along the following lines:

- personalized topic -oriented initialization of map like visualization of the selected document space model (also rebuilding of a component model is possible, treating the user profile as a modifier of term weights of all documents);
- personalized identification of key words, document space / map cell labeling, query expansion;

- document recommendation based on document membership degree in client profile context;

- recommendation of map cells;

- recommendation of other users (measuring the histogram distances between user profiles);

- clustering of users as well as users and contexts.

Present day search engines are characterized by a static information model. This means that textual data bases are updated in a heavily discontinuous way which results in abrupt changes of query results (after each cycle of indexing new documents). On the other hand the data organization and search model does not take into account the user profile information for the given document base and the given user query. Hence the reply is frequently identical, independent of the user.

The experimental search engine BEATCA [10] exhibits several capabilities that can become a starting point for a radical change of this situation.

- reduced processing time, scalability of the adaptive contextual approach, reduced memory requirements of the implemented clustering algorithms (contextual reduction of the vector space) and search (inverted lists compression);

- possibility of construction and maintenance of multiple models/maps representing diverse views of the same document collection (and fitting the map to the query);

- possibility of inclusion of system-user interaction history into the algorithm of map initialization (e.g. by strengthening / weakening of terms from documents evaluated by the user as more or less interesting);

- possibility of inclusion of user preference profiles into the modeling process itself by taking into account the automatically collected information on user walk through the collection or provided externally.

We presented a new concept of document cluster characterization via term (importance) distribution histograms. This idea allows the clustering process to have a deeper insight into the role played by the term in formation of a particular cluster. So a full profit can be taken from our earlier idea of "contextual clustering", that is of representing different document clusters in different subspaces of a global vector space. We have also elaborated incremental methods of document cluster models based both on GNG model properties and histogram-based context adaptation. Such an approach to mining high dimensional datasets proved to be an effective solution to the problem of massive data clustering. The contextual approach appears to be fast, of good quality and scalable (with the data size and dimension). Additionally, the histogram-based characterization of document clusters proved to be a stabilizing factor in creating the clustering structure, and well suited for document classification. As a side effect, a new internal cluster quality measure, based on histograms, has been developed.

We believe that the idea of histogram-based subspace identification and evaluation can be efficiently applied not only to textual, but also other challenging high dimensional datasets (especially those characterized by attributes from heterogeneous or correlated distributions).

Contextual approach leads to many interesting research issues, such as context-dependent dictionary reduction and keywords identification, topic-sensitive document summarization, subjective model visualization based on particular user's information requirements, dynamic adaptation of the document representation and local similarity measure computation. Especially, the user-oriented, contextual data visualization can be a major step on the way to information retrieval personalization in search engines.

*T*he main lesson from our research is that personalization related to any Web data processing requires

- optimization of the entire process itself,
- a clear conceptual separation of the aspects that are objective (common for all envisaged users, not depending on their personal characteristics) from those that are really related to personal experience, balancing towards reduction of the latter ones in favour for the first ones,
- a data process level separation of procedures related to these two aspects,
- special care for optimization of processes driven by the person-related aspects of the data processing,
- personalization related processing should be always conceptually separated from group experience processing as that latter does not need to be run on-line, whereas the former to a large extent has to.

Our further research on personalization in document maps will be directed towards:

- chronology (sequence) of visiting the documents by the user versus the incremental growth of document collection;
- capturing the relationship between contexts and users and identification of *topical trends* versus random walk;
- exploitation of link structure (between documents and contexts) as modifiers of clustering space.

References

[1] A. Basu, I. R. Harris, and S. Basu. Minimum distance estimation: The approach using density-based distances. In G. S. Maddala and C. R. Rao, editors, Handbook of Statistics, volume 15, pages 21-48. North-Holland, 1997.

[2] J.C. Bezdek, S.K. Pal, Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data, IEEE, New York, 1992

[3] K. Ciesielski, S. Wierzchon, M. Klopotek, An Immune Network for Contextual Text Data Clustering, in: H.Bersini, J.Carneiro (Eds.), 5th International Conference on Artificial Immune Systems (ICARIS-2006), Oeiras, LNCS 4163, Springer-Verlag, 2006, pp.432-445

[4] K. Ciesielski, M. Klopotek, Towards Adaptive Web Mining: Histograms and Contexts in Text Data Clustering, to appear in: M.R.Berthold, J.Shawe-Taylor (Eds.), Intelligent Data Analysis – Proceedings of IDA-2007, Ljulbjana, September 2007, Springer-Verlag, LNCS 4723, pp.284-295

[5] K. Ciesielski, M. Klopotek, Text Data Clustering by Contextual Graphs, in: L.Todorovski, N.Lavrac, K.P.Jantke, 9th International Conf. on Discovery Science (ALT/DS 2006), Barcelona, LNAI 4265, Springer-Verlag, 2006, pp.65-76

[6] S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, R.A. Harshman, Indexing by Latent Semantic Analysis, Journal of the American Society of Information Science, 41(1990)6, ppp. 391-407, `citeseer.nj.nec.com/deerwester90indexing.html`

[7] M. Dittenbach, A. Rauber, D. Merkl, Uncovering hierarchical structure in data using the Growing Hierarchical Self-Organizing Map. Neurocomputing 48 (1-4)2002, pp. 199-216.

[8] B. Fritzke, A growing neural gas network learns topologies, in: G. Tesauro, D.S. Touretzky, and T.K. Leen (Eds.) Advances in Neural Information Processing Systems 7, MIT Press Cambridge, MA, 1995, pp. 625-632.

[9] C. Hung, S. Wermter, A constructive and hierarchical self-organising model in a non-stationary environment, Int.Joint Conference in Neural Networks, 2005

[10] M. Klopotek, S. Wierzchon, K. Ciesielski, M. Draminski, D. Czerski, Techniques and Technologies Behind Maps of Internet and Intranet Document Collections, in: Lu, Jie; Ruan, Da; Zhang, Guangquan (Eds.): E-Service Intelligence – Methodologies, Technologies and Applications. Springer-Verlag Series: Studies in Computational Intelligence, Vol. 37, 2007, X, 711 p., 190 illus., Hardcover, ISBN-10: 3-540-37015-3, ISBN-13: 978-3-540-37015-4

[11] T. Kohonen, Self-Organizing Maps, Springer Series in Information Sciences, vol. 30, Springer-Verlag, 2001

[12] T. Kohonen, S. Kaski, P. Somervuo, K. Lagus, M. Oja, V. Paatero, Self-organization of very large document collections, Helsinki University of Technology technical report, 2003, `http://www.cis.hut.fi/research/reports/biennial02-03`

[13] A. Rauber, Cluster Visualization in Unsupervised Neural Networks, Diplomarbeit, Technische Universität Wien, Austria, 1996

[14] J. Timmis, aiVIS: Artificial Immune Network Visualization, in: Proceedings of EuroGraphics UK 2001 Conference, Univeristy College London 2001, pp.61-69

3 Curriculum modelling, planning and validation

The birth of the Semantic Web brought along standard models, languages, and tools for representing and dealing with machine-interpretable semantic descriptions of Web resources, giving a strong new impulse to research on personalization. The introduction of machine-processable semantics makes the use of a variety of reasoning techniques, for implementing personalization functionalities, possible, thus widening the range of the forms that personalization can assume.

So far, reasoning in the Semantic Web is mostly *reasoning about knowledge*, expressed in some ontology. However personalization may involve also other kinds of reasoning and representations of knowledge, that conceptually lie at the *logic* and *proof* layers of the Semantic Web tower. Moreover, the next Web generation promises to deliver Semantic Web Services, that can be retrieved and *combined* in a way that satisfies the user. This perspective opens the way to many forms of *service-oriented personalization.* In fact, web services provide an ideal infrastructure for enabling *interoperability* among personalization applications and for constructing Plug&Play-like environments, where the user can select and combine the kinds of services he or she prefers. Personalization can be obtained by taking different approaches, e.g. by developing services that offer personalization functionalities as well as by personalizing the way in which services are selected, and composed in order to meet specific user's requirements.

In the last years we have carried on a research in the educational domain, focussing on *semantic web* representations of learning resources and on *automated reasoning* techniques for enabling different and complementary personalization functionalities, such as curriculum sequencing [5, 4] and the verification of the compliance of a curriculum against some "course design" goals [6]. Our current aim is to implement such results in an organic system, where different personalization services, that exploit reasoning over semantic web representations of *learning resources*, can be combined to support the user in the task of building a personalized curriculum.

While in early times learning resources were simply considered as "contents", which required a specific platform to be used, recently, greater and greater attention has been posed on the issues of: (1) the *re*-use of learning resources, and (2) the development of standards that allow a *cross-platform* usage. The solution that we propose is to adopt a *semantic annotation* of contents based on standard languages, e.g. RDF and LOM. Hereafter, we will consider a *learning resource* as formed by *educational contents* plus *semantic meta-data*, which supply information on the resources at a *knowledge level*, i.e. on the basis of concepts, which have been taken from an ontology describing the educational domain. In particular, we rely on the interpretation of learning resources as *actions* discussed in [5, 4]: the meta-data captures the *learning objectives* of the learning resource and its *pre-requisites.* By doing so, one can rely on a classical theory of actions and apply different reasoning methods –like *planning*– for building personalized curricula [5, 4].

Curriculum planning and validation offer a useful support in many practical contexts and can fruitfully be combined for helping students as well as educational institutions. Often a student knows what competency he/she would like to acquire but has no knowledge of which courses will help acquiring it. Moreover, taking courses at different Universities is becoming more and more common in Europe. As a consequence, building a curriculum might become a complicated task for students, who must deal with an enormous set of courses across the European countries, each described in different languages and on the basis of different keywords.

The need of personalizing the sequencing of learning resource, w.r.t. the student's interests and context, has often to be *combined* with the ability to check that the resulting curriculum *complies* to some abstract *specification*, which encodes the *curricula-design goals*, expressed by the teachers or by the institution offering the courses. One possible scenario is the following: a student builds a personalized curriculum, either by modifying a curriculum, which has been suggested by the institution or by an automatic goal-driven system, or by writing one by him-/herself, based on personal criteria. In both cases, the obtained (personalized) curriculum is to be proved consistent, from an educational point of view. To this aim, it would be helpful to give the curriculum in input to some validation service, and check its compliance to the desired curricula model. Generally speaking, curricula models specify general rules for building learning paths. We interpret curricula models as constraints, designed by the University (or other institution) for guaranteeing the acquisition of certain competencies. These constraints are to be expressed in terms of *knowledge elements*, and maybe also on features that characterize the resources. Consider, now, a University, which needs to certify that the curricula, offered by it for acquiring certain competencies (built upon the courses offered locally by the University itself), respect the latest guidelines supplied by the European Community. Also in this case, we could define the guidelines as a set of constraints posed on the set of offered competencies. Given this representation, the verification could be performed automatically, by means of a proper reasoner. Finally, the automatic checking of compliance combined with curriculum planning could be used for implementing processes like cooperation among institutes in curricula design and integration, which are actually the focus of the so called *Bologna Process* [10], promoted by the EU. Given our requirements, it is a natural choice to settle our implementation in the Personal Reader (PR) framework [18]. The PR relies on a service-oriented architecture enabling personalization, via the use of semantic *Personalization Services*. Each service offers a different personalization functionality, e.g. recommendations tailored to the needs of specific users, pointers to related (or interesting or more detailed/general) information, and so on. These semantic web services communicate solely based on RDF documents. In the following we present our achievements both from a theoretical point of view, and by presenting the implementation of a Planning service and a Validation service, which can interoperate within the PR Framework [18].

3.1 Curricula representation and reasoning

Let us begin with the introduction of our approach to the representation of learning resources, curricula, and curricula models. The basic idea is to describe all the different kinds of objects, that we need to tackle and that we will introduce hereafter, on the basis of a set of predefined *competencies*, i.e. terms identifying specific *knowledge elements*. We will use the two terms as synonyms. Competencies can be thought of, and implemented, as concepts in a shared ontology. In particular, for what concerns the application system described here, competencies were extracted by means of a semi-automatic process and stored as an RDF file (see Section 3.3.1 for details).

Given a predefined set of competencies, the initial knowledge of a student can be represented as a set of such concepts. This set changes, typically it grows, as the student studies and learns. In the same way, a user, who accesses a repository of learning resources, does it with the aim of finding materials that will allow him/her to acquire some knowledge of interest. Also this knowledge, that we identify by the term *learning goal*, can be represented as a set of knowledge elements. The learning goal is to be taken into account in a variety of tasks. For instance, the construction of a personalized curriculum is, actually, the construction of a curriculum which allows the achievement of a learning goal expressed by the user. In Section 3.3 we will describe a *curricula planning service* for accomplishing this task.

3.1.1 Learning resources and curricula

A *curriculum* is a sequence of *learning resources* that are homogeneous in their representation. Based on work in [5, 4], we rely on an *action theory*, and take the abstraction of resources as *simple actions*. More specifically, a learning resource is modelled as an action for acquiring some competencies (called *effects*). In order to understand the contents supplied by a learning resource, the user is sometimes required to own other competencies, that we call *preconditions.* Both preconditions and effects can be expressed by means of a *semantic annotation* of the learning resource [4]. In the following we will often refer to learning resources as "courses" because in our work we have focussed on the specific case of University curricula. As a simple example of "learning resource as action", let us, then, report the possible representation (in a classical STRIPS-like notation) of the course "databases for biotechnologies" (*db_for_biotech* for short):

ACTION: db_for_biothec(),
PREREQ: relational_db, EFFECTS: scientific_db

The prequisites to this action is to have knowledge about *relational databases.* Its effect is to supply knowledge about *scientific databases.*

Given the above interpretation of learning resources, a *curriculum* can be interpreted as a *plan*, i.e. as a sequence of actions, whose execution causes transitions from a state to another, until some final state is reached. The *initial state* contains all the competences that we suppose available before the curriculum is taken, e.g. the knowledge that the student already has. This set can also be empty. The *final state* is sometimes required to contain specific knowledge elements, for instance, all those that compose the user's learning goal. Indeed, often curricula are designed so to allow the achievement of a well-defined *learning goal.* A transition between two states is due to the application of the action corresponding to a learning resource. Of course, for an action to be applicable, its preconditions must hold in the state to which it should be applied. The application of the action consists in an *update* of the state.

We assume that competences can only be added to states. Formally, this amount to assume that the domain is *monotonic.* The intuition behind this assumption is that the act of using a new learning resource (e.g. attending a course) will not erase from the student's memory the concepts acquired so far. Knowledge grows incrementally.

3.1.2 DCML: A Declarative Curricula Model Language

Let us, now, describe the *Declarative Curricula Model Language* (DCML, for short), a graphical language that we have developed to represent the specification of curricula models. The advantage of a graphical language is that the fact of *drawing*, rather than *writing*, constraints *facilitates the user*, who needs to represent curricula models, allowing a general overview of the relations between concepts. We present here, the most general version of DCML, in which a distinction is made between the terms *competence* and *competency.* These two terms are respectively used, in the literature concerning professional curricula and e-learning, to denote the "effective performance within a domain at some level of proficiency" and "any form of knowledge, skill, attitude, ability or learning objective that can be described in a context of learning, education or training". Previous versions can be found in [6, 9].

DCML is inspired by DecSerFlow, the Declarative Service Flow Language to specify, enact, and monitor web service flows by van der Aalst and Pesic [1]. DCML, as well as DecSerFlow, is grounded in Linear Temporal Logic [13] and allows a curricula model to be described in an easy way maintaining at the same time a rigorous and precise meaning given by the logic representation. LTL includes temporal operators such as next-time ($\bigcirc\varphi$, the formula φ holds in the immediately following state of the run), eventually ($\Diamond\varphi$, φ is guaranteed to eventually become true), always ($\Box\varphi$, the formula φ remains invariably true throughout a run), until

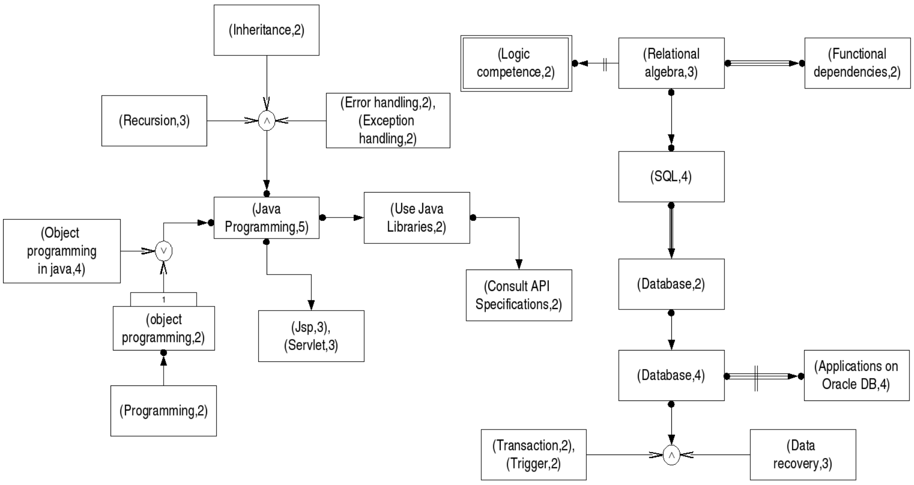


Figure 1: An example of curricula model in DCML.

(α U β, the formula α remains true until β), see also [19, Chapter 6]. The set of LTL formulas obtained for a curricula model are, then, used to verify whether a curriculum will respect it [6].

As an example, Fig. 1 shows a curricula model expressed in DCML. Every box contains at least one competence. Boxes/competences are connected by arrows, which represent (mainly) temporal constraints among the times at which they are to be acquired. Altogether the constraints describe a curricula model.

DCML, as mentioned, includes the representation of the *proficiency level* at which a competency is owned or supplied. To this aim, we associate to each competency a variable k, having the same name as the competency, which can be assigned natural numbers as values. The value of k denotes the proficiency level; zero means absence of knowledge. Therefore, k encodes a *competence*, Fig. 2(a).

On top of competences, in DCML it is possible to define three basic *constraints*. The "*level of competence*" constraint, Fig. 2(c), imposes that a certain competency k must be acquired at least at level l. It is represented by the LTL formula $\Diamond(k \geq l)$. Similarly, a course designer can impose that a competency must never appear in a curriculum with a proficiency level higher than l. This is possible by means of the "*always less than level*" constraint, shown in Fig. 2(d). The LTL formula $\Box(k < l)$ expresses this fact (it is the negation of the previous one). As a special case, when the level l is one ($\Box(k < 1)$), the competency k must never appear in a curriculum.

The third constraint, represented by a double box, see Fig. 2 (b), specifies that k must belong to the initial knowledge with, at least, level l. In other words, the simple logic formula $(k \geq l)$ must hold in the initial state.

To specify relations among concepts, other elements are needed. In particular, in DCML it is possible to represent *Disjunctive Normal Form* (DNF) formulas as *conjunctions* and *disjunctions* of concepts. For lack of space, we do not describe the notation here, however, an example can be seen in Fig. 1.

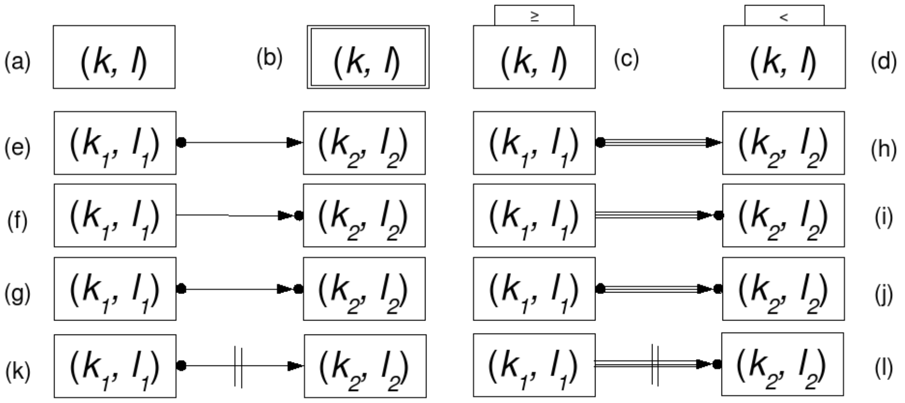


Figure 2: Competences (a) and basic constraints (b), (c), and (d). Relations among competences: (a) implication, (b) before, (c) succession, (d) immediate implication, (e) immediate before, (f) immediate succession, (g) not implication, (h) not immediate before.

Besides the representation of competences and of constraints on competences, DCML allows to represent *relations* among competences. For simplicity, in the following presentations we will always relate simple competences, it is, however, of course possible to connect DNF formulas. We will denote by (k, l) the fact that competence k is required to have at least level l (i.e. $k \geq l$) and by $\neg(k, l)$ the fact that k is required to be less than l.

Arrows ending with a little-ball, Fig. 2(f), express the *before* temporal constraint between two competences, that amount to require that (k_1, l_1) holds *before* (k_2, l_2). This constraint can be used to express that to understand some topic, some proficiency of another is required as precondition. It is important to underline that if the antecedent never becomes true, also the consequent must be invariably false; this is expressed by the LTL formula $\neg(k_2, l_2) \mathsf{U} (k_1, l_1)$, i.e. $(k_2 < l_2) \mathsf{U} (k_1 \geq l_1)$. It is also possible to express that a competence must be acquired *immediate before* some other. This is represented by means of a triple line arrow that ends with a little-ball, see Fig. 2(i). The constraint (k_1, l_1) *immediate before* (k_2, l_2) imposes that (k_1, l_1) holds before (k_2, l_2) and the latter either is true in the next state w.r.t. the one in which (k_1, l_1) becomes true or k_2 *never* reaches the level l_2. The difference w.r.t the *before* constraint is that it imposes that the two competences are acquired *in sequence*. The corresponding LTL formula is "(k_1, l_1) *before* (k_2, l_2)" $\wedge\Box((k_1, l_1) \supset (\bigcirc(k_2, l_2) \vee \Box\neg(k_2, l_2)))$.

Both of the two previous relations represent temporal constraints between competences. The *implication* relation (Fig. 2(e)) specifies, instead, that if a competency k_1 holds at least at the level l_1, some other competency k_2 must be acquired sooner or later at least at the level l_2. The main characteristic of the implication, is that the acquisition of the consequent is imposed by the truth value of the antecedent, but, in case this one is true, it does not specify when the consequent must be achieved (it could be before, after or in the same state of the antecedent). This is expressed by the LTL formula $\Diamond(k_1, l_1) \supset \Diamond(k_2, l_2)$. The *immediate implication* (Fig. 2(h)), instead, specifies that the consequent must *hold* in the state right after the one in which the antecedent is acquired. Note that, this does not mean that it must be *acquired* in that state, but only that it cannot be acquired after. This is expressed by the LTL implication formula in conjunction with the constraint that whenever $k_1 \geq l_1$ holds, $k_2 \geq l_2$

holds in the next state: $\Diamond(k_1, l_1) \supset \Diamond(k_2, l_2) \wedge \Box((k_1, l_1) \supset \bigcirc(k_2, l_2))$.

The last two kinds of temporal constraint are *succession* (Fig. 2(g)) and *immediate succession* (Fig. 2(j)). The *succession* relation specifies that if (k_1, l_1) is acquired, afterwards (k_2, l_2) is also achieved; otherwise, the level of k_2 is not important. This is a difference w.r.t. the *before* constraint where, when the antecedent is never acquired, the consequent must be invariably false. Indeed, the *succession* specifies a condition of the kind *if* $k_1 \geq l_1$ *then* $k_2 \geq l_2$, while *before* represents a constraint without any conditional premise. Instead, the fact that the consequent must be acquired after the antecedent is what differentiates *implication* from *succession*. Succession constraint is expressed by the LTL formula $\Diamond(k_1, l_1) \supset (\Diamond(k_2, l_2) \wedge (\neg(k_2, l_2) \mathsf{U} (k_1, l_1)))$. In the same way, the *immediate succession* imposes that the consequent either is acquired in the same state as the antecedent or in the state immediately after (not before nor later). The immediate succession LTL formula is "(k_1, l_1) *succession* (k_2, l_2)" $\wedge \Box((k_1, l_1) \supset \bigcirc(k_2, l_2))$.

After the "positive relations" among competences, let us now introduce the graphical notations for "negative relations". The graphical representation is very intuitive: two vertical lines break the arrow that represents the constraint, see Fig. 2(k)-(l). (k_1, l_1) *not before* (k_2, l_2) specifies that k_1 cannot be acquired up to level l_1 before or in the same state when (k_2, l_2) is acquired. The corresponding LTL formula is $\neg(k_1, l_1) \mathsf{U} ((k_2, l_2) \wedge \neg(k_1, l_1))$. Notice that this is not obtained by simply negating the before relation but it is weaker; the negation of *before* would *impose the acquisition* of the concepts specified as consequents (in fact, the formula would contain a strong until instead of a weak until), the *not before* does not. The *not immediate before* is translated exactly in the same way as the *not before*. Indeed, it is a special case because our domain is monotonic, that is a competency acquired at a certain level cannot be forgotten.

(k_1, l_1) *not implies* (k_2, l_2) expresses that if (k_1, l_1) is acquired k_2 cannot be acquired at level l_2; as an LTL formula: $\Diamond(k_1, l_1) \supset \Box\neg(k_2, l_2)$. Again, we choose to use a weaker formula than the natural negation of the implication relation because the simple negation of formulas would impose the presence of certain concepts. (k_1, l_1) *not immediate implies* (k_2, l_2) imposes that when (k_1, l_1) holds in a state, $k_2 \geq l_2$ must be false in the immediately subsequent state. Afterwards, the proficiency level of k_2 does not matter. The corresponding LTL formula is $\Diamond(k_1, l_1) \supset (\Box\neg(k_2, l_2) \vee \Diamond((k_1, l_1) \wedge \bigcirc\neg(k_2, l_2)))$, that is weaker than the "classical negation" of the *immediate implies*.

The last relations are *not succession*, and *not immediate succession*. The first imposes that a certain competence cannot be acquired after another, (either it was acquired before, or it will never be acquired). As LTL formula, it is $\Diamond(k_1, l_1) \supset (\Box\neg(k_2, l_2) \vee$ "(k_1, l_1) *not before* (k_2, l_2)"). The second imposes that if a competence is acquired in a certain state, in the state that follows, another competence must be false, that is $\Diamond(k_1, l_1) \supset (\Box\neg(k_2, l_2) \vee$ "(k_1, l_1) *not before* (k_2, l_2)" $\vee \Diamond((k_1, l_1) \wedge \bigcirc\neg(k_2, l_2)))$.

3.1.3 Representing curricula as activity diagrams

Let us now consider specific curricula. In the line of [5, 3, 6], curricula are, in our model, sequences of learning resources (courses), taking the abstraction of learning resources (courses) as simple actions, which cause subsequent state transitions, from the initial set of competences of a user (which can possibly be empty) up to a final state, which contains all the competences that are owned by the user in the end.

We represent curricula as UML *activity diagrams* [24], normally used for representing *business processes*. We decided to do so, because activity diagrams allow to capture in a natural

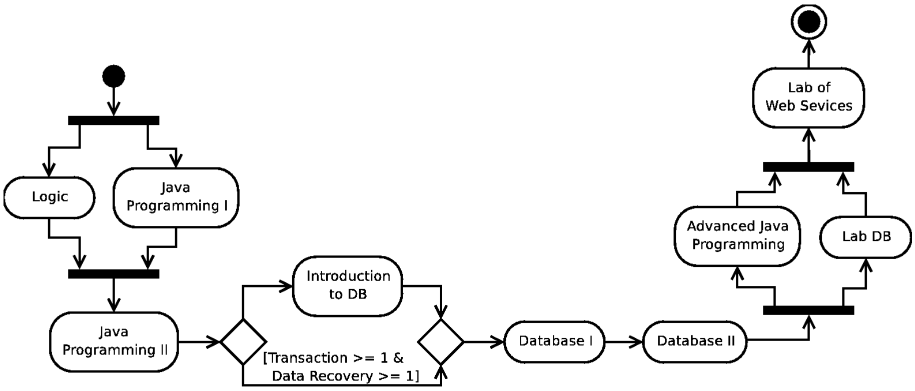


Figure 3: Activity diagram representing a set of eight different curricula. Notice that *Logic* and *Java Programming I* can be attended in any order (even in parallel), as well as *Advanced Java Programming* and *Lab DB*, while *Introduction to DB* will be considered only if the guard *Transaction* and *Data Recovery* is false.

way the simple sequencing of courses as well as the possibility of attending courses in *parallel* or in possibly conditioned *alternatives.*

An example is reported in Fig. 3. Besides the initial and the final nodes, the graphical elements used in an activity diagram are: *activity nodes* (rounded rectangle) that represent activities (attending courses) that occur; *flow/edge* (arrows) that represent activity flows; *fork* (black bar with one incoming edge and several outgoing edges) and *join nodes* (black bar with several incoming edges and one outgoing edge) to denote parallel activities; and *decision* (diamonds with one incoming edge and several outgoing edges) and *merge nodes* (diamonds with several incoming edges and one outgoing edge) to choose between alternative flows.

In the modeling of *learning processes*, we use activities to represent the act of attending courses (or using learning resources). For example, by fork and join nodes we represent the fact that two (or more) courses or sub-curricula are not related and, it is possible for the student to attend them in parallel. This is the case of *Java Programming I* and *Logic*, as well as *Advanced Java Programming* and *Lab. of DB* showed in Fig. 3. Till all parallel branches have not been attended successfully, the student cannot attend other courses, even if some of the parallel branches have been completed. Parallel branches can also be used when we want to express that the order among courses of different branches does not matter.

Decision and merge nodes can be used to represent alternative paths. The student will follow only one of the possible branches. Alternative paths can also be conditioned, in this case a *guard*, a boolean condition, is added at the beginning of the branch. Guards should be mutually exclusive. In our domain, the conditions are expressed in terms of concepts that must hold, otherwise a branch is not accessible. If no guard is present, the student can choose one (and only one) of the possible paths. In the example in Fig. 3, the guard consists of two competences: *Transaction* and *Data Recovery.* If one of these does not hold the student has to attend the course *Introduction to DB*, otherwise he/she does not.

3.2 Planning and Validation

Given a semantic annotation with preconditions and effects of the courses, classical planning techniques are exploited for creating *personalized curricula*, in the spirit of the work in [4]. Intuitively the idea is that, given a repository of learning resources, which have been semantically annotated, the user expresses a *learning goal* as a set of *knowledge elements* he/she would like to acquire, and possibly also a set of already owned competencies. Then, the system applies planning to build a sequence of learning resources that, read in sequence, will allow him/her to achieve the goal.

The particular planning methodology that we implemented (see Section 3.3.3 for details) is a simple *depth-first forward planning* (an early prototype was presented in [2]), where actions cannot be applied more than once. The algorithm is simple:

1. Starting from the initial state, the set of *applicable* actions (those whose preconditions are contained in the current state) is identified.

2. One of such actions is selected and its application is simulated leading to a new state.

3. The new state is obtained by adding to the previous one the competencies supplied as effects of the selected action.

4. The procedure is repeated until either the goal is reached or a state is reached, in which no action can be applied and the learning goal is not satisfied.

5. In the latter situation, backtracking is applied to look for another solution.

The procedure will eventually end because the set of possible actions is finite and each is applied at most once. If the goal is achieved, the sequence of actions that label the transitions leading from the initial to the final state is returned as the resulting *curriculum.* If desired, the backtracking mechanism allows to collect a set of alternative solutions to present to the user.

Besides the capability of automatically building personalized curricula, it is also interesting to perform a set of verification tasks on curricula and curricula models. The simplest form of verification consists in *checking the soundness* of curricula which are built by hand by users themselves, reflecting their own personal interests and needs. Of course, not all sequences which can be built starting from a set of learning resources are lawful. Constraints, imposed by courses themselves, in terms of preconditions and effects, must be respected. In other words, a course can appear at a certain point in a sequence only if it is *applicable* at that point, leaving no *competency gaps.* These implicit "applicability constraints" capture precedences and dependencies that are innate to the nature of the taught concepts. In particular, it is important to verify that all the *competencies*, that are necessary to fully understand the contents, offered by a learning resource, are introduced or available before that learning resource is accessed. Usually, this verification, as stated in [20], is performed manually by the learning designer, with hardly any guidelines or support.

A recent proposal for automatizing the competence gap verification is done in [4] where an analysis of pre- and post-requisite annotations of the Learning Objects (LO), representing the learning resources, is proposed. A logic based validation engine can use these annotations in order to validate the curriculum/LO composition. Melia and Pahl's proposal is inspired by the CocoA system [11], that allows to perform the analysis and the consistency check of static web-based courses. Competence gaps are checked by a prerequisite checker for *linear courses*, simulating the process of teaching with an overlay student model. Pre- and post-requisites are represented as "concepts".

Together with the verification of consistence gaps, there are other kinds of verification. Brusilovsky and Vassileva [11] sketch some of them. In our opinion, two are particularly important: (a) verifying that the curriculum allows to achieve the users' *learning goals*, i.e. that the user will acquire the desired knowledge, and (b) verifying that the curriculum is compliant against the *course design goals*. Manually or automatically supplied curricula, developed to reach a learning goal, should match the "design document", a *curricula model*, specified by the institution that offers the possibility of personalizing curricula. Curricula models specify general rules for designing sequences of learning resources (courses). We interpret them as *constraints*, that are expressed in terms of concepts and, in general, are not directly associated to learning resources, as instead is done for pre-requisites. They constrain the process of acquisition of concepts, independently from the resources.

Given the interpretation of resources as actions, the verification of the *soundness of a curriculum*, w.r.t. the learning dependencies and the learning goal, can be interpreted as an *executability check* of the curriculum. Also in this case, the algorithm is simple:

1. Given an initial state, representing the knowledge available before the curriculum is attended, a simulation is executed, in which all the actions in the curriculum are (virtually) executed one after the other.

2. An action (representing a course) can be executed only if the current state contains all the concepts that are in the course precondition. Intuitively, it will be applied only if the student owns the notions that are required for understanding the topics of the course.

3. If, at a certain point, an action that should be applied is *not applicable* because some precondition does not hold, the verification fails and the reasons of such failure can be reported to the user.

4. Given that all the courses in the sequence can be applied, one after the other, the final state that is reached must be compared with the learning goal of the student: all the desired goal concepts must be achieved, so the corresponding knowledge elements must be contained in the final state.

Such verification task can be accomplished by the service described in Section 3.3.4.

Concerning compliance against course design goals, it amounts to determine if a curriculum satisfies the constraints posed by a specification given by the designer. This specification is captured by a "curricula model". A curricula model imposes constraints on *what* to achieve and *how* achieving it. In our proposal the verification is done by applying *model checking* and, in particular, the SPIN model checker [19]. SPIN is used for verifying systems that can be represented by *finite state structures*, where the specification is given in an LTL logic. The verification algorithm is based on the exploration of the state space. This is exactly what we need for performing all the validation tests that we mentioned, provided that we can translate the curriculum in the internal representation used by the model checker. In the case of SPIN, the internal representation is given in the Promela language. The implementation of the validation service is described in Section 3.3.4.

For representing curricula models, we have developed DCML, see Section 3.1.2. We have seen in this section that the consraints, given in DCML, can be translated into formulas in a temporal logic (LTL, linear temporal logic [13]). This logic allows the verification of properties of interest for all the possible executions of a model, which in our case corresponds to the curriculum at issue. To exploit this feature, we translate the activity diagram, that represents a curriculum to be checked, in a *Promela* program [19]. Afterwards, we check that the program satisfies the LTL formulas corresponding to the curricula model of interest. As in [20], in the representation that we have developed we distinguish between *competency* and *competence*, where by the first term we denote a concept (or skill) while by the second we denote a competency plus the level of proficiency at which it is learnt or known or supplied. So far, we do not yet tackle with "contexts", as defined in the competence model proposed in [20], which will be part of future work.

The approach to the representation and reasoning about curricula moels, that we have briefly described here, differs from previous work [5], where we presented an adaptive tutoring system, that exploits *reasoning about actions and changes* to plan and verify curricula. The approach was based on abstract representations, capturing the *structure* of a curriculum, and implemented by means of prolog-like logic clauses. Such representations were applied a procedure-driven form of planning, in order to build personalized curricula. In this context, we proposed also some forms of verification, of competence gaps, of learning goal achievement, and of whether a curriculum, given by a user, is compliant to the "course design" goals. The use of procedure clauses is, however, limiting because they, besides having a *prescriptive* nature, pose very strong constraints on the sequencing of learning resources. In particular, clauses represent what is "legal" and whatever sequence is not foreseen by the clauses is "illegal". However, in an open environment where resources are extremely various, they are added/removed dynamically, and their number is huge, this approach becomes unfeasible: the clauses would be too complex, it would be impossible to consider all the alternatives and the clauses should change along time.

For this reason we considered as appropriate to take another perspective and represent only those constraints which are strictly necessary, in a way that is inspired by the so called *social approach* proposed by Singh for multi-agent and service-oriented communication protocols [21, 22]. In this approach only the *obligations* are represented. In our application context, obligations capture relations among the times at which different competencies are to be acquired. The advantage of this representation is that we do not have to represent all that is legal but only those *necessary conditions* that characterize a legal solution. To make an example, by means of constraints we can request that a certain knowledge is acquired before some other knowledge, without expressing what else is to be done in between. If we used the clause-based approach, instead, we should have described also what can legally be contained between the two times at which the two pieces of knowledge are acquired. Generally, the constraints-based approach is more flexible and more suitable to an open environment.

3.3 Implementation in the Personal Reader Framework

The Personal Reader Framework has been developed with the aim of offering a uniform entry point for accessing the Semantic Web, and in particular Semantic Web Services. Indeed it offers an environment for designing, implementing and realizing Web content readers in a service-oriented approach, for a more detailed description, see [18] (`http://www.personal-reader.de/`).

In applications based on the Personal Reader Framework, a user can select and combine —plug together— which personalized support he or she wants to receive. The framework has already been used for developing Web Content Readers that present online material in an embedded context [8, 17].

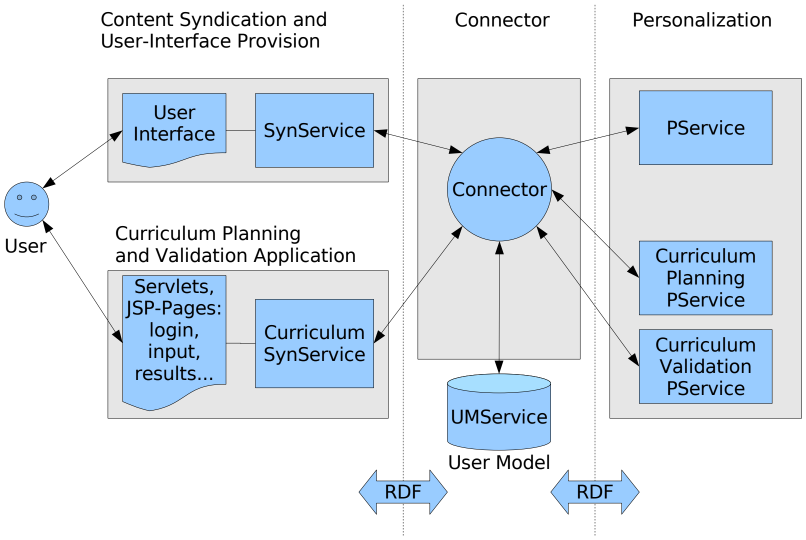


Figure 4: Personal Reader Framework Overview

Besides a user-interface, as shown in figure 4, a Personal Reader application consists of three types of *services*. *Personalization services*(PService) provide personalization functionalities: they deliver personalized recommendations for content, as requested by the user and obtained or extracted from the Semantic Web. *Syndication Services* (SynService) allow for some interoperability with the other services in the framework, e.g. for the discovery of the applications interfaces by a portal. The *Connector* is a single central instance responsible for all the communication between user interface and personalization services. It selects services based on their semantic description and on the requirements by the SynService. The Connector protects –by means of a public-key-infrastructure (PKI)– the communication among the involved parties. It also supports the customization and invocation of services and interacts with a user modelling service, called the *UMService*, which maintains a central user model.

3.3.1 Metadata Description of Courses

In order to create the corpus of courses, we started with information collected from an existing database of courses. We used the Lixto [7] tool to extract the needed data from the web-pages provided by the HIS-LSF (http://www.his.de/) system of the University of Hannover. This approach was chosen based on our experience with Lixto in the *Personal Publication Reader* [8] project, where we used Lixto for creating the publications database by crawling the publication pages of the project partners. The effort to adapt our existing tool for the new data source was only small. From the extracted metadata we created an RDF document, containing course names, course catalog identifier, semester, number of credit points, effects and preconditions, and the type of course, e.g. laboratory, seminar or regular course with examinations in the end, as illustrated in Figure 5.

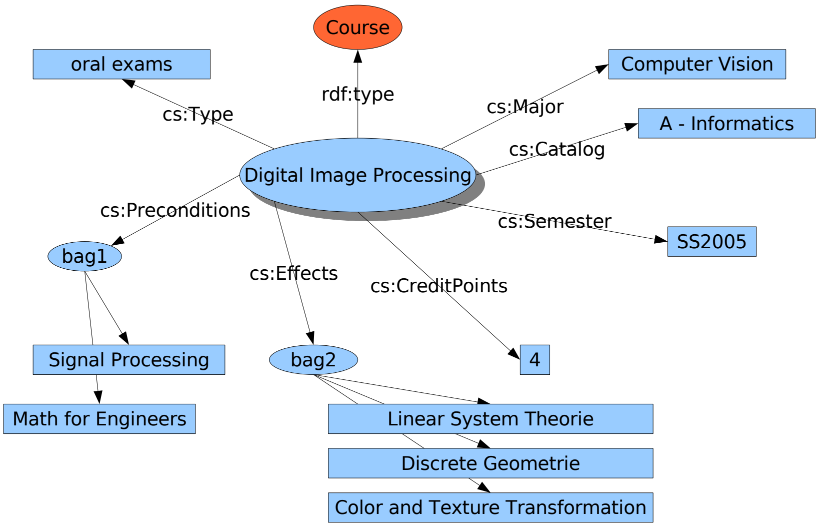


Figure 5: An annotated course from the Hannover course database

The larger problem was that the quality of most of the information in the database turned out to be insufficient, mostly due to inconsistencies in the description of prerequisites and effects of the courses. Additionally the corpus was not annotated using a common set of terms, but authors and department secretaries used a slightly varying vocabulary for each of their course descriptions, instead of relying on a common classification system, like e.g. the ACM CCS for computer science.

As a consequence, we focussed only on a subset of the courses (computer science and engineering courses), and manually post-processed the data. Courses are annotated with prerequisites and effects, that can be seen as knowledge concepts or competences, i.e. ontology terms. After automatic extraction of effects and preconditions, the collected terms were translated into proper English language, synonyms were removed and annotations were corrected where necessary. The resulting corpus had a total of 65 courses left, with 390 effects and 146 preconditions.

3.3.2 The User Interface and Syndication Service

In our implementation, the user interface (see figure 6) is responsible for identifying the user, presenting the user an interface to select the knowledge she wants to acquire, and to display the results of the planning and validation step, allowing further refinement of created plans. The creation of curriculum sequences and the validation are implemented as two independent

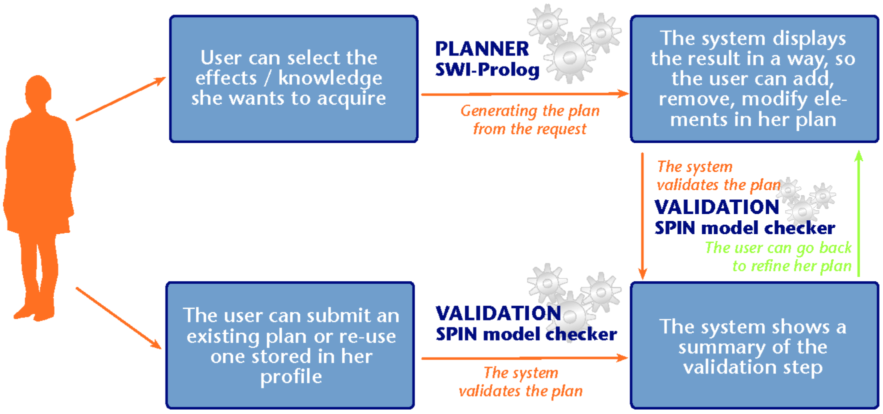

User can select the effects / knowledge she wants to acquire
PLANNER
SWI-Prolog
Generating the plan from the request
The system displays the result in a way, so the user can add, remove, modify elements in her plan
The system validates the plan
VALIDATION
SPIN model checker
The user can go back to refine her plan
The user can submit an existing plan or re-use one stored in her profile
VALIDATION
SPIN model checker
The system validates the plan
The system shows a summary of the validation step

Figure 6: The Actions supported by the User Interface

Personalization Services, the "Curriculum Planning PService", and the "Curriculum Validation PService". Because of the plug-and-play nature of the infrastructure, the two PServices can be used by other applications (SynServices) as well (Fig. 6). Also possible is that PServices, which provide additional planning and validation capabilities can be used in our application. The current and upcoming future implementations of the Curriculum Planning and Validation Prototype are available at `http://semweb2.kbs.uni-hannover.de:8080/plannersvc`.

3.3.3 The Curriculum Planning PService

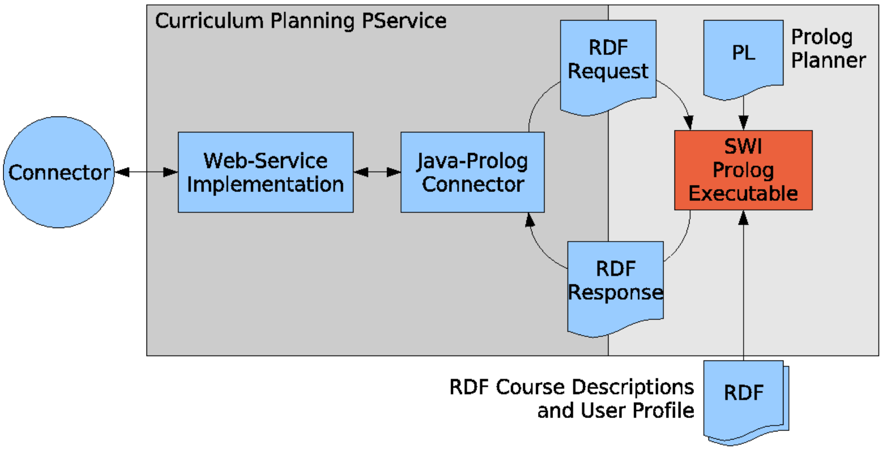

Curriculum Planning PService
RDF Request
PL
Prolog Planner
Connector
Web-Service Implementation
Java-Prolog Connector
SWI Prolog Executable
RDF Response
RDF Course Descriptions and User Profile
RDF

Figure 7: Curriculum Planning Web Service

In order to integrate the Planning Service as a plug-and-play personalization service in the Personal Reader architecture we worked at embedding the Prolog reasoner into a web service.

Figure 7 gives an overview over the components in the current implementation. The web service implements the Personalization Service (*PService* [18]) interface, defined by the Personal Reader framework, which allows for the processing of RDF documents and for inquiring about the services capabilities. The *Java-to-Prolog Connector* runs the SWI-Prolog executable in a sub-process; essentially it passes the RDF document containing the request *as-is* to the Prolog system, and collects the results, already represented as RDF.

The curriculum planning task itself is accomplished by a reasoning engine, which has been implemented in SWI Prolog[2]. The interesting thing of using SWI Prolog is that it contains a semantic web library allowing to deal with RDF statements. Since all the inputs are sent to the reasoner in a *RDF request document*, it actually simplifies the process of interfacing the planner with the Personal Reader. In particular the request document contains: a) links to the RDF document containing the database of courses, annotated with metadata, b) a reference to the user's context c) the user's actual learning goal, i.e. a set of knowledge concepts that the user would like to acquire, and that are part of the *domain ontology* used for the semantic annotation of the actual courses. The reasoner can also deal with information about credits provided by the courses, when the user sets a credit constraint together with the learning goal.

Given a request, the reasoner runs the Prolog planning engine on the database of courses annotated with prerequisites and effects. The initial state is set by using information about the user's context, which is maintained by the User Modelling component of the PR. In fact such user's context includes information about what is considered as already learnt by the student (attended courses, learnt concepts) and such information is included in the request document. The Prolog planning engine has been implemented by using a classical depth-first search algorithm. This algorithm is extremely simple to implement in declarative languages as Prolog.

At the end of the process, a *RDF response document* is returned as an output. It contains a list of plans (sequences of courses) that fulfill the user's learning goals and profile. The maximum number of possible solutions can be set by the user in the request document. Notice that further information stored in the user profile is used at this stage for adapting the presentation of the solutions, here simple hints are used to *rank higher* those plans that include topics that the user has an expressed special interest in.

3.3.4 The Curriculum Validation PService

In this section we discuss how to validate a curriculum. As explained, three kinds of verifications have to be performed: (1) verifying that a curriculum does not have competence gaps, (2) verifying that a curriculum supplies the user's learning goals, and (3) verifying that a curriculum satisfies the course design goals, i.e. the constraints imposed by the curricula model. To do this, we use *model checking techniques* [12].

By means of a *model checker*, it is possible to generate and analyze all the possible states of a program exhaustively to verify whether no execution path satisfies a certain property, usually expressed by a temporal logic, such as LTL. When a model checker refuses the negation of a property, it produces a *counterexample* that shows the violation. SPIN, by G. J. Holzmann [19], is the most representative tool of this kind. Our idea is to translate the activity diagram, that represents a set of curricula, in a Promela (the language used by SPIN) program, and, then to verify whether it satisfies the LTL formulas that represents the curricula model.

[2] `http://www.swi-prolog.org/`

In the literature, we can find some proposals to translate UML activity diagrams into Promela programs, such as [14, 16]. However, these proposals have a different purpose than ours and they cannot be used to perform the translation that we need to perform the verifications we list above. Generally, their aim is debugging UML designs, by helping UML designers to write sound diagrams. The translation proposed in the following, instead, aims to simulate, by a Promela program the acquisition of competencies by attending courses contained into the curricula represented by an activity diagram.

Given a curriculum as an activity diagram, we represent all the competences involved by its courses as *integer variables*. In the beginning, only those variables that represent the initial knowledge owned by the student are set to a value greater than zero. *Courses* are represented as actions that can modify the value of such variables. Since our application domain is monotonic, the value of a variable can only grow.

The Promela program consists of two main processes: one is called *CurriculumVerification* and the other *UpdateState*. While the former contains the actual translation of the activity diagram and simulates the acquisition of the competences for *all* curricula represented by the translated activity diagram, the latter contains the code for updating the state, i.e. the competences achieved so far, according to the definition in terms of preconditions and effects of each course. The processes *CurriculumVerification* and *UpdateState* communicate by means of the channel *attend*. The notation *attend!courseName* represents the fact that the course with name "courseName" is to be attended. On the other hand, the notation *attend?courseName* represents the possibility for a process of receiving a message. For example, the process *CurriculumVerification* for the activity diagram of Fig. 3 is defined as follows:

```
 proctype CurriculumVerification() {
   activity_forkjoin_1();
   course_java_programming_II();
   activity_decisionmerge_1();
   course_database_I();
   course_database_II();
   activity_forkjoin_2();
   course_lab_of_web_services();
   attend!stop;
}
```

If the simulation of all its possible executions end, then, there are no competence gaps; *attend!stop* communicates this fact and starts the verification of user's learning goal, that, if passed, ends the process. Each *course* is represented by its preconditions and its effects. For example, the course "Laboratory of Web Services" is as follows:

```
 inline preconditions_course_lab_of_web_services() {
   assert(N_tier_architectures >= 4 && sql >= 2);
 }

 inline effects_course_lab_of_web_services() {
   SetCompetenceState(jsp, 4); [...]
   SetCompetenceState(markup_language, 5);
 }

 inline course_lab_of_web_services() {
   attend!lab_of_web_services;
 }
```

assert verifies the truth value of its condition, which in our case is the precondition to the course. If violated, SPIN interrupts its execution and reports about it. *SetCompetenceState* increases the level of the passed competence if its current level is lower than the second parameter. If all the curricula represented by the translated activity diagram have *no competence gaps*, no assertion violation will be detected. Otherwise, a counterexample will be returned that corresponds to an effective sequence of courses leading to the violation, giving a precise feedback to the student/teacher/course designer of the submitted set of curricula.

The *fork/join nodes* are simulated by activating as many parallel processes as their branches. Each process translates recursively the corresponding sub-activity diagram. Thus, SPIN simulates and verifies *all possible interleavings* of the courses (we can say that the curriculum is only one but it has different executions). The join nodes are translated by means of the synchronization message *done* that each activated process must send to the father process when it finishes its activity:

```
proctype activity_joinfork_11() {
  course_java_programming_I(); joinfork_11!done;
}

proctype activity_joinfork_12() {
  course_logic(); joinfork_12!done;
}

inline activity_joinfork_1() {
  run activity_joinfork_11(); run activity_joinfork_12();
  joinfork_11?done; joinfork_12?done;
}
```

Finally, *decision and merge nodes* are encoded by either conditioned or non-deterministic *if*. Each such *if* statement refers to a set of alternative sub-activity diagrams (sub-curricula). Only one will be effectively attended but all of them will be verified:

```
inline activity_decisionmerge_11() {
  course_introduction_to_database();
}

inline activity_decisionmerge_12() {
  skip;
}

inline activity_decisionmerge_1() {
  if
  :: (transaction >= 1 && data_recovery >= 1) ->
       activity_decisionmerge_12();
  :: else -> activity_decisionmerge_11();
  fi
}
```

On the other hand, the process *UpdateState*, after setting the initial competences, checks if the preconditions of the courses communicated by *CurriculumVerification* hold in the current state. If a course is applicable it also updates the state. The test of the preconditions and the update of the state are performed as an atomic operation. In the end if everything is right it sends a feedback to *CurriculumVerification* (*feedback!done*):

```
proctype UpdateState() {
  SetInitialSituation();
  do [ ... ]
  :: attend?lab_of_web_services -> atomic {
        preconditions_course_lab_of_web_services();
        effects_course_lab_of_web_services(); }
  :: attend?stop -> LearningGoal(); break;
  od
}
```

When *attend?stop* (see above) is received, the check of the user's learning goal is performed. This just corresponds to a test on the knowledge in the ending state:

```
inline LearningGoal() {
  assert(advanced_java_programming>=5 && N_tier_architectures
         >= 4 && relational_algebra>=2 && ER_language>=2);
}
```

To check if the curriculum complies to a curricula model, we check if every possibly sequence of execution of the Promela program satisfies the LTL formulas, now transformed into *never claims* directly by SPIN. For example, the curriculum shown in Fig. 3 respects all the constraints imposed by the curricula model described in Fig. 1, taking into account the description of the courses supplied at the URL above. The assertion verification takes very few seconds on an old notebook; the automaton generated from the Promela program on that example has more than four-hundred states, indeed, it is very tractable. Also the verification of the temporal constraints is not hard if we check the constraints one at the time.

3.3.5 Conclusions

We have described the work carried on in the last years about the representation of and reasoning applied curricula, reporting also the current state of advancement of the integration of semantic personalization web services for Curriculum Planning and Validation within the Personal Reader Framework.

We have introduced a graphical language to describe curricula models as temporal constraints posed on the acquisition of competences (supplied by courses), therefore, taking into account both the concepts supplied/required and the proficiency level. We have also shown how model checking techniques can be used to verify that a curriculum complies to a curricula model, and also that a curriculum both allows the achievement of the user's learning goals and that it has no competence gaps. This use of model checking is inspired by [1], where LTL formulas are used to describe and verify the properties of a composition of Web Services. Another recent work, though in a different setting, that inspired this proposal is [23], where medical guidelines, represented by means of the GLARE graphical language, are translated in a Promela program, whose properties are verified by using SPIN. Similarly to [23], the use of SPIN, gives an *automa-based semantics* to a curriculum (the automaton generated by SPIN from the Promela program) and gives a declarative, formal, representation of curricula models (the set of temporal constraints) in terms of a LTL theory that enables other forms of reasoning. In fact, as for all logical theories, we can use an inference engine to derive other theorems or to discovery inconsistencies in the theory itself. The presented proposal is an evolution of earlier works [4, 3, 5], where we applied semantic annotations to learning objects, with the aim of building compositions of new learning objects, based on the user's learning goals and exploiting planning techniques. That proposal was based on a different approach that relied on the experience of the authors in the use of techniques for reasoning about actions and changes which, however, suffers of the limitations discussed in the introduction.

We have also reported about the integration of the above approach into the Personal Reader Framework [18]. In this latter context, the goal of personalization is to create sequences of courses that fit the specific context and the learning goal of individual students. Despite some manual post-processing for fixing inconsistencies, we used real information from the Hannover University database of courses for extracting the meta-data. Currently the courses are annotated also by meta-data concerning the schedule and location of courses, like for instance room-numbers, addresses and teaching hours. As a further development, it would be interesting to let our Curriculum Planning Service to make use also of such metadata in order to find a solution that fits the desires and the needs of the user in a more complete way.

The Curriculum Planning Service has been integrated as *a new plug-and-play personalization service* in the Personal Reader framework. The Curriculum Validation Service has been designed. An early prototype of the validation system based on the model checker SPIN has been developed [6] and is currently being embedded in the same framework. The Personal Reader Platform provides a natural framework for implementing a service-oriented approach to personalization in the Semantic Web, allowing to investigate how (semantic) web service technologies can provide a suitable infrastructure for building personalization applications, that consist of re-usable and interoperable personalization functionalities. The idea of taking a service oriented approach to personalization is quite new and was born within the personalization working group of the Network of Excellence REWERSE.

References

[1] W. M. P. van der Aalst and M. Pesic. DecSerFlow: Towards a Truly Declarative Service Flow Language. In Mario Bravetti and Gialuigi Zavattaro, editors, Proc. of WS-FM, LNCS, Vienna, September 2006. Springer.

[2] M. Baldoni, C. Baroglio, I. Brunkhorst, N. Henze, E. Marengo, and V. Patti. A Personalization Service for Curriculum Planning. In E. Herder and D. Heckmann, editors, Proc. of the 14th Workshop on Adaptivity and User Modeling in Interactive Systems, ABIS 2006, pages 17-20, Hildesheim, Germany, October 2006.

[3] M. Baldoni, C. Baroglio, and N. Henze. Personalization for the Semantic Web. In N. Eisinger and J. Maluszynski, editors, Reasoning Web, First International REWERSE Summer School 2005, volume 3564 of LNCS Tutorials, pages 173-212. Springer-Verlag, Malta, July 2005.

[4] M. Baldoni, C. Baroglio, V. Patti, and L. Torasso. Reasoning about learning object metadata for adapting SCORM courseware. In L. Aroyo and C. Tasso, editors, AH 2004: Workshop Proceedings, Part I, International Workshop on Engineering the Adaptive Web, EAW'04: Methods and Technologies for personalization and Adaptation in the Semantic Web, pages 4-13, Eindhoven, The Netherlands, August 2004. Technische Universiteit Eindhoven.

[5] M. Baldoni, C. Baroglio, and V. Patti. Web-based adaptive tutoring: an approach based on logic agents and reasoning about actions. Artificial Intelligence Review, 22(1):3-39, 2004.

[6] M. Baldoni, C. Baroglio, A. Martelli, V. Patti, and L. Torasso. Verifying the compliance of personalized curricula to curricula models in the semantic web. In M. Bouzid and N. Henze, editors, Proc. of the Semantic Web Personalization Workshop, held in conjuction with the 3rd European Semantic Web Conference, pages 53-62, Budva, Montenegro, 2006.

[7] R. Baumgartner and S. Flesca and G. Gottlob. Visual Web Information Extraction with Lixto. The VLDB Journal, 119–128, 2001.

[8] R. Baumgartner, N. Henze, and M. Herzog: The Personal Publication Reader: Illustrating Web Data Extraction, Personalization and Reasoning for the Semantic Web. European Semantic Web Conference ESWC 2005, Heraklion, Greece, May 29 - June 1 2005.

[9] M. Baldoni and E. Marengo. Curriculum Model Checking: Declarative Representation and Verification of Properties. In E. Duval, R. Klamma, and M. Wolpers, editors, Proc. of EC-TEL 2007 - Second European Conference on Technology Enhanced Learning, number 4753 in LNCS, pages 432-437. Springer, 2007.

[10] European Commission, Education and Training: The BOLOGNA Process, url-http://europa.eu.int/comm/education/policies/educ/bologna/bologna_en.html

[11] P. Brusilovsky and J. Vassileva. Course sequencing techniques for large-scale web-based education. Int. J. Cont. Engineering Education and Lifelong learning, 13(1/2):75-94, 2003.

[12] O. E. M. Clarke and D. Peled. Model checking. MIT Press, Cambridge, MA, USA, 2001.

[13] E. A. Emerson. Temporal and model logic. In Handbook of Theoretical Computer Science, volume B, pages 997-1072. Elsevier, 1990.

[14] M. del Mar Gallardo, P. Merino, and E. Pimentel. Debugging UML Designs with Model Checking. Journal of Object Technology, 1(2):101¿117, July-August 2002.

[15] N. Guelfi and A. Mammar. A Formal Approach for the Veri¿cation of E-business Processes Using The Promela Language. Technical Report TR-SE2C-04-10, SE2C, Software Engineering Competence Center, University of Luxembourg, 2004.

[16] N. Guelfi and A. Mammar. A Formal Semantics of Timed Activity Diagrams and its PROMELA Translation. In Proc. of the 12th Asia-Pacific Software Engineering Conference (APSEC'05), pages 283-290. IEEE Computer Society, 2005.

[17] N. Henze: Personal Readers: Personalized Learning Object Readers for the Semantic Web. 12th al Conference on Artificial Intelligence in Education, AIED'05, 18-22 July 2005, Amsterdam, The Netherlands.

[18] N. Henze and D. Krause. Personalized access to web services in the semantic web. In The 3rd International Semantic Web User Interaction Workshop (SWUI, collocated with ISWC 2006, November 2006.

[19] G. J. Holzmann. The SPIN Model Checker, Primer and Reference Manual. Addison-Wesley, 2003.

[20] J. L. De Coi, E. Herder, A. Koesling, C. Lofi, D. Olmedilla, O. Papapetrou, and W. Siberski. A model for competence gap analysis. In WEBIST 2007, Proceedings of the Third International Conference on Web Information Systems and Technologies: Internet Technology / Web Interface and Applications, Barcelona, Spain, Mar 2007. INSTICC Press.

[21] M. P. Singh. Agent communication languages: Rethinking the principles. IEEE Computer, 31(12):40-47, 1998.

[22] M. P. Singh. A social semantics for agent communication languages. In In Issues in Agent Communication, number 1916 in LNCS, pages 31-45. Springer, 2000.

[23] P. Terenziani, L. Giordano, A. Bottrighi, S. Montani, and L. Donzella. SPIN Model Checking for the Verification of Clinical Guidelines. In Proc. of ECAI 2006 Workshop on AI techniques in healthcare: evidence-based guidelines and protocols, Riva del Garda, August 2006.

[24] Unified Modeling Language: Superstructure, version 2.1.1. OMG, Object Management Group, February 2007.

4 DYNAWeb

Industrial maintenance is one of the most important tasks in the industry because its cost is too high, usually due to poor maintenance decisions. Traditionally, corrective maintenance and preventive maintenance are performed, but both of them, the excessive and the lacking maintenance can be harmful. In the last years, CBM (Condition Based Maintenance) technology or predictive maintenance has appeared in order to establish whether the system will fail during some future period and then take actions to avoid consequences.

Within these years, we have developed the e-maintenance platform DYNAWeb, which is also part of DYNAMITE project. DYNAWeb develops a CBM system based on OSA-CBM standard over MIMOSA comprising many capabilities, like sensing and data acquisition, signal processing, health assessment, and prognosis.

This platform ensures the integration of all the components (software and hardware) using different technologies (e.g. sensor technologies, wireless communication technologyand providing them with agents and (Semantic) Web Services to allow the integration and the reuse among different applications.

The DYNAMITE vision aims at promoting a major change in the focus of condition based maintenance, essentially taking full advantage of recent advanced information technologies, that are related to hardware, software and *semantic* information modelling. Special attention is also given to the identification of cost-effectiveness related to the upgraded CBM strategies, as well as to the inclusion of innovative technologies within CBM. It is expected that the combination of the use of new technologies with a clear indication of cost-benefit trade-off will facilitate the upgrade into CBM, in many cases where non-critical machinery exists, and especially for the vast majority of SME companies that feel the distance between planned maintenance and condition based is too wide.

In order to provide the most convenient analysis flow, information processing is understood as a distributed and collaborative system, where three different levels of entities can undertake intelligence tasks. At the lower end, sensors can provide certain degree of reasoning, taking into account the “local” scope of this processing. At a medium level, smart PDAs (mobile agents) will provide higher communication interfaces with sensors, intermediate processing capabilities and a smart end for human interface to remote web services centres that will compose a distributed web platform system at the higher end of the processing hierarchy [1]. Finally, wireless data transmission between sensor devices and information processing layers will be implemented.

4.1 Ontology

The real challenge is to match the semantic web concept to the maintenance function. In this way, information used over internet must be specified in ontologies. The ontology represents the knowledge in internet [3], defining in a formal way the concepts of the different domains and relationships, with ability to perform reasoning over this knowledge. The definition of this ontologies has been performed starting from the standard CRIS (Common Relational Information Schema) defined by MIMOSA 3 (Machinery Information Management Open System Alliance).

CRIS represents a static view of the data produced by a CBM system, where every OSA-CBM layer has been associated to an ontology [2]. OSA-CBM was developed around MIMOSA CRIS (Common Relational Information Schema) that provides coverage of the information (data) that will be managed within a condition based maintenance system. It defines a relational database schema with about 200 of tables approximately for machinery maintenance information. In short, CRIS is the core of MIMOSA which aim is the development and publication of open conventions for information exchange between plant and machinery maintenance information system. In this sense DYNAMITE investigate the way to improve these ontologies defined in XML language to other richer semantic ontology languages as RDF or OWL.

4.2 Semantic web services

DYNAWeb (Semantic Web) Services are the core software modules which support the e- maintenance platform and provide a services oriented functionality for the Semantic Web. The description of these services goes far beyond the idea of an interface since we may find internal reasoning, state handling and many other elements. The set of DYNAWeb Web Services has been divided into different groups of Web Services depending on associated OSA-CBM layer, so it has defined three types of different Web Services (condition monitoring, diagnosis and prognosis). All of these Web Services could be invoked by PDAs or CMOpS. The condition monitoring will perform in sensors and PDAs but Web Services have been designed to provide this service in case not having information. So, PDA or CMOpS could request the Web Service and depending on data received by this, it could be performed four different functions related to condition monitoring:

- Absolute: this function retrieves alerts depending on fixed limits. The limits define subsets which belong to specific alert regions.
- Relative: this function issued alerts by means of the deviations between the measured current values and a reference values. Depending on the variation, it is issued an alert with a specific severity.
- Two Dimensions: the limits which define the alert regions depend on another parameter.
- Dynamic Limits: this function calculates the mean and standard deviation of historic values within it is possible to define the interval of confidence.

The Web Services for diagnosis assess the health of the component. There are five different services for diagnosis:

- Spindle Diagnosis: this function applies Bayesian Network to retrieve a list of failures with probabilities associated using vibration information.
- Hydraulic System: this function uses CLIPS expert system to retrieve diagnosis status by means of oil information.
- Cogenerator Diagnosis: this function uses means of vibration, oil and efficiency parameters within FMECA and retrieves diagnosis status.
- Reciprocating Engine Diagnosis: this function retrieves diagnosis status using CLIPS expert system within means of vibration and oil information.
- Reliability Diagnosis: this function is able to perform a diagnosis by means of FTA (Fault tree Analysis) information of the component.

Finally, prognosis Web Services implements functionality to compute the remaining useful life of the component submitted to a degradation mode. The design of these Web Services is focused on two specific cases of prognosis based on:

- Reliability: this function uses a degradation model based on reliability;
- Condition monitoring: this function applies a data modelling approach (trend analysis, pattern recognition, etc).

References

[1] A. Arnaiz, C. Emmanouilidis, B. Iung, and E. Jantunen (2006). Mobile Maintenance Management. Journal of International Technology and Information Management, Vol 15 (4) 11-22

[2] M. Lebold, K. Reichard, C. S. Byington, R. Orsagh. OSA-CBM Architecture Development with Emphasis on XML Implementations. MARCON 2002.

[3] A. Lozano: Ontologies in the semantic web. Jornadas de Ingeniería web. (2001)

[4] M. Melia and C. Pahl. Automatic Validation of Learning Object Compositions. In Information Technology and Telecommunications Conference IT&T'2005: Doctoral Symposium, Carlow, Ireland, 2006.

5 Conclusion

This deliverable accounts for the most recent advancement of the research line on Testbeds of the working group A3. The presented testbeds are at different development stages; in particular, the work on the DYNAWeb platform is still in an early phase. The other two applications, instead, are quite at an advanced stage and witness the integration, within the Personal Reader framework, of Personalization services that base upon totally different theoretical bases (unsupervised neural networks and temporal reasoning in the specific case). These integration come out as being, nevertheless, pretty easy, thanks to the service-oriented nature of the Personal Reader framework and to the choice of decoupling as much as possible the application logic of the personalization services from the interaction with the user and the interface components, as well as from the fact that all the components of the system interact with one another based on a semantics basis. Indeed, all information (e.g. encoding the user profiles) are kept and exchanged in the form of RDF documents.

Acknowledgement

This research has been co-funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. `http://rewerse.net`).

6 Appendix

Hereafter you will find included the following scientific papers:

1. *Scalability and evaluation of contextual immune model for Web mining*, S.T. Wierzchoń, K. Ciesielski, M.A. Klopotek. In:, Aboul-Ella Hassanien Janusz Kacprzyk, Ajith Abraham (eds): Computational Intelligence in Multimedia Processing: Recent Advances, Soft Computing Series, Springer, 2008.

2. *Term Distribution-based Initialization of Fuzzy Text Clustering*, K. Ciesielski, M.A. Klopotek, S.T. Wierzcho'n. To appear in Proc. 17th Int. Symposium on Methodologies for Intelligent Systems (ISMIS'08), Toronto, Canada, May 20-23, 2008, LNAI, Springer.

3. *A Service-Oriented Approach for Curriculum Planning and Validation*, by Matteo Baldoni, Cristina Baroglio, Ingo Brunkhorst, Elisa Marengo, Viviana Patti, in the Proc. of MALLOW-AWESOME 2007.

4. *Curricula Modeling and Checking*, by Matteo Baldoni, Cristina Baroglio, and Elisa Marengo, in the Proc. of AI*IA 2007, LNAI 4733, Springer, 2007.

5. *Web Services System for distributed technology upgrade within an e-maintenance framework*, by Eduardo Gilabert, Susana Ferreiro, Aitor Arnaiz, OTM Workshops, 2007.

6. *Semantic Web Services for advanced maintenance strategies*, by S. Ferreiro, E. Gilabert, A. Arnaiz, 2007.

Scalability and Evaluation of Contextual Immune Model for Web Mining

Sławomir T. Wierzchoń[1,2], Krzysztof Ciesielski[1], and Mieczysław A. Kłopotek[1,3]

[1] Institute of Computer Science, Polish Academy of Sciences,
Ordona 21, 01-237 Warszawa,Poland
stw,kciesiel,klopotek@ipipan.waw.pl

[2] Faculty of Mathematics, Physics and Informatics, Gdańsk University,
Wita Stwosza 57, 80-952 Gdańsk-Oliwa

[3] Institute of Computer Science, University of Podlasie,
Konarskiego 2, 08-110 Siedlce

Summary. In this chapter we focus on some problems concerning application of an immune-based algorithm to extraction and visualization of cluster structure. Particularly a hierarchical, topic-sensitive approach is proposed; it appears to be a robust solution to the problem of scalability of document map generation process (both in terms of time and space complexity). This approach relies upon extraction of a hierarchy of concepts, i.e. almost homogenous groups of documents described by unique sets of terms. To represent the content of each context a modified version the aiNet [9] algorithm is employed; it was chosen because of its natural ability to represent internal patterns existing in a training set. Careful evaluation of the effectiveness of the novel text clustering procedure is presented in section reporting experiments.

1 Introduction

When analyzing the number of terms per query in one billion accesses to the Altavista site, [12], extraordinary results were observed by Alan Gilchrist: (a) in 20.6% queries no term was entered, (b) in almost 25% queries only one term was used in a search, and (c) the average was not much higher than two terms! This justifies our interest in looking for a more "user-friendly" interfaces to web-browsers.

A first stage in improving the effectiveness of Information Retrieval (IR) systems was to apply the idea of clustering inspired by earlier studies of Salton, [21], and reinvigorated by Rijsbergen's Cluster Hypothesis [24]. According to this hypothesis, relevant documents tend to be highly similar to each other, and therefore tend to appear in the same clusters. Thus, it is possible to reduce the number of documents that need to be compared to a given query, as it suffices to match the query against cluster representatives first. However such an approach offers only technical improvement in searching relevant documents. A more radical improvement can be gained by using so-called document maps, [2], where a graphical representation allows additionally to convey information about the relationships of individual documents or group of documents. Document maps are primarily oriented towards visualization of a certain similarity of a collection of documents, although other usage of such the maps is possible – consult Chapter 5 in [2] for details.

The most prominent representative of this direction is the WEBSOM project[4]. Here the Self-Organizing Map, or SOM, algorithm [19] is used to organize miscellaneous text documents onto a 2-dimensional grid so that related documents appear close to each other. Each grid unit contains a set of closely related documents. The color intensity reflects dissimilarity among neighboring units: the lighter shade the more similar neighboring units are. Unfortunately this approach is time and space consuming, and rises questions of scaling and updating of document maps (although some improvements are reported in [20]). To overcome some of these problems the DocMINER system was proposed in [2]. It composes a number of methods from explorative data analysis to support effectively information access for knowledge management tasks. Particularly, a given collection of documents represented as vectors in highly dimensional vector space is moved – by a multidimensional scaling algorithm – to so-called semantic document space in which document similarities are reinforced. Then the topological structure of the semantic document space is mapped to a 2-dimensional grid using the SOM algorithm.

Still, the profound problem of map-like representation of document collections is the issue of scalability which is strongly related to high dimensionality. While multidimensional scaling and other specialized techniques, like PCA, versions of SVD etc., reduce the dimensionality of the space formally, they may result in increased complexity of document representation (which had a low number of non-zero coordinates in the high-dimensional space, and has more non-zero coordinates in the reduced space). So some other way of dimensionality reduction, via feature selection and not feature construction, should be pursued.

Note that the map of documents collection is a new kind of clustering, where not only the documents are split into groups, but also there exists a structural relationship between clusters, reflected by the topology of a map. We can say we have to do with a cluster networking. This affects the closely related issue of evaluation of the quality of the obtained clusters. Usually the quality evaluation function is a driving factor behind the clustering algorithm and hence partially determines its complexity and success. While the conventional external and internal cluster evaluations criteria (like class purity, class uniformity, inter-class dissimilarity) are abundant, they are primarily devised

[4] Details and full bibliography concerning WEBSOM can be found at the web-page `http://websom.hut.fi/websom/`.

to evaluate the sets of independent (not linked) clusters, there exist no satisfactory evaluation criteria for cluster network quality. Beside SOM, there are other clustering methods like growing neural gas (GNG) [11] or artificial immune systems (AIS) [9], [25] that face similar problems.

In our research project BEATCA, [18], oriented towards exploration and navigation in large collections of documents a fully-fledged search engine capable of representing on-line replies to queries in graphical form on a document map has been designed and constructed [16]. A number of machine-learning techniques, like fast algorithm for Bayesian networks construction [18], SVD analysis, (GNG) [11], SOM algorithm, etc., have been employed to realize the project. BEATCA extends the main goals of WEBSOM by a multilingual approach, new forms of geometrical representation (besides rectangular maps, projections onto sphere and torus surface are possible).

The process of document map creation is rather complicated and consists of the following main stages: (1) document crawling, (2) indexing, (3) topics identification, (4) document grouping, (5) group-to-map transformation, (6) map region identification, (7) group and region labeling, and finally, (8) visualization. At each of theses stages various decisions should be made implying different views of the document map.

Within such a framework, in this chapter we propose a new solution to the problem of scalability and of evaluation of the quality of the cluster network. In particular, the contribution of this chapter concerns: (1) invention of a new artificial immune algorithm for handling large-scale document collections, to replace the traditional SOM in document map formation, (2) invention of a new representation of the document space, in which instead of single point statistics of terms their distributions (histograms) are exploited, (3) invention of a measure of quality of networked clustering of document collections, which is based on the above-mentioned histograms, and which evaluates the quality of both the clustering of documents into the groups as well as usefulness of the inter-group links. These new features are of particular value within our framework of contextual document space representation, described in earlier publications, allowing for a more radical intrinsic dimensionality reduction, permitting efficient and predominantly local processing of documents.

In section 2 we present our hierarchical, topic-sensitive approach, which appears to be a robust solution to the problem of scalability of map generation process (both in terms of time complexity and space requirements). It relies upon extraction of a hierarchy of concepts, i.e. almost homogenous groups[5] of documents. Any homogenous group is called here a "context", in which further document processing steps – like computation of term-frequency related measures, keyword extraction, and dimensionality reduction – are carried out, so that each context is described by unique set of terms. To represent the content of each context a modified version of the aiNet algorithm [10] was employed –

[5] By a homogegous group we understand hereafter a set of documents belonging to a single cluster after a clustering process.

see section 3. This algorithm was chosen because of its ability of representing internal patterns existing in a training set. More precisely, the aiNet produces a compressed data representation for the vectors through the process resembling data edition. Next this reduced representation is clustered; the original aiNet algorithm uses hierarchical clustering, [10], while we propose original and much more efficient procedure.

Further, the method of representing documents and groups of documents in the vector space was enriched: Instead of traditional single point measure we apply the histograms of term occurrence distributions in some conceptual space so that the document content patterns would be matched in a more refined way – see section 4 for details.

To evaluate the effectiveness of the novel text clustering procedure it has been compared to the aiNet and SOM algorithms in section 5. In the experimental sections 5.6 – 5.8 we have also investigated issues such as evaluation of immune network structure and the influence of the chosen antibody/antigen representation on the resulting immune memory model. Final conclusions are given in section 7.

1.1 Document maps

Before going into details let us devote a little bit attention to the concept of a document map as such. Formally, a document map can be understood as a 2-dimensional rectangle (or any other geometrical figure) split into disjoint areas, usually squares or hexagons[6], called "cells". To each cell a set of documents is assigned, thus a single cell may be viewed as a kind of document cluster. The cells are frequently clustered into so-called regions on the ground of similarity of their content. The cells (and regions) are labeled by the keywords best-describing cell/region content, where "best-describing" is intended to mean entire characteristic of the cell/region, but distinguishing it from surrounding cells/regions. A document map is visualized in such a way that cell colors (or textures) represent the number of documents it contains, or the degree of similarity to the surrounding cells, the importance of documents (e.g. PageRank), the count of documents retrieved in the recent query, or any other feature significant from the point of view of the user. The labels of some cell/region are also displayed, but with a "density" not prohibiting the overall readibility. Optionally, labels may be displayed in "mouse-over" fashion.

2 Contextual local networks

In our approach – like in many traditional IR systems – documents are mapped into T-dimensional term vector space. The points (documents) in this space

[6] For non-Euclidian geometries other possibilities exist – cf. [18]

are of the form $(w_{1,d}, ..., w_{T,d})$ where T stands for the number of terms, and each $w_{t,d}$ is a weight for term t in document d, so-called term frequency/inverse document frequency, *tfidf*, weight:

$$w_{t,d} = w(t,d) = f_{td} \cdot log\left(\frac{N}{f_t}\right) \tag{1}$$

where f_{td} is the number of occurrences of term t in document d, f_t is the number of documents containing term t and N is the total number of documents.

The vector space model has been criticized for some disadvantages, polysemy and synonymy, among others, [3]. To overcome these disadvantages a contextual approach has been proposed [18] relying upon dividing the set of documents into a number of homogenous and disjoint subgroups (clusters). During the dimensionality reduction process, each of the clusters, called also "contexts" (for reasons obvious later), will be described by a unique subset of terms.

In the sequel we will distinguish between *hierarchical* and *contextual* model of documents treatment. In the former the dimensionality reduction process is run globally, for the whole collection of documents, so that the terms used for document description are identical for each subgroup of documents, and the computation of *tfidf* weights, defined in equation (1) is based on the whole document collection. In the later model, for each subgroup the dimensionality reduction process is run separately, so that each subgroup may be described by a different subset of terms weighted in accordance with the equation (4). Finally, whenever we do not carry out a clustering of documents and we construct a single, "flat", representation for entire collection – we will speak about a *global* model. [7]

The contextual approach consists of two main stages. At first stage a hierarchical model is built, i.e. a collection D of documents is recurrently divided – by using Fuzzy ISODATA algorithm [4] – into homogenous groups consisting of approximately identical number of elements. Such a procedure results in a hierarchy represented by a tree of clusters. The process of partitioning halts when the number of documents inside each group meets predefined criteria[8]. To compute the distance $dist(d,v)$ of a document d from a cluster centroid v, the cosine distance was used:

$$dist(d,v) = 1 - <d/||d||, v/||v||> = 1 - (d/\|d\|)^{\mathrm{T}}(v/\|v\|) \tag{2}$$

where the symbol $< \cdot, \cdot >$ stands for the dot-product of two vectors. Given m_{dG}, the degree of membership of a document d to a group G, (obtained via

[7] The principal difference between the "hierarchical" and the "global" models is that in the hierarchical model we distinguish a number of clusters, while in the global model we treat the whole collection as a single cluster.

[8] Currently a single criterion saying that the cardinality c_i of i-th cluster cannot exceed a given boundaries $[c_{min}, c_{max}]$. This way the maps created for each group at the same level of a given hierarchy will contain similar number of documents.

the Fuzzy-ISODATA algorithm) this document is assigned to the group with highest value of m_{dG}.

The second phase of contextual document processing relies upon division of terms space (dictionary) into – possibly overlapping – subspaces of terms specific to each context (i.e. the group extracted in previous stage). The fuzzy membership level, m_{tG}, representing importance of a particular term t in a given context G is computed as:

$$m_{tG} = \frac{\sum_{d \in G} (f_{td} \cdot m_{dG})}{f_G \cdot \sum_{d \in G} m_{dG}} \tag{3}$$

where f_G is the number of documents in the cluster G, m_{dG} is the degree of membership of document d to group G, f_{td} is the number of occurrences of term t in document d. We assume that a term t is relevant for a given context G if $m_{tG} > \epsilon$, where ϵ is a parameter.

Removing non-relevant terms leads to the topic-sensitive reduction of the dimension of the terms space. This reduction results in a new vector representation of documents; each component of the vector is computed according to the equation:

$$w_{tdG} = f_{td} \cdot m_{tG} \cdot log \left(\frac{f_G}{f_{tG} \cdot m_{tG}} \right) \tag{4}$$

where f_{tG} is the number of documents in the group G containing term t.

To depict similarity relation between contexts (represented by a set of contextual models), additional "global" map is required. Such a model becomes the root of contextual maps hierarchy. Main map is created in a manner similar to previously created maps with one distinction: an example in training data is a weighted centroid of referential vectors of the corresponding contextual model: $\widetilde{v_i} = \sum_{c \in C_i} (|c| \cdot v_c)$, where C_i is the set of antibodies [9] in i-th contextual model (obtained from Fuzzy-ISODATA), $|c|$ is the density of the antibody, i.e. the number of assigned documents and v_c is its referential vector.

The whole process of learning contextual model (summarized in the pseudocode 1) is to some extent similar to the hierarchical learning [13]. However, in our approach each constituent model, and the corresponding contextual map, can be processed independently (particularly, in parallel). Also a partial incremental update of such a model appears to be much easier to perform, both in terms of model quality, stability and time complexity. The possibility of incremental learning stems from the fact that the very nature of the learning process is iterative. So if new documents come, we can consider the learning process as having been stopped at some stage and it is resumed now with all the documents. We claim that it is not necessary to start the learning process from scratch neither in the case that the new documents "fit" the distribution

[9] This notion is explained in section 3.1.

of the previous ones nor when their term distribution is significantly different. This claim is supported by experimental results presented e.g in [18].

Algorithm 1 Scheme of the meta-algorithm of contextual processing

1. Index the whole set of documents and collect global frequency statistics for terms
2. Create global vector representation of documents and identify globally significant terms (global reduction of dimensionality)
3. Identify major themes in the document collection
4. Based on global representation and major themes carry out fuzzy splitting of the document collection and reduce the term space
5. Create initial contextual groups and compute contextual statistics for the terms
6. Identify locally significant terms and create contextual vector representation for the individual groups
7. Create the contextual model (a hierarchy of network models, based on local vector representations)
8. Create map-like visualization of the contextual model and find labels for document groups (network nodes and map cells)
9. Adapt the existing model in response to changes of objective factors (data changes) or subjective factors (personalization, response to changes in user profile):
    a) Modify local statistics of individual contexts and modify vector representations taking into account the significance of terms.
    b) Modify the existent split into contexts
    c) Start incremental learning of existing contextual models
    d) Create a new map-like visualization of the modified contextual model and update the group and cell labels

3 Immune approach to text data clustering

One of main goals of the BEATCA project was to create multidimensional document maps in which geometrical vicinity would reflect conceptual closeness of documents in a given document set. Additional navigational information (based on hyperlinks between documents) can be introduced to visualize directions and strength of between-group topical connections.

Clustering and content labeling is the crucial issue for understanding the two-dimensional map by a user. We started our research with the WEBSOM approach which, appeared to be unsatisfactory: both speed and clustering stability were not very encouraging.

In SOM algorithm, [19] each unit of an $K \times K$ grid contains so-called reference vector v_i, whose dimension agrees with the dimension of training examples. The training examples are repeatedly presented to the network until a termination criterion is satisfied. When an example $x(t)$ is presented at time t to the network, its reference vectors are updated according to the rule

$$v_i(t+1) = v_i(t) + \alpha_i(t) \cdot (x(t) - v_i(t)) \, , i = 1, ..., |K| \times |K| \tag{5}$$

where $\alpha_i(t)$ is so-called learning rate varying according to the equation:

$$\alpha_i(t) = \epsilon(t) \cdot \exp\left(-\frac{dist(i,w)}{\sigma^2(t)} \right) \tag{6}$$

Here $\epsilon(t)$ and $\sigma(t)$ are two user defined monotone decreasing functions of time called, respectively, step size (or cooling schedule) and neighborhood radius. The symbol $dist(i,w)$ stands for the distance (usually Manhattan distance) between i-th unit and so-called winner unit (i.e. the unit which reference vector is most similar to the example $x(t)$).

The main deficiencies of SOM are (cf. [1]): (a) it is order dependent, i.e. the components of final weight vectors are affected by the order in which training examples are presented, (b) the components of these vectors may be severely affected by noise and outliers, (c) the size of the grid, the step size and the size of the neighborhood must be tuned individually for each data-set to achieve useful results, (d) high computational complexity.

GNG [11] uses the same equation (5) to update reference vectors but with fixed learning rate α. Further its output is rather graph and not a grid. The main idea is such that starting from very few nodes (typically, two), one new node is inserted ever λ iterations near the node featuring the local local error measurement. There is also a possibility to remove nodes: every λ iterations the node with lowest utility for error reduction is removed. The main disadvantages of GNG are (cf. [1]): (a) in comparison with SOM it requires larger number of control parameters which should be tuned, (b) because of fixed learning rate it lacks stability, (c) rather elaborated technique for visualizing resulting graph must be invented.

An immune algorithm is able to generate the reference vectors (called antibodies) each of which summarizes basic properties of a small group of documents treated here as antigens[10]. This way the clusters in the immune network spanned over the set of antibodies will serve as internal images, responsible for mapping existing clusters in the document collection into network clusters. In essence, this approach can be viewed as a successful instance of exemplar-based learning giving an answer to the question "what examples to store for use during generalization, in order to avoid excessive storage and time complexity, and possibly to improve generalization accuracy by avoiding noise and overfitting", [26].

[10] Intuitively by antigens we understand any substance threatening proper functioning of the host organism while antibodies are protein molecules produced to bind antigens. A detailed description of these concepts can be found in [9].

3.1 aiNet algorithm for data clustering

The artificial immune system aiNet [10] mimics the processes of clonal selection, maturation and apoptosis [9] observed in the natural immune system. Its aim is to produce a set of antibodies binding a given set of antigens (i.e. documents). The efficient antibodies form a kind of immune memory capable to bind new antigens sufficiently similar to these from the training set.

Like in SOM and GNG, the antigens are repeatedly presented to the memory cells (being matured antibodies) until a termination criterion is satisfied. More precisely, a memory structure M consisting of matured antibodies is initiated randomly with few cells. When an antigen ag_i is presented to the system, its affinity $\mathit{aff}(ag_i, ab_j)$ to all the memory cells is computed. The value of $\mathit{aff}(ag_i, ab_j)$ expresses how strongly the antibody ab_j binds the antigen ag_i. From a practical point of view $\mathit{aff}(ag_i, ab_j)$ can be treated as a degree of similarity between these two cells[11]. The greater affinity $\mathit{aff}(ag_i, ab_j)$, the more stimulated ab_j is.

The idea of clonal selection and maturation translates into next steps (here σ_d, and σ_s are parameters). The cells which are most stimulated by the antigen are subjected to clonal selection (i.e. each cell produces a number of copies proportionally to the degree of its stimulation), and each clone is subjected to mutation (the intensity of mutation is inversely proportional to the degree of stimulation of the mother cell). Only clones cl which can cope successfully with the antigen (i.e. $\mathit{aff}(ag_i, cl) > \sigma_d$) survive. They are added to a tentative memory, M_t, and the process of clonal suppression starts: an antibody ab_j too similar to another antibody ab_k (i.e. $\mathit{aff}(ab_j, ab_k) > \sigma_s$) is removed from M_t. Remaining cells are added to the global memory M.

These steps are repeated until all antigens are presented to the system. Next the degree of affinity between all pairs $ab_j, ab_k \in M$ is computed and again too similar – in fact: redundant – cells are removed from the memory. This step represents network suppression of the immune cells. Lastly $r\%$ (one more parameter) of the worst individuals in M are replaced by freshly generated cells. This ends one epoch, and next epoch begins until a termination condition is met.

Among all the parameters mentioned above the crucial one seems to be the σ_s as it critically influences the size of the global memory. Each memory cell can be viewed as an exemplar which summarizes important features of "bundles" of antigens stimulating it.

[11] In practical applications this measure can be derived from any metric dissimilarity measure $dist$ as $\mathit{aff}(ag_i, ab_j) = \frac{d_{max} - dist(ag_i, ab_j)}{d_{max}}$, where d_{max} stands for the maximal dissimilarity between two cells. Another possibility – used in our approach – is to assume that the affinity is inversely proportional to the distance between corresponding molecules.

3.2 Identification of redundant antibodies

Clonal suppression stage requires $|M_t| \cdot (|M_t| - 1)/2$ calculations of the affinity between all pairs of the cells in M_t. To reduce time complexity of this step we refer to the agglomerative clustering approach. The crucial concept here is to manage matrix of distances in a smart way and to update only those distances which have really changed after merging two clusters. Among many possible solutions, we have applied so-called partial similarity matrix and update algorithm presented in [14]. Authors have shown that the expected complexity of a single-step update is of order of $O(2 \cdot N \cdot G \cdot g)$, where N is the number of objects, G is the maximum number of clusters, $g << G$ is the maximal number of column rescanning and modifications after clusters merging step. It is significantly less than the $O(N^3)$ complexity of a naive approach. Finally, the reduced antibodies are replaced by a single cell being the center of gravity of the set of removed antibodies. Thus, we not only reduce the size of the immune network, but presumably compress an information contained in a set of specialized antibodies to the new, universal antibody.

3.3 Robust construction of mutated antibodies

In case of high-dimensional data, such as text data represented in vector space, calculation of stimulation level is quite costly (proportional to the number of different terms in dictionary). Thus, the complexity of an immune algorithm can be significantly reduced if we could restrict the number of required expensive recalculations of stimulation level. The direct, high-dimensional calculations can be replaced by operations on scalar values on the basis of the simple geometrical observation that a stimulation of a mutated antibody clone can be expressed in terms of original antibody stimulation.

Such an optimization is based on the generalized Pythagoras theorem: if v_1, v_2, v_3 are the sides of a triangle $(v_1 + v_2 + v_3 = 0)$ then $|v_3|^2 = |v_1|^2 + |v_2|^2 - 2 \cdot |v_1| \cdot |v_2| \cdot cos(v_1, v_2)$. We can define mutated clone $\widetilde{m}$ as: $\widetilde{m} = \kappa \cdot d + (1 - \kappa) \cdot \widetilde{c}$, where $\widetilde{c}$ is cloned antibody, d is antigen (document) and κ is the random mutation level.

Taking advantage of equation (5) and Pythagoras theorem (where $v_1 := d' = \kappa \cdot d$, $v_2 := \widetilde{c'} = (1 - \kappa) \cdot \widetilde{c}$, $v_3 := -m$) and having calculated original antibody stimulation $\mathit{aff}(\widetilde{c}, d)$, we can calculate mutated clone stimulation level $\mathit{aff}(\widetilde{m}, d)$ as follows. Let

$$P = cos(\widetilde{c'}, d') = cos(\widetilde{c}, d) = 1 - \mathit{aff}(\widetilde{c}, d) \tag{7}$$

and the scalar product

$$\langle \widetilde{c}, d \rangle = P \cdot |\widetilde{c}| \cdot |d| \tag{8}$$

Then the norm of the mutated antibody is

$$|\widetilde{m}|^2 = |d'|^2+|\widetilde{c'}|^2+2\cdot P\cdot|\widetilde{c'}|\cdot|d'| = \kappa^2\cdot|d|^2+(1-\kappa)^2\cdot|\widetilde{c}|^2+2\cdot\kappa\cdot(1-\kappa)\cdot P\cdot|\widetilde{c}|\cdot|d| \tag{9}$$

Let us further define

$$s = \kappa\cdot|d|^2+(1-\kappa)\cdot|\widetilde{c}|\cdot|d| = \kappa|d|^2+(1-\kappa)\cdot P\cdot|\widetilde{c}|\cdot|d| \tag{10}$$

Finally,

$$\mathit{aff}(\widetilde{m},d) = \frac{s}{|\widetilde{m}|\cdot|d|} \tag{11}$$

Dually, we can find mutation threshold κ so that mutated antibody clone stimulation $\mathit{aff}\,(\widetilde{m},d) < \sigma_d$. Precisely, we are looking for constant value K such that $\mathit{aff}(\widetilde{m},d) = \sigma_d$. Subsequently K can be used to create mutated antibody for random mutation level $\kappa \in (0,K)$. The advantage of such an approach is the reduction of the number of inefficient (too specific) antibodies, which would be created and immediately removed from the clonal memory.

Analogically to the previous inference, if we define

$$p = \mathit{aff}(\widetilde{c},d)$$

$$x = -p\cdot|d|+p^2\cdot|\widetilde{c}|+\sigma_d^2\cdot(p\cdot|d|-\widetilde{c})$$

$$y = |d|^2-2\cdot p\cdot|\widetilde{c}|\cdot|d|+p^2\cdot|\widetilde{c}|^2-\sigma_d^2\cdot(|d|^2-|\widetilde{c}|^2+2\cdot p\cdot|\widetilde{c}|\cdot|d|)$$

$$z = \sigma_d\cdot|d|\sqrt{(p^2-1)\cdot(\sigma_d^2-1)}$$

then the sought value of mutation threshold K is

$$K = \frac{|\widetilde{c}|\cdot(x+z)}{y} \tag{12}$$

3.4 Stabilization via time-dependent parameters

Typical problem with immune based algorithms is the stabilization of the size of the memory cells set. This explains why we decided to use time dependent parameters. For each parameter p, we defined its initial value p_0 and the final value p_1 as well as the time-dependent function $f(t)$, such that $p(t) = f(t)$ and $p(0) = p_0$, $p(T) = p_1$ where T is the number of learning iterations.

In particular, both $\sigma_s(t)$ and $\sigma_d(t)$ are reciprocally increased, while $m_b(t)$ – the number of clones produced by a cell – is linearly decreased with time: $\sigma(t) = \sigma_0+(\sigma_1-\sigma_0)\cdot\frac{t\cdot(T+1)}{T\cdot(t+1)}$ and $m_b(t) = m_0+\frac{m_1-m_0}{T}\cdot t$, where $\sigma_0 = 0.05$, $\sigma_1 = 0.25$ for $\sigma_s(t)$; $\sigma_0 = 0.1$, $\sigma_1 = 0.4$ for $\sigma_d(t)$; $m_0 = 3$, $m_1 = 1$ for $m_b(t)$.

3.5 Robust antibody search in immune network

One of the most computationally demanding parts of any AIS algorithm is the search for the best fitted (most stimulated) antibodies. Especially, in application to web documents, where both the text corpus size and the number of cells in the immune, called also *idiotypic*, network is huge, the cost of even a single global search phase in the network is prohibitive.

Unfortunately, experiments showed that neither local search method (i.e. searching through the graph edges of the idiotypic network from last-iteration's starting cell) nor joint-winner search method (our own approach devoted to SOM learning [16]) are directly applicable to idiotypic networks.

We propose a replacement of the global search approach with a modified local search[12]. The modification relies upon remembering most stimulated cell for more than one connected component of the idiotypic network and to conduct in parallel a single local-winner search thread for each component. Obviously, it requires one-for-iteration recalculation of connected components, but this is not very expensive: the complexity of this process is of order $O(V+E)$, where V is the number of cells and E is the number of connections (graph edges).

A special case is the possibility of an antibody removal during the t-th learning iteration. When the previous iteration's most stimulated antibody for a given document (antigen) has been removed from the model (i.e. current system's memory), we activate search processes (in parallel threads) from each of its direct neighbors in the previous iteration's graph.

We have developed another, slightly complicated, but more accurate method. It exploits well-known Clustering Feature Tree (CF-Tree, [27]) to group similar network cells in dense clusters. Antibody clusters are arranged in the hierarchy and stored in a balanced search tree. Thus, finding most stimulated (similar) antibody for a document requires $O(log_t V)$ comparisons, where t is the tree branching factor (refer to [27] for details). Amortized tree structure maintenance cost (insertion and removal) is also proportional to $O(log_t V)$.

3.6 Adaptive visualization of the AIS network

Despite many advantages over SOM approach, AIS networks have one serious drawback: high-dimensional networks cannot be easily visualized. In our approach the immune cells are projected onto a regular Kohonen grid. To initialize such a grid properly, a given group of documents is divided into small number of disjoint group (main topics) by using fast ETC Bayesian tree [15]. The centers of the main topics are uniformly spread over the map surface,

[12] In such a procedure searching for the most stimulated antibody in t-th iteration starts from the most stimulated antibody identified in $(t-1)$-th iteration and next the graph edges of the idiotypic network are traversed appropriately.

and remaining cells of the grid are initialized with intermediate topics calculated as the weighted average of main topics, with the weight proportional to the Euclidean distance from the corresponding cells representing main topics. This way geographical neighborhood on he grid corresponds to the graphical neighborhood in the immune network.

After initialization, the map is learned with the standard SOM algorithm [19]. Finally, we adopt attraction-repelling algorithm [23] to adjust the position of AIS antibodies on the SOM projection map, so that the distance on the map reflects as close as possible the similarity of the adjacent cells. The topical initialization of the map is crucial here to assure the stability of the final visualization [16]. The resulting map visualizes AIS network with resolution depending on the SOM size (a single SOM cell can gather more than one AIS antibody).

4 Histograms in Vector Spaces

As has been said in previous sections, the coordinate value referring to the term t_i in the vector d_j representing the whole document is equal to the value of the pondering (term-weighting) function $f(t_i, d_j)$. This function may ponder the term globally, like *tfidf* defined in Eq. (1)), or locally, like the contextual function $f_G(t_i, d_j) = w_{tdG}$, defined in Eq. (4).

In the following subsection we will extend this representation by an information about the distribution of pondering function values for individual dimensions in the vector space. Subsequently we will describe possible applications of this information.

4.1 Distributions of the function pondering the terms

Properties of each term can be considered individually (for a single document), or in the broader context of a given (sub)set of documents D. In the latter case we can consider the values of the pondering function for a given term t for each document $d \in D$ as observed values of a random variable with underlying continuous probability distribution. In practical cases the continuous distribution will be approximated by a discrete one, so that the information about the random variable distribution for the term t can be summarized as a histogram $H_{t,D}$.

Let us consider the document $d \in D$ and the pondering function f. We shall represent this document by a normalized vector $d = \left[f'_{0,d}, \ldots, f'_{T,d}\right]$, where $f'_{t,d} = \|d\|^{-1} \cdot f_{t,d}$ for $t = 0, \ldots, T$. After normalization, all the documents are located within the unit hypercube $[0, 1]^T$.

For a fixed number $Q_{t,D}$ of intervals of the histogram $H_{t,D}$ we define the discretization $\Delta_{t,D} : [0, 1] \mapsto \{0, \ldots, Q_{t,D} - 1\}$, i.e. the transformation of the normalized pondering function into the index of interval.

In the simplest case it can be a uniform split of the interval $[0,1]$ into segments of equal length $\Delta_{t,D}(f'_{t,d}) = \lfloor (Q_{t,D} - 1) \cdot f'_{t,d} \rfloor$. An efficient discretization, however, should take into account the fact, that the pondering function for a fixed term takes values in only a subset of the unit interval (like in the case of splitting the set of documents into homogenous subsets, as done in contextual approach). Optimal discretization should also approximate quantile-based split of the underlying pondering function distribution.

Having defined the discretization $\Delta_{t,D}$ and a fixed interval q, let us define the characteristic function:

$$\chi\left(\Delta_{t,D}\left(f'_{t,d}\right), q\right) = \begin{cases} 1 & \text{if } \Delta_{t,D}\left(f'_{t,d}\right) = q \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Then we compute the value assigned to the interval q of the histogram $H_{t,D}$ for the term t in document collection D as:

$$H_{t,D}(q) = \sum_{d \in D} \chi\left(\Delta_{t,D}\left(f'_{t,d}\right), q\right) \tag{14}$$

So individual intervals of the histogram $H_{t,D}$ represent the number of occurrences of a discretized value of the pondering function f for the term t in the document collection D. The interval values can be in turn transformed to relative frequencies via the plain normalization $H'_{t,D}(q) = H_{t,D}(q)/T_{t,D}$, where $T_{t,D} = \sum_{i \in \Theta} H_{t,D}(q)$ is the total number of documents $d \in D$ containing the term t. The frequency distribution approximates the probability distribution of the unknown variable describing the weight of occurrence of term t in a randomly chosen document $d \in D$. A term not occurring in any document of the collection will be represented by an "empty histogram", having zero frequencies assigned to all intervals.

Note that the definition of the characteristic function (13) can be generalized to a more elaborated variant in which adding a document into to a context will be associated with:

- the degree of similarity of the document to the context (cf. section 4.4)
- document quality (as an aggregation of quality values for terms contained in a document)
- both above-mentioned factors

In this general case instead of unit counts, the respective counts related to intervals of individual histograms would be increased by a quantity proportional to document quality and the degree of membership to a given context. Such an approach allows also to take into account the fuzzy membership of some documents, prohibiting from assignment to a unique context.

4.2 Practical aspects of usage of histograms

The collection of histograms $H(t, D)$ for $t \in T_D$, obtained from the document collection D, via the previously described transformation, can be treated as an aggregated information on the (sub)space, in which the documents reside. Subsequently we show how this information can be exploited.

First we shall stress that the maintenance of the histograms is not a big burden for the computer memory even for large document collections, due to:

- initial reduction of term space dimensionality [17]
- additional reduction and clustering of terms in case of contextual processing
- compact carrier (set of non-zero valued histogram elements) for majority of terms dla

This implies that histograms may be represented by by sparse matrices or (in a still more efficient way) by) by cyclic tables, indexing all intervals with non-zero frequencies (carrier compactness assumption). For example a full histogram description (histograms with the dictionary) for 20000 documents from the 20NewsGroups collection divided into 20 contextual groups requires as little as 6 MB RAM (about 300 KB for a single context, the number of terms spanning the vector space: 7030).

4.3 Significance of a term in a context

With the rapid growth of dictionary size T, the most important task is the identification of the most significant terms, most strongly affecting clustering and classification of documents as well as the description of the resulting clusters (keywords extraction). Also the impact of irrelevant terms needs to be bounded, since their number grows much more rapidly than the number of significant terms.

The first stage in differentiation of the term significance is the dictionary reduction process. It is a kind of "binary" differentiation: non-significant terms are simply removed from further stages of document processing. The dictionary reduction can be conducted in two phases: the global and contextual.

Beside dictionary reduction (removal of least important terms), introduction of contextual pondering function (see Eq. (4) in section 2) leads also to diversification of the significance of the remaining terms. We are interested also in similar diversification expressed as a function of features of term histograms. The advantage of such a term description, as we will see in subsequent sections, is not only an effective application in tasks like document classification or identification of keywords for clusters of documents and contexts, but also the possibility of dynamic adaptation understood here as the change of the pondering function (hence also of the vector representation of documents) parallel to the process of incremental clustering.

Intuitively, less significant terms are represented by histograms with the following features:

- high value of curtosis (a histogram with high peaks), which is especially visible for terms that are frequent and occur uniformly in the document collection, hence are less characteristic

- the domain (the carrier) of the histogram is relatively "compact", having few intervals with non-zero coordinates meaning low variability of pondering function values
- non-zero values occur only for intervals with low indices (corresponding to low values of pondering function)
- appear in just a few documents or, in almost every document (e.g. function words)

Dually, the significant terms are those that are not too common (but also not to rare), have high variability of values of the pondering function, with many non-zero intervals, and at the same time the pondering function values are high (non-zero values of intervals with high indices).

Therefore, we define the significance of a term t in a given context, determined by the set of documents D, as follows:

$$w_{t,D} = \frac{1}{Q_{t,D}} \sum_{q \in \Theta} (q+1) \cdot \log\left(H_{t,D}(q)\right) \tag{15}$$

The weight $w_{t,D}$ takes its values in the interval $\left[0, \sum_{q \in \Theta} \log\left(H_{t,D}(q)\right)\right]$ being a subset of $[0, T_{t,D}]$.

The above measure has an additional advantage that it can be computed at the low computational cost and can be updated in $O(|d|)$ (where $|d|$ is the number of distinct terms in the document d) when a document appear in or disappear from a given subspace or context. It is very important property in case of incremental clustering of dynamically changing text/web data (see also section 4.5).

4.4 Determining the degree of membership of a document to a context

A document fits well to a given contextual subspace if the distribution of some measurable features of term occurrence is typical for the "majority" of documents in this space. Generally, we can look here at features like correlations or co-occurrences of some terms or location-based statistics (e.g. deviation of distances between repeated occurrences of a term in the document content from the expected number of occurrences under the assumption of term indifference from a given context; key terms occurrence patterns should differ from those of functional words).

Qualitative features could potentially be taken into account, like style characteristics (dominant usage of a synonym or non-typical inflection) or even features not directly related to the textual content (e.g. link structure between hypertext documents). In this paper we restrict ourselves to the analysis of frequency of term occurrence and to a definition of "typicality" based on histograms of pondering function for individual terms in a given context. Hence we can talk about an approach similar to statistical maximum likelihood estimation, in which, based on observed values, we construct a parametric function approximating parameters of an unknown conditional probability distribution $f \propto P(D|\Theta)$. The likelihood function should maximize the probability of observed data, and on the other hand it should highly valuate unseen data similar to ones in the training sample (the "generalization" capability of the model). We proceed the same way in our case. A document is considered as "typical" for which the values of the pondering function for the *majority* of terms are frequent ones in the given context.

Additionally, to avoid domination of the aggregated term-based function evaluating document "typicality" by less important (but more numerous) terms, the aggregation should take into account the formerly defined term significance in a given context. Therefore, the similarity (degree of membership) of the document d to the context determined by the document collection D is defined as follows:

$$m_f(d', D) = \frac{\sum_{t \in d'} w_{t,D} \cdot H'_{t,D}(q)}{\sum_{t \in d'} w_{t,D}} \tag{16}$$

where $w_{t,D}$ is the significance of a term (eq.(15)), $H'_{t,D}$ is the normalized histogram for the term t (see section 4.1), and $q = \Delta_{t,D}\,(f_D t, d')$ is the sequential index of the interval, determined for a fixed normalized pondering function f_D and discretization $\Delta_{t,D}$ (which transforms the value $f_D t, d'$ into the index q). The function $m_f(d', D)$ takes its values in $[0, 1]$.

It should be noted that the cost of computing the similarity function $m_f(d', D)$ is $O(|d'|)$, and it is proportional to the number of distinct terms in the document and equal to the complexity of the cosine measure calculation.

Having determined the similarity of a document to individual contexts in the contextual model, we obtain the vector of fuzzy memberships of a document to the contexts, similarly to known methods of fuzzy clustering (e.g. Fuzzy-ISODATA). In the next section we explain, how such a vector is used to achieve incremental updates of the contextual clustering model.

4.5 Incremental adaptation of contexts

While the topic distribution within the stream of documents is dynamically changing in time (e.g. some Internet or intranet documents appear, disappear or have its content modified) also the contextual clustering models have to be adapted correspondingly. Such adaptation is performed both on the level of individual documents and the document clusters, represented by antibodies (or GNG cells, in case of GNG-based model). So a new document can be assigned to a context, and within it to an antibody. A modified document may be moved from one antibody to another, in the same, or in another context. As a result, antibodies may not fit their original contextual immune model and it may be necessary to move them elsewhere, as a side effect of the so-called reclassification.

When a single new document is appearing, its similarity to every existing context is calculated by equation (16) and the document is assigned to its "most similar" context[13]. Whenever document context is modified, it may eventually be removed from its previous context and assigned to a new one.

Important aspect of context adaptation is that the measure of contextual term importance (cf. Eq.(15)) can be efficiently updated as documents are added or removed from a given context. Constant-time update of the importance of each term t which appears in document d requires only to keep separately numerator and denumerator from equation (15) and to update them adequately. Denominator is increased or decreased by one, while nominator by $i+1$, where i is the index of the updated interval in the histogram $H_{t,D}(q)$. Index i is computed by the discretization $\Delta_{t,D}\,(f_{t,d'})$ (conf. section 4.1).

After any of the contextual models has converged to a stable state, the reclassification procedure is applied. Each document group is represented by reference vector within an antibody, which can be treated as a pseudo-document d_{v_i}. The similarity of d_{v_i} to every other (temporally fixed) context is calculated (eq.(16)). If the "most similar" context is different from the current context then the antibody (with assigned documents) is relocated to corresponding contextual model. The relocated antibody is connected to the most stimulated antibody in the new immune model (and eventually merged with it in the subsequent phases).

There is no room to go into details, so we only mention that also the whole context can be eventually incorporated within some other context, on the basis of our between-context similarity measure, based on Hellinger divergence. Finally, we obtain incremental text data meta-clustering model, based both on adaptive properties of modified clustering model (within-context adaptation, [8]) and on dynamically modified contexts, which allows for clustering scalability and adaptation also on inter-context level.

5 Experimental results

In the following sections, the overall experimental design as well as quality measures are described. Since immune network can be treated both as a clustering and a meta-clustering (clusters of clusters) model, beside commonly used clustering quality measures (unsupervised and supervised), we have also investigated immune network structure. The discussion of results is given in Sect. 5.4 – 5.8.

5.1 Quality Measures of the Clustering

Various measures of quality have been developed in the literature, covering diverse aspects of the clustering process. The clustering process is frequently

[13] One could also consider assignment of a single document to more than one context, i.e. fuzzy assignment.

referred as ”learning without a teacher”, or ”unsupervised learning”, and is driven by some kind of similarity measure.

The optimized criterion is intended to reflect some esthetic preferences, like: uniform split into groups (topological continuity) or appropriate split of documents with known a priori categorization. As the criterion is somehow hidden, we need tests if the clustering process really fits the expectations. In particular, we have accommodated for our purposes and investigated the following well known quality measures of clustering [28, 6]:

Average Document Quantization: average cosine distance (dissimilarity) for the learning set between a document and the cell it was classified into. The goal is to measure the quality of clustering at the level of a single cell: $AvgDocQ = \frac{1}{|C|}\sum_{c \in C}\left(\frac{1}{|D_c|}\sum_{d \in D_c} dist(d, c)\right)$, where D_c is the set of documents assigned to the cell c.

This measure has values in the [0,1] interval, the lower values correspond respectively to more ”smooth” inter-cluster transitions and more ”compact” clusters. The two subsequent measures evaluate the agreement between the clustering and the a priori categorization of documents (i.e. particular newsgroup in case of newsgroups messages).

Average Weighted Cluster Purity: average ”category purity” of a cell (cell weight is equal to its density, i.e. the number of assigned documents): $AvgPurity = \frac{1}{|D|}\sum_{c \in C} max_k\left(|D_{k,c}|\right)$, where D is the set of all documents in the corpus and $D_{k,c}$ is the set of documents from category k assigned to the cell c. Similarly, *Average Weighted Cluster Entropy* measure can be calculated, where the $D_{k,c}$ term is replaced with the entropy of the categories frequency distribution.

Normalized Mutual Information: the quotient of the entropy with respect to the categories and clusters frequency to the square root of the product of category and cluster entropies for individual clusters [6].

$$NMI = \frac{\sum_{C \in C}\sum_{k \in K} |D_{k,c}| \ \log\left(\frac{|D_{k,c}| \ |D|}{|D_c| \ |D_c|}\right)}{\sqrt{\left(\sum_{c \in C} |D_c| \ \log\left(\frac{|D_c|}{|D|}\right)\right)\left(\sum_{k \in K} |D_k| \ \log\left(\frac{|D_k|}{|D|}\right)\right)}} \tag{17}$$

where N is the set of graph cells, D is the set of all documents in the corpus, D_c is the set of documents assigned to the cell c, D_k is the set of all documents from category k and $D_{k,c}$ is the set of documents from category k assigned to the cell c.

Again, both measures have values in the [0,1] interval. The higher the value is, the better agreement between clusters and *a priori* given categories.

5.2 Quality of the Immune Network

Beside the clustering structure represented by cells, idiotypic network should be also treated as a meta-clustering model. Similarity between individual clusters is expressed by graph edges, linking referential vectors in antibodies. Thus, there is a need to evaluate quality of the structure of the edges.

There is a number of ways to evaluate idiotypic model structure. In this paper we present the one which we have found to be the most clear for interpretation. This approach is based on the analysis of the edge lengths of the minimal spanning tree (MST) constructed over the set of antibodies, in each iteration of the learning process.

5.3 Histogram-based reclassification measures

Each contextual model (as well as subgraph or map area) represents some topically consistent (meta-)cluster of documents. Traditionally, such a cluster is represented by a single element (e.g. centroid, medoid, reference vector in GNG/SOM, antibody in immune model). Alternative representation of a group of documents have been presented in section 4. It has been shown in section 4.2 that both computational and space complexity of such representation is low. It has numerous advantages such as abandoning of the assumption of spherical shape of clusters and efficient adaptation during incremental learning on dynamically modified data sets. It also allows for the construction of various measures for subspace clusters evaluation. Here we focus only on one such measure, evaluating reclassification properties of contextual groups.

Reclassification aims at measuring stability of the existing structure of the clustering model (both on the meta-level of contexts and on the level of document groups in some subgraphs and map areas). Reclassification measures also the consistency of the histogram-based subspace description with the model-based clustering. For the fixed clustering structure (e.g. some split of the document collection into contexts) we can describe each cluster by a set of histograms, like in section 4.1. Having such histograms built, we can classify each document to its ”most similar” histogram-based space, like in section 4.4. Finally, we can investigate the level of agreement between original (model-based) and the new (histogram-based) grouping.

In the ideal case we expect both groupings to be equal. To assess how far from ideal agreement two groupings are, we construct contingency table. Since the group indexes in original and the new grouping are left unchanged, correctly reclassified objects appear in the diagonal elements of the contingency matrix. Finally, we can calculate measures traditionally used for evaluation of classifiers performance (precision, recall, F-statistics, etc.).

In general case, to take the meta-clustering information into account, we discriminate between different kind of misclassifications. Since contexts and subspaces are also similar on meta-level and this similarity is reflected by the graph edges, we exploit shortest-path algorithm for weighted graphs with non-negative weights (Diskstra algorithm). Likewise binary agreement approach, proper reclassification is assigned with distance equal to 0. Each improper reclassification gets the distance equal to the sum of edges weights on the shortest path between model-based and histogram-based cluster (i.e. context or cell in the graph).

5.4 Experimental settings

The architecture of BEATCA system supports comparative studies of clustering methods at the various stages of the process (i.e. initial document grouping, initial topic identification, incremental clustering, graph model projection to 2D map and visualization, identification of topical areas on the map and its labeling) – consult [18] for details. In particular, we conducted series of experiments to compare the quality and stability of AIS, GNG and SOM models for various model initialization methods, cell/antibody search methods and learning parameters [18]. In this paper we focus only on the evaluation and comparison of the immune models.

This study required manually labelled documents, so the experiments were executed on a widely-used 20 Newsgroups document collection[14] of approximately 20 thousands newsgroup messages, partitioned into 20 different newsgroups (about 1000 messages each). As a data preprocessing step in BEATCA system, entropy-based dimensionality reduction techniques are applied [16], so the training data dimensionality (the number of distinct terms used) was 4419.

Each immune model have been trained for 100 iterations, using previously described algorithms and methods : contexts extraction (section 2), agglomerative identification of redundant antibodies [18], robust construction of the mutated antibodies (section 3.3), time dependent parameters (section 3.4) and CF-Tree based antibody search method [7].

5.5 Impact of the time-dependent parameters

In the first two series of experiment, we compared models built with time-dependent parameters $\sigma_s(t)$ and $\sigma_d(t)$ with the constant, a priori defined values of σ_s and σ_d. As a reference case we took a model where $\sigma_s(t)$ was changed from the initial value 0.05 up to 0.25 and $\sigma_d(t)$ from 0.1 up to 0.4 (cf. section 3.4).

First, we compare the reference model and the four models with constant σ_d. Parameter σ_s has been changed identically as in reference model. The values of σ_d varied from the starting value in the reference model (0.1) up to the final value (0.4) by 0.1 step. The results[15] are presented in Figure 1.

[14] `http://people.csail.mit.edu/jrennie/20Newsgroups/`

[15] All figures present average values of the respective measures in 20 contextual nets

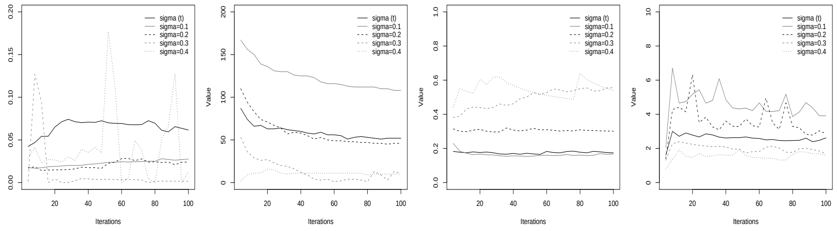


Fig. 1. Time-dependent σ_d: (a) edge length variance (b) network size (c) quantization error (d) learning time

Fig. 1(a) presents variance of the edge length in the minimal spanning tree built over the set of antibodies in the immune memory in i^{th} iteration of the learning process. At first glance one can notice instability of this measure for high values of σ_d. Comparing stable values, we notice that the variance for the reference network has the highest value. It means that the idiotypic network contains both short edges, connecting clusters of more similar antibodies and longer edges, linking more distant antibodies, probably stimulated by different subsets of documents (antigens). Such meta-clustering structure is desirable and preferred over networks with equidistant antibodies (and, thus, low edge length variance).

Comparing network sizes, Fig. 1(b), and quantization error, Fig. 1(c), we observe that for the highest values of σ_d, the set of antibodies reduces to just a few entities; on the other hand - for the lowest values almost all antibodies (universal and over-specialized) are retained in the system's memory. It is not surprising that the quantization error for a huge network (e.g. $\sigma_d = 0.1$) is much lower than for smaller nets. Still, the time-dependent $\sigma_d(t)$ gives similarly low quantization error for moderate network size. Also, both measures stabilize quickly during learning process. Learning time, Figure 1(d), is – to some extent – a function of network size. Thus, for the reference model, it is not only low but very stable over all iterations.

In the next experiment – dually – we compare reference model and another five models with constant σ_s (and varying σ_d). Analogically to the first case, the values of σ_s varied from the initial value 0.05 up to the final value in the reference model 0.25 by 0.05 step. The results are presented in Fig. 2. Due to the space limitations, we restrict the discussion of the results to the conclusion that also in this case time-dependent parameter $\sigma_s(t)$ had a strong, positive influence on the resulting immune model.

A weakness of the approach seems to be the difficulty in selecting appropriate values of the parameters for a given dataset. We investigated independently changes to the values of both parameters, but it turns out that they should be changed "consistently"; that is the antibodies should not be removed too quickly, nor aggregated too quickly. However, once found, there is

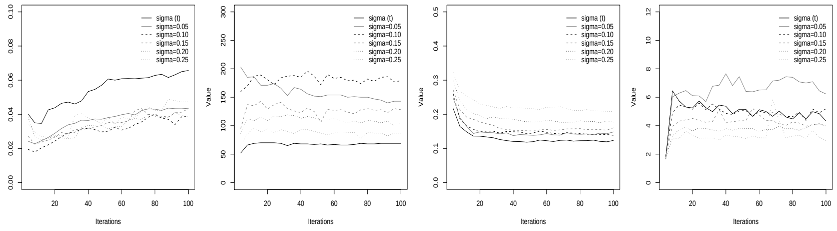


Fig. 2. Time-dependent σ_s: (a) edge length variance (b) network size (c) quantization error (d) learning time

a justified hope that for an incrementally growing collection of documents the parameters do not need to be sought anew, but rather gradually adopted.

5.6 Scalability and comparison with global models

Comparing hierarchical and contextual models described in section 2, with a "flat", global model the most noticeable difference is the learning time[16]. The total time for 20 contextual networks accounted for about 10 minutes, against over 50 minutes for hierarchical network and almost 20 hours (*sic*!) for a global network. Another disadvantage of the global model is high variance of the learning time at single iteration as well as the size of the network. The learning time varied from 150 seconds to 1500 seconds (10 times more!) and the final network consisted of 1927 antibodies (two times more than for contextual model). It should also be noted that in our experimental setting, each model (local and global) has been trained for 100 iterations, but it can be seen (e.g. Figure 5) that the local model stabilizes much faster. Recalling that each local network in the hierarchy can be processed independently and in parallel, it makes contextual approach robust and scalable[17] alternative to the global immune model.

One of the reasons for such differences of the learning time is the representation of antibodies in the immune model. The referential vector in an antibody is represented as a balanced red-black tree of term weights. If a single cell tries to occupy "too big" portion of a document-term vector space (i.e. it covers documents belonging to different topics), many terms which rarely co-occur in a single document have to be represented by a single red-black tree. Thus, it becomes less sparse and - simply - bigger. On the other hand, better separation of terms which are likely to appear in various topics and increasing "crispness" of topical areas during model training leads to faster convergence

[16] By learning time we understand the time needed to create an immune memory consisting of the set of antibodies representing the set of antigens (documents).

[17] Especially with respect to growing dimensionality of data, what - empirically - seem to be the most difficult problem for immune-based approach

and better models, in terms of previously defined quality measures. While the quantization error is similar for global and contextual model (0.149 versus 0.145, respectively), then both supervised measures - showing correspondence between documents labels (categories) and clustering structure - are in favor to contextual model. The final value of the Normalized Mutual Information was 0.605 for the global model and 0.855 for the contextual model and Average Weighted Cluster Purity: 0.71 versus 0.882 respectively.

One can also observe the positive impact of homogeneity of the distribution of term frequencies in documents grouped to a single antibody. Such homogeneity is - to some extent - acquired by initial split of a document collection into contexts. Another cause of the learning time reduction is the contextual reduction of vector representation dimensionality, described in the section 2.

It can be seen that model stabilizes quite fast; actually, most models converged to final state in less than 20 iterations. The fast convergence is mainly due to topical initialization. It should also be noted here that the proper topical initialization can be obtained for well-defined topics, which is the case in contextual model.

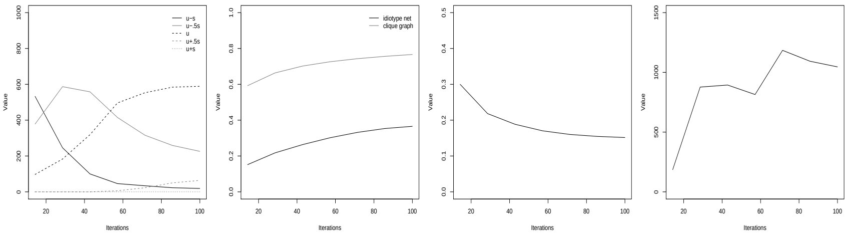


Fig. 3. Global model: (a) edge length distribution (b) clique average edge length (c) quantization error (d) learning time

We have also executed experiments comparing presented immune approach with SOM models: flat (i.e. standard, global Kohonen's map) and our own variant of contextual approach - the hierarchy of contextual maps (C-SOM). To compare immune network structure, with the static grid of SOM model, we have built minimal spanning tree on the SOM grid. Summary of the results can be seen in Figure 4. Again, global model turned out to be of lower quality than both contextual SOM and contextual AIS model. Similarly to the global immune model, also in this case the learning time (over 2 hours) was significantly higher than for the contextual models. Surprisingly, the average edge in contextual SOM model was much longer than in case of contextual immune network and standard SOM, what may be the result of the limitations of the rigid model topology (2D grid). The discussion of the edge length distribution (Figure 4(b)) we defer to the section 5.8.

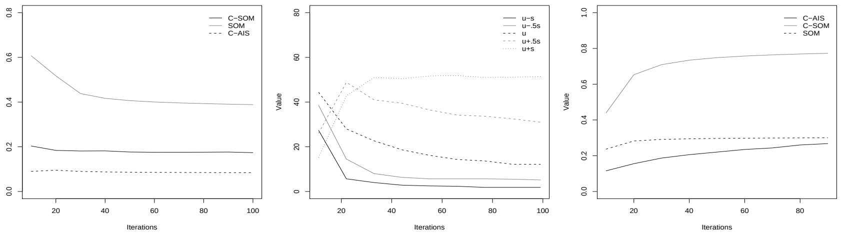


Fig. 4. Immune model vs. SOM: (a) quantization error (b) SOM (MST on SOM grid) edge length distribution (c) average edge length

5.7 Contextual versus Hierarchical model

The next series of experiments compared contextual model with hierarchical model. Figures 5(a) and 5(b) presents network sizes and convergence (wrt Average Document Quantization measure) of the contextual model (represented by black line) and hierarchical model (grey line).

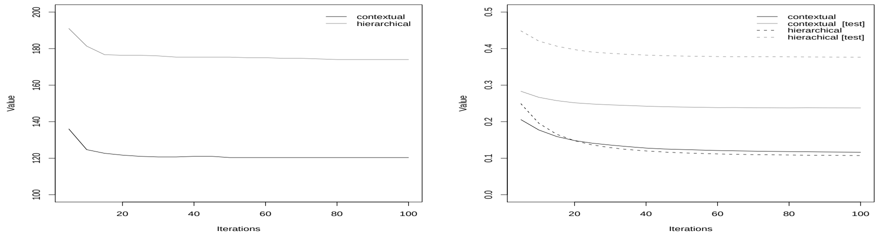


Fig. 5. Contextual vs. hierarchical model: (a) network size (b) quantization error

Although convergence to the stable state is fast in both cases and the quantization error is similar, it should be noted that this error is acquired for noticeably smaller network in contextual case (and in shorter time, as mentioned in previous section).

However, the most significant difference is the generalization capability of both models. For this experiment, we have partitioned each context (group of documents) into training and test subsets (in proportion 10:1). Training documents were used during learning process only, while the quantization error was computed for both subsets. The results are shown in Figure 5(b) – respective learning data sets are depicted with black lines while test data sets with grey lines. Nevertheless quantization error for learning document sets are similar, the difference lies in test sets and the hierarchical network is clearly *overfitted*. Again, there's no room to go into detailed study here, but it can be shown that this undesirable behavior is the result of the noised information brought by additional terms, which finally appears to be not meaningful in the particular context (and thus are disregarded in contextual weights w_{dtG}).

5.8 Immune network structure investigation

To compare robustness of different variants of immune-based models, in each learning iteration, for each of the immune networks: contextual [Fig. 6(b)], hierarchical [Fig. 6(c)], global [Fig. 6(d)] and MST built on SOM grid [Fig. 4(c)], the distributions of the edge lengths have been computed. Next, the average length u and the standard deviation s of the length have been calculated and edges have been classified into five categories, depending on their length, l: shortest edges with $l \leq u{-}s$, short with $l \in (u{-}s, u{-}0.5s]$, medium with $l \in (u{-}0.5s, u{+}0.5s]$, long with $l \in (u{+}0.5s, u{+}s]$ and very long edges with $l > u{+}s$. First category consists of the edges no longer that $u{-}s$, i.e. the shortest edges. Second category contains edges with lengths in interval $(u{-}s, u{-}0.5s]$, next - "average length" edges between $u{-}0.5s$ and $u{+}0.5s$. Last two categories contain longer edges: 4^{th} - edges shorter that $u + s$ and the last one - longer than $u{+}s$.

Additionally, in Figure 6(a), we can see average length of the edges for hierarchical and contextual immune networks (dashed and solid black lines, respectively) and complete graphs on both models' antibodies (cliques - depicted with grey lines). Actually, in both cases clustering structure has emerged and the average length of the edge in the immune network is much lower than in the complete graph. However, the average length for the contextual network is lower, whereas variance of this length is higher. It signifies more explicit clustering structure.

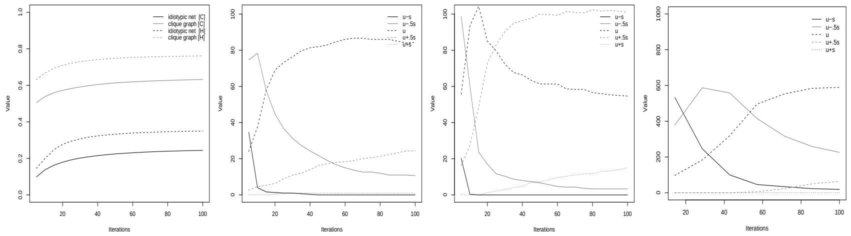


Fig. 6. Edge length distrib.: (a) complete (b) contextual (c) hierarchical (d) global net

There are quite a few differences in edge length distribution. One can notice than in all models, the number of shortest edges diminishes with time. It is coherent with the intention of gradual elimination of the redundant antibodies from the model. However, such elimination is much slower in case of the global model, what is another reason of slow convergence and high learning time. Also in case of SOM model, which has a static topology and no removal of inefficient cells is possible, we can see that the model slowly reduces the number of redundancies, represented by too similar referential vectors.

On the extreme side, the dynamics of the longest edges' distribution is similar in case of the contextual and the global model, but distinct in case of the hierarchical model. This last contains much more very long edges. Recalling that the variance of the edge lengths has been low for this model and the average length has been high, we can conclude that hierarchical model is generally more discontinuous. The same is true for the SOM model, which is another indication of the imperfection of the static grid topology.

6 Related Work

Application of artificial immune systems, especially of the aiNet algorithm, to document clustering is not new. For example, in [22], integrates Principal Component Analysis (PCA) with the original aiNet to reduce the time complexity, with results preferential to hierarchical agglomerative clustering and K-means. A novel hierarchical method of immune system application to text processing was suggested in [5]. In our opinion, our method offers a radical improvement, reducing the space dimensionality not only formally, but also intrinsically - in terms of complexity of the document description (we do not generate denser description vectors like the PCA). An important difference to both of these approaches is our histogram representation of document space dimensions. It allows to represent not only the document affinity, but also the diversity.

The aiNet approach to document clustering belongs to a wider family of clustering techniques that can be called "networked clustering" as the clusters are not independent, but rather form a network of more or less related items. Other prominent technologies in this class include self-organizing maps [19], growing neural gas [11] and similar techniques. Like aiNet, also this category of systems suffers from problems of scalability. Our idea of contextual clustering can serve as a remedy for the performance of theses systems also.

The networked clustering methods lack generally good measures of quality. Neither the supervised (external) quality measures nor unsupervised (internal) ones derived for independent clusters do not reflect the quality of link structure between the clusters. Kohonen proposed therefore map continuity measures, based on differences between neighboring map points. In this chapter we presented a generalization of his proposals taking into account other structures than the pure planar map.

Let us note that some document classification methods like ones based on ID3 of Quinlan use in some sense "contextual" information when selecting the next attribute (term) for split of document collection. Our contextual method exploits the split much more thoroughly, extracting more valuable local information about the term collection.

Many document clustering techniques suffer from the tendency of forming spherical clusters. The histogram-based characterization of clusters allows for departing from this shortcoming not only in case of artificial immune systems.

7 Concluding remarks

The contextual model described in this paper admits a number of interesting and valuable features in comparison with global and hierarchical models used traditionally to represent a given collection of documents. Further, when applying immune algorithm to clustering the collection of documents, a number of improvements was proposed. These improvements obey:

- Identification of redundant antibodies by means of the fast agglomerative clustering algorithm [18].
- Fast generation of mutated clones without computation of their stimulation by currently presented antigen. These mutants can be characterized by presumed ability of generalization (cf. section 3.3).
- Time-dependent parameters σ_d and σ_s. In general we have no a recipe allowing to tune both the parameters to a given dataset. In original approach [10] a trial-and-error method was suggested. We observed that in highly dimensional space the value of σ_d is almost as critical as the value of σ_s. Hence we propose a "consistent" tuning of these parameters – cf. section 3.4. The general recipe is: carefully (i.e. not to fast) remove weakly stimulated and too specific antibodies and carefully splice redundant (too similar) antibodies.
- Application of the CF-trees [27] for fast identification of winners (most stimulated memory cells) [7].

With these improvements we proposed a new approach to mining high dimensional datasets. The contextual approach described in section 2 appears to be fast, of good quality (in term of indices introduced in sections 5.1 and 5.2) and scalable (with the data size and dimension).

Clustering high dimensional data is both of practical importance and at the same time a big challenge, in particular for large collections of text documents. The paper presents a novel approach, based on artificial immune systems, within the broad stream of map type clustering methods. Such approach leads to many interesting research issues, such as context-dependent dictionary reduction and keywords identification, topic-sensitive document summarization, subjective model visualization based on particular user's information requirements, dynamic adaptation of the document representation and local similarity measure computation. We plan to tackle these problems in our future work. It has to be stressed that not only textual, but also any other high dimensional data may be clustered using the presented method.

Acknowledgements

This research was partially supported by the European Commission and by the Swiss Federal Office for Education and Science with the 6th Framework Programme project REWERSE no. 506779 "Reasoning on the Web with Rules and Semantics" (cf. `http://rewerse.net`). Krzysztof Ciesielski was partially supported by MNiSW grant no. N516 005 31/0646.

References

1. A. Baraldi and P. Blonda. A survey of fuzzy clustering algorithms for pattern recognition. *IEEE Trans. on Systems, Man and Cybernetics*, 29B:786–801, 1999.
2. A. Becks. Visual knowledge management with adaptable document maps. *GMD research series*, 15, 2001.
3. M.W. Berry, Z. Drmač, and E.R. Jessup. Matrices, vector spaces and information retrieval. *SIAM Review*, 41(2):335–362, 1999.
4. J.C. Bezdek and S.K. Pal. *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data.* IEEE, New York, 1992.
5. G. B. P. Bezerra, T. V. Barra, M. F. Hamilton, and F. J. von Zuben. A hierarchical immune-inspired approach for text clustering. In *Proc. Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'2006)*, volume 1, pages 2530–2537, 2006.
6. C. Boulis and M. Ostendorf. Combining multiple clustering systems. In *Proc. of 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2004)*, pages 63–74. Springer-Verlag, LNAI 3202, 2004.
7. K. Ciesielski, M. Dramiński, M. Kłopotek, D.Czerski, and S.T. Wierzchoń. Adaptive document maps. In *Proceedings of the Intelligent Advances in Soft Computing 5*, pages 109–120. Springer-Verlag, 2006.
8. K. Ciesielski and M. Kłopotek. Text data clustering by contextual graphs. In L. Todorovski, N. Lavrac, and K.P. Jantke, editors, *Discovery Science*, pages 65–76. Springer-Verlag, LNAI 4265, 2006.
9. L.N. de Castro and J. Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach.* Springer, 2002.
10. L.N. de Castro and F.J. von Zuben. An evolutionary immune network for data clustering. In *SBRN'2000*, pages 84–89. IEEE Computer Society Press, 2000.
11. B. Fritzke. Some competitive learning methods, 1997. `http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper`.
12. M. Gilchrist. Taxonomies for business: Description of a research project. In *11 Nordic Conference on Information and Documentation*, Reykjavik, Iceland, May 30 - June 1 2001. `http://www.bokis.is/iod2001/papers/Gilchrist_paper.doc`.
13. C. Hung and S. Wermter. A constructive and hierarchical self-organising model in a non-stationary environment. In *Int. Joint Conference in Neural Networks*, pages 2948–2953, 2005.
14. S.Y. Jung and K. Taek-Soo. An incremental similarity computation method in agglomerative hierarchical clustering. *Journal of Fuzzy Logic and Intelligent System*, December 2001.

15. M. Kłopotek. A new bayesian tree learning method with reduced time and space complexity. *Fundamenta Informaticae*, 49(4):349–367, 2002.
16. M. Kłopotek, M. Dramiński, K. Ciesielski, M. Kujawiak, and S.T. Wierzchoń. Mining document maps. In M. Gori, M. Ceci, and M. Nanni, editors, *Proc. of Statistical Approaches to Web Mining Workshop (SAWM) at PKDD'04*, pages 87–98, Pisa, Italy, 2004.
17. M. Kłopotek, S. Wierzchoń, K. Ciesielski, M. Dramiński, and D. Czerski. *E-Service Intelligence – Methodologies, Technologies and Applications. Part II: Methodologies, Technologies and Systems*, volume 37 of *Studies in Computational Intelligence*, chapter Techniques and technologies behind maps of Internet and Intranet document collections. Springer, 2007.
18. M. Kłopotek, S. Wierzchoń, K. Ciesielski, M. Dramiński, and D. Czerski. *Conceptual Maps of Document Collections in Internet and Intranet. Coping with the Technological Challenge.* IPI PAN Publishing House, Warszawa 2007.
19. T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences.* Springer, Berlin, Heidelberg, New York, 2001.
20. K. Lagus, S. Kaski, and T. Kohonen. Mining massive document collections by the WEBSOM method. *Information Sciences*, 163/1-3:135–156, 2004.
21. G. Salton. *The SMART Retrieval System – Experiments in Automatic Document Processing.* Prentice-Hall, Upper Saddle River, NJ, USA, 1971.
22. N. Tang and V. R. Vemuri. An artificial immune system approach to document clustering. In *Proceedings of the 2005 ACM symposium on Applied Computing Santa Fe, New Mexico*, pages 918–922, 2005.
23. J. Timmis. aiVIS: Artificial immune network visualization. In *Proceedings of EuroGraphics UK 2001 Conference*, pages 61–69. Univeristy College, London, 2001.
24. C.J. van Rijsbergen. *Information Retrieval.* Butterworths, London, 1979. `http://www.dcs.gla.ac.uk/Keith/Preface.html`.
25. S.T. Wierzchoń. *Artificial immune systems. Theory and applications (in Polish).* Akademicka Oficyna Wydawnicza EXIT Publishing, Warszawa, 2001.
26. D.R. Wilson and T.R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–286, 2000.
27. T. Zhang, R. Ramakrishan, and M. Livny. Birch: Efficient data clustering method for large databases. In *Proc. ACM SIGMOD Int. Conf. on Data Management*, pages 103–114, 1997.
28. Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. `http://www-users.cs.umn.edu/~karypis/publications/ir.html`.

Index

Term Distribution-based Initialization of Fuzzy Text Clustering

Krzysztof Ciesielski[1], Mieczysław A. Kłopotek[1,3], Sławomir T. Wierzchoń[1,2]

[1] Institute of Computer Science, Polish Academy of Sciences,
ul. Ordona 21, 01-237 Warszawa, Poland
[2] Institute of Informatics, Univ. of Gdansk, Wita Stwosza 57, 80-952 Gdansk.
[3] Institute of Informatics, Univ. of Podlasie in Siedlce.
kciesiel,klopotek,stw@ipipan.waw.pl

Abstract. We investigate the impact of an initialization strategy on the quality of fuzzy-based clustering, applied to creation of maps of text document collection. In particular, we study the effectiveness of bootstrapping as compared to traditional "randomized" initialization. We show that the idea is effective both for traditional Fuzzy K-Means algorithm and for a new one, applying histogram-based cluster description.

1 Introduction

In this research clustering is proposed as a tool allowing to reveal the internal structure of a collection, e.g., we are interested in grouping documents by topics, subtopics, etc. When using clustering we subsume so-called *cluster hypothesis*, according to which the documents in the same cluster behave similarly with respect to relevance to information needs.

A common approach to document representation and processing relies upon treating each textual document as a set of terms, i.e. words or phrases extracted automatically from the documents themselves. Then to each term in a given document we assign a numeric weight , representing an estimate of this term usefulness when distinguishing the document from other documents in the same collection. The weights assigned to the terms in a given document d can then be interpreted as the coordinates of d in the document space, i.e. $d = (w_1, \ldots, w_{|T|})$ is a point in $|T|$-dimensional document space D; here $|T|$ is the cardinality of the set of terms T . Among many existing weighting schemes the most popular is so-called *tfidf* scheme, see Sect. 3 for details. Here *tf*, i.e. term frequency, is a document-specific statistic, while the inverse document frequency *idf* is a global statistics characterizing a given term within an entire collection of documents. To get a deeper insight into the nature of different sub-collections of documents, we propose a context dependent weighting scheme; besides the two statistics so-called term specificity is additionally introduced. This allows to characterize the documents by a varying sets of weights $(w_1, \ldots, w_{|C(T)|})$ in different contexts[4];

[4] By a context one shall intuitively understand a sufficiently large (for statistics sake) set of documents with sufficiently uniform topics.

here $|C(T)|$ stands for the cardinality of the set of terms specific for a context C. In this paper we will examine the usefulness of this new "contextual" weighting scheme against the "global" scheme expressed by the *tfidf* weights.

With the vector representation we can adopt any clustering algorithm to reveal the internal structure of the documents collection. A popular approach to clustering relies upon minimizing an objective function expressing (weighted) distance between objects from a given cluster (group) and a characteristic point of this group (e.g. its center of gravity). Although simple and efficient, these algorithms heavily depend on the initial partition subjected further modification. In this paper we present a new, boosting-like, initialization scheme.

The paper is organized as follows: in section 2, fuzzy clustering approach and its drawbacks are briefly outlined. In section 3 the concept of clustering space is presented. Following that concept, FKH clustering algorithm is proposed. The key contribution of this paper is boosting-like initialization algorithm, described in section 4. Section 5 gives experimental results on FKH algorithm and proposed initialization method. Section 6 summarizes the paper.

2 Fuzzy clustering algorithm

Let $D = \{d_1, \ldots, d_n\}$ be a set of objects (documents in our case). Each object is described by a set of real-valued features, $d_i = (w_{i,1}, \ldots, w_{i,|T|})$, i.e. it can be viewed as a point in $|T|$-dimensional Euclidean space, $\Re^{|T|}$. The aim of cluster analysis is to partition the set D into a (usually predefined) number $K > 1$ of homogenous groups. The notion of homogeneity is understood in such a way that two objects assigned to the same class are much more similar one to another than any two objects assigned to different classes. Usually the similarity between pairs of objects is measured in terms of a distance metrics. Such an approach causes troubles in highly dimensional feature spaces, however, because, as shown in [1], with increasing dimensionality, the "contrast", i.e. relative difference between the closest and the farthest neighbor, is decreasing. In case of documents grouping the cosine measure is commonly used; it can be converted to a distance by the mapping $dist(d_i, d_j) = 1 - \cos(d_i, d_j)$.

In such a context, when similarity between pairs of objects is measured in terms of their mutual distance, a natural method for inducing partition of the set D is to use the objective function [2]

$$J_\alpha = \sum_{d \in D} \sum_{j=1}^{K} dist^2(d, \overline{v}_j) u^\alpha_{d,C(\overline{v}_j)} \tag{1}$$

where $u_{d,C(\overline{v}_j)}$ is a degree of membership of object d to j-th class, $\overline{v}_j$ is a characteristic of group C, usually defined as a (weighted) center of gravity called centroid, and $\alpha \geq 1$ is a parameter. Assuming that $u_{d,C(\overline{v}_j)} \in \{0, 1\}$, $\alpha = 1$ and that we are searching for the minimum of J_α, we obtain well known crisp K-means algorithm. When $u_{d,C(\overline{v}_j)} \in [0, 1]$ and $\alpha > 1$, minimization of the index J_α leads to the fuzzy K-means algorithm, called hereafter FKM for brevity [2].

Both these algorithms iteratively improve the initial partition matrix $U = [u_{d,C(\overline{v}_j)}]$ by modifying the centroids after the objects have been reassigned to the groups[5] computed and new cluster centers are determined. The quality of resulting partition is measured e.g. in terms of the index [2]

$$F_K(U) = \frac{1}{n} \cdot trace\left(U \cdot U^{\mathrm{T}}\right) \tag{2}$$

measuring the degree of fuzziness of the partition represented by the matrix U. This measure decreases when documents belong to more than one class and takes the lowest value for documents belonging to the same degree to all the classes.

An attractive feature of these algorithms is their numerical simplicity and linear time complexity with respect to the number of objects, what allows to process large collections of objects. They admit a number of drawbacks, however, like: (a) final partition heavily depends on the data order and on the initial data partition, (b) the algorithms are sensitive to the outliers, (c) the number of clusters must be known in advance, and finally, (d) these algorithm can be used only in case of numerical data representation.

To cope with the first drawback a number of approaches was proposed, like [6], [8], or [7]. In this paper a new method (inspired by the boosting algorithm used in machine learning) is proposed. It is especially useful when processing large datasets since it clusters only a sample of objects and re-clustering is performed when there are objects which do not fit to the existing partition. Before introducing this method, some remarks on contextual clustering are needed.

3 Clustering Space

Recall first, that when processing text documents the entries of the vectors d_i are so-called *tfidf* indices defined as, see e.g. [9]:

$$w_{t,d} = f_{t,d} \times \log \frac{n}{f_t^{(D)}} \tag{3}$$

where n is the number of documents in the collection D, $f_{t,d}$ is the number of occurrences of the term t in the document d, and $f_t^{(D)}$ is the number of documents in the collection D containing at least one occurrence of the term t.

The scheme (3) assumes that all the terms are weighted from the same (global) perspective. But it is obvious that the weight of a term can vary according to the context in which it is used. Hence we left the notion of cluster center, and we assume "relativistic" point of view treating the collection D as a continuous space, called hereafter clustering space, where each point p can be viewed as a cluster center. In such a space we define a proximity relationship among the documents, and $\mu_{d,C(p)}$ quantifies a normalized degree of membership of document d to the cluster $C(p)$ with center p, where normalization is

[5] More precisely, for a given partitioning group centers $\overline{v}_j, j = 1, \ldots, K$ are calculated and the objects are assigned to the groups with most similar centroids. Then the cycle repeats: centers $\overline{v}_j$ are updated and the objects are reassigned to the groups.

over all $C(p)$ [5]. Thus, $\mu_{d,C(p)}$ plays role analogous to $u_{d,C(p)}$ in fuzzy clustering approach. Denoting $|C(p)| = \sum_{d \in D} \mu_{d,C(p)}$ the fuzzy cardinality of this cluster, and combining it with $f_{t,d}$ and $\mu_{d,C(p)}$, we introduce a notion of specificity $s_{t,C(p)}$ of the term t in the cluster $C(p)$ as $s_{t,C(p)} = f_{t,D}^{-1} \cdot \sum_{d \in D} f_{t,d} \cdot \mu_{d,C(p)}$, where $f_{t,D}$ is the total frequency of term t in the collection D. With these notions we introduce new contextual term weighting formula

$$w_{t,d,C(p)} = s_{t,C(p)} \times f_{t,d} \times \log \frac{|C(p)|}{f_{t,C(p)}} \tag{4}$$

where $f_{t,C(p)} = \sum_{\{d: f_{t,d} > 0\}} \mu_{d,C(p)}$ is the fuzzy count of documents in the collection $C(p)$ containing at least one occurrence of the term t; we assume that $w_{t,d,C(p)} = 0$ if $f_{t,C(p)} = 0$.

Now, the context-free weight *tfidf* given by the equation (3) is replaced by an averaged local weight

$$w_{t,d} = l \cdot \sum_{p \in HS} \left(\mu_{d,C(p)} \cdot w_{t,d,C(p)} \right) \tag{5}$$

where HS is the unit hyper-sphere and l is a normalizing constant placing the document d on the unit hyper-sphere.

3.1 Histograms of term weights

Furthermore, we can consider properties of each term individually (for a single document), or in a broader context of a given subset of documents, e.g. cluster $C(p)$ as a whole. In the latter case, the values of the weight w for a given term t for each document $d \in C(p)$ [6] are treated as observed values of a random variable with underlying continuous probability distribution. In practical cases, the continuous distribution will be approximated by a discrete one, so that the information about the random variable distribution for the term t can be summarized as a histogram $h_{t,C}$[7].

Single interval of a histogram $h_{t,C}$ represents the number of occurrences of a discretized value of the term weighting function w for the term t in the document collection C. The interval values can be in turn transformed to the relative frequencies via the plain normalization $h'_{t,C}(q) = f_{t,C}^{-1} \cdot h_{t,C}(q)$, where $f_{t,C} = \sum_{i=1}^{Q} h_{t,C}(q)$ is the total number of documents $d \in C$ containing term t. The normalized frequency distribution approximates the probability distribution of an unknown variable describing the weight of occurrence of the term t in randomly chosen document $d \in C$.

3.2 Histogram-based clustering algorithm

Finally, we can define histogram-based document membership in a given context. A document d fits well to a given context (represented by a fuzzy cluster C) if its

[6] where fuzzy cluster $C(p)$ is defined as $C(p) = \{d \in D : \mu_{d,C(p)} > 0\}$

[7] a deeper discussion of the histograms of term weights is given in [5]

observed term weights $w_{t,d}$ are typical for the "majority" of documents in space C, i.e. the histogram-based probability of the weights observed in d is high. We expect such a document to follow some topic-specific term distribution. Thus, instead of fuzzy degree of membership u, defined in section 2 on the basis of distance from the single fuzzy center of gravity, we can define histogram-based (unnormalized) degree of membership $m_{d,C(p)}$:

$$m_{d,C(p)} = \frac{\sum_{t \in d} m_{t,C(p)} \cdot h_{t,C(p)}(q)}{\sum_{t \in d} m_{t,C(p)}} \tag{6}$$

where $m_{t,C(p)}$ is some measure of term significance in context $C(p)$[8], $h_{t,C(p)}$ is a histogram for term t and histogram index q is computed by discretization Δ of term weight $w_{t,d,C(p)}$ (i.e. $q = \Delta\left(w_{t,d,C(p)}\right)$). After normalization over all $C(p)$, unnormalized degrees $m_{d,C(p)}$ become $\mu_{d,C(p)}$.

Such a distribution-based degree of membership, computed for every context $C(p)$, after normalization gives us a vector of fuzzy-like probabilities $\mu_{d,C(p)}$, analogously to $u_{d,C(p)}$ from section 2. In section 4, we will use a matrix U of the normalized distribution-based contextual membership $\mu_{d,C(p)}$ to produce set of clusters covering diverse topical aspects of the whole document collection D.

Enriched fuzzy clusters description, based on sets of term histograms, are employed in an incremental clustering algorithm, see Algorithm 1, called Fuzzy K-Histograms, or FKH for brevity that we propose in this paper. It joins two known paradigms of clustering: the fuzzy clustering and the subspace clustering. The method differs essentially from fuzzy clustering in that it is designed solely for text data and is based on contextual vector representation and histogram-based description of vector subspaces.

Algorithm 1 Fuzzy K-Histograms, **FKH**

```
1. Fix the number of clusters, $K \geq 2$ and initialize the fuzzy partition matrix $U(\tau_0)$,
   $\mu_{d,C}(\tau_0) \in [0,1]$
2. Repeat steps 3-5 until the partition matrix stabilizes, i.e. $||U(\tau) - U(\tau-1)|| =$
   $max_{d,C} |\mu_{d,C}(\tau) - \mu_{d,C}(\tau-1)| \leq \varepsilon$)
3. For each term $t$ in the document $d$, having fixed matrix $U(\tau)$, compute a new
   vector representation $w_{t,d,C}$ in current context $C = C(\tau)$.
4. For each context $C(\tau)$ construct a new histogram description $\{h_{t,C} : t = 1, \ldots, T\}$
5. Calculate new degrees of membership $m_{d,C}$ and normalize them to $\mu_{d,C}$ in order
   to get new partition matrix $U(\tau+1)$
```

Like FKM, the FKH algorithm starts with an initial split into subgroups, represented by a matrix $U(\tau_0)$, rows of which represent documents, and columns represent topical groups. Iteratively, we adapt (a) the document representation, (b) the histogram description of contextual groups, and (c) membership degree of documents and term significance in the individual groups. These modifications

[8] In the simplest case, term significance could be equal to one for every term t. However, since majority of terms in a given context is expected to be irrelevant, in [5] we propose another measure, proportional to the weighted field under histogram plot.

recursively lead to a precise description of a contextual subspace in terms of the membership degree of documents and significance of terms in a particular context; further an insight into the documents similarity is offered. So we can view the algorithm as a kind of reinforcement learning.

Both FKM and FKH algorithms are very sensitive to the choice of the initial fuzzy clusters memberships $\mu_{d,C}$. We can start without any knowledge of document similarity, via a random assignment of documents to a number of groups and global term weighing. But through the iterative process some terms specific for a group would be strengthened, so that class membership of documents would be modified, hence also their vector representation and, indirectly, similarity definition. As experimental results show, FKH initialized with the sampling-based method presented in the next section behaves much more stable (i.e. clusters are retained over different cross-validated document subsets) and it outperforms FKM algorithm in terms of supervised cluster quality (cf. section 5).

4 Sampling-based initialization algorithm

Let us return to the main idea of boosting-like initialization algorithm. It starts from a random sample of documents drawn from the set D in case of global clustering, or from a contextual group C in case of contextual clustering. The sample size is $M = \min(1000, 0.1n)$ and the documents are drawn from the uniform distribution. The sample is clustered into K groups, where K is a parameter.

Surely, the sample drawn in such a way may appear to be too arbitrary. Thus, in case of the FKM algorithm the values of J_α and of F_K, defined in equations (1) and (2) respectively, are computed and a decision is made if the resulting partition can be accepted. It was assumed that the sample is rejected if $|F_K(U^{end}) - F_K(U^{start})| \leq \varepsilon$, where U^{start} is an initial partition matrix, while U^{end} is the partition matrix returned by the algorithm.

Next, for each cluster a corresponding histogram description is created and the spanning-tree of clusters is constructed.The tree plays a control role, e.g. it is used when the clusters are merged or their number is reduced.

If the distance (measured in terms of so-called Hellinger's divergence[9]) between any two clusters is less than a pre-specified threshold value ε – these clusters are merged. The resulting set of clusters S is saved. This ends a single step of the algorithm.

In the subsequent steps new samples of documents are drawn randomly, but this time the probability of including drawn document d into new sample equals:

$$P(d) = 1 - max\{m_{d,C(p)}\colon C(p) \in S\} \tag{7}$$

[9] which is – contrary to Kullback-Leibler measure – a symmetric distance measure between the two distributions, cf. e.g. A.Basu, I.R.Harris, S.Basu, *Minimum distance estimation: The approach using density-based distances*. Handbook of Statistics, 15, 1997, pp. 21–48.

where S the set of clusters computed in previous step, and $m_{d,C(p)}$ is the degree of membership of the document d into the cluster $C(p)$, computed according to the contextual weighting scheme [4].

This way new sample is formed by the documents with low degree of membership to the existing set of clusters. The aim of such a procedure is recurrent improvement of the partition which should represent all the themes from the collection D.

Surely, this new sample is subjected to clustering and new clusters are added to the set S. Then both the contextual descriptions and the spanning tree of clusters are updated. The procedure terminates after a given number of steps or when the set S stabilizes, i.e. no cluster is modified and no new cluster is added.

5 Experimental results

In this section we briefly comment our experiments concerning the quality of the contextual clustering by means of the Fuzzy K Means (FKM) algorithm and adaptive histogram-based algorithm (FKH). Further, we compare the influence of the boosting-like initialization on both algorithms.

In these experiments we use the following sets of documents: *20 Newsgroups*[10], *12 Newsgroups*[11], *WebKb*[12] and *Reuters*[13]. The *box-and-whisker* plot depicted on Figures 1 and 2 illustrates the process of global clustering of the *Reuters* data. Here a single box presents the distribution of the F_K index defined in equation (2) in subsequent iterations with respect to different divisions into training and testing sets (10-fold cross-validation). The horizontal bars represent median and the lower and upper sides of the box represent 25% and 75% quartiles; lastly the "whiskers" represent lower and upper extreme values and the dots – the outliers.

Let us comment variations of the F_K measure in consecutive iterations of particular clustering algorithms. In case of the FKM with random initialization (Fig. 1(a)) the algorithm almost immediately get stuck in a local minima. Another consequence of random initialization is large variance of this measure for particular validating data sets. Similar phenomena, although in a smaller scale is observed in case of the FHC algorithm, see Fig. 2(a), where we observe minor improvements of the F_K measure, but the results are saddled with large variance and highly depend on the content of the sets created during cross-validation.

The quality of the results improves remarkably when we use our boosting-like initialization. However even in this case we observe rather large variation of the degree of fuzziness characterizing particular clusters. Similar behavior has been observed for the remaining test sets.

[10] `http://people.csail.mit.edu/jrennie/20Newsgroups/`

[11] a sample of 8094 messages from 12 groups; sizes vary from 326 to 1000 messages

[12] `http://www.cs.cmu.edu/~TextLearning/datasets.html`

[13] `http://www.ics.uci.edu/~kdd/databases/reuters21578/reuters21578.html`

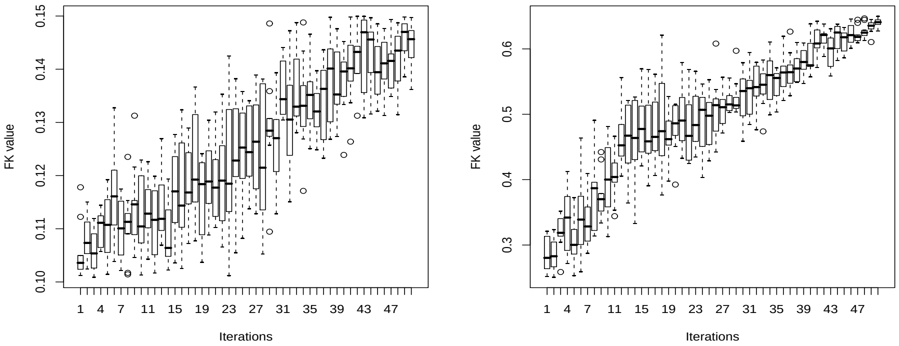


Fig. 1. Distribution of the F_K measure in consecutive iterations of FKM algorithm tested on *Reuters* data set: (a) random initialization (b) boosting-like initialization.

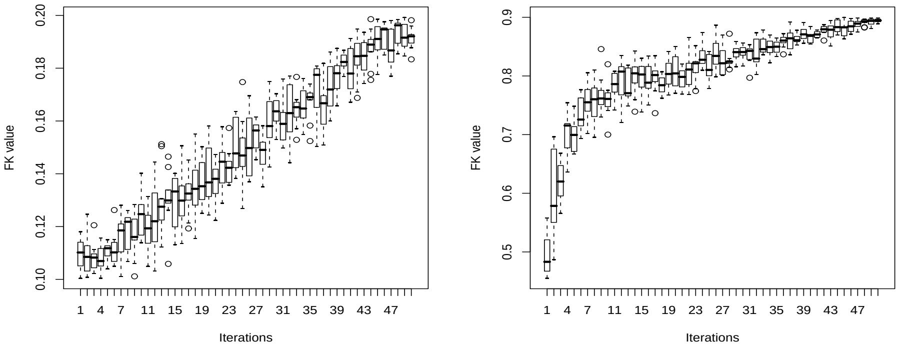


Fig. 2. Distribution of the F_K measure in consecutive iterations of FKH algorithm tested on *Reuters* data set: (a) random initialization (b) boosting-like initialization

Tables 1 and 2 present final values of the normalized mutual information (cf. e.g. [3]), measuring agreement between a priori given document categories and the identified clusters for respective test datasets. By *random* we denote variant in which cross-validated samples are not initialized at all (i.e. initial fuzzy partition, represented by matrix U is random). On the other hand, *initialized* means initialization via the sampling algorithm described in section 4. We also

NMI K-Means	*12News*	*20News*	*Reuters*	*WebKb*
random/direct	0.135 ± 0.09	0.146 ± 0.08	0.102 ± 0.08	0.19 ± 0.09
initialized/direct	0.323 ± 0.06	0.364 ± 0.02	0.151 ± 0.02	0.621 ± 0.08
random/recursive	0.161 ± 0.08	0.176 ± 0.08	0.111 ± 0.1	0.189 ± 0.1
initialized/recursive	0.371 ± 0.04	0.329 ± 0.01	0.128 ± 0.03	0.674 ± 0.05

Table 1. Normalized Mutual Information for the contextual groups generated by FKM

NMI Histograms	*12News*	*20News*	*Reuters*	*WebKb*
random/direct	0.197 ± 0.07	0.208 ± 0.06	0.125 ± 0.08	0.587 ± 0.04
initialized/direct	0.392 ± 0.02	0.457 ± 0.01	0.326 ± 0.04	0.752 ± 0.02
random/recursive	0.256 ± 0.08	0.371 ± 0.06	0.117 ± 0.07	0.634 ± 0.04
initialized/recursive	0.453 ± 0.01	0.495 ± 0.01	0.306 ± 0.03	0.727 ± 0.02

Table 2. Normalized Mutual Information for the contextual groups generated by FKH

considered two strategies of partitioning the data: a *direct* one, where for both FKM and FKH algorithms the number K of clusters was given as an input parameter, and *recursive* one, where the recursive splitting of large clusters into smaller ones was driven by the external criterion of thematic homogeneity. The criterion, exploiting Hellinger divergence between respective term distributions in the clusters of documents, relaxes the requirement of fixing in advance the number K of clusters. In case of four variants displayed in tables, one should note the influence of the initialization on the final results. Recurrent methods prove their advantage over the direct methods with respect to the mean value and the variance of the final result.

The robustness of the proposed approach is confirmed by in-depth analysis of the interrelationships among topical clusters identified by FKH algorithm and the document categories of examplary *20 Newsgroups* collection. The number of clusters identified by the FKH algorithm is lower than the total number of newsgroups in the collection (9 vs. 20), however, one can notice that a single cluster gathers documents from categories (i.e. newsgroups) which are closely related. For instance, one cluster is comprised mostly of documents related to religion (e.g. `alt.atheism`, `talk.religion.misc`, `talk.politics.mideast`), the other cluster contains the documents related to computer issues (`comp.graphics`, `comp.os.ms-windows.misc`, `comp.sys.ibm.pc.hardware`, `comp.sys.mac.hardware`, `comp.windows.x`), and other clusters consist of predominantly sport-related, motorization-related and science-related documents. In BEATCA search engine[14], clusters identified by FKH algorithm are used later on to construct thematical maps and visualize more subtle intertopical proximities in graphical form. Thus, the property of grouping similar newsgroups together is an additional advantage, since all of the thematically related messages will be presented on a single map (so-called contextual map).

6 Conclusions

In this paper we explore the consequences of document cluster characterization via term (importance) distribution histograms . This idea offers a deeper insight into the role played by the terms in formation of a particular cluster. So a full profit can be taken from our earlier idea of ”contextual clustering”, that is of representing different document clusters in different subspaces of a global vector space. Histogram-based approach leads to many efficient algorithms for textual

[14] cf. e.g. `http://www.ipipan.eu/~klopotek/BEATCA/pdf/AIMSA2006.pdf`

data processing purposes, to mention only vector space dimensionality reduction or keyword and keyphrases identification. In this paper we focus on the proposal of a fuzzy text clustering algorithm based on local (”contextual”) document-to-cluster similarity and fuzzy clustering initialization via iterative sampling-based method in the vein of boosting.

We have observed the following properties of the presented algorithms:

- advantage of quality and stability of clustering structure identified by our FKH over the known FKM algorithm, both in direct and in recursive variant of these algorithms
- positive impact of proposed initialization method on clustering stability (i.e. reduction of variance over cross-validated data samples), in case of both FKM and FKH algorithm
- positive impact of proposed initialization method on supervised clustering quality (agreement between a priori given labels and identified clusters)
- positive impact of recursive variant of FKH algorithm on thematical homogeneity clustered together as a single contextual group

Acknowledgements: research partly supported by the Polish state grant No. N516 01532/1906 and co-financed with the 6FP project REWERSE no. 506779.

References

1. C.C. Aggarwal, A. Hinneburg, and D.A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *Proc. 8th International Conference Database Theory (ICDT 2001)*, pages 420–430. Springer-Verlag, LNCS 1973, 2001.
2. J.C. Bezdek and S.K. Pal. Fuzzy models for pattern recognition: Methods that search for structures in data. *IEEE, New York*, 1992.
3. C. Boulis and M. Ostendorf. Combining multiple clustering systems. In *Proc. of 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2004)*, pages 63–74. Springer-Verlag, LNAI 3202, 2004.
4. K. Ciesielski and M. Kłopotek. Text data clustering by contextual graphs. In L. Todorovski, N. Lavrac, and K.P. Jantke, editors, *Discovery Science (DS-2006)*, pages 65–76, Barcelona, Spain, 2006. Springer-Verlag, LNAI 4265.
5. K. Ciesielski and M. Kłopotek. Towards adaptive web mining: Histograms and contexts in text data clustering. In M.R. Berthold and J. Shawe-Taylor, editors, *Intelligent Data Analysis (IDA-2007)*, pages 284–295, Ljubljana, Slovenia, September 2007. Springer-Verlag, LNCS 4723.
6. E. Forgy. Cluster analysis of multivariate data: Efficiency versus interpretability of classification. *Biometrics*, 21:768–780, 1965.
7. L. Kaufman and P.J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis.* Wiley series in probability and mathematical statistics : Applied probability and statistics. Wiley, New York, 1990.
8. J.B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley, 1967. University of California Press.
9. G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, 1983.

A Service-Oriented Approach for Curriculum Planning and Validation

Matteo Baldoni[1], Cristina Baroglio[1], Ingo Brunkhorst[2],
Elisa Marengo[1], Viviana Patti[1]

[1] Dipartimento di Informatica — Università degli Studi di Torino
c.so Svizzera, 185, I-10149 Torino (Italy)
{baldoni,baroglio,patti}@di.unito.it, elisa.mrng@gmail.com
[2] L3S Research Center, University of Hannover
D-30539 Hannover, Germany
brunkhorst@l3s.de

Abstract. We present a service-oriented personalization system, set in an educational framework, based on a semantic annotation of courses, given at a knowledge level (what the course teaches, what is requested to know for attending it in a profitable way). The system supports users in building personalized curricula, formalized by means of an action theory. It is also possible to verify the compliance of curricula w.r.t. a model, expressing constraints at a knowledge level. For what concerns the first task, classical planning techniques are adopted, which take into account both the student's initial knowledge and her learning goal. Instead, curricula validation is done against a model, formalized as a set of temporal constraints. We have developed a prototype of the planning and validation services, by using -as reasoning engines- SWI-Prolog and the SPIN model checker. Such services will be supplied and combined as plug-and-play personalization services in the Personal Reader framework.

1 Introduction and Motivation

The birth of the Semantic Web brought along standard models, languages, and tools for representing and dealing with machine-interpretable semantic descriptions of Web resources, by giving a strong new impulse to research on personalization. The introduction of machine-processable semantics makes the use of a variety of reasoning techniques for implementing personalization functionalities possible, widening the range of the forms that personalization can assume. So far, reasoning in the Semantic Web is mostly reasoning about knowledge expressed in some ontology. However personalization may involve also other kinds of reasoning and knowledge representation, that conceptually lie at the logic and proof layers of the Semantic Web tower.

Moreover, the next Web generation promises to deliver Semantic Web Services, that can be retrieved and *combined* in a way that satisfies the user. It opens the way to many forms of *service-oriented personalization*. Web services provide an ideal infrastructure for enabling *interoperability* among personalization applications and for constructing Plug&Play-like environments, where the user can select and combine the kinds of services he or she prefers. Personalization can be obtained by taking different approaches, e.g. by developing services that offer personalization functionalities as well as by personalizing the way in which services are selected, and *composed* in order to meet specific user's requirements.

In the last years we carried on a research in the educational domain, by focussing on *semantic web* representations of learning resources and on *automated reasoning* techniques for enabling different and complementary personalization functionalities, e.g. curriculum sequencing [6, 7] and verification of the compliance of a curriculum against some course design goals [5]. Our current aim is to implement such results in an organic system, where different personalization services, that exploit semantic web reasoning, can be combined to support the user in the task of building a curriculum, based on *learning resources* that represent courses.

While in early times learning resources were simply considered as "contents", strictly tied to the platform used for accessing them, recently, greater and greater attention has been posed on the issue of *re-use* and of a *cross-platform* use of educational contents. The proposed solution is to adopt a *semantic annotation* of contents based on standard languages, e.g. RDF and LOM. Hereafter, we will consider a *learning resource* as formed by *educational contents* plus *semantic meta-data*, which supply information on the resources at a *knowledge level*, i.e. on the basis of concepts taken from an ontology that describes the educational domain. In particular we rely on the interpretation of learning resources as *actions* discussed in [6, 7]: the meta-data captures the *learning objectives* of the learning resource and its *pre-requisites*. By doing so,one can rely on a classical theory of actions and apply different reasoning methods -like *planning*- for building personalized curricula [6, 7]. The modeling of learning resources as actions also enables the use of model checking techniques for developing a validation service that detects if a user-given curriculum is compliant w.r.t an abstract model, given as a set of constraints. In the following we present our achievements in the implementation of a Planning service and a Validation service that can interoperate within the Personal Reader Framework [18].

Curriculum planning and validation offer a useful support in many practical contexts and can be fruitfully combined for helping students or teaching institutions. Often a student knows what competency he/she would like to acquire but has no knowledge of which courses will help him/her acquiring it. Moreover, taking courses at different Universities is becoming more and more common in Europe. As a consequence, building a curriculum might become a complicated task for students, who must deal with an enormous set of courses across the European countries, each described in different languages and on the basis of different keywords.

The need of personalizing the sequencing of learning resource, w.r.t. the student's interests and context, has often to be *combined* with the ability to check that the resulting curriculum *complies* against some abstract *curricula specification*, which encodes the *curricula-design goals* expressed by the teachers or by the institution offering the courses. Consider a student, who wants to build a valid curriculum with the support of our automatic system. The student can either use as a basis the suggestion returned by the system or he/she can design the curriculum by hand, based on own criteria. In both cases a personalized curriculum is obtained and can be given in input to the validation service for checking the compliance against a curricula model. Curricula models specify general rules for building learning paths and can be interpreted as constraints designed by the University for guaranteeing the achievement of certain learning goals. These constraints are to be expressed in terms of knowledge elements, and maybe also on features that characterize the resources.

Consider now a university which needs to certify that the specific curricula, that it offers for achieving a certain educational goal, and that are built upon the courses offered locally by the university itself, respect some European guidelines. In this case, we could, in fact, define the guidelines as a set of constraints at an abstract level, i.e. as relations among a set of competencies which should be offered in a way that meets some given scheme. At this point the verification could be performed automatically, by means of a proper reasoner. Finally, the automatic checking of compliance combined with curriculum planning could be used for implementing processes like cooperation among institutes in curricula design and integration, which are actually the focus of the so called *Bologna Process* [15], promoted by the EU.

While SCORM [2] and Learning Design [19, 20] represent the most important steps in the direction of managing and using e-learning based courses and workflows among a group of actors participating in learning activities, most of the available tools lack the machine-interpretable information about the learning resources, and as a result they are not yet open for reasoning-based personalization and automatic composition and verification. Given our requirements, it is a natural choice to settle our implementation in the Personal Reader (PR) framework. The PR relies on a service-oriented architecture enabling personalization, via the use of semantic *Personalization Services*. Each service offers a different personalization functionality, e.g. recommendations tailored to the needs of specific users, pointers to related (or interesting or more detailed/general) information, and so on. These semantic web services communicate solely based on RDF documents.

The paper is organized as follows. Section 2 describes our approach to the representation and reasoning about learning resources, curricula, and curricula models. The implementation of the two services and their integration into the PR Framework is discussed in section 3. We finish with conclusions and hints on future work in Section 4.

2 Curricula representation and reasoning

Let us begin with the introduction of our approach to the representation of learning resources, curricula, and curricula models. The basic idea is to describe all the different kinds of objects, that we need to tackle and that we will introduce hereafter, on the basis of a set of predefined *competencies*, i.e. terms identifying specific *knowledge elements*. We will use the two terms as synonyms. Competencies can be thought of, and implemented, as concepts in a shared ontology. In particular, for what concerns the application system described here, competencies were extracted by means of a semi-automatic process and stored as an RDF file (see Section 3.1 for details).

Given a predefined set of competencies, the initial knowledge of a student can be represented as a set of such concepts. This set changes, typically it grows, as the student studies and learns. In the same way, a user, who accesses a repository of learning resources, does it with the aim of finding materials that will allow him/her to acquire some knowledge of interest. Also this knowledge, that we identify by the term *learning goal*, can be represented as a set of knowledge elements. The learning goal is to be taken into account in a variety of tasks. For instance, the construction of a personalized curriculum is, actually, the construction of a curriculum which allows the achievement of a learning goal expressed by the user. In Section 3 we will describe a *curricula planning service* for accomplishing this task.

2.1 Learning resources and curricula

A *curriculum* is a sequence of *learning resources* that are homogeneous in their representation. Based on work in [6, 7], we rely on an *action theory*, and take the abstraction of resources as *simple actions*. More specifically, a learning resource is modelled as an action for acquiring some competencies (called *effects*). In order to understand the contents supplied by a learning resource, the user is sometimes required to own other competencies, that we call *preconditions*. Both preconditions and effects can be expressed by means of a *semantic annotation* of the learning resource [7]. In the following we will often refer to learning resources as "courses" due to the particular application domain that we have considered (university curricula).

As a simple example of "learning resource as action", let us, then, report the possible representation (in a classical STRIPS-like notation) of the course "databases for biotechnologies" (*db_for_biotech* for short):

```
ACTION: db_for_biothec(),
    PREREQ: relational_db, EFFECTS: scientific_db
```

The prequisites to this action is to have knowledge about *relational databases*. Its effect is to supply knowledge about *scientific databases*.

Given the above interpretation of learning resources, a *curriculum* can be interpreted as a *plan*, i.e. as a sequence of actions, whose execution causes transitions from a state to another, until some final state is reached. The *initial state* contains all the competences that we suppose available before the curriculum is taken, e.g. the knowledge that the student already has. This set can also be empty. The *final state* is sometimes required to contain specific knowledge elements, for instance, all those that compose the user's learning goal. Indeed, often curricula are designed so to allow the achievement of a well-defined *learning goal*.

A transition between two states is due to the application of the action corresponding to a learning resource. Of course, for an action to be applicable, its preconditions must hold in the state to which it should be applied. The application of the action consists in an *update* of the state. We assume that competences can only be added to states. Formally, we assume that the domain is monotonic. The intuition behind this assumption is that the act of using a new resource will never erase from the students' memory the concepts acquired insofar. Knowledge grows incrementally.

2.2 Curricula models

Curricula models consist in sets of constraints that specify desired properties of curricula. Curricula models are to be defined on the basis of knowledge elements as well as of learning resources (courses). In particular, we would like to restrict the set of possible sequences of resources corresponding to curricula. This will be done by imposing constraints on the *order* by which knowledge elements are added to the states (e.g. "a knowledge element α is to be acquired before a knowledge element β"), or by specifying some *educational objectives* to be achieved, in terms of knowledge that must be contained in the final state (e.g. "a knowledge element α must be acquired sooner or later"). Therefore, we represent a curricula model as a set of *temporal constraints*. Being defined on knowledge elements, a curricula model is *independent* from the specific resources that are taken into account, for this reason, it can be *reused* in different contexts and it is suitable to open and dynamic environments like the web.

The possibility of *verifying the compliance of curricula to models* is extremely important in many applicative contexts, as explained by examples in the introduction. In some cases these checks could be integrated into the curriculum construction process; nevertheless, it is important to be able to perform the verification independently from the construction process. Let us consider again our simple scenario concerning a university, which offers a set of curricula that are proved to satisfy the guidelines given by the EU for a certain year. After a few years, the EU guidelines change: our University has the need to check if the curricula that it offers, still satisfy the guidelines, without rebuilding them.

A natural choice for representing temporal constraints on action paths is linear-time temporal logic (LTL) [14]. This kind of logic allows to verify if a property of interest is true for all the possible executions of a model (in our case the specific curriculum). This is often done by means of model checking techniques [12].

The curricula as we represent them are, actually, Kripke structures. Briefly, a Kripke structure identifies a set of states with a transition relation that allows passing from a state to another. In our case, the states contain the knowledge items that are owned at a certain moment. Since the domain is monotonic (as explained above we can assume that knowledge only grows), states will always contain *all* the competencies acquired up to that moment. The transition relation is given by the actions that are contained in the curriculum that is being checked.

It is possible to use the LTL logic to verify if a given formula holds starting from a state or if it holds for a set of states.

For example, in order to specify in the curricula model constraints on *what* to achieve, we can use the formula $\Diamond\alpha$, where $\Diamond$ is the eventually operator. Intuitively, such a formula expresses the fact that a set of knowledge elements will be acquired sooner or later. Moreover, constraints concerning *how* to achieve the educational objectives, such as "a knowledge element β cannot be acquired before the knowledge element α is acquired", can, for instance, be expressed by the LTL temporal formula $\neg\beta\ U\ \alpha$, where U is the *weak until* operator. Given a set of knowledge elements to be acquired, such constraints specify a partial ordering of the same elements.

2.3 Planning and Validation

Given a semantic annotation with preconditions and effects of the courses, classical planning techniques are exploited for creating *personalized curricula*, in the spirit of the work in [6, 7]. Intuitively the idea is that, given a repository of learning resources, which have been semantically annotated as described, the user expresses a *learning goal* as a set of *knowledge elements* he/she would like to acquire, and possibly also a set of already owned competencies. Then, the system applies planning to build a sequence of learning resources that, read in sequence, will allow him/her to achieve the goal.

The particular planning methodology that we implemented (see Section 3.3 for details) is a simple *depth-first forward planning* (an early prototype was presented in [3]), where actions cannot be applied more than once. The algorithm is simple:

1. Starting from the initial state, the set of *applicable* actions (those whose preconditions are contained in the current state) is identified.
2. One of such actions is selected and its application is simulated leading to a new state.
3. The new state is obtained by adding to the previous one the competencies supplied as effects of the selected action.
4. The procedure is repeated until either the goal is reached or a state is reached, in which no action can be applied and the learning goal is not satisfied.
5. In the latter situation, backtracking is applied to look for another solution.

The procedure will eventually end because the set of possible actions is finite and each is applied at most once. If the goal is achieved, the sequence of actions that label the transitions leading from the initial to the final state is returned as the resulting *curriculum*. If desired, the backtracking mechanism allows to collect a set of alternative solutions to present to the user.

Besides the capability of automatically building personalized curricula, it is also interesting to perform a set of verification tasks on curricula and curricula models. The simplest form of verification consists in *checking the soundness* of curricula which are built by hand by users themselves, reflecting their own personal interests and needs. Of course, not all sequences which can be built starting from a set of learning resources are lawful. Learning dependencies, imposed by courses themselves in terms of preconditions and effects, must be respected. In other words, a course can appear at a certain point in a sequence only if it is *applicable* at that point, therefore, there are no *competency gaps*. These implicit "applicability constraints" capture precedences and dependencies that are innate to the nature of the taught concepts. In particular, it is important to verify that all the *competencies*, that are necessary to fully understand the contents, offered by a learning resource, are introduced or available before that learning resource is accessed. Usually, this verification, as stated in [13], is performed manually by the learning designer, with hardly any guidelines or support.

Given the interpretation of resources as actions, the verification of the *soundness of a curriculum*, w.r.t. the learning dependencies and the learning goal, can be interpreted as an *executability check* of the curriculum. Also in this case, the algorithm is simple:

1. Given an initial state, representing the knowledge available before the curriculum is attended, a simulation is executed, in which all the actions in the curriculum are (virtually) executed one after the other.
2. An action (representing a course) can be executed only if the current state contains all the concepts that are in the course precondition. Intuitively, it will be applied only if the student owns the notions that are required for understanding the topics of the course.
3. If, at a certain point, an action that should be applied is *not applicable* because some precondition does not hold, the verification fails and the reasons of such failure can be reported to the user.
4. Given that all the courses in the sequence can be applied, one after the other, the final state that is reached must be compared with the learning goal of the student: all the desired goal concepts must be achieved, so the corresponding knowledge elements must be contained in the final state.

This latter task actually corresponds to another basic form of verification, i.e. to check whether a (possibly hand-made) curriculum allows the *achievement of the desired learning goal*. These forms of basic verifications can be accomplished by the service described in Section 3.4.

Another interesting verification task consists in checking if a *personalized curriculum is valid w.r.t. a particular curricula model* or, following Brusilovski's terminology, checking if the curriculum is *compliant against the course design goals* [11]. Indeed, a personalized curriculum that is proved to be executable, cannot automatically be considered as being *valid* w.r.t. a particular *curricula model*. A curricula model, in fact, imposes further constraints on *what* to achieve and *how* achieving it. We will return to this kind of verification in Section 3.4.

3 Implementation in the Personal Reader Framework

The Personal Reader Framework has been developed with the aim of offering a uniform entry point for accessing the Semantic Web, and in particular Semantic Web Services. Indeed it offers an environment for designing, implementing and realizing Web content readers in a service-oriented approach, for a more detailed description, see [18] (`http://www.personal-reader.de/`).

In applications based on the Personal Reader Framework, a user can select and combine —plug together— which personalized support he or she wants to receive. The framework has already been used for developing Web Content Readers that present online material in an embedded context [10, 1, 17]. Besides a user-interface, as shown in figure 1, a Personal Reader application consists of three types of *services*. *Personalization services* (PService) provide personalization functionalities: they deliver personalized recommendations for content, as requested by the user and obtained or extracted from the Semantic Web. *Syndication Services* (SynService) allow for some interoperability with the other services in the framework, e.g. for the discovery of the applications interfaces by a portal. The *Connector* is a single central instance responsible for all the communication between user interface and personalization services. It selects services based on their semantic description and on the requirements by the SynService. The Connector protects –by means of a public-key-infrastructure (PKI)– the communication among the involved parties. It also supports the customization and invocation of services and interacts with a user modelling service, called the *UMService*, which maintains a central user model.

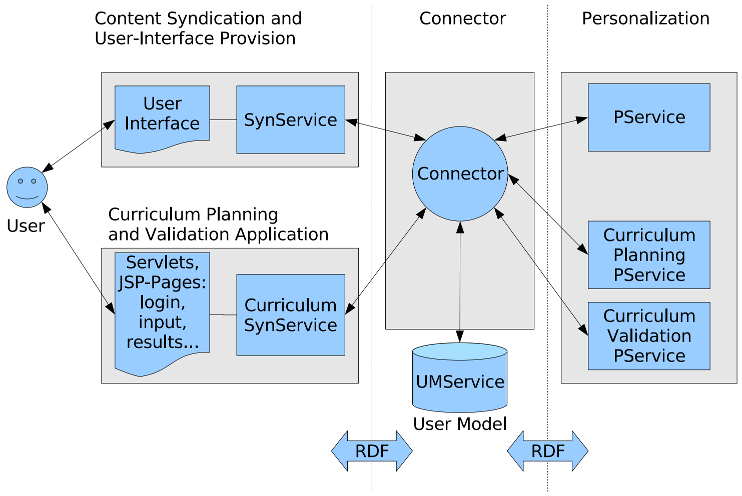


Fig. 1. Personal Reader Framework Overview

3.1 Metadata Description of Courses

In order to create the corpus of courses, we started with information collected from an existing database of courses. We used the Lixto [9] tool to extract the needed data from the web-pages provided by the HIS-LSF (http://www.his.de/) system of the University of Hannover. This approach was chosen based on our experience with Lixto in the *Personal Publication Reader* [10] project, where we used Lixto for creating the publications database by crawling the publication pages of the project partners. The effort to adapt our existing tool for the new data source was only small. From the extracted metadata we created an RDF document, containing course names, course catalog identifier, semester, number of credit points, effects and preconditions, and the type of course, e.g. laboratory, seminar or regular course with examinations in the end, as illustrated in Figure 2.

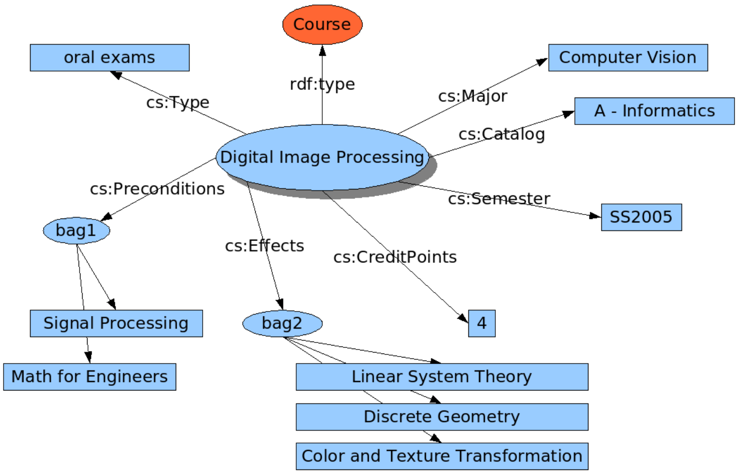
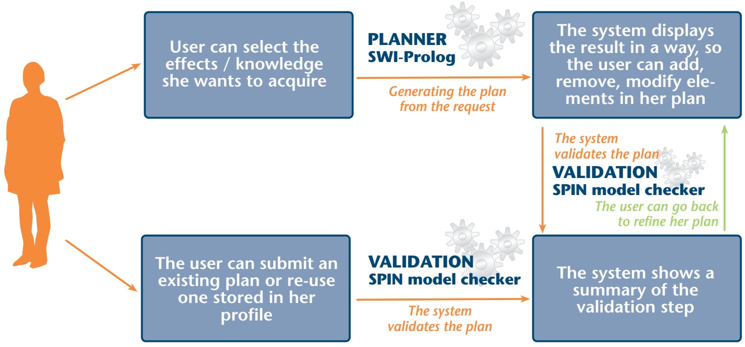


Fig. 2. An annotated course from the Hannover course database

The larger problem was that the quality of most of the information in the database turned out to be insufficient, mostly due to inconsistencies in the description of prerequisites and effects of the courses. Additionally the corpus was not annotated using a common set of terms, but authors and department secretaries used a slightly varying vocabulary for each of their course descriptions, instead of relying on a common classification system, like e.g. the ACM CCS for computer science.

As a consequence, we focussed only on a subset of the courses (computer science and engineering courses), and manually post-processed the data. Courses are annotated with prerequisites and effects, that can be seen as knowledge concepts or competences, i.e. ontology terms. After automatic extraction of effects and preconditions, the collected terms were translated into proper English language, synonyms were removed and annotations were corrected where necessary. The resulting corpus had a total of 65 courses left, with 390 effects and 146 preconditions.

3.2 The User Interface and Syndication Service



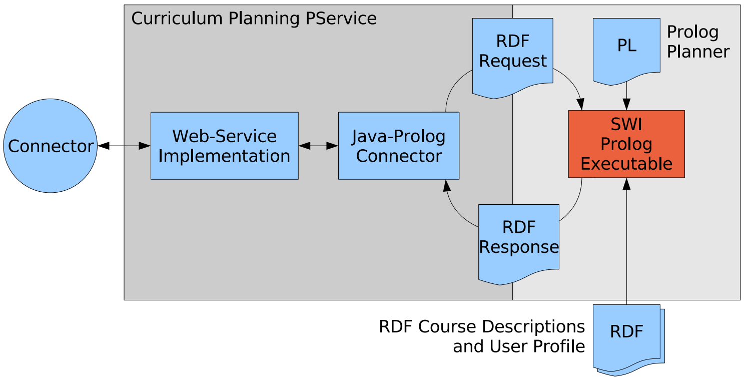
User can select the effects / knowledge she wants to acquire
PLANNER
SWI-Prolog
Generating the plan from the request
The system displays the result in a way, so the user can add, remove, modify elements in her plan
The system validates the plan
VALIDATION
SPIN model checker
The user can go back to refine her plan
The user can submit an existing plan or re-use one stored in her profile
VALIDATION
SPIN model checker
The system validates the plan
The system shows a summary of the validation step

Fig. 3. The Actions supported by the User Interface

In our implementation, the user interface (see figure 3) is responsible for identifying the user, presenting the user an interface to select the knowledge she wants to acquire, and to display the results of the planning and validation step, allowing further refinement of created plans. The creation of curriculum sequences and the validation are implemented as two independent Personalization Services, the “Curriculum Planning PService”, and the “Curriculum Validation PService”. Because of the plug-and-play nature of the infrastructure, the two PServices can be used by other applications (SynServices) as well (Fig. 3). Also possible is that PServices, which provide additional planning and validation capabilities can be used in our application. The current and upcoming future implementations of the Curriculum Planning and Validation Prototype are available at `http://semweb2.kbs.uni-hannover.de:8080/plannersvc`.

3.3 The Curriculum Planning PService

In order to integrate the Planning Service as a plug-and-play personalization service in the Personal Reader architecture we worked at embedding the Prolog reasoner into a web service. Figure 4 gives an overview over the components in the current implementation. The web service implements the Personalization



Curriculum Planning PService
Connector
Web-Service Implementation
Java-Prolog Connector
RDF Request
PL
Prolog Planner
SWI Prolog Executable
RDF Response
RDF Course Descriptions and User Profile
RDF

Fig. 4. Curriculum Planning Web Service

Service (*PService* [18]) interface, defined by the Personal Reader framework, which allows for the processing of RDF documents and for inquiring about the services capabilities. The *Java-to-Prolog Connector* runs the SWI-Prolog executable in a sub-process; essentially it passes the RDF document containing the request *as-is* to the Prolog system, and collects the results, already represented as RDF.

The curriculum planning task itself is accomplished by a reasoning engine, which has been implemented in SWI Prolog[3]. The interesting thing of using SWI Prolog is that it contains a semantic web library allowing to deal with RDF statements. Since all the inputs are sent to the reasoner in a *RDF request document*, it actually simplifies the process of interfacing the planner with the Personal Reader. In particular the request document contains: a) links to the RDF document containing the database of courses, annotated with metadata, b) a reference to the user's context c) the user's actual learning goal, i.e. a set of knowledge concepts that the user would like to acquire, and that are part of the *domain ontology* used for the semantic annotation of the actual courses. The reasoner can also deal with information about credits provided by the courses, when the user sets a credit constraint together with the learning goal.

Given a request, the reasoner runs the Prolog planning engine on the database of courses annotated with prerequisites and effects. The initial state is set by using information about the user's context, which is maintained by the User Modelling component of the PR. In fact such user's context includes information about what is considered as already learnt by the student (attended courses, learnt concepts) and such information is included in the request document. The Prolog planning engine has been implemented by using a classical depth-first search algorithm [22]. This algorithm is extremely simple to implement in declarative languages as Prolog.

[3] `http://www.swi-prolog.org/`

At the end of the process, a *RDF response document* is returned as an output. It contains a list of plans (sequences of courses) that fulfill the user's learning goals and profile. The maximum number of possible solutions can be set by the user in the request document. Notice that further information stored in the user profile is used at this stage for adapting the presentation of the solutions, here simple hints are used to *rank higher* those plans that include topics that the user has an expressed special interest in.

3.4 The Curriculum Validation PService

In order to verify if a curriculum is valid w.r.t. a curricula model, we adopt *model checking* techniques, by using SPIN. To check a curriculum with SPIN, this must be translated in the Promela language. *Competencies* are represented as *boolean variables*. In the beginning, only those variables that represent the initial knowledge of the student are true. *Courses* are implemented as actions that can modify the value of the variables. Since our application domain is monotonic, only those variables, whose value is false in the initial state, can be modified.

The Promela program consists of two processes: one is named *CurriculumVerification* and the other *UpdateState*. While the former contains a representation of the curriculum itself, and simulates its execution, the latter contains the code for updating the state (i.e. the set of competencies achieved so far) step by step along the simulation of the execution of the curriculum. The two processes communicate by means of two channels, *attend* and *feedback*. The notation *attend!courseName* represents the fact that the course with name *courseName* is to be attended. In this case the sender process is *CurriculumVerification* and the receiver is *UpdateState*. *UpdateState* will check the preconditions of the course in the current state and will send a feedback to *CurriculumVerification* after updating the state. On the other hand, the notation *feedback?feedbackMsg* represents the possibility for the process *Curriculum* of receiving a feedback of kind *feedbackMsg* from the process *UpdateState*.

Given these two processes, it is possible to perform a test, aimed at verifying the possible presence of competency gaps. This test is implemented as a *deadlock* verification: if the sequence is correct w.r.t. the action theory, no deadlock arises, otherwise a deadlock will be detected. The *curricula model* is to be supplied apart, as a set of temporal logic formulas, possibly obtained by an automatic translation process from a DCML representation. Notice that curricula can contain branching points. The branching points are encoded by either conditioned or non-deterministic *if*; each such *if* statement refers to a set of alternative courses (e.g. *languagesEnvironmentProg* and *programmingLanguages*). Depending on the course communicated by the channel *attend*, it updates the state. The process continues until the message *stop* is communicated. Then the learning goal is checked.

Let us see how to use the model checker to verify the *temporal constraints* that make a curricula model. Model checking is the algorithmic verification of the fact that a finite state system complies to its specification. In our case the specification is given by the curricula model and consists of a set of temporal constraints, while the finite state system is the curriculum to be verified.

SPIN allows to specify and verify every kind of LTL formulas and it also allows to deal with curricula that at some points contain alternatives. This makes the system suitable to more realistic application scenarios. In fact, for what concerns curricula written by hand, users often do not have a clear mind and, thus, it is difficult for them to write a single sequence. In the case of curricula built by an automatic system, there are planners that are able to produce sets of alternative solutions gathered in a tree structure.

The following are examples of constraints, expressed as LTL formulas, that could be part of a curricula model:

(1) $\neg jdbc \ \mathsf{U}\ (sql \wedge relational\_algebra)$,
(2) $\neg op\_systems \ \mathsf{U}\ basis\_of\_prog$,
(3) $\neg basis\_of\_oo \ \mathsf{U}\ basis\_of\_prog$,
(4) $\diamond basis\_of\_prog \supset \diamond basis\_of\_java\_prog$,
(5) $\diamond database$,
(6) $\diamond web\_services$.

The first constraint means that before learning *jdbc* the student must own Knowledge about *sql* and about *relational algebra*. The following two constraints are of the same kind but involve different competencies. Constraint (4) means that if the student acquires knowledge about "basis of programming", he/she will also have knowledge about "basis of java programming" but the two events are not temporally related. Constraints (5) and (6) mean that soon or later knowledge about databases and web services must be acquired.

4 Conclusion, Further and Related Works

In this work we have described the current state of the integration of semantic personalization web services for Curriculum Planning and Validation within the Personal Reader Framework. The goal of personalization is to create sequences of courses that fit the specific context and the learning goal of individual students. Despite some manual post-processing for fixing inconsistencies, we used real information from the Hannover University database of courses for extracting the meta-data. Currently the courses are annotated also by meta-data concerning the schedule and location of courses, like for instance room-numbers, addresses and teaching hours. As a further development, it would be interesting to let our Curriculum Planning Service to make use also of such metadata in order to find a solution that fits the desires and the needs of the user in a more complete way.

The Curriculum Planning Service has been integrated as a new plug-and-play personalization service in the Personal Reader framework. In the current implementation, the learning goal corresponds to a set of hard constraints; that is to say that the planner returns only plans that satisfy them *all*. A different choice would be to consider the constraints given by the goal as *soft* constraints, and allow the return of plans which do satisfy the goal only partially. This would be approapriate, for instance, in the case in which a student would like to acquire a range of competencies of interest but it is not possible to build, on top of a given repository of course descriptions, a curriculum for achieving them all. Nevertheless, it would be possible to build a curriculum for achieving *part* of them. In some circumstances, it would anyway be helpful for the student to receive this information as a feedback. Of course, in this case many questions arise, e.g. the issue of ranking the goals based on the actual interest of the requestor, so to know what can possibly be discarded and what is mandatory. From an implementation perspective, the spirit of the SOA infrustructure given to the Personal Reader is, indeed, meant to easily allow extensions by adding new Personalization Services. We can, therefore, think to develop and add a soft-goal planning service, to be used in these circumstances. The new planner would inherit the wrapping and interaction part from the current planning service but implement an algorithm like for instance [16].

The Curriculum Validation Service has been designed. An early prototype of the validation system based on the model checker SPIN has been developed [5] and is currently being embedded in the same framework. The choice of relying on SPIN, rather than developing a simpler and ad hoc checking system, is due to the need of rapidly developing a prototype. For this reason we have decided to rely on already exisiting and well-established technology. The engineering of the developed services should be tailored to the specific kinds of constraint that can be used to design the model. Analogous considerations can be done for the planning algorithm. The one that has been used is the simplest that can be thought of. Of course, there are many possible optimizations and extensions (e.g. the adoption of soft goals mentioned above) that could be done, and many algorithms are already available in the literature. Our choice has been motivated by the desire of quickly testing our ideas rather than developing a system thought for real use.

The Personal Reader Platform provides a natural framework for implementing a service-oriented approach to personalization in the Semantic Web, allowing to investigate how (semantic) web service technologies can provide a suitable infrastructure for building personalization applications, that consist of re-usable and interoperable personalization functionalities. The idea of taking a service oriented approach to personalization is quite new and was born within the personalization working group of the Network of Excellence REWERSE (Reasoning on the Web with Rules and Semantics, http://rewerse.net).

Writing curricula models directly in LTL is not an easy task for the user. For this reason, we have recently developed a graphical language, called DCML (Declarative Curricula Model Language) [8, 4], inspired by DecSerFlow, the Declarative Service Flow Language by van der Aalst and Pesic [23]. DCML allows to express the temporal relations between the times of acquisition of the concepts. The advantage of a graphical language is that *drawing*, rather than *writing*, constraints facilitates the user, who needs to represent curricula models, allowing a general overview of the relations which exist between concepts. At the same time, a rigorous and precise meaning is also given, due to the logic grounding of the language. Moreover, in [4] we represent curricula as UML activity diagrams and include the possibility of handling the concurrent attending of courses. Also in this case curricula can be translated in Promela programs so that it becomes possible to perform all the kinds of verification that we have described.

DCML, besides being a graphical language, has also a textual representation. We are currently working at an integration of this new more sophisticated solution into the Personal Reader Framework by implementing an automatic system for translating DCML textual representations into LTL, for translating curricula (activity diagrams) in Promela, and then run the checks.

Another recent proposal for automatizing the competency gap verification is done in [21] where an analysis of pre- and post-requisite annotations of the Learning Objects (LO), representing the learning resources, is proposed. In this approach, whenever an error will be detected by the validation phase, a correction engine will be activated. This engine will use a "Correction Model" to produce suggestions for correcting the wrong curriculum, by means of a reasoning-by-cases approach. The suggestions will, then, be presented to the course developer, who is in charge to decide which ones to adopt (if any). Once a curriculum will have been corrected, it will have to be validated again, because the corrections might introduce new errors. Melia and Pahl's proposal is inspired by the CocoA system [11], that allows to perform the analysis and the consistency check of static web-based courses. Competency gaps are checked by a prerequisite checker for *linear courses*, simulating the process of teaching with an overlay student model. Pre- and post-requisites are represented by knowledge elements.

Acknowledgement This research has partially been funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. `http://rewerse.net`).

References

1. F. Abel, I. Brunkhorst, N. Henze, D. Krause, K. Mushtaq, P. Nasirifar, and K. Tomaschweski. Personal reader agent: Personalized access to configurable web services. Technical report, Distributed Systems Institute, Semantic Web Group, University of Hannover, 2006.
2. Advanced Distributed Learning Network. SCORM: The sharable content object reference model, 2001. http://www.adlnet.org/Scorm/scorm.cfm.
3. M. Baldoni, C. Baroglio, I. Brunkhorst, N. Henze, E. Marengo, and V. Patti. A Personalization Service for Curriculum Planning. In E. Herder and D. Heckmann, editors, *Proc. of the 14th Workshop ABIS*, pages 17–20, Hildesheim, Germany, October 2006.
4. M. Baldoni, C. Baroglio, and E. Marengo. Curricula Modeling and Checking. In *Proc. of AI*IA 2007: Advances in Artificial Intelligence*, volume 4733 of *LNAI*, pages 471–482. Springer, 2007.
5. M. Baldoni, C. Baroglio, A. Martelli, V. Patti, and L. Torasso. Verifying the compliance of personalized curricula to curricula models in the semantic web. In *Proc.*

of the Semantic Web Personalization Workshop, pages 53–62, Budva, Montenegro, 2006.

6. M. Baldoni, C. Baroglio, and V. Patti. Web-based adaptive tutoring: An approach based on logic agents and reasoning about actions. *Artificial Intelligence Review*, 1(22):3–39, 2004.
7. M. Baldoni, C. Baroglio, V. Patti, and L. Torasso. Reasoning about learning object metadata for adapting SCORM courseware. In L. Aroyo and C. Tasso, editors, *Int. Workshop on Engineering the Adaptive Web, EAW'04*, pages 4–13, 2004.
8. M. Baldoni and E. Marengo. Curriculum Model Checking: Declarative Representation and Verification of Properties. In *Proc. of 2nd Eur. Conf. EC-TEL*, volume 4753 of *LNCS*, pages 432–437. Springer, 2007.
9. R. Baumgartner, S. Flesca, and G. Gottlob. Visual web information extraction with lixto. In Peter M. G. Apers, Paolo Atzeni, Stefano Ceri, Stefano Paraboschi, Kotagiri Ramamohanarao, and Richard T. Snodgrass, editors, *VLDB*, pages 119–128. Morgan Kaufmann, 2001.
10. R. Baumgartner, N. Henze, and M. Herzog. The personal publication reader: Illustrating web data extraction, personalization and reasoning for the semantic web. In *ESWC*, pages 515–530, 2005.
11. P. Brusilovsky and J. Vassileva. Course sequencing techniques for large-scale web-based education. *Int. J. Cont. Engineering Education and Lifelong learning*, 13(1/2):75–94, 2003.
12. O. E. M. Clarke and D. Peled. *Model checking.* MIT Press, Cambridge, MA, USA, 2001.
13. Juri L. De Coi, Eelco Herder, Arne Koesling, Christoph Lofi, Daniel Olmedilla, Odysseas Papapetrou, and Wolf Sibershi. A model for competence gap analysis. In *Proc. of WEBIST 2007*, 2007.
14. E. A. Emerson. Temporal and model logic. In *Handbook of Theoretical Computer Science*, volume B, pages 997–1072. Elsevier, 1990.
15. European Commission, Education and Training. The Bologna process. `http://ec.europa.eu/education/policies/educ/bologna/bologna_en.html`.
16. E. Giunchiglia and M. Maratea. SAT-based planning with minimal-♯actions plans and "soft" goals. In *Proc. of AI*IA 2007: Advances in Artificial Intelligence*, volume 4733 of *LNAI*. Springer, 2007.
17. N. Henze. Personal readers: Personalized learning object readers for the semantic web. In *12th International Conference on Artificial Intelligence in Education, AIED05*, Amsterdam, The Netherlands, 2005.
18. N. Henze and D. Krause. Personalized access to web services in the semantic web. In *The 3rd International Semantic Web User Interaction Workshop (SWUI, collocated with ISWC 2006*, November 2006.
19. IMSGlobal. Learning design specifications. Available at `http://www.imsglobal.org/learningdesign/`.
20. R. Koper and C. Tattersall. *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training.* Springer Verlag, 2005.
21. M. Melia and C. Pahl. Automatic Validation of Learning Object Compositions. In *Information Technology and Telecommunications Conference IT&T'2005: Doctoral Symposium*, Carlow, Ireland, 2006.
22. S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 1995.
23. W. M. P. van der Aalst and M. Pesic. DecSerFlow: Towards a Truly Declarative Service Flow Language. In Mario Bravetti and Gialuigi Zavattaro, editors, *Proc. of WS-FM*, LNCS, Vienna, September 2006. Springer.

Curricula Modeling and Checking

Matteo Baldoni, Cristina Baroglio, and Elisa Marengo

Dipartimento di Informatica — Università degli Studi di Torino
C.so Svizzera, 185 — I-10149 Torino (Italy)
{baldoni,baroglio}@di.unito.it, elisa.mrng@gmail.com

Abstract. In this work, we present a constrained-based representation for specifying the goals of "course design", that we call curricula model, and introduce a graphical language, grounded into Linear Time Logic, to design curricula models which include knowledge of proficiency levels. Based on this representation, we show how model checking techniques can be used to verify that the user's learning goal is supplied by a curriculum, that a curriculum is compliant to a curricula model, and that competence gaps are avoided.

1 Introduction and Motivations

As recently underlined by other authors, there is a strong relationship between the development of peer-to-peer, (web) service technologies and e-learning technologies [17]. The more learning resources are freely available through the Web, the more modern e-learning management systems (LMSs) should be able to take advantage from this richness: LMSs should offer the means for easily retrieving and assembling e-learning resources so to satisfy specific users' learning goals, similarly to how (web) services are retrieved and composed [12]. As in a composition of web services it is necessary to verify that, at every point, all the information necessary to the subsequent invocation will be available, in a learning domain, it is important to verify that all the *competencies*, i.e. the *knowledge*, necessary to fully understand a learning resource are introduced or available before that learning resource is accessed. The composition of learning resources, a curriculum, does not have to show any *competence gap*. Unfortunately, this verification, as stated in [10], is usually performed *manually* by the learning designer, with hardly any guidelines or support.

A recent proposal for automatizing the competence gap verification is done in [17] where an analysis of pre- and post-requisite annotations of the Learning Objects (LO), representing the learning resources, is proposed. A logic based validation engine can use these annotations in order to validate the curriculum/LO composition. Melia and Pahl's proposal is inspired by the CocoA system [8], that allows to perform the analysis and the consistency check of static web-based courses. Competence gaps are checked by a prerequisite checker for *linear courses*, simulating the process of teaching with an overlay student model. Pre- and post-requisites are represented as "concepts".

R. Basili and M.T. Pazienza (Eds.): AI*IA 2007, LNAI 4733, pp. 471–482, 2007.

Together with the verification of consistence gaps, there are other kinds of verification. Brusilovsky and Vassileva [8] sketch some of them. In our opinion, two are particularly important: (a) verifying that the curriculum allows to achieve the users' *learning goals*, i.e. that the user will acquire the desired knowledge, and (b) verifying that the curriculum is compliant against the *course design goals.* Manually or automatically supplied curricula, developed to reach a learning goal, should match the "design document", a *curricula model*, specified by the institution that offers the possibility of personalizing curricula. Curricula models specify general rules for designing sequences of learning resources (courses). We interpret them as *constraints*, that are expressed in terms of concepts and, in general, are not directly associated to learning resources, as instead is done for pre-requisites. They constrain the process of acquisition of concepts, independently from the resources.

More specifically, in this paper we present a constraint-based representation of curricula models. Constraints are expressed as formulas in a temporal logic (LTL, linear temporal logic [11]) represented by means of a simple graphical language that we call DCML (*Declarative Curricula Model Language*). This logic allows the verification of properties of interest for all the possible executions of a model, which in our case corresponds to the specific curriculum. Curricula are represented as *activity diagrams* [1]. We translate an activity diagram, that represents a curriculum, in a *Promela* program [16] and we check, by means of the well-known SPIN Model Checker [16], that it respect the model by verifying that the set of LTL formulas are satisfied by the Promela program. Moreover, we check that learning goals are achieved, and that the curriculum does not contain competence gaps. As in [10], we distinguish between *competency* and *competence*, where by the first term we denote a concept (or skill) while by the second we denote a competency plus the level of proficiency at which it is learnt or known or supplied. So far, we do not yet tackle with "contexts", as defined in the competence model proposed in [10], which will be part of future work.

This approach differs from previous work [5], where we presented an adaptive tutoring system, that exploits *reasoning about actions and changes* to plan and verify curricula. The approach was based on abstract representations, capturing the *structure* of a curriculum, and implemented by means of prolog-like logic clauses. Such representations were applied a procedure-driven form of planning, in order to build personalized curricula. In this context, we proposed also some forms of verification, of competence gaps, of learning goal achievement, and of whether a curriculum, given by a user, is compliant to the "course design" goals. The use of procedure clauses is, however, limiting because they, besides having a *prescriptive* nature, pose very strong constraints on the sequencing of learning resources. In particular, clauses represent what is "legal" and whatever sequence is not foreseen by the clauses is "illegal". However, in an open environment where resources are extremely various, they are added/removed dynamically, and their number is huge, this approach becomes unfeasible: the clauses would be too complex, it would be impossible to consider all the alternatives and the clauses should change along time.

For this reason we considered as appropriate to take another perspective and represent only those constraints which are strictly necessary, in a way that is inspired by the so called *social approach* proposed by Singh for multi-agent and service-oriented communication protocols [18,19]. In this approach only the *obligations* are represented. In our application context, obligations capture relations among the times at which different competencies are to be acquired. The advantage of this representation is that we do not have to represent all that is legal but only those *necessary conditions* that characterize a legal solution. To make an example, by means of constraints we can request that a certain knowledge is acquired before some other knowledge, without expressing what else is to be done in between. If we used the clause-based approach, instead, we should have described also what can legally be contained between the two times at which the two pieces of knowledge are acquired. Generally, the constraints-based approach is more flexible and more suitable to an open environment.

2 DCML: A Declarative Curricula Model Language

In this section we describe the *Declarative Curricula Model Language* (DCML, for short), a graphical language to represent the specification of a curricula model (the course design goals). The advantage of a graphical language is that drawing, rather than writing, constraints facilitates the user, who needs to represent curricula models, allowing a general overview of the relations which exist between concepts. DCML is inspired by DecSerFlow, the Declarative Service Flow Language to specify, enact, and monitor web service flows by van der Aalst and Pesic [21]. DCML, as well as DecSerFlow, is grounded in Linear Temporal Logic [11] and allows a curricula model to be described in an easy way maintaining at the same time a rigorous and precise meaning given by the logic representation. LTL includes temporal operators such as next-time ($\bigcirc\varphi$, the formula φ holds in the immediately following state of the run), eventually ($\Diamond\varphi$, φ is guaranteed to eventually become true), always ($\Box\varphi$, the formula φ remains invariably true throughout a run), until ($\alpha \mathsf{U} \beta$, the formula α remains true until β), see also [16, Chapter 6]. The set of LTL formulas obtained for a curricula model are, then, used to verify whether a curriculum will respect it [4]. As an example, Fig. 1 shows a curricula model expressed in DCML. Every box contains at least one competence. Boxes/competences are related by arrows, which represent (mainly) temporal constraints among the times at which they are to be acquired. Altogether the constraints describe a curricula model.

2.1 Competence, Competency, and Basic Constraints

The terms *competence* and *competency* are used, in the literature concerning professional curricula and e-learning, to denote the "effective performance within a domain at some level of proficiency" and "any form of knowledge, skill, attitude, ability or learning objective that can be described in a context of learning, education or training". In the following, we extend a previous proposal [4,7] so

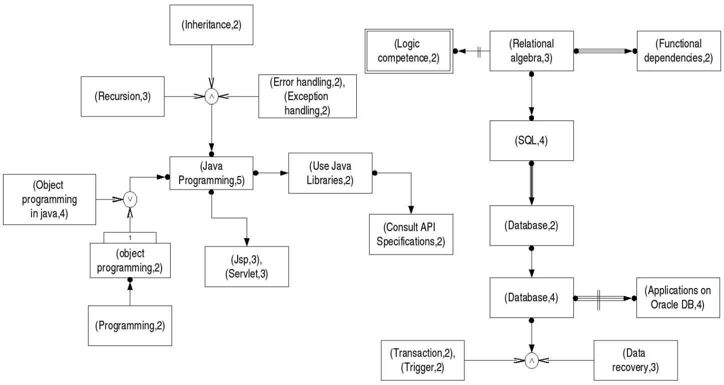


Fig. 1. An example of curricula model in DCML

as to include a representation of the *proficiency level* at which a competency is owned or supplied. To this aim, we associate to each competency a variable k, having the same name as the competency, which can be assigned natural numbers as values. The value of k denotes the proficiency level; zero means absence of knowledge. Therefore, k encodes a *competence*, Fig. 2(a). On competences, we can define three basic *constraints*.

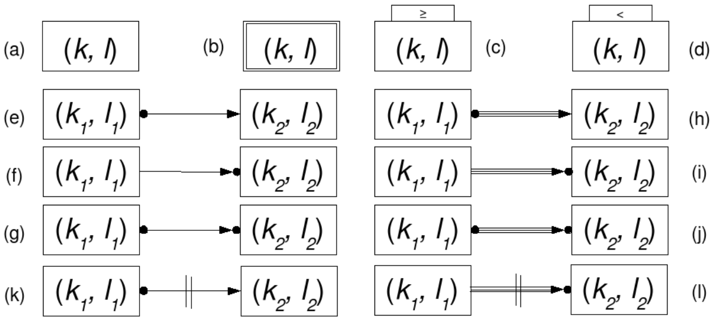


Fig. 2. Competences (a) and basic constraints (b), (c), and (d). Relations among competences: (a) implication, (b) before, (c) succession, (d) immediate implication, (e) immediate before, (f) immediate succession, (g) not implication, (h) not immediate before.

The "*level of competence*" constraint, Fig. 2(c), imposes that a certain competency k must be acquired at least at level l. It is represented by the LTL formula $\Diamond(k \geq l)$. Similarly, a course designer can impose that a competency must never appear in a curriculum with a proficiency level higher than l. This is possible by means of the "*always less than level*" constraint, shown in Fig. 2(d). The LTL formula $\Box(k < l)$ expresses this fact (it is the negation of the previous one). As a special case, when the level l is one ($\Box(k < 1)$), the competency k must never appear in a curriculum.

The third constraint, represented by a double box, see Fig. 2 (b), specifies that k must belong to the initial knowledge with, at least, level l. In other words, the simple logic formula $(k \geq l)$ must hold in the initial state.

To specify relations among concepts, other elements are needed. In particular, in DCML it is possible to represent *Disjunctive Normal Form* (DNF) formulas as *conjunctions* and *disjunctions* of concepts. For lack of space, we do not describe the notation here, however, an example can be seen in Fig. 1.

2.2 Positive and Negative Relations Among Competences

Besides the representation of competences and of constraints on competences, DCML allows to represent *relations* among competences. For simplicity, in the following presentations we will always relate simple competences, it is, however, of course possible to connect DNF formulas. We will denote by (k, l) the fact that competence k is required to have at least level l (i.e. $k \geq l$) and by $\neg(k, l)$ the fact that k is required to be less than l.

Arrows ending with a little-ball, Fig. 2(f), express the *before* temporal constraint between two competences, that amount to require that (k_1, l_1) holds *before* (k_2, l_2). This constraint can be used to express that to understand some topic, some proficiency of another is required as precondition. It is important to underline that if the antecedent never becomes true, also the consequent must be invariably false; this is expressed by the LTL formula $\neg(k_2, l_2) \ \mathsf{U} \ (k_1, l_1)$, i.e. $(k_2 < l_2) \ \mathsf{U} \ (k_1 \geq l_1)$. It is also possible to express that a competence must be acquired *immediate before* some other. This is represented by means of a triple line arrow that ends with a little-ball, see Fig. 2(i). The constraint (k_1, l_1) *immediate before* (k_2, l_2) imposes that (k_1, l_1) holds before (k_2, l_2) and the latter either is true in the next state w.r.t. the one in which (k_1, l_1) becomes true or k_2 *never* reaches the level l_2. The difference w.r.t the *before* constraint is that it imposes that the two competences are acquired *in sequence*. The corresponding LTL formula is "(k_1, l_1) *before* (k_2, l_2)" $\wedge\Box((k_1, l_1) \supset (\bigcirc(k_2, l_2) \vee \Box\neg(k_2, l_2)))$.

Both of the two previous relations represent temporal constraints between competences. The *implication* relation (Fig. 2(e)) specifies, instead, that if a competency k_1 holds at least at the level l_1, some other competency k_2 must be acquired sooner or later at least at the level l_2. The main characteristic of the implication, is that the acquisition of the consequent is imposed by the truth value of the antecedent, but, in case this one is true, it does not specify when the consequent must be achieved (it could be before, after or in the same state of the antecedent). This is expressed by the LTL formula $\Diamond(k_1, l_1) \supset \Diamond(k_2, l_2)$. The *immediate implication* (Fig. 2(h)), instead, specifies that the consequent must *hold* in the state right after the one in which the antecedent is acquired. Note that, this does not mean that it must be *acquired* in that state, but only that it cannot be acquired after. This is expressed by the LTL implication formula in conjunction with the constraint that whenever $k_1 \geq l_1$ holds, $k_2 \geq l_2$ holds in the next state: $\Diamond(k_1, l_1) \supset \Diamond(k_2, l_2) \wedge \Box((k_1, l_1) \supset \bigcirc(k_2, l_2))$.

The last two kinds of temporal constraint are *succession* (Fig. 2(g)) and *immediate succession* (Fig. 2(j)). The *succession* relation specifies that if (k_1, l_1) is acquired, afterwards (k_2, l_2) is also achieved; otherwise, the level of k_2 is not important. This is a difference w.r.t. the *before* constraint where, when the antecedent is never acquired, the consequent must be invariably false. Indeed, the *succession* specifies a condition of the kind *if* $k_1 \geq l_1$ *then* $k_2 \geq l_2$, while *before* represents a constraint without any conditional premise. Instead, the fact that the consequent must be acquired after the antecedent is what differentiates *implication* from *succession.* Succession constraint is expressed by the LTL formula $\Diamond(k_1, l_1) \supset (\Diamond(k_2, l_2) \wedge (\neg(k_2, l_2) \mathsf{U} (k_1, l_1)))$. In the same way, the *immediate succession* imposes that the consequent either is acquired in the same state as the antecedent or in the state immediately after (not before nor later). The immediate succession LTL formula is "(k_1, l_1) *succession* (k_2, l_2)" $\wedge\Box((k_1, l_1) \supset \bigcirc(k_2, l_2))$.

After the "positive relations" among competences, let us now introduce the graphical notations for "negative relations". The graphical representation is very intuitive: two vertical lines break the arrow that represents the constraint, see Fig. 2(k)-(l). (k_1, l_1) *not before* (k_2, l_2) specifies that k_1 cannot be acquired up to level l_1 before or in the same state when (k_2, l_2) is acquired. The corresponding LTL formula is $\neg(k_1, l_1) \mathsf{U} ((k_2, l_2) \wedge \neg(k_1, l_1))$. Notice that this is not obtained by simply negating the before relation but it is weaker; the negation of *before* would *impose the acquisition* of the concepts specified as consequents (in fact, the formula would contain a strong until instead of a weak until), the *not before* does not. The *not immediate before* is translated exactly in the same way as the *not before.* Indeed, it is a special case because our domain is monotonic, that is a competency acquired at a certain level cannot be forgotten.

(k_1, l_1) *not implies* (k_2, l_2) expresses that if (k_1, l_1) is acquired k_2 cannot be acquired at level l_2; as an LTL formula: $\Diamond(k_1, l_1) \supset \Box\neg(k_2, l_2)$. Again, we choose to use a weaker formula than the natural negation of the implication relation because the simple negation of formulas would impose the presence of certain concepts. (k_1, l_1) *not immediate implies* (k_2, l_2) imposes that when (k_1, l_1) holds in a state, $k_2 \geq l_2$ must be false in the immediately subsequent state. Afterwards, the proficiency level of k_2 does not matter. The corresponding LTL formula is $\Diamond(k_1, l_1) \supset (\Box\neg(k_2, l_2) \vee \Diamond((k_1, l_1) \wedge \bigcirc\neg(k_2, l_2)))$, that is weaker than the "classical negation" of the *immediate implies.*

The last relations are *not succession*, and *not immediate succession.* The first imposes that a certain competence cannot be acquired after another, (either it was acquired before, or it will never be acquired). As LTL formula, it is $\Diamond(k_1, l_1) \supset (\Box\neg(k_2, l_2) \vee$ "(k_1, l_1) *not before* (k_2, l_2)"). The second imposes that if a competence is acquired in a certain state, in the state that follows, another competence must be false, that is $\Diamond(k_1, l_1) \supset (\Box\neg(k_2, l_2) \vee$ "(k_1, l_1) *not before* (k_2, l_2)" $\vee\Diamond((k_1, l_1) \wedge \bigcirc\neg(k_2, l_2)))$.

3 Representing Curricula as Activity Diagrams

Let us now consider specific curricula. In the line of [5,3,4], we represent curricula as sequences of courses/resources, taking the abstraction of courses as simple actions. Any action can be executed given that a set of preconditions holds; by executing it, a set of post-conditions, the effects, will become true. In our case, we represent courses as actions for acquiring some concepts (*effects*) if the user owns some competences (*preconditions*). So, a curriculum is seen as a sequence of actions that causes *transitions* from the initial set of competences (possibly empty) of a user up to a final state that will contain all the competences owned by the user in the end. We assume that concepts can only be added to states and competence level can only grow by executing the actions of attending courses (or more in general reading a learning material). The intuition behind this assumption is that no new course erases from the students memory the concepts acquired in previous courses, thus knowledge grows incrementally. We represent curricula as *activity diagrams* [1], normally used for representing *business processes*. We decided to do so, because they allow to capture in a natural way the simple sequencing of courses as well as the possibility of attending courses in *parallel* or in possibly conditioned *alternatives*. An example is reported in Fig. 3. Besides the initial and the final nodes, the graphical elements used in an activity diagram are: *activity nodes* (rounded rectangle) that represent activities (attending courses) that occur; *flow/edge* (arrows) that represent activity flows; *fork* (black bar with one incoming edge and several outgoing edges) and *join nodes* (black bar with several incoming edges and one outgoing edge) to denote parallel activities; and *decision* (diamonds with one incoming edge and several outgoing edges) and *merge nodes* (diamonds with several incoming edges and one outgoing edge) to choose between alternative flows.

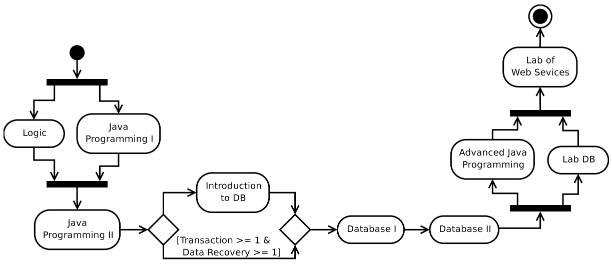


Fig. 3. Activity diagram representing a set of eight different curricula. Notice that *Logic* and *Java Programming I* can be attended in any order (even in parallel), as well as *Advanced Java Programming* and *Lab DB*, while *Introduction to DB* will be considered only if the guard *Transaction* and *Data Recovery* is false.

In the modeling of *learning processes*, we use activities to represent attending courses (or reading learning resources). For example, by fork and join nodes we represent the fact that two (or more) courses or sub-curricula are not related and, it is possible for the student to attend them in parallel. This is the case of *Java Programming I* and *Logic*, as well as *Advanced Java Programming* and *Lab. of DB* showed in Fig. 3. Till all parallel branches have not been attended successfully, the student cannot attend other courses, even if some of the parallel branches have been completed. Parallel branches can also be used when we want to express that the order among courses of different branches does not matter.

Decision and merge nodes can be used to represent alternative paths. The student will choose only one of these. Alternative paths can also be conditioned, in this case a *guard*, a boolean condition, is added at the beginning of the branch. Guards should be mutually exclusive. In our domain, the conditions are expressed in terms of concepts that must hold, otherwise a branch is not accessible. If no guards are present, the student can choose one (and only one) of the possible paths. In the example in Fig. 3, the guard consists of two copetences: *Transaction* and *Data Recovery.* If one of these does not hold the student has to attend the course *Introduction to DB*, otherwise does not.

4 Verifying Curricula by Means of SPIN Model Checker

In this section we discuss how to validate a curriculum. As explained, three kinds of verifications have to be performed: (1) verifying that a curriculum does not have competence gaps, (2) verifying that a curriculum supplies the user's learning goals, and (3) verifying that a curriculum satisfies the course design goals, i.e. the constraints imposed by the curricula model. To do this, we use *model checking techniques* [9].

By means of a *model checker*, it is possible to generate and analyze all the possible states of a program exhaustively to verify whether no execution path satisfies a certain property, usually expressed by a temporal logic, such as LTL. When a model checker refuses the negation of a property, it produces a *counterexample* that shows the violation. SPIN, by G. J. Holzmann [16], is the most representative tool of this kind. Our idea is to translate the activity diagram, that represents a set of curricula, in a Promela (the language used by SPIN) program, and, then to verify whether it satisfies the LTL formulas that represents the curricula model.

In the literature, we can find some proposals to translate UML activity diagrams into Promela programs, such as [13,14]. However, these proposals have a different purpose than ours and they cannot be used to perform the translation that we need to perform the verifications we list above. Generally, their aim is debugging UML designs, by helping UML designers to write sound diagrams. The translation proposed in the following, instead, aims to simulate, by a Promela program the acquisition of competencies by attending courses contained into the curricula represented by an activity diagram.

Given a curriculum as an activity diagram, we represent all the competences involved by its courses as *integer variables.* In the beginning, only those variables that represent the initial knowledge owned by the student are set to a value greater than zero. *Courses* are represented as actions that can modify the value of such variables. Since our application domain is monotonic, the value of a variable can only grow.

The Promela program consists of two main processes: one is called *CurriculumVerification* and the other *UpdateState*. While the former contains the actual translation of the activity diagram and simulates the acquisition of the competences for *all* curricula represented by the translated activity diagram, the latter contains the code for updating the state, i.e. the competences achieved so far, according to the definition in terms of preconditions and effects of each course. The processes *CurriculumVerification* and *UpdateState* communicate by means of the channel *attend*. The notation *attend!courseName* represents the fact that the course with name "courseName" is to be attended. On the other hand, the notation *attend?courseName* represents the possibility for a process of receiving a message. For example, the process *CurriculumVerification* for the activity diagram of Fig. 3 is defined as follows:

```
proctype CurriculumVerification()
{ activity_forkjoin_1();
  course_java_programming_II();
  activity_decisionmerge_1();
  course_database_I();
  course_database_II();
  activity_forkjoin_2();
  course_lab_of_web_services();
  attend!stop; }
```

If the simulation of all its possible executions end, then, there are no competence gaps; *attend!stop* communicates this fact and starts the verification of user's learning goal, that, if passed, ends the process. Each *course* is represented by its preconditions and its effects. For example, the course "Laboratory of Web Services" is as follows:

```
inline preconditions_course_lab_of_web_services()
{ assert(N_tier_architectures >= 4 && sql >= 2); }
inline effects_course_lab_of_web_services()
{ SetCompetenceState(jsp, 4); [...]
  SetCompetenceState(markup_language, 5); }
inline course_lab_of_web_services()
{ attend!lab_of_web_services; }
```

assert verifies the truth value of its condition, which in our case is the precondition to the course. If violated, SPIN interrupts its execution and reports about it. *SetCompetenceState* increases the level of the passed competence if its current level is lower than the second parameter. If all the curricula represented by the translated activity diagram have *no competence gaps*, no assertion violation will be detected. Otherwise, a counterexample will be returned that corresponds to an effective sequence of courses leading to the violation, giving a precise feedback to the student/teacher/course designer of the submitted set of curricula.

The *fork/join nodes* are simulated by activating as many parallel processes as their branches. Each process translates recursively the corresponding sub-activity diagram. Thus, SPIN simulates and verifies *all possible interleavings* of the courses (we can say that the curriculum is only one but it has different executions). The join nodes are translated by means of the synchronization message *done* that each activated process must send to the father process when it finishes its activity:

```
proctype activity_joinfork_11()
{ course_java_programming_I(); joinfork_11!done; }
proctype activity_joinfork_12()
{ course_logic(); joinfork_12!done; }
inline activity_joinfork_1()
{ run activity_joinfork_11(); run activity_joinfork_12();
  joinfork_11?done; joinfork_12?done; }
```

Finally, *decision and merge nodes* are encoded by either conditioned or non-deterministic *if*. Each such *if* statement refers to a set of alternative sub-activity diagrams (sub-curricula). Only one will be effectively attended but all of them will be verified:

```
inline activity_decisionmerge_11()
{ course_introduction_to_database(); }
inline activity_decisionmerge_12() { skip; }
inline activity_decisionmerge_1()
{ if
  :: (transaction >= 1 && data_recovery >= 1) ->
       activity_decisionmerge_12();
  :: else -> activity_decisionmerge_11();
  fi }
```

On the other hand, the process *UpdateState*, after setting the initial competences, checks if the preconditions of the courses communicated by *CurriculumVerification* hold in the current state. If a course is applicable it also updates the state. The test of the preconditions and the update of the state are performed as an atomic operation. In the end if everything is right it sends a feedback to *CurriculumVerification* (*feedback!done*):

```
proctype UpdateState() { SetInitialSituation();
  do [ ... ]
  :: attend?lab_of_web_services -> atomic {
        preconditions_course_lab_of_web_services();
        effects_course_lab_of_web_services(); }
  :: attend?stop -> LearningGoal(); break;
  od }
```

When *attend?stop* (see above) is received, the check of the user's learning goal is performed. This just corresponds to a test on the knowledge in the ending state:

```
inline LearningGoal()
{ assert(advanced_java_programming>=5 && N_tier_architectures
         >= 4 && relational_algebra>=2 && ER_language>=2); }
```

To check if the curriculum complies to a curricula model, we check if every possibly sequence of execution of the Promela program satisfies the LTL formulas, now transformed into *never claims* directly by SPIN. For example, the curriculum shown in Fig. 3 respects all the constraints imposed by the curricula model described in Fig. 1, taking into account the description of the courses supplied at the URL above. The assertion verification takes very few seconds on an old notebook; the automaton generated from the Promela program on that example has more than four-hundred states, indeed, it is very tractable. Also the verification of the temporal constraints is not hard if we check the constraints one at the time. The above example is available for download at the URL `http://www.di.unito.it/~baldoni/DCML/AIIA07`.

5 Conclusions

In this paper we have introduced a graphical language to describe curricula models as temporal constraints posed on the acquisition of competences (supplied by courses), therefore, taking into account both the concepts supplied/required and the proficiency level. We have also shown how model checking techniques can be used to verify that a curriculum complies to a curricula model, and also that a curriculum both allows the achievement of the user's learning goals and that it has no competence gaps. This use of model checking is inspired by [21], where LTL formulas are used to describe and verify the properties of a composition of Web Services. Another recent work, though in a different setting, that inspired this proposal is [20], where medical guidelines, represented by means of the GLARE graphical language, are translated in a Promela program, whose properties are verified by using SPIN. Similarly to [20], the use of SPIN, gives an *automa-based semantics* to a curriculum (the automaton generated by SPIN from the Promela program) and gives a declarative, formal, representation of curricula models (the set of temporal constraints) in terms of a LTL theory that enables other forms of reasoning. In fact, as for all logical theories, we can use an inference engine to derive other theorems or to discovery inconsistencies in the theory itself.

The presented proposal is an evolution of earlier works [6,3,5], where we applied semantic annotations to learning objects, with the aim of building compositions of new learning objects, based on the user's learning goals and exploiting planning techniques. That proposal was based on a different approach that relied on the experience of the authors in the use of techniques for reasoning about actions and changes which, however, suffers of the limitations discussed in the introduction. We are currently working on the automatic translation from a textual representation of DCML curricula models into the corresponding set of LTL formulas and from a textual representation of an activity diagram, that describes a curriculum (comprehensive of the description of all courses involved with their preconditions and effects), into the corresponding Promela program. We are also going to realize a graphical tool to define curricula models by means of DCML. We think to use the Eclipse framework, by IBM, to do this. In [2], we discuss the integration into the Personal Reader Framework [15] of a web service that implements an earlier version of the techniques explained here, which does not include proficiency levels.

Acknowledgements. The authors would like to thank Viviana Patti for the helpful discussions. This research has partially been funded by the 6th Framework Programme project REWERSE number 506779 and by the Italian MIUR PRIN 2005.

References

1. Unified Modeling Language: Superstructure, version 2.1.1. OMG (February 2007)
2. Baldoni, M., Baroglio, C., Brunkhorst, I., Marengo, E., Patti, V.: Curriculum Sequencing and Validation: Integration in a Service-Oriented Architecture. In: Proc. of EC-TEL'07. LNCS, Springer, Heidelberg (2007)
3. Baldoni, M., Baroglio, C., Henze, N.: Personalization for the Semantic Web. In: Eisinger, N., Małuszyński, J. (eds.) Reasoning Web. LNCS, vol. 3564, pp. 173–212. Springer, Heidelberg (2005)
4. Baldoni, M., Baroglio, C., Martelli, A., Patti, V., Torasso, L.: Verifying the compliance of personalized curricula to curricula models in the semantic web. In: Proc. of Int.l Workshop SWP'06, at ESWC'06, pp. 53–62 (2006)
5. Baldoni, M., Baroglio, C., Patti, V.: Web-based adaptive tutoring: an approach based on logic agents and reasoning about actions. Artificial Intelligence Review 22(1), 3–39 (2004)
6. Baldoni, M., Baroglio, C., Patti, V., Torasso, L.: Reasoning about learning object metadata for adapting SCORM courseware. In: Proc. of Int.l Workshop EAW'04, at AH 2004, Eindhoven, The Netherlands, August 2004, pp. 4–13 (2004)
7. Baldoni, M., Marengo, E.: Curricula model checking: declarative representation and verification of properties. In: Proc. of EC-TEL'07. LNCS, Springer, Heidelberg (2007)
8. Brusilovsky, P., Vassileva, J.: Course sequencing techniques for large-scale web-based education. Int. J. Cont. Engineering Education and Lifelong learning 13(1/2), 75–94 (2003)
9. Clarke, O.E.M., Peled, D.: Model checking. MIT Press, Cambridge (2001)
10. De Coi, J.L., Herder, E., Koesling, A., Lofi, C., Olmedilla, D., Papapetrou, O., Sibershi, W.: A model for competence gap analysis. In: Proc. of WEBIST 2007 (2007)
11. Emerson, E.A.: Temporal and model logic. In: Handbook of Theoretical Computer Science, vol. B, pp. 997–1072. Elsevier, Amsterdam (1990)
12. Farrell, R., Liburd, S.D., Thomas, J.C.: Dynamic assebly of learning objects. In: Proc. of WWW 2004, New York, USA (May 2004)
13. del Mar Gallardo, M., Merino, P., Pimentel, E.: Debugging UML Designs with Model Checking. Journal of Object Technology 1(2), 101–117 (2002)
14. Guelfi, N., Mammar, A.: A Formal Semantics of Timed Activity Diagrams and its PROMELA Translation. In: Proc. of APSEC'05, pp. 283–290 (2005)
15. Henze, N., Krause, D.: Personalized access to web services in the semantic web. In: The 3rd Int.l Workshop SWUI, at ISWC 2006 (2006)
16. Holzmann, G.J.: The SPIN Model Checker. Addison-Wesley, Reading (2003)
17. Melia, M., Pahl, C.: Automatic Validation of Learning Object Compositions. In: Proc. of *IT&T'2005: Doctoral Symposium*, Carlow, Ireland (2006)
18. Singh, M.P.: Agent communication languages: Rethinking the principles. IEEE Computer 31(12), 40–47 (1998)
19. Singh, M.P.: A social semantics for agent communication languages. In: Dignum, F.P.M., Greaves, M. (eds.) Issues in Agent Communication. LNCS, vol. 1916, pp. 31–45. Springer, Heidelberg (2000)
20. Terenziani, P., Giordano, L., Bottrighi, A., Montani, S., Donzella, L.: SPIN Model Checking for the Verification of Clinical Guidelines. In: Proc. of ECAI 2006 Workshop on AI techniques in healthcare, Riva del Garda (August 2006)
21. van der Aalst, W.M.P., Pesic, M.: DecSerFlow: Towards a Truly Declarative Service Flow Language. In: Bravetti, M., Núñez, M., Zavattaro, G. (eds.) WS-FM 2006. LNCS, vol. 4184, Springer, Heidelberg (2006)

Web Services System for distributed technology upgrade within an e-maintenance framework.

Eduardo Gilabert, Susana Ferreiro, Aitor Arnaiz

Fundación Tekniker, Av. Otaola 20,
20600 Eibar, Spain
{egilabert, sferreiro, aarnaiz}@tekniker.es

Abstract. Nowadays, industrial maintenance is one of the most important tasks in the industry because its cost is too high, usually due to poor maintenance decisions. Traditionally, corrective maintenance and preventive maintenance are performed, but both of them, the excessive and the lacking maintenance can be harmful. In the last years, CBM (Condition Based Maintenance) technology or predictive maintenance has appeared in order to establish whether the system will fail during some future period and then take actions to avoid consequences. This paper shows the e-maintenance platform nicknamed DYNAWeb which is part of DYNAMITE project. DYNAWeb develops a CBM system based on OSA-CBM standard over MIMOSA comprising broad of capabilities like sensing and data acquisition, signal processing, health assessment, prognosis... This platform ensures the integration of all the components (software and hardware) using different technologies (sensor technologies, wireless communication technology…) and providing them with agents and (Semantic) Web Services to allow the integration and the reuse among different applications.

Keywords: e-maintenance, CBM, OSA-CBM, MIMOSA, ontology, semantic web services, agent.

1 Introduction

Nowadays, maintenance is going through major changes. The industry is realising that the efficient use of industrial assets is a key issue in supporting our current standard of living. In this context, efficiency means producing good quality products without interrupting the production for unnecessary breakdowns.

A demand for improvements on system productivity, availability and safety is increasing, as well as product quality and customer satisfaction. Taking into account the trend for decrease in profit margins, the importance of implementing efficient maintenance strategies becomes unquestionable. In this picture the maintenance function plays a critical role in a company's ability to compete on the basis of cost, quality and delivery performance and maintenance is taken into account in production requirements: [1],[2].

To support this role, the maintenance concept must undergone through several major developments involving proactive considerations, which require changes in transforming traditional “fail and fix” maintenance practices to “predict and prevent” e-maintenance strategies. Such an approach takes into account the potential impact on service to customer, product quality and cost reduction [3].

E-Maintenance provides the opportunity for the 3rd generation maintenance and is a sub-concept of e-manufacturing and e-business for supporting next generation manufacturing practices (NGMS). The success of this NGMS is based on the inclusion and application of right support technologies to lower the set up costs as well as to facilitate the integration of such technologies with existing material and personal resources. As a consequence, one of the main aims of the new EU-funded Integrated Project DYNAMITE - Dynamic Decisions in Maintenance, is to bring together a series of technologies that can be integrated in a structured way, yet flexible enough to allow the selection of a particular subset of the technologies. The DYNAWeb concept is then a platform that designs an operational interaction between technologies in the framework of a distributed information scenario, where technologies of interest may vary from a company to another.

The organisation of this paper is as follows. First, it introduces OSA-CBM architecture, a non proprietary standard to standardize a condition based maintenance system. Next it is detailed a global view of DYNAWeb, a standardization of a new framework to be developed in DYNAMITE (Dynamic Decisions and Maintenance). DYNAWeb is based on Semantic Web Services for the e-maintenance, and its architecture concerning communication view is presented. Finally, it is included a description of main components interacting in DYNAWeb Platform: HMI, agents and Web Services, supported by ontologies.

2 OSA-CBM architecture

OSA-CBM[1] (Open System Architecture for Condition Based Maintenance) is designed as an open non-proprietary CBM communications framework to provide a functional platform flexible enough to suit a broad range of applications. Standardization of a networking protocol within the community of CBM developers and users will, ideally, drive CBM suppliers to produce interchangeable hardware and software components.

The goal of OSA-CBM is the development of architecture (and data exchange conventions) that enables interoperability of CBM components. Specifications are written in different languages, such as the Unified Modelling Language and correspond to a standard architecture for moving information in a condition-based maintenance system for software engineers. The basics of the architecture are described according to seven functional layers [4]:

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

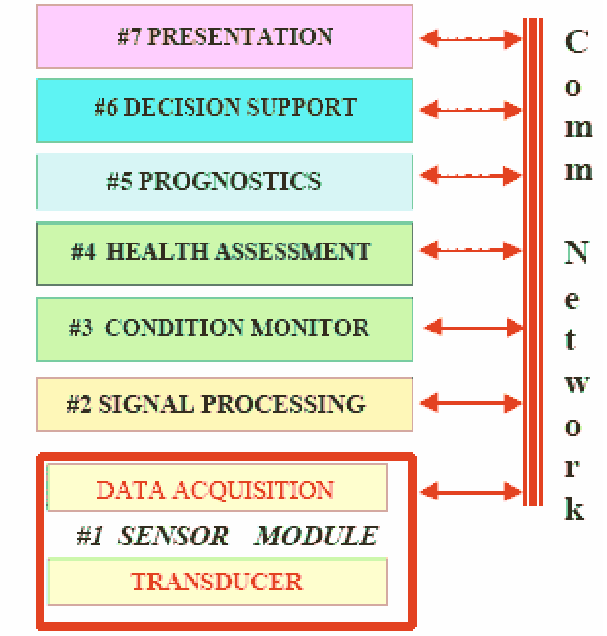


Fig. 1: OSA-CBM architecture defined in 7 layers

The implementation of a CBM system usually requires the integration of a variety of hardware and software components. Therefore, a complete CBM system may be composed of a number of functional layers or capabilities:

Layer 1 – Data Acquisition: it provides the CBM system with digitized sensor or transducer data.

Layer 2 – Data Manipulation: it performs signal transformations.

Layer 3 – Condition monitoring: it receives data from sensor modules, compares data with expected values or operations limits and generates alerts based on these limits.

Layer 4 – Health assessment: it receives data from condition monitoring and prescribes if the health in the monitoring component, sub-system or system is degraded. Besides, it is able to generate diagnostic (based upon trends in the health history, operational status and loading and maintenance history) and propose fault possibilities too.

Layer 5 – Prognosis: it plans the health state of equipment into the future or estimates the remaining useful life (RUL), taking into account estimates of future usage profiles.

Layer 6 – Decision support: it generates recommended actions (related with maintenance or how to run the asset until the current mission is completed without occurrence of breakdown) and alternatives. It takes into account operational history, current and future mission profile, high-level unit objectives and resource constraints.

Layer 7 – Presentation layer: Human System Interface (HSI) is required to provide a means of displaying vital information and provide user access to the system.

3 The DYNAWeb platform

Nowadays, a more research effort is required to face up to the challenges for modern e-maintenance. One focused research direction is offered by the ongoing EU-funded Integrated Project DYNAMITE - Dynamic Decisions in Maintenance. The partnership is composed by six research institutes in the UK, France, Spain, Sweden and Finland, two car manufacturers FIAT and VOLVO, the machine tool manufacturer GORATU, the automation and maintenance services provider Zenon, and seven SME's representing related business areas.

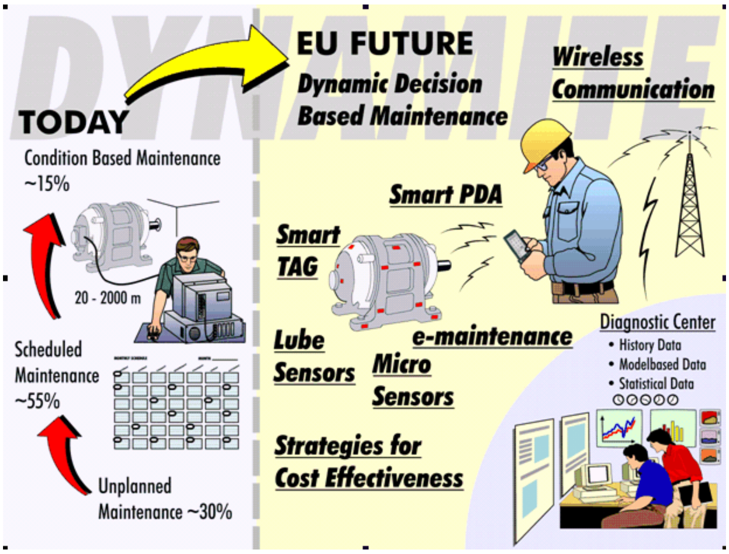


Fig. 2: The European DYAMITE concept for future IT-based maintenance. It aims at promoting a major change in the focus of condition based maintenance, essentially taking full advantage of recent advanced information technologies related to hardware, software and semantic information modelling.

The main technologies expected to facilitate this upgrade are wireless devices, such as smart tags and hand-held computing devices, micro-size MEMS sensors especially designed for maintenance purposes, and low-cost on-line lubrication analysis sensors. Inside DYNAMITE project, the DYNAWeb platform [8] refers to the ICT architecture concerning software web services and communication architecture that intends to provide support to the new maintenance concept, related mainly to the lower maintenance layer indicated in next figure.

The system architecture has been defined using the standard UML[2] (Unified modelling language). In particular, use case diagrams (UCDs) have been used to describe the system functionality, and the interactions between actors and the required functions.

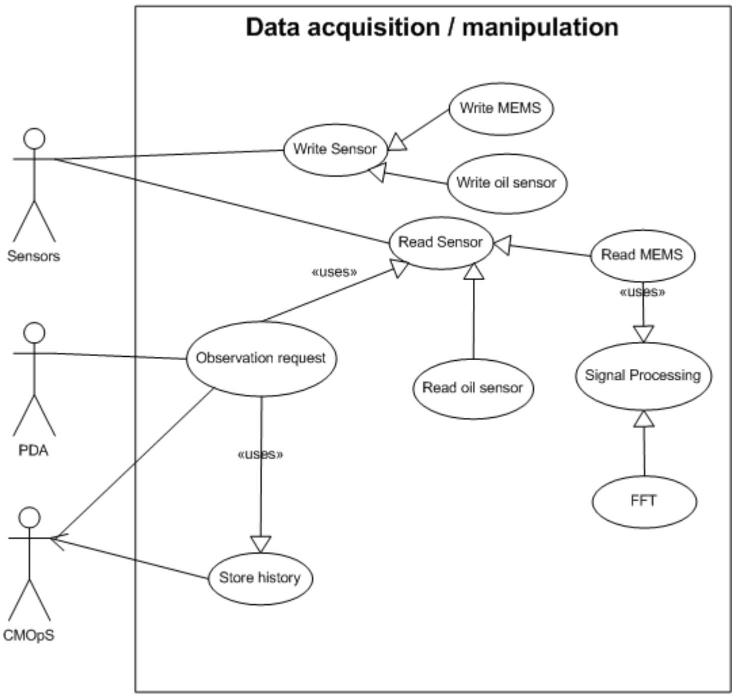


Fig. 3 : Use case diagram for Data acquisition & manipulation. It covers layers 1 and 2 of OSA-CBM standard: Sensor Module, which provides the system with digitized or transducer data, and Signal Processing, that receives signals and data from the sensor layer and the output includes digitally filtered sensor data, frequency spectra, virtual sensor signals and other CBM features.

In order to provide the most convenient analysis flow, information processing is understood as a distributed and collaborative system, where there are different levels of entities can undertake intelligence tasks. The UCD for data acquisition &

[2] http://www.uml.org

manipulation (see Fig. 3 : Use case diagram for Data acquisition & manipulation. It covers layers 1 and 2 of OSA-CBM standard) depicts the lower end of the system architecture. It corresponds to the machine and identifies sensors and smart tags associated to this level of interoperation. It is expected to perform some intelligence tasks. Sensors can provide certain degree of reasoning, taking into account the 'local' scope of this processing. It is also expected that sensors hold temporal information concerning current condition values, with little or no historical information attached. On the other hand, it is argued that PDAs can hold temporal information concerning operator activities and input values, and that the Conditional Maintenance Operational System (CMOpS) will hold historical records on selected condition information.

Smart PDAs will provide higher communication interfaces with sensors, intermediate processing capabilities and a smart end for human interface to remote web services centres that will compose a distributed web platform system at the higher end of the processing hierarchy [5]. Finally, wireless data transmission between sensor devices and information processing layers will be implemented.

Another UCD has been defined for Operation, Evaluation and execution of Tasks (see Fig. 4: Use case diagram for Operation, Evaluation and Execution. It covers 3 layers of OSA-CBM standard: Condition Monitoring, health assessment and Prognostics. The diagram sets the relationships among actors and expected functionality.). The specification of this UCD includes 3 layers or modules of OSA-CBM standard [4]:

- *Condition monitoring*: The condition monitor receives data from the sensor modules, the signal processing modules and other condition monitors. Its primary focus is to compare data with expected values. The condition monitor should also be able to generate alerts based on preset operational limits.
- *Health Assessment*: it receives data from different condition monitors or from heath assessment modules. The primary focus of the health assessment module is to prescribe if the health in the monitored component, sub-system or system has degraded. The health assessment layer should be able to generate diagnosis records and propose fault possibilities. The diagnosis should be based upon trends in the health history, operational status and loading and maintenance history.
- *Prognostics*: this module should have the possibility to take account data from all the prior layers. The primary focus of the prognostic module is to calculate the future health of an asset, with account taken to the future usage profiles. The module should report the failure health status of a specified time or the remaining useful life (RUL)

Another important operation is "Schedule work orders". CMMS schedules work orders based on component predictions. After that it has to distribute work orders to different operators (PDAs). PDA need read the smart tags to know component environment.

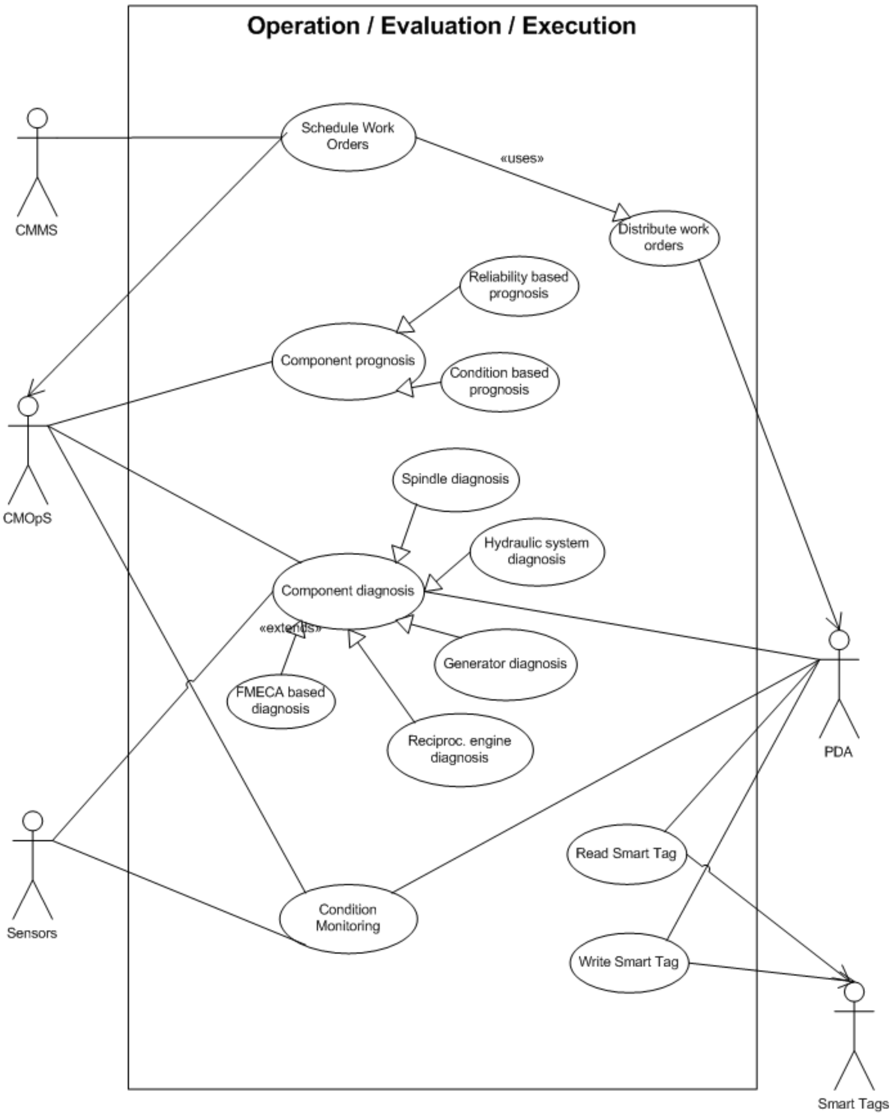


Fig. 4: Use case diagram for Operation, Evaluation and Execution. It covers 3 layers of OSA-CBM standard: Condition Monitoring, health assessment and Prognostics. The diagram sets the relationships among actors and expected functionality.

4 Ontology

The real challenge is to match the semantic web concept to the maintenance function. In this way, information used over internet must be specified in ontologies. The ontology represents the knowledge in internet [6], defining in a formal way the concepts of the different domains and relationships, with ability to perform reasoning over this knowledge. The definition of this ontologies has been performed starting from the standard CRIS (Common Relational Information Schema) defined by MIMOSA[3] (Machinery Information Management Open System Alliance).

CRIS represents a static view of the data produced by a CBM system, where every OSA-CBM layer has been associated to an ontology [7]. OSA-CBM was developed around MIMOSA CRIS (Common Relational Information Schema) that provides coverage of the information (data) that will be managed within a condition based maintenance system. It defines a relational database schema with about 200 of tables approximately for machinery maintenance information. In short, CRIS is the core of MIMOSA which aim is the development and publication of open conventions for information exchange between plant and machinery maintenance information system.

In this sense DYNAMITE investigate the way to improve these ontologies defined in XML language to other richer semantic ontology languages as RDF or OWL.

5 Semantic web services

Web services[4] are a well known technology and widely extended which is becoming to be used in industrial environments. They provide interoperability between independent software applications over internet by means of SOAP protocol which enables the communication.

With regard to the usage of these web services, there are defined three main elements which take part in the communication:

- HMI’s actor, that is, the CMOpS, PDA or other software allocated at plant level which interacts with operators or management staff and need the performance of a web service.
- Agent for communicating with DYNAWeb web services. Agent is able to get needed data from other sources, translating it into the ontology language. In this way, Agent acts as an interface between HMI and Web service.
- Web service, performing the requested service, supported by ontologies.

A basic web service for the different OSA-CBM levels is being developed, to demonstrate feasibility, including several implementations for different components and information coming from different sensor technologies. It is also designed a group of several agents an HMI ‘patches’ or ‘add-ins’ to show feasibility of communications and interfacing with legacy systems already in companies.

[3] http://www.mimosa.org

[4] http://en.wikipedia.org/wiki/Web_services

6 Distributed Agents

Associated to the actors, there are agents which perform the operations related to the communications with the semantic web services, supported by ontologies. According to this, the main functionalities of the agent in DYNAWeb are:

- Collect data from different distributed databases in the plant.
- Translate data to Ontology language (XML, RDF, OWL)
- Request an operation to a web service, sending the data needed.
- Translate results from ontology language to normal data.
- Store data results into the appropriate databases
- Communication with other agents

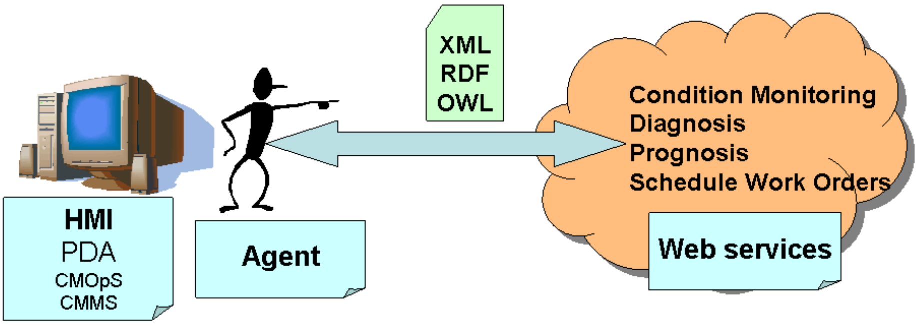


Fig. 5: Main components in DYNAWeb: HMIs, Agents and Web Services.

In this sense, the agent acts as a link between HMI of different software installed on the plant or management offices and the semantic web services supported by ontologies. In conclusion, the agent would be associated to a specific actor (PDA, CMOpS, CMMS) inside the plant, whereas the web services would be as generic as possible, covering a functionality for a wide range of different companies.

7 Conclusions

This paper has show the DYNAWeb platform refers to the ICT architecture concerning software web services and communication architecture that intends to provide support to the new maintenance concept, in DYNAMITE project. The aim is to be able to extend the use of a selected set of rising technologies within the area of maintenance activities. This effort is performed within a flexible architecture concept to provide flexible data and information management, where core concepts such as e-maintenance and OSA-CBM architecture are followed. Web services provide generic functionalities supported by ontologies. At client side, the agents act as links among actors and the semantic web services.

Acknowledgments. The authors gratefully acknowledge the support of the European Commission Sixth Framework programme for Research and Technological Development. This paper summarises work performed as part of FP6 Integrated Project IP017498 DYNAMITE "Dynamic Decisions in Maintenance".

References

1. Al-Najjar B., Alsyouf I.: Selecting the most efficient maintenance approach using fuzzy multiple criteria decision making. International Journal of Production Economics 84, 85-100 (2003)

2. Crespo Marquez A., Gupta J.N.D.: Contemporary Maintenance Management: Process, Framework and Supporting Pillars. Omega. 34/3, 325-338 (2006)

3. Lee J.: A framework for web-enabled e-Maintenance Systems. Proceedings of the second international symposium on environmentally conscious design and inverse manufacturing. EcoDesign 2001.

4. Bengtsson M.: Standardization issues in condition based maintenance. Department of Innovation, Design and Product Development, Mälardalen University, Sweden (2003).

5. Arnaiz A., Emmanouilidis C., Iung B. and Jantunen E.: Mobile Maintenance Management. Journal of International Technology and Information Management, Vol 15 (4) 11-22 (2006).

6. Lozano A.: Ontologies in the semantic web. Jornadas de Ingeniería web. (2001)

7. Lebold M., Reichard K., S.Byington C., Orsagh R.. OSA-CBM Architecture Development with Emphasis on XML Implementations. MARCON 2002.

8. Arnaiz A. Iung B., Jantunen E., Levat E., Gilabert E.: DYNAWeb. A web platform for flexible provision of e-maintenance services. Harrogate. 2007.

Semantic Web Services for advanced maintenance strategies

Authors: S. Ferreiro, E. Gilabert, A. Arnaiz

Abstract:

Nowadays industrial maintenance activity is mainly driven by traditional corrective strategies, combined with preventive time-based. Moreover, the cost of industries maintenance is too high and about 20% of this cost is wasted because of a poor maintenance.

During the last decades 'Condition Based Maintenance (CBM)' strategy has appeared with the purpose to eliminate breakdowns and adjust better the maintenance intervals. Its standardization 'Open System Architecture for Condition Based Maintenance (OSA-CBM)' divides a CBM system in seven different layers in order to facilitate the integration and interoperability between CBM components.

This paper presents the development of a (Semantic) Web Services platform which provides several intelligent processing capabilities to carry out some tasks. This platform is being developed according to OSA-CBM standard using an XML based implementation, from Condition Monitoring to Prognosis, and according to MIMOSA CRIS specification that provides broad coverage of the data types for machinery maintenance managed within CBM. Moreover, this software platform has a flexible communication infrastructure, where a generic wireless device is being also developed between sensors and some company areas where other communication options are not available.

Finally, based on this architecture, a small research about several existing possibilities is presented in order to add more semantics to these Web Services in the future.

This development nicknamed DYNAWeb Core, is still ongoing and it forms part of DYNAMITE project.

Keywords: E-maintenance, Semantic Web, Web Services, CBM, OSA-CBM, MIMOSA CRIS, UML, ontology, XML, RDF, OWL.

1. Introduction

Today maintenance is going through major changes. The industry and also the public are realising that the efficient use of industrial assets is a key issue in supporting our current standard of living. In this context, efficiency means producing good quality products without interrupting the production for unnecessary breakdowns.

There is a growing demand for improvements on system productivity, availability and safety, product quality and customer satisfaction In this setting the maintenance function becomes a critical task in a company's ability to compete on the basis of cost, quality and delivery performance (Al-Najjar and Alsyouf 2003). To support this task, the maintenance concept must undergone through several major developments involving proactive considerations, which require changes in transforming traditional "fail and fix" maintenance practices to "predict and prevent" e-maintenance strategies (Iung and Crespo 2006). Such an approach takes into account the potential impact on service to customer, product quality and cost reduction (Lee 2004). The key advantage is that maintenance is performed only when a certain level of equipment deterioration occurs rather than after a specified period of time or usage. In other words, there is a shift away from current mean-time-between failure (MTBF) practices to mean-time-between-degradation (MTBD) technologies.

E-Maintenance provides the opportunity for the 3rd generation maintenance and is a sub-concept of e-manufacturing and e-business for supporting next generation manufacturing practices. The success of this 3rd generation maintenance in a wide range of companies with non-critical machinery is based on the inclusion and application of right support technologies to lower the set up costs as well as to facilitate the integration of such technologies with existing material and personal resources. As a consequence, one of the main aims of the new EU-funded Integrated Project DYNAMITE - Dynamic Decisions in Maintenance, is to bring together a series of technologies that can be integrated in a structured way, yet flexible enough to allow the selection of a particular subset of the technologies. The design of such a flexible structure is supported on the assumption of numerous companies that can benefit from a subset of the technologies addressed in the project, providing customised plug and play to the desired upgrades with respect to each company's existing maintenance activities. A key point included in this article indicates software must also participate on this plug-and-play idea. In order to achieve this, a web services system is deployed, based on current standards that implement intelligent services, ready to be accessible at different company levels, and ready to be extended semantically.

The organisation of this paper is as follows. First, it introduces to a global view of DYNAMITE (Dynamic Decisions and Maintenance) project to develop a standardization of a new framework based on Semantic Web for the e-maintenance and its architecture concerning communication view. Next it is detailed OSA-CBM architecture, a non proprietary standard to standardize a condition based maintenance system and a description of the Web Services concerning the overall platform. After that, a description of Web Services implementation follows, with an example of a (Condition Monitoring) Web Service implementation. Finally, it is included an overview on the possibilities to extend the semantic of the DYNAWeb Web Services.

2. The dynamite approach

As indicated in the introduction, more research effort is required to face up to the challenges for modern e-maintenance. One focused research direction is offered by the ongoing EU-funded Integrated Project DYNAMITE - Dynamic Decisions in Maintenance. It includes six research institutes in the UK, France, Spain, Sweden and Finland, two car manufacturers FIAT and VOLVO, the machine tool manufacturer GORATU, the automation and maintenance services provider Zenon, and seven SME's representing related business areas.

The DYNAMITE vision (Figure 1) aims at promoting a major change in the focus of condition based maintenance, essentially taking full advantage of recent advanced information technologies related to hardware, software and semantic information modelling. Special attention is also given to the identification of cost-effectiveness related to the upgraded CBM strategies, as well as to the inclusion of innovative technologies within CBM. It is expected that the combination of the use of new technologies with a clear indication of cost-benefit trade-off will facilitate the upgrade into CBM, in many cases where non-critical machinery exists, and especially for the vast majority of SME companies that feel the distance between planned maintenance and condition based is too wide.

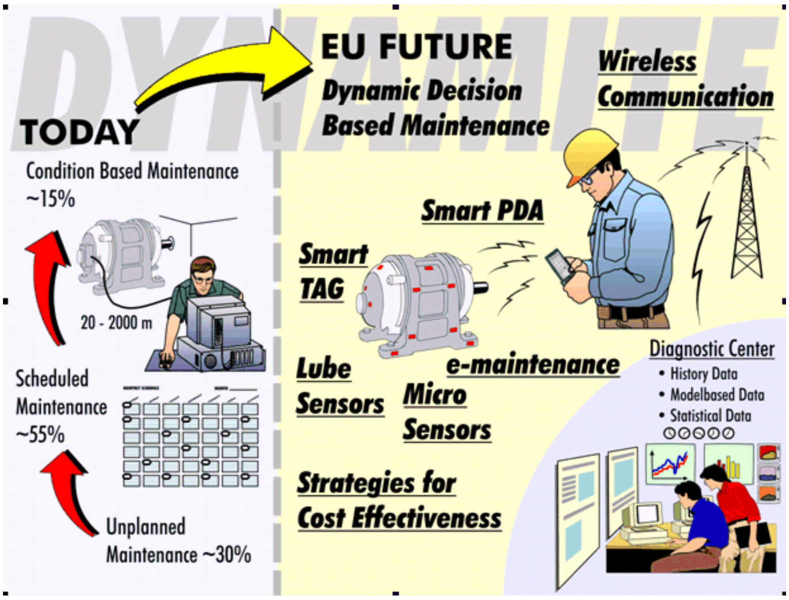


Figure 1: The European DYNAMITE concept for future IT-based maintenance.

The main technologies expected to facilitate this upgrade are wireless devices, such as smart tags and hand-held computing devices, micro-size MEMS sensors especially designed for maintenance purposes, and low-cost on-line lubrication analysis sensors. On the other hand, adequate information processing tools should take care of the continuous data flow and suggest appropriate actions to the operators.

In order to provide the most convenient analysis flow, information processing is understood as a distributed and collaborative system, where three different levels of entities can undertake intelligence tasks. At the lower end, sensors can provide certain degree of reasoning, taking into account the ‘local’ scope of this processing. At a medium level, smart PDAs (mobile agents) will provide higher communication interfaces with sensors, intermediate processing capabilities and a smart end for human interface to remote web services centres that will compose a distributed web platform system at the higher end of the processing hierarchy (Arnaiz et al, 2006). Finally, wireless data transmission between sensor devices and information processing layers will be implemented.

3. DYNAWEB platform

DYNAWeb refers to the ICT architecture concerning software web services and communication architecture that intends to provide support to the new maintenance concept, related mainly to the lower maintenance layer indicated in next figure.

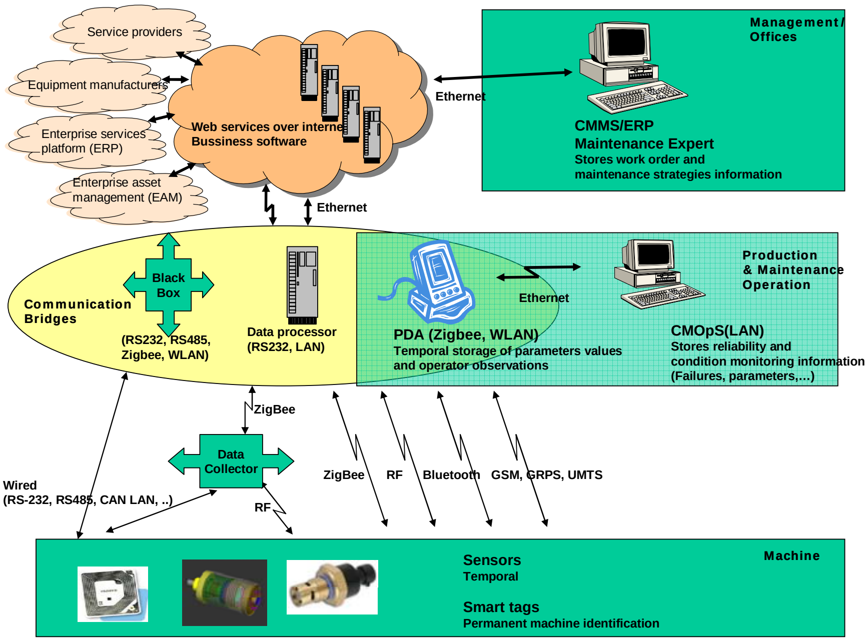

Service providers
Equipment manufacturers
Enterprise services platform (ERP)
Enterprise asset management (EAM)
Web services over internet
Bussiness software
Ethernet
Management/ Offices
CMMS/ERP
Maintenance Expert
Stores work order and maintenance strategies information
Ethernet
Communication Bridges
Black Box
(RS232, RS485, Zigbee, WLAN)
Data processor
(RS232, LAN)
Ethernet
Production & Maintenance Operation
PDA (Zigbee, WLAN)
Temporal storage of parameters values and operator observations
CMOpS(LAN)
Stores reliability and condition monitoring information (Failures, parameters,...)
ZigBee
Data Collector
Wired
(RS-232, RS485, CAN LAN, ..)
RF
ZigBee
RF
Bluetooth
GSM, GRPS, UMTS
Sensors
Temporal
Smart tags
Permanent machine identification
Machine

Figure 2: DYNAWeb platform

This figure specifies DYNAMITE architecture concerning communication view in the form of a graphical layout of the main actors (web services, PDAs, sensors) and expected communication channels between the actors.

The graphical layout indicates typical communication architecture with respect to a company where information and actors are distributed. This communication view identifies the existence of three layers (squared blue) of interaction of actors with respect to the company.

- First level corresponds to the machine and identifies sensors and smart tags as associated to this level of interoperation. It is also expected that sensors hold temporal information concerning current condition values, with little or no historical information attached.
- Second level corresponds to the production shop floor and identifies two main actors: The PDA and the Computer and maintenance Operational support. It is argued that these two

PDAs can hold temporal information concerning operator activities and input values, and that CMOpS will hold historical records on selected condition information.

- Third level corresponds to headquarters and management staff, where both tactical and strategic decisions are made. CMMS as well as Maintenance expert agents are located at this level, together with information concerning scheduled operations and maintenance strategies.

4. OSA_CBM architecture

The implementation of a CBM system usually requires the integration of a variety of hardware and software components. Therefore, a complete CBM system may be composed of a number of functional blocks or capabilities: sensing and data acquisition, data manipulation, condition monitoring, health assessment/diagnostics, prognostics, and decision reasoning. In addition, some form of a Human System Interface (HSI) is required to provide a means of displaying vital information and provide user access to the system. Thus, there is a broad range of system level requirements that include: communication and integration with legacy systems, protection of proprietary data and algorithms, need for upgradeability, and reduction of engineering design time and costs.

With these requirements in mind, OSA-CBM[1] (Open System Architecture for Condition Based Maintenance) is designed as an open non-proprietary CBM communications framework to provide a functional platform flexible enough to suit a broad range of applications. Standardization of a networking protocol within the community of CBM developers and users will, ideally, drive CBM suppliers to produce interchangeable hardware and software components.

The goal of OSA-CBM is the development of architecture (and data exchange conventions) that enables interoperability of CBM components. This primer is intended to cross the gap between computer scientists and program managers and systems integrators. The basis of the architecture is described according to seven functional layers (Bengtsson 2003):

Layer 1 – Data Acquisition: it provides the CBM system with digitized sensor or transducer data.

Layer 2 – Data Manipulation: it performs signal transformations.

Layer 3 – Condition monitoring: it receives data from sensor modules, compares data with expected values or operation limits and generates alerts based on these limits.

Layer 4 – Health assessment: it receives data from condition monitoring and prescribes if the health in the monitoring component, sub-system or system is degraded. Besides, it is able to generate diagnostic (based upon trends in the health history, operational status and loading and maintenance history) and propose fault possibilities too.

Layer 5 – Prognosis: it plans the health state of equipment into the future or estimates the remaining useful life (RUL), taking into account estimates of future usage profiles.

Layer 6 – Decision support: it generates recommended actions (related with maintenance or how to run the asset until the current mission is completed without occurrence of breakdown) and alternatives. It takes into account operational history, current and future mission profile, high-level unit objectives and resource constraints.

Layer 7 – Presentation layer

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

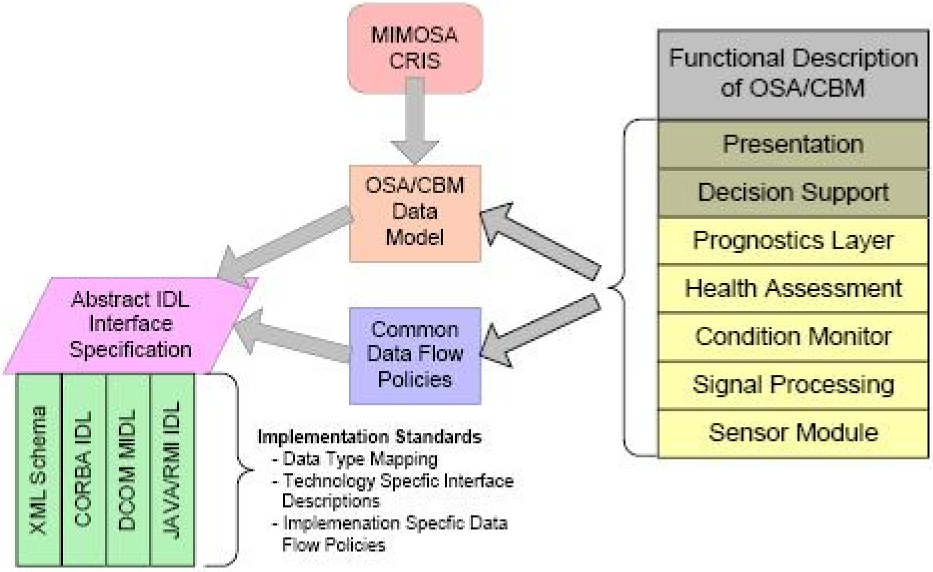


Figure 3: OSA/CBM Architecture (Lebold et. al.)

5. DYNAWeb (Semantic Web) Services

DYNAWeb (Semantic Web) Services are the core software modules which support the e-maintenance platform and provide a services oriented functionality for the Semantic Web. The description of these services goes far beyond the idea of an interface since we may find internal reasoning, state handling and many other elements.

The set of DYNAWeb Web Services has been divided into different groups of Web Services depending on associated OSA-CBM layer, so it has defined three types of different Web Services (condition monitoring, diagnosis and prognosis). All of these Web Services could be invoked by PDAs or CMOpS.

The condition monitoring will perform in sensors and PDAs but Web Services have been designed to provide this service in case not having information. So, PDA or CMOpS could request the Web Service and depending on data received by this, it could be performed four different functions related to condition monitoring:

- Absolute: this function retrieves alerts depending on fixed limits. The limits define subsets which belong to specific alert regions.
- Relative: this function issued alerts by means of the deviations between the measured current values and a reference values. Depending on the variation, it is issued an alert with a specific severity.
- Two Dimensions: the limits which define the alert regions depend on another parameter.
- Dynamic Limits: this function calculates the mean and standard deviation of historic values within it is possible to define the interval of confidence.

The Web Services for diagnosis assess the health of the component. There are five different services for diagnosis:

- Spindle Diagnosis: this function applies Bayesian Network to retrieve a list of failures with probabilities associated using vibration information.
- Hydraulic System: this function uses CLIPS expert system to retrieve diagnosis status by means of oil information.
- Cogenerator Diagnosis: this function uses means of vibration, oil and efficiency parameters within FMECA and retrieves diagnosis status.
- Reciprocating Engine Diagnosis: this function retrieves diagnosis status using CLIPS expert system within means of vibration and oil information.

- Reliability Diagnosis: this function is able to perform a diagnosis by means of FTA (Fault tree Analysis) information of the component.

Finally, prognosis Web Services implements functionality to compute the remaining useful life of the component submitted to a degradation mode. The design of these Web Services is focused on two specific cases of prognosis based on:

- Reliability: this function uses a degradation model based on reliability-
- Condition monitoring: this function applies a data modelling approach (trend analysis, pattern recognition, etc)

The following section supplies a simplified example of a Condition Monitoring Web Service.

6. MIMOSA bases

OSA-CBM was developed around MIMOSA CRIS (Common Relational Information Schema) that provides coverage of the information (data) that will be managed within a condition based maintenance system (see Figure 3).

CRIS represents a static view of the data produced by a CBM system, that is, it defines a relational database schema with about 200 of tables approximately for machinery maintenance information. Moreover, it provides broad coverage of the type of data need to be managed within CBM domain like (description of the configuration of the system being monitored, list of specific assets being tracked, characteristics of monitoring components, description of failure mode effects, alarm limits… etc).

In short, CRIS is the core of MIMOSA (Machinery Information Management Open System Alliance) which aim is the development and publication of open conventions for information exchange between plant and machinery maintenance information system.

DYNAWeb platform uses a set of MIMOSA tables to modelling OSA-CBM architecture like AlertRegion (that defines the conditions under which an alert is produced), NumAlert (the alert which is produced in response to an alert region), AlertSeverity (the alert severity defined in a scale from 0 to 10 with 0 being to information level and 10 being maximum severity), AlertType (the type of alert), Asset (the actual piece of machinery)… etc.

For moving this information in a condition-based maintenance system for software engineers, the specifications for each layer are written in Unified Modelling Language as shown in the next figure that designs the DYNAWeb Condition Monitoring layer.

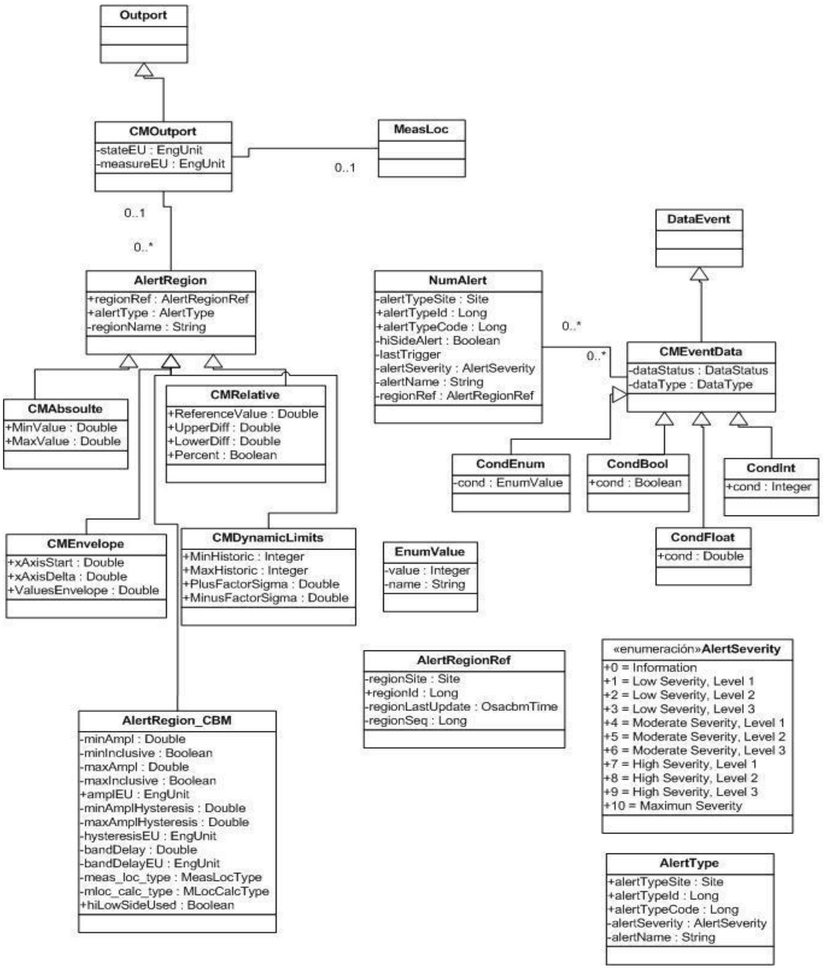


Figure 4: UML Condition monitoring layer

The following section supplies a simplified example of a Web Service.

7. An example of interaction (Condition Monitoring)

All the data necessary to carry out Condition Monitoring Web Service is represented in the UML model for condition monitoring shown in the Figure 4. For an easier implementation this model is divided in two different models, the first one represents the configuration and the input data for the Web Service and the second one the output data based on a group of alerts generated.

As mentioned earlier, UML data model which provide abstract specifications is defined for each OSA-CBM layer, but it is necessary to convert these specifications to other middleware language for a software implementation.

DYNAWeb Web Services provide a development of OSA-CBM system based on XML technology using SOAP as communication protocol.

These XML schemas describe the structure of the information and they are used to test the validity and to process the data, at the Web Service when it receives a request and at the client when it receives the response from the Web Service. This is an important aspect because the Web Services are receiving and sending information to and from many sources and they should not process invalid data.

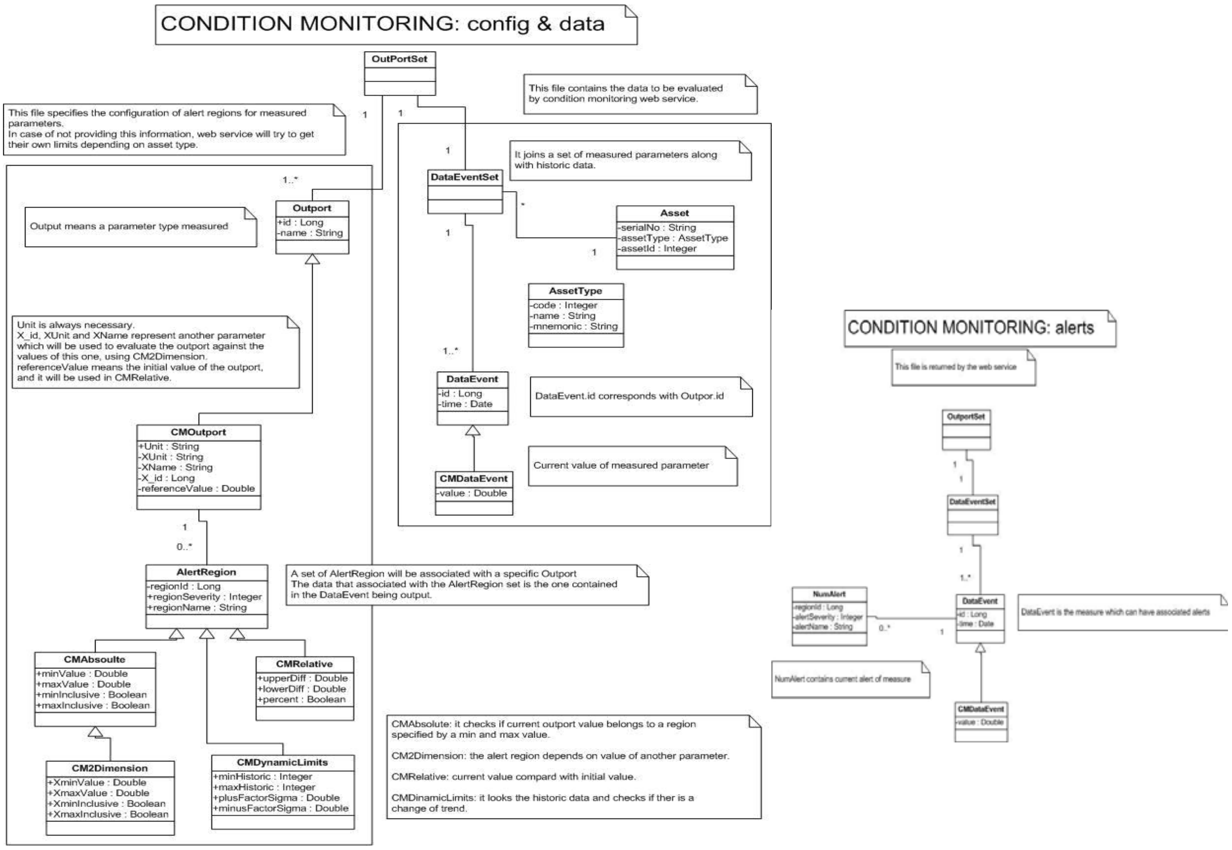
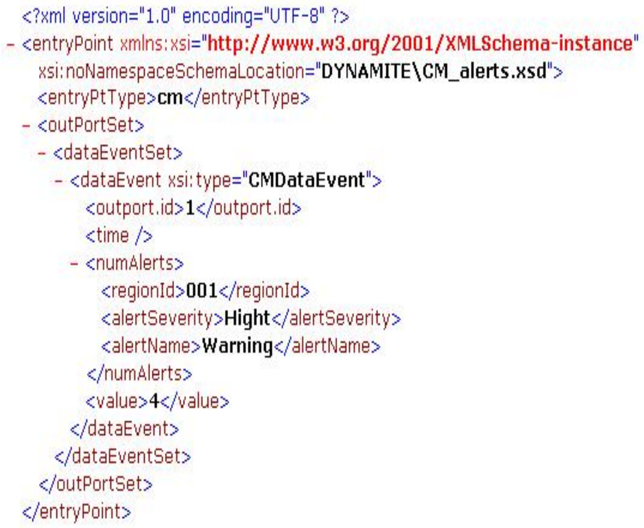

CONDITION MONITORING: config & data
OutPortSet
This file contains the data to be evaluated by condition monitoring web service.
This file specifies the configuration of alert regions for measured parameters.
In case of not providing this information, web service will try to get their own limits depending on asset type.
1
1
1
It joins a set of measured parameters along with historic data.
1..*
DataEventSet
Outport
+id : Long
-name : String
Output means a parameter type measured
*
Asset
-serialNo : String
-assetType : AssetType
-assetId : Integer
1
1
AssetType
-code : Integer
-name : String
-mnemonic : String
Unit is always necessary.
X_id, XUnit and XName represent another parameter which will be used to evaluate the outport against the values of this one, using CM2Dimension.
referenceValue means the initial value of the outport, and it will be used in CMRelative.
1..*
DataEvent
-id : Long
-time : Date
DataEvent.id corresponds with Outpor.id
CMOutport
+Unit : String
-XUnit : String
-XName : String
-X_id : Long
-referenceValue : Double
Current value of measured parameter
CMDataEvent
-value : Double
1
0..*
AlertRegion
-regionId : Long
+regionSeverity : Integer
+regionName : String
A set of AlertRegion will be associated with a specific Outport
The data that associated with the AlertRegion set is the one contained in the DataEvent being output.
CMAbsoulte
+minValue : Double
+maxValue : Double
+minInclusive : Boolean
+maxInclusive : Boolean
CMRelative
+upperDiff : Double
+lowerDiff : Double
+percent : Boolean
CM2Dimension
+XminValue : Double
+XmaxValue : Double
+XminInclusive : Boolean
+XmaxInclusive : Boolean
CMDynamicLimits
+minHistoric : Integer
+maxHistoric : Integer
+plusFactorSigma : Double
+minusFactorSigma : Double
CMAbsolute: it checks if current outport value belongs to a region specified by a min and max value.
CM2Dimension: the alert region depends on value of another parameter.
CMRelative: current value compard with initial value.
CMDinamicLimits: it looks the historic data and checks if ther is a change of trend.
CONDITION MONITORING: alerts
This file is returned by the web service
OutportSet
1
1
DataEventSet
1
1..*
NumAlert
-regionId : Long
-alertSeverity : Integer
-alertName : String
0..*
1
DataEvent
-id : Long
-time : Date
DataEvent is the measure which can have associated alerts
NumAlert contains current alert of measure
CMDataEvent
-value : Double

Figure 5: DYNAWeb condition monitoring layer

After PDA or CMOpS invokes the Web Service sending all information available with current measure, historic data, derived data, limits values, component information, etc, the Web Service receives data, validates it with the XML schema. Depending on information it calls the appropriate function and gives back alerts in case of being necessary (see next figure). If it does not provide any alert, it can also request information concerning the specifics of the component (such as component type) and then look up for information concerning these specifics.



```
 <?xml version="1.0" encoding="UTF-8" ?>
- <entryPoint xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="DYNAMITE\CM_alerts.xsd">
    <entryPtType>cm</entryPtType>
  - <outPortSet>
    - <dataEventSet>
      - <dataEvent xsi:type="CMDataEvent">
          <outport.id>1</outport.id>
          <time />
        - <numAlerts>
            <regionId>001</regionId>
            <alertSeverity>Hight</alertSeverity>
            <alertName>Warning</alertName>
          </numAlerts>
          <value>4</value>
        </dataEvent>
      </dataEventSet>
    </outPortSet>
  </entryPoint>
```

Figure 6: CM_alerts.xml

8. Conclusions and further research

The development of intelligent web services within an e-maintenance architecture has been outlined, and a small example concerning the web service for condition monitoring task automation has been shown. This is the first part of a series of intelligent web services to be developed next 12 months that will complete a skeleton of intelligent add-ins for extended maintenance operations

Even tough current services are based on standards (especially OSA-CBM and MIMOSA), final objective is attempting to create a subset of the XML ontology in richer semantic languages (RDF and OWL).

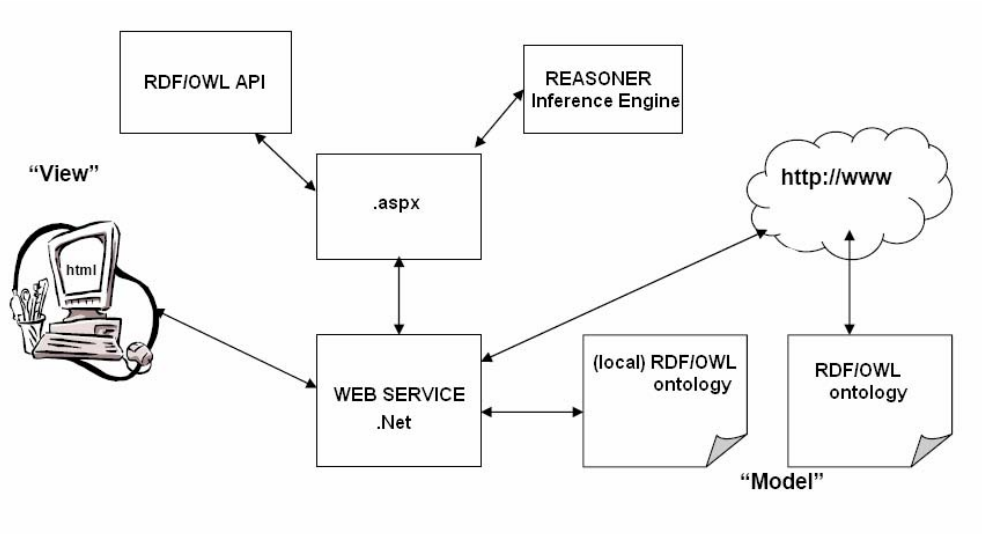


Figure 7: Extended Architecture view

Unfortunately, .NET framework (the compulsory framework within DYNAMITE project) only supports RDF management with an extended set of compatible libraries, and nothing concerning OWL up to now, so a semantic knowledge inference could only be integrated as part of DYNAWeb system using this language. Therefore, the final purpose is to introduce some functionality concerning semantic web services (diagnosis and/or prognosis) using RDF.

	.Net	**Java**
RDF API	SemWeb, Drive, VicSoft.Rdf parser	Jena
OWL API	--	Jena
REASONER	--	Racer, Fact, Fact ++, Pellet,

Table 1. Available options for semantic development.

References

Arnaiz A., Emmanouilidis C., Iung B. and Jantunen E. (2006). Mobile Maintenance Management. *Journal of International Technology and Information Management*, Vol 15 (4) 11-22

Al-Najjar B. and Alsyouf I. (2003). Selecting the most efficient maintenance approach using fuzzy multiple criteria decision making. International Journal of Production Economics 84, 85-100

Bengtsson M. (2003). “Standardization issues in condition based maintenance”, Department of Innovation, Design and Product Development, Mälardalen University, Sweden.

Crespo Marquez A. and Gupta J.N.D. (2006). Contemporary Maintenance Management: Process, Framework and Supporting Pillars. Omega. 34/3, 325-338

Egea-Lopez E., Martinez-Sala A., Vales-Alonso J., Garcia-Haro J., Malgosa-Sanahuja J. (2005) Wireless communications deployment in industry: a review of issues, options and technologies, Computers in Industry, Volume 56, Issue 1, January 2005, pp 29-53.

Harnett B.M., Doarn C.R., Zhao X., Merrell R.C. (2004) Redundant wireless communication technologies for real-time surveillance, Telematics and Informatics, Volume 21, Issue 4, November 2004, pp 375-386.

Iung B., Crespo Marquez A. (2006), Special issue on emaintenance, Computers in Industry, 57(6), pp. 473-606

Komonen K. (2005). Käynnissä- ja kunnossapidon kehitys 2000-luvulla Suomessa [Development of availability and maintenance in Finland in the present decade]. Kunnossapito 2, 52-55. (in Finnish)

Lee J., J. Ni, D. Djurdjanovic, H. Qiu and H. Liao (2006), Intelligent prognostics tools and e-maintenance, Computers in Industry, Special issue on e-maintenance, 57(6), pp 476489

Lee J. (2001). A framework for web-enabled e-Maintenance Systems. Proceedings of the second international symposium on environmentally conscious design and inverse manufacturing, EcoDesign’01

Wang, H. (2002). A survey of maintenance policies of deteriorating systems. European Journal of Operational Research, 139, p. 469-489.

Takata S., F. Kimura, F.J.A.M. van Houten, E. Westkämper, M. Shpitalni, D. Ceglarek, J. Lee (2004), Maintenance: Changing Role in Life Cycle Management, Annals of the CIRP, 53/2, pp 643 – 656

Van Houten, F.J.A.M., Tomiyama, T. and Salomons, O.W. (1998). Product modelling for model-based maintenance, Annals of the CIRP, 47/1, p. 123–129.

Adolfo Lozano. (2001). Ontologies in the semantic web. Jornadas de Ingeniería web.

Mitchell Lebold, Karl Reichard, Carl S.Byington, Rolf Orsagh. OSA-CBM Architecture Development with Emphasis on XML Implementations.

Mitchell Lebold, Michael Thurston. Open Standards for Condition-Based Maintenance and Prognostic Systems.