DESIGN OF THE NARRATOR SYSTEM: PROCESSING, STORING AND RETRIEVING MEDICAL NARRATIVE DATA

Leo Wolf
Regina Overberg
Pieter Toussaint
Leiden University Medical Centre, Department of Clinical Informatics, Leiden, The Netherlands

Eduard Hoenkamp
Nijmegen Institute for Cognition and Information, Nijmegen, The Netherlands

Hilke Reckman
Leiden Centre for Linguistics, Leiden, The Netherlands

In the context of patients communicating about their disease, there are several channels along which this can be done. Most of these channels do not take the patient as primary input, but provide authoritative information. The Narrator system supplies patients with information extracted from personal stories in plain text format called "narratives". These will be processed and stored using techniques from both Information Retrieval and Natural Language Processing. As such, the system will be set up as a toolbox implementing different approaches while a Service Oriented Architecture provides the framework for integration. In this paper such approaches are described together with efforts to combine them within a suitable architecture. Furthermore, some of the important implementation details are discussed. As a starting point for the system, experiments have been carried out with initial narratives, the results of which are discussed.

Keywords: *computational linguistics, information retrieval, probabilistic methods, semantic web.*

1. Introduction

The Narrator project is part of the Dutch research program ToKeN (former ToKeN2000), which was set up and funded by the Netherlands Organization for Scientific Research, Dutch abbreviation: NWO (NWO, 2000). From the start three parties were involved in this project:

- the Leiden Centre for Linguistics (ULCL)
- the J.F.Schouten School for User-System Interaction of the Technical University Eindhoven (IPO institute)
- the Clinical Informatics group of the Leiden University Medical Centre (LUMC)

The goal of the project is to *define and possibly implement as a prototype, a system that is able to support patient communication in such a way that it provides patients (and their relatives) with information about their specific disease based on narratives supplied by other (fellow) patients*. A fundamental question is whether such a system has added value compared to existing channels in the health

care domain as described in (Toussaint *et al.,* 2002). Since the ToKeN program has a strong focus on access to knowledge, the initial question was reformulated in 3 separate questions:

(1) what kind of information are patients interested in? This depends on the type of illness; in the case of the Narrator project the patient group at hand was defined as women suffering from breast cancer. Questions to answer are:
   - what topics do these patients want to read about in fellow patients' stories?
   - what personal features do they want to know from authors i.e. fellow patients, in order to identify with them

   The relational aspect is of prime importance, it has high emotional value. These matters are investigated in the LUMC research activity.

(2) how should the information available preferably be presented to users i.e. fellow patients? How do patients delivering content interact with the system? These aspects concern human computer interaction and this field is covered by the activity of the J.F.Schouten School.

(3) what knowledge is needed to supply patients with the desired information and can we extract features from the information put in by fellow patients. This asks for a thorough semantic analysis of the texts at hand and is taken care of by the linguists from the ULCL.

Each of these activities is expected to result in a thesis, for which a period of four years is available.

Later in the project, the Nijmegen Institute for Cognition and Information (NICI) was added to it. Their expertise concerns Information Retrieval (IR) techniques; moreover, they were interested in applying these to the medical domain.

Soon after the project started (some 2 years ago) it was realized that none of the disciplines mentioned was able to deal with the research issues on its own:

- characterizing the various texts using linguistic means proved to be difficult, due to differences in length, style, use of grammar and vocabulary
- finding information based on analysis was not a mere information retrieval task: focus was needed to supply patients with exactly the information that best suited their specific situation. At the same time similar problems were noticed in the world wide web, leading to the Semantic Web Initiative
- studying the patient group at hand and the way they communicate with the system provides useful requirements for graphical interface design. However, we should also take into account the way the system is accessed and where information of what kind of is coming from, in order to find and present the information sought for by graphical means.

To find out whether results from each research domain could actually be applied in the process of patient communication, it was decided that a prototype should be built, with the LUMC Clinical Informatics group being responsible for this activity. This prototype was to be set up as a toolbox in which techniques from Natural Language Processing (NLP), Human Computer Interaction (HCI) and Information Retrieval (IR) can be explored, implemented, integrated and tested. If successful, the prototype could be extended to a complete i.e. operational, system.

The purpose of the Narrator system can thus be understood as to offer solutions from each of the areas mentioned by integrating them in a functional as well as technical manner. The purpose of this paper is to describe the explorative nature of the project, the bottlenecks encountered and the solutions used to realize the prototype of the system envisioned.

The paper is organized as follows: in section 2 the system is introduced along a chronological line: from the early stages of analysis resulting in the requirements, followed by a more concrete view in functional terms and finally to the proposed technical design. The functional view deals mainly with

process flow between yet undefined, subsystems (henceforth: components); we distinguish offline processing as a preparation stage for online retrieval. The architecture presented serves as a starting point for the integration of the various components.

In section 3 the several components are described from their respective disciplines: computational linguistics and information retrieval. In the last part of that section some (early) experiments with narrative texts carried out are described. These have already led to useful insights about integrating i.e. combining different techniques.

In section 4 implementation issues are discussed, mainly regarding the use of techniques from the Semantic Web (W3C Semantic Web Activity, 2001). Also, some characteristics of frameworks are described that have been investigated to implement the Narrator system.

Finally, in section 5, the results so far are summarized, together with preliminary conclusions and discussion of further work.

2. Narrator: a general description

2.1. The system analysis phase

Referring to the emotional issues in the Introduction, in terms of information retrieval this can be formulated as a demand for high precision and low recall. After all, it is better to return a limited amount of documents, some of which are really relevant, instead of overwhelming and thereby confusing patients with hundreds of hits. The aspect of emotion is once again crucial here. It also means that the system should provide an intuitive (low threshold) environment. Although this is a rather subjective criterion, for the Narrator system to have a real added value, we have to deal with it appropriately.

The above discussion resulted in the following requirements:

(1) the information sought for should be returned with high precision
(2) the retrieval process is constrained by time limits since it is interactive
(3) the system should be easy accessible, both from home, hospital and maybe elsewhere

To realize high precision a thorough semantic analysis of the narrative documents is vital. At first, a natural language parser was brought forward for this purpose; at the same it was expected that linguistic means were able to support the online retrieval, as part of a Dialogue Management facility. The latter proved to be another research area in itself; moreover, it implies an interactive i.e. real-time, environment. Since the linguistic analysis turned out to be expensive in computational terms, this led to the idea to restrict it to a non-interactive context, while actual information retrieval could be done in real-time.

It then became apparent that the narrative data at hand were very different from any text analysed by the parser before: they could not be classified as (semi-) structured data. The already noted incomplete and fragmentary nature of narratives demanded extensive enhancements of the parser and its lexicon. These are described in section 3.1

At the same time, the problem was viewed from a pure IR perspective: if narrative text could be suitably indexed, the resulting semantic space (see: section 3.2) could be searched to resolve a query against. This approach turned out to yield results that were difficult to interpret. Early experiments done for this purpose are described in section 3.3.

All of these considerations led to the following solution:

(1) split the system in an offline and an online mode of usage
(2) perform the expensive semantic annotation and indexing offline
(3) do online retrieval in two steps:
    - find relevant cluster from the generated index
    - identify relevant stories within this cluster through its semantic annotation

(4) set up the system as a Web based application. This ensures easy access, security becomes an important issue then.

Although some (prototypes of) components were already available, they were developed for a purpose too specific to be either used in another context or combined with other applications. In terms of design: they lacked a suitable interface to communicate with. This justifies the fact that the prototype was designed in a top-down manner, in order to carefully analyse the nature of data streams and how to define interfaces that handle them. As a consequence, the design of the prototype had to focus on the aspect of *integration*. In particular, how should output data from one module be transformed into a suitable format for the next module. Moreover, since the components were developed and available at different geographical locations, *distribution* had to be taken into account.

2.2. Narrator, a functional view

As mentioned in the Introduction: each of the research areas and their associated components alone were not able to cope with the specific nature of patient communication by means of narratives. This resulted in the idea to combine them, such that they could strengthen one another. In section 3.4 the considerations regarding this approach are discussed in detail. Together with the solution discussed in the previous section, this leads to the functional view of the system as depicted in Figure 1:

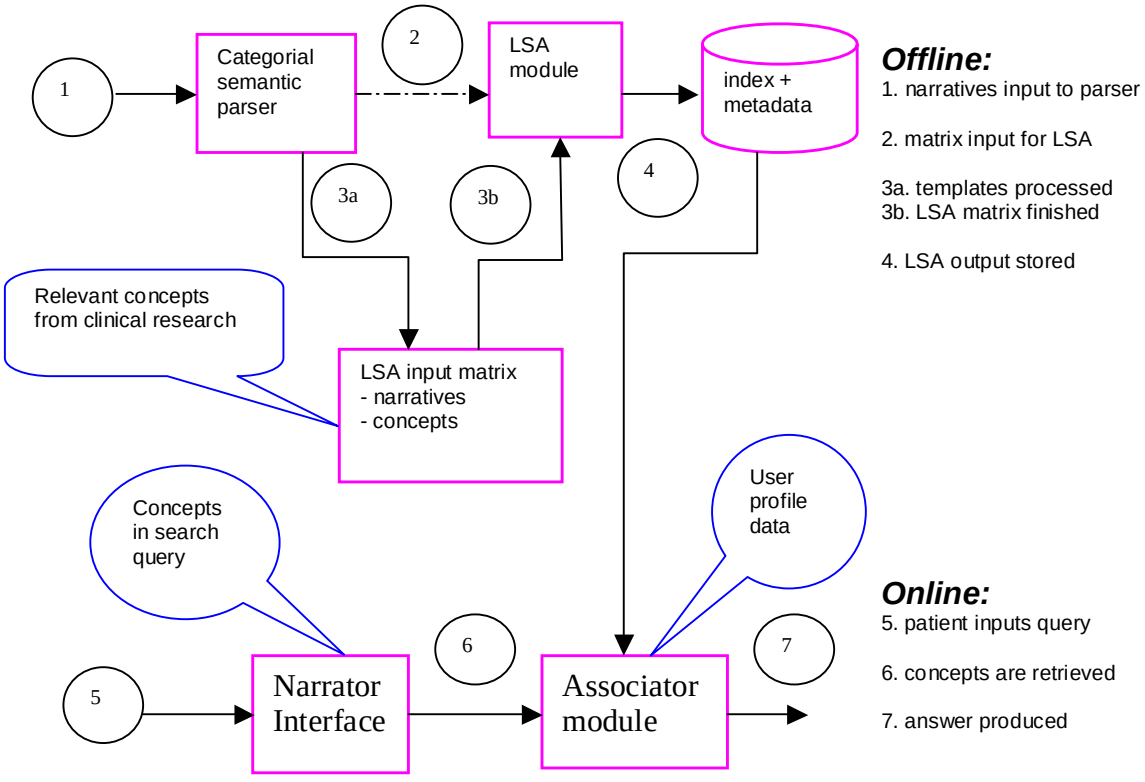

Fig. 1 The Narrator system in offline and online mode.

The components are shown in a process flow for two different modes of usage: offline and online. Figure 1 gives an impression of *how* these components, most of which are just black boxes at this stage, interact in terms of input/output streams.

In the offline mode, a patient supplies narrative data to the system; these are processed (steps 2 and 3) and stored (step 4) in specific formats. To a large extent these are XML formats used to annotate the linguistic data produced by the parser and the cluster data produced by the probabilistic LSA module (described in section 3.2). The online mode starts with step 5, where a patient poses a query to the system. After extracting conceptual data from the query in step 6, an answer is produced and presented to the user in step 7. To summarize: offline processing prepares the system for online usage.

It should be noted at this point that the arrows suggest direct connections between the modules, but they merely serve to indicate the direction of the flow within the whole process. In practice data transformations are needed to actually connect the various modules. Moreover, all the steps of the process need to be controlled by some central component. We come back to this point in the next section and in section 4.2.

The Associator module acts as a kind of mediator between the two modes: it uses structures generated in the offline process as well as query information from the user interface to produce an answer. Information from previous online sessions (in the form of profiles) is kept to further assists in the search and retrieval process.

Both the clinical and the query related concepts are investigated in the research activities designated in the introduction with 1 and 2 respectively. The first category results from studying the narratives and from interviewing patients; these concepts play a central role in the offline processing. In the online situation the patient supplies query data by means of GUI controls provided by the interface from which the other concepts are extracted. The Associator takes care of mapping these concepts onto the document space, which is semantically indexed as a result of the offline processing. As such, the Associator behaves like an intelligent agent, managing some kind of internal model of the knowledge involved and taking complex decisions based of several sources of information, as described in (Russell *et al.,* 2003).

2.3. Narrator, a technical view

In the preceding section the functional integration of the components was described in a qualitative way. After the functional analysis is finished, decisions have to be taken regarding the implementation aspects, the *how* part of the design. At this stage, integration has to be realized using existing and reliable technical solutions. At the same time, distribution becomes a crucial issue. Referring to a general 3-tier architecture, common to many Web applications that are inherently distributed, Figure 2 below fills in some of the details of the implementation. Since we are dealing with the Web, the initial starting point will be an HTTP client (be it desktop, laptop or mobile devices). Such clients typically connect to an HTTP server; from then on a multitude of (server side) techniques can be used. Here, a choice is made for a Java environment for the following reasons:

- it provides a flexible, distributed architecture (Sun Microsystems J2EE, 2005)
- it is widely supported by development tools
- most of the API's in used in the Semantic Web are written in Java as is true for the majority of XML API's

Since much of the processing as well as storage formats deal with XML, Java fits nicely in this scenario. Last but not least: all the different system parts need to be integrated. Since the introduction of Web Services (W3C Web Services, 2002) there is a renewed interest in service based architectures, now called Service Oriented Architectures (SOA's). They should cater for a framework in which appli-

cations can be called and activated from other applications, thereby creating a network of services. Whether and how this goal will be met are questions to be answered by building the prototype.

All of these considerations have led to the layered structure of figure 2:

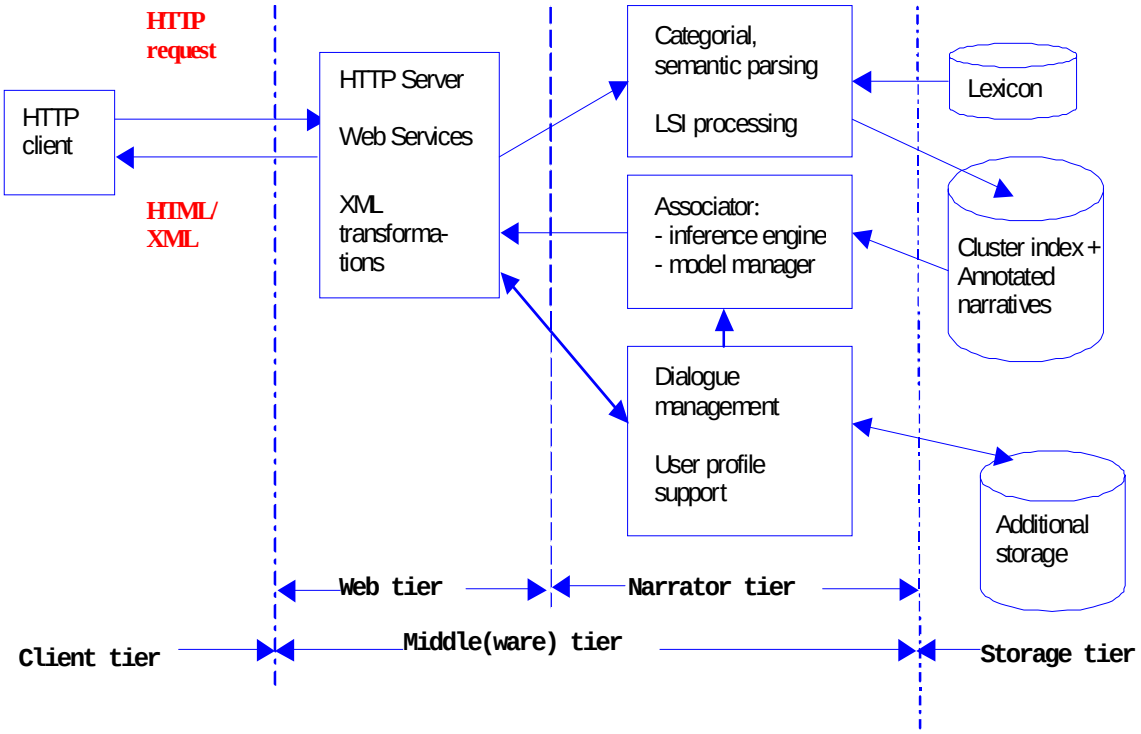

Fig. 2 The Narrator prototype as a 4-tier system.

Since the middle(ware) layer deals with everything from user interface to storage aspects, this layer tends to become a layered structure itself, where several aspects of business logic find a place in sublayers. From an architectural point of view, since boundaries are defined when and at places where this is deemed necessary, this changes the system from 3-tier to 4-tier. This leads to a Web tier, where both Web Services and XML presentation processing are implemented and a Narrator tier mainly devoted to language processing and intelligent search facilities. It also contains the Associator module that can be seen as a controller component in terms of the Model-View-Controller (MVC) design pattern. In the same parlour the web tier deals with most of the View functionality whereas the Model is implemented both by interfaces to storage capabilities and the storage facilities themselves. Right now, (open source) frameworks based on Java technology implement this pattern; further implementation details are given in section 4.2.2. It must be emphasized that the separation between layers is not fixed from now on; they can and certainly will be subject to changes and adaptations in the future.

3. Narrator components in some detail

In section 2.2 some of the components, were depicted much like labelled black boxes: input/output streams are drawn without a description of the relation between them. In the previous section they were situated in the business layer. In this section the components will be explained in more detail. They

cover two major areas of interest: processing plain text input and processing and setting up a semantically indexed document space.

3.1. Natural language parsing: Delilah

The NLP part of the system is implemented using an application called Delilah. It is a semantic parser for Dutch, developed by the ULCL. This parser is programmed in SICStus (formerly Quintus) Prolog and is based on the concept of a combinatory categorial grammar (Cremers *et al.*, 1996) and (Cremers *et al.*, 1997). At first it does the normal syntactical parsing of the sentences in the text and builds the associated parse tree that includes a semantic representation for each sentence; it is then used to construct semantic language templates. The Lexicon, an active dictionary that is able to deal with words and useful and common conjugations like collocations and constructions related to the idiom, supports the parsing process. In the Narrator system the Delilah application parses narrative data and produces output in XML format.

Since its Prolog version 3.10 the Swedish Institute of Computer Science (SICS) provides a Java API called PrologBeans. This API uses socket communication, details of which are hidden in both a Prolog library and a Java archive. At the same time the API is set up to support sessions from several clients; it uses a Java servlet for that purpose. The Delilah program was developed in the context of purely linguistic research. In the Narrator project it is adapted for the healthcare context, at the same time it was made more robust in order to cope with incomplete sentences. On the implementation level it was extended to support XML output streams.

3.1.1. Linguistic semantic analysis

The Delilah application performs so-called deep parsing, this results in very detailed semantic representations. These are logical formulas containing predicates, variables, constants and quantifiers that can be seen as a mathematical way of expressing meaning. Not only do they reflect the words that occur in the text by representing them as concepts, but they also specify relations between these concepts. In (Davidson, 1967) Davidson introduced the concept of action verbs with event arguments. Of the many extensions to his approach, the Delilah semantics is based on the so-called neo-Davidsonian event semantics (Parsons, 1994).Typical for this approach is to represent subjects and objects as separate conjuncts as well. As a result, all elementary thematic and adjunctive relations can be expressed as separate conjuncts. As a simple example consider the sentence “Henk werkt.” (Henk works). After analysis, the logical representation looks like:

$\exists$e.work(e) & event(e) & agent_of(e, henk) & attime(e, present)

where an event is characterized in terms of:

- type/name: a working event here
- an agent (the one who does it) named Henk
- time: the event occurs in the present time

The “agent_of” relation illustrates very well how relations between concepts can be represented. Furthermore, the “attime” relation separates the time aspects from the event itself, thereby yielding a canonical form (similar to the stem for verbs) describing the action. In the end, all (combinations of all) conjuncts are now logically derivable; this facilitates inference.

The main reason why the Narrator system needs these detailed representations is that the narratives are often pretty similar to each other i.e. they contain the same elements. This means that details matter to find the most relevant narratives for a specific user. Next, a few examples of such details and their use for retrieval are given.

(1) An analysis based on the linguistic structure can, for example, compute the scope of an operator like negation. This is important because it helps distinguish narratives in which a certain thing *is* the case, from narratives in which that same thing *is not* the case. In probabilistic approaches one can of course detect the presence of a negation, finding words like 'not', but without access to the structure of the sentence it is difficult to determine what exactly is negated.

(2) Another advantage, especially of the event based approach, is that it can identify concepts across different syntactic categories. For example, "Ik werd geopereerd." (I was operated upon.) and "Ik onderging een operatie." (I underwent surgery.) can be seen as different ways of saying the same thing. Both sentences tell that there was an operating event and that the person telling the story was the patient, the one who underwent the event. Delilah gives both sentences the same semantic representation, even though "operatie" is a noun and "geopereerd" is a verb. The representations are derived fully compositionally. In this particular example the verb and the noun get the same kind of event semantics in the lexicon with slots available for their participants. The verb "ondergaan" (undergo) is analyzed as selecting an eventive nominalization and letting its subject bind the object-argument of the event. The temporal information encoded in the verb "ondergaan" is also applied to the surgery event. Lexical specifications thus provide the basis for this type of parallel interpretation.

(3) Combined with the previous point (about negation) one can retrieve now, for example, only narratives by/about people who didn't have surgery. This enables the formulation of quite specific requirements regarding which narratives the system should look for.

(4) Similarly, there is the possibility of lexical decomposition of conceptually complex terms. One can choose to spell out the meaning of certain words in a lot of detail. For example, medical treatments that are referred to by one word but involve manipulation of different things in certain parts of the body in a particular way. All the concepts involved and the relationships between them can in such case be included in the semantic representation.

In the end, Delilah yields so-called predefined templates, containing various linguistic entities. Since natural language is (mainly due to redundancy) inherently ambiguous, one sentence may result in more than one logical construct; on the other hand, syntactically different sentences may result in the same construct. This is not a matter of coincidence: every logical representation is deterministically calculated. Deciding which construct to use cannot be done without additional means. This is also true when an inference tool uses the semantic representations for the purpose of retrieval: these tools are still quite limited in what they can handle. To solve these problems the Narrator system may use a probabilistic technique like LSA (described in the next section). The combination of the different approaches is discussed in section 3.4 and depicted in Figure 2 in the upper-left corner.

3.2. Latent Semantic Analysis: the document space

One advantage of a linguistic representation as described in the previous section, is that information that is only implicit in the text can sometimes be made explicit by using logical deduction. Also, in principle at least, one could check if two documents tell the same story by checking if the stories are logically equivalent. Or one could check if documents contain equivalent passages that are just expressed in different sentences. So with the logical representation one can establish whether two documents (or parts of them) are semantically the same.

Unfortunately, it is very difficult to know whether documents are almost the same, or not so similar, or in how far they tell the same story. An approach that tries to answer exactly that very question can be found in the area of Information Retrieval (IR) and its use of search engines. The search engines that most people are familiar with take little heed of the structure of documents, and mostly consider a

document as just a collection, or bag, of words. Much of the relative success of search engines comes down to a clever balancing of word frequencies within the document, and the distribution of these words over all documents under consideration. This led to the definition of the *document space*. This space is a vector space that has the words (more generally the 'terms') as coordinate axes, and the documents as points in that space. Normally, some scheme of adding weighs to the dimensions is used. These vary from simple term frequencies (per document) to more elaborate functions. A well known example is the so-called tf/idf measure, defined as: (term frequency/inverse document frequency), where the denominator refers to the number of documents a term appears in. All of the terms and documents are put in a matrix, the element values corresponding to the calculated weights. From then on, computations on documents become linear algebra manipulations on the *document space*.

In IR, distance between documents is usually measured as the cosine between the document vectors. Hence, the documents that a search engine returns are those that make a small angle with the query. Other computations are used to avoid the lexicon problem (the problem of synonyms and polysemic words). Most notably, a technique of *dimension reduction* is applied: instead of taking the very high dimensional space where each word represents a separate dimension, the space is reduced to lower dimension of *latent semantic factors*. For a complete overview of the approach that was originally launched as Latent Semantic Indexing (LSI), but was recently renamed to Latent Semantic Analysis (LSA), the reader is referred to (Deerwester *et al.,* 1990). The technique is comparable to a factor analysis on high dimensional data; an overview and comparison of techniques like Singular Value Decomposition (SVD) and Haar (wavelet) transform can be found in (Hoenkamp, 2003). This way, for example synonyms that originally are different dimensions can reduce to one dimension representing the underlying meaning. The end result is a word-by-document representation which is the state of the art in search engine technology.

A search engine takes a query and returns documents relevant to the query by measuring their distances to the query. Likewise, within the context of the Narrator system, for a given story, the distances to other stories is measured, and the ones with the shortest distance are taken to tell a similar story. So, conceptually, the only difference with a routine Web search is that the query is much more elaborate and precise, namely a whole story instead of two or three keywords.

As an illustration of query elaboration, Figure 3 depicts part of the document space with several stories projected on the unit sphere in 3D, so that it can be visualized (Hoenkamp *et al.*, 2005).

The dimensions chosen are the largest latent semantic factors of the original space. Suppose a new story were incrementally added to the document space. The first paragraph could be close to Story 1, but as more of the story is added, it travels through the document space, to end near Story 5 when it is complete.

If similar stories were all that is required, a straight forward clustering technique, such as *k-means* with cosine distance would be appropriate. And finding compatible others would be analog to finding relevant documents given a query. So in the example of Figure 3, stories 4, 5 and 7 would be selected as compatible with the new story. This is not necessarily the most compatible from the patient's viewpoint, as will be explained in a moment.

3.3. Initial experiment with patients' diary fragments using LSA

Let us summarise the approaches so far, and add a third one:

- The first approach parses documents into an underlying logical representation. A document is thus represented as a sequence of logical sentences.
- The second approach tries to discover implicit, or latent, semantics underlying documents. A document is then represented as point in a metric space defined on the latent factors.

- We added a third approach that takes the documents under study and tries to find the most important and general concepts that the documents have in common. Much of this work is done by hand, but usually involves the use of thesauri and techniques for text clustering.

Our research uses all three approaches. The next section reports on the first experiment we conducted to see if the factors produced by the LSA algorithm could be related to concepts that were constructed manually. For this a small sample of the narratives was split in meaningful parts, which were treated as isolated 'documents'.

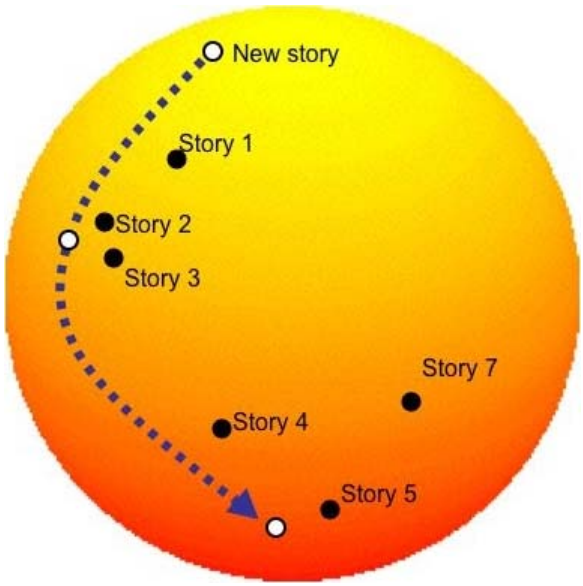

Fig. 3 Part of the document space containing several stories, and after dimension reduction to the three largest latent semantic factors. The arrow shows how a new story might travel through the document space, were it added in increments, from the first paragraph to the complete story at the arrowhead.

3.3.1. Objective and expectations

The Narrator system aims at providing patients with fellow patients' illness stories i.e. narratives, that match their profile and information needs. Illness stories have to be clustered according to several aspects in order to achieve accurate retrieval. Therefore, an experiment was performed in which clustering of cancer patients' diary fragments using LSA, was examined. It was expected that LSA clustered on:

(1) content i.e. concepts

Furthermore, we wondered if LSA was also capable to cluster on aspects like:

(2) moment in course of disease
(3) writing style, an author specific outcome
(4) any combination of 1,2 and 3. We may expect inner (nested) clusters here

The basic question to be answered by this experiment was: can we find a clustering in the narratives that makes sense i.e. is suitable to resolve search queries against. As such, it must be regarded as a first measurement to see what results LSA processing yields on raw data.

3.3.2. Some remarks regarding clustering

Clustering is an example of unsupervised learning, where one has to decide on:

- a measure of (dis)similarity; much used distance measures are: cosine, Euclidean, city block (Manhattan), Mahalanobis[1]
- the number of clusters, most of the time an empirical value
- a distance measure between groups: the several possibilities are discussed below.

The strategy of clustering we chose was *hierarchical*: start with every story in a cluster, then calculate the distance matrix (according to one of the above mentioned measures) and combine (see below) the clusters closest to one another. Continue until the desired number is reached.

Some criteria on which clusters can be combined are:

- single linkage, "nearest neighbor": $d(A, B) = \min \{d(a,b) \mid a \in A, b \in B\}$
- complete linkage, "farthest neighbor": $d(A, B) = \max \{d(a,b) \mid a \in A, b \in B\}$
- average linkage, calculates: $d(A, B) = \text{mean} \{d(a,b) \mid a \in A, b \in B\}$
- Ward clustering, seeks for partitioning that minimizes 'information loss' using a sum-of-squares error criterion. A partitioning here is every computed set of clusters (partition), up to some predefined number.

where: A, B are clusters, (a,b) pairs of objects i.e. stories, in these clusters and d(a,b) the elements of the distance matrix. For an experiment like the one we conducted, this is usually visualized by means of so-called dendograms, like those in Figure 4.

3.3.3. Materials and methods

For this experiment 128 cancer patients' diary fragments compiled in a Dutch booklet (Van den Borne *et al.*, 1987) were used. The diary fragments were mainly written down by six female cancer patients who differ with respect to age, marital status, and type of cancer. The diary fragments focus on feelings and thoughts, cover four different moments in the course of the disease, and range from a few sentences to a whole page. Editorial additions were removed, such as clarifications of medical terms and introductions to fragments.

The diary fragments were manually provided with an alphanumeric label that includes information about:

- the author of a fragment
- the moment in the course of the disease
- the number of fragments written by this author about this specific moment in the course of the disease

Moreover, all 128 diary fragments were read separately from each other and important topics were noted. From this topic list 17 concepts were extracted in four main categories. In addition, a fifth cate-

[1] The Manhattan measure sums differences between absolute values, while the Euclidian measure takes the square root of the sum of squared differences of the component values. The Mahalanobis measure takes covariances into account; we do not use this measure.

gory was used for fragments for which the context was not clear. To each fragment one or several concepts were attached.

The LSA algorithm was executed on the diary fragments without the performer being aware of the labelling. Two different n-dimensional spaces were used:

(1) based on the separate words in the diary fragments, "bag-of-words" approach
(2) based on the concepts attached to the diary fragments, "concepts" approach

The number of clusters that resulted from the bag-of-words approach was defined à priori to be 30. This number lies between 1 (all fragments in one cluster) and 128 (each fragment in a separate cluster). We chose n = 30 because it turned out that it resulted in a clear cluster graph. The number of clusters in the concepts approach was found to be 4.

The Ward technique was used to cluster the diary fragments, and Euclidian distances were calculated to determine distances between the diary fragments in space. In addition, the tf/idf weights, discussed in the previous section, were used in the bag-of-words approach.

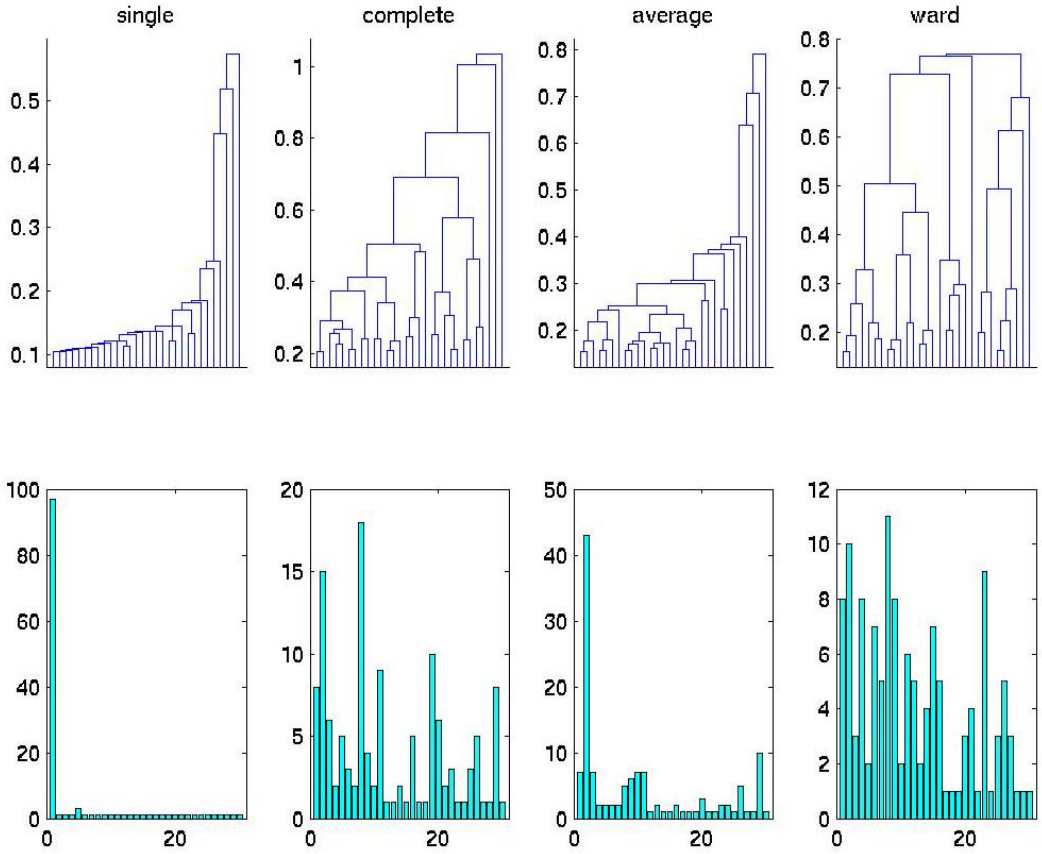

Fig. 4 The upper row shows dendograms to a level of 30 clusters, the lower row gives the number of documents per cluster. Note: the clusters are not in the same order.

3.3.4. Results

3.3.4.1 After LSA

The output of the algorithm in bag-of-words approach was mapped onto a 2-dimen-sional space, as shown in Figure 5. Note that, documents that are close together is this space, might be far apart when projected on a sphere like the one in Figure 4.

Given the fact that the stories are chosen without any knowledge of concepts or bias of any sort, this can only mean that either the stories are rather identical (apart from the outliers) or they are different but the LSA algorithm did not discover that fact. In the first case we expect to find one big cluster because finding more clusters of reasonable size would contradict the fact that each of the stories is just a random sample of words. In the second case, clustering will be a difficult job and will not distinguish the documents in a meaningful manner.

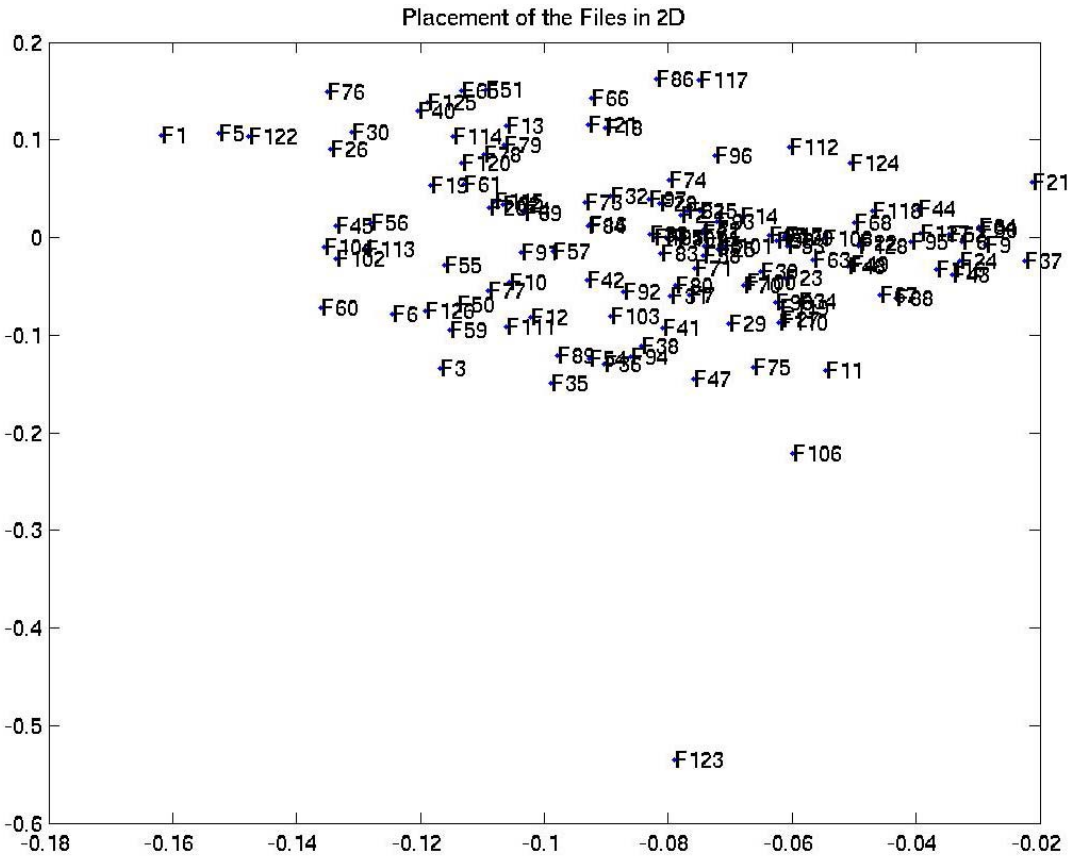

Fig. 5 Distribution of the documents, arbitrarily numbered from F1 to F128.

3.3.4.2 After clustering

The number of diary fragments in the 30 clusters resulting from the bag-of-words approach varies from 2 - 8. The concepts approach resulted in 4 clearly distinguishable clusters in a two dimensional space in which the number of diary fragments varies from 7 - 71. Table 1 shows the number of clusters found, categorized by the aspects mentioned in section 3.3.1. The number of fragments in the clusters

is not mentioned in table 1, only an extreme low or high number of fragments is reported in the table footnotes. Moreover, only outcome 1 is split in the categories: similar, rather similar, and totally different; for the other ones only the category similar is displayed. This reflects the fact that the number of clusters found is too low to be meaningful, which was the only matter of interest. Also, for this reason the values for outcome 1 add to 100 percent, while the others do not.

Table 1. Clustering of cancer patients' diary fragments by LSA in the bag-of-words approach and the concepts approach

Outcomes		Clusters, # (%)	
		bag-of-words (n=30)	concepts (n=4)
1	similar concepts [a]	0 (0)	3 (75)
	rather similar concepts [b]	4 (13)	0 (0)
	totally different concepts	26 (87)	1 (25) [e]
2	similar moment in course of disease	3 (10)	0 (0)
3	similar author	5 (17)	0 (0)
4	similar moment in course of disease, and similar author [c]	2 (7) [d]	0 (0)

Notes:

a) a certain concept is found in all diary fragments in the cluster; in several fragments in the cluster also other concepts can be discovered
b) a certain concept is attached to at least 70% of diary fragments in a cluster
c) only this combination is reported, because the other combinations (similar concepts, moment in course of disease, and author; similar concepts and moment in course of disease; similar concepts and author) resulted in zero clusters for the bag-of-words approach as well as the concepts approach
d) each of these two clusters consists of two diary fragments
e) this cluster consists of 71 diary fragments

3.3.5. Discussion of results

From Table 1 we see that the bag-of-words approach clustered cancer patients' diary fragments on moment in course of disease, author, or a combination of these two in such a limited degree that this clustering must be seen as coincidentally and not as structurally. In addition, for none of these three as-

pects clusters were found in the concepts approach. Thus, we can conclude that LSA does not seem suitable with respect to clustering on these attributes.

Both the bag-of-words approach and the concepts approach are, in the way conducted, not usable for clustering on content. Table 1 clearly shows that in the bag-of-words approach LSA did not cluster on content: 87% of the clusters contain diary fragment with totally different concepts. This conclusion can also be drawn from the fact that the bag-of-words approach and the concepts approach generate very different clusters, implying that the bag-of-words approach did not cluster on the basis of the concepts (i.e. content). One explanation could be the small amount of diary fragments (128). Or, it may be caused by the choices made regarding: adding weights to words, calculating distances between diary fragments in space, and clustering techniques used. From the fact that the generated clusters in the bag-of-words approach differ from those in the concepts approach, it can be concluded that the weights that are used in the bag-of-words approach were not appropriate to cluster the diary fragments on the basis of the concepts.

In the concepts approach 75 percent of the generated clusters contains diary fragments with similar concepts, where two sets of concepts share a third one (we have a nested cluster here). However, the biggest cluster with 71 diary fragments contains fragments with totally different concepts, suggesting that the majority of diary fragments discusses concepts that fall outside the first three sets. This may be due to the fact that the n-dimensional space in which the diary fragments are plotted are based on the concepts.

For the other outcomes, with no clusters found, conclusion cannot be easily drawn. Suppose all authors tend to write about different stages of their disease, while these were not explicitly mentioned, then there maybe no latent factors discovered by LSA that could be related to these stages. A similar argument could hold for the specific author characteristics that we were hoping to find: when a lot of different topics are discussed, this may obscure latent factors concerning the author. However, the added value of the concepts approach is that it will be clear if and how the concepts are related to one another. The fact that only four clusters are generated in a two-dimensional space on the basis of 17 concepts indicates that the 17 concepts are related to each other in some way. This implies that the diary fragments can be described with less than the 17 concepts used.

To summarize: in this experiment both the bag-of-words approach and the concepts approach did not cluster cancer patients' diary fragments on content in a meaningful way. However, the bag-of-words approach could result in more accurate clustering on content in this domain if more diary fragments are used and other choices are made in adding weights to words, calculating distances, and clustering techniques. In addition, with the use of an all-embracing, representative set of concepts that is as small as possible, clustering on content in the concepts approach could be rather accurate.

Final note: the various outcomes are not always readily explained; this is typical for the LSA/clustering approach, which in a sense is a combination of trial-and-error and experimentation. Probably other combinations of weighting scheme, distance measure and/or clustering technique will do much better. That is why further such experiments have been conducted, especially with other data sets and parameters. See e.g. (Hoenkamp *et al.,* 2006).

3.4. NLP and LSA: best of both worlds?

The initial experiments with LSA to find out if relevant clustering of narrative texts could be achieved showed it is difficult to interpret its results, both in the case of a bag-of-words approach as well as an approach in which texts are labelled with concepts relevant in the domain.

On the other hand, NLP techniques alone raised problems like the ones mentioned in section 3.1. So the idea emerged to combine the two techniques in such a way that they could strengthen one another. The basic idea of this combination is summarized in the following flow of control (see also Figure 1):

(1) In the first step, narratives are analysed by the Delilah software. So, all texts are transformed into semantic representations of their content
(2) These semantically represented texts are then fed through the LSA-module, which compiles an index based on the clustering of the annotated texts
(3) This index is used in the first step of the retrieval process. A cluster of texts is identified that has the best match with the user query.
(4) The texts within this cluster are then filtered using a moderate inference engine.

So, we see that in this approach the LSA technique and the semantic parsing technique of Delilah, are combined in several ways:

(1) Clustering by means of LSA is done on the basis of semantically analysed texts;
(2) The retrieval process is a two-step approach. In the first step the outcome of the clustering algorithm is used, and in the second step inferences on the symbolic representation of the semantic content of the texts is used.

4. The Narrator prototype: implementation issues

This section deals with two aspects of the implementation of the prototype: the use of XML technology and the integration of components in a distributed environment. The emphasis will be on the interconnection of the different components making up the prototype, as if they were tools from a toolbox. As already mentioned in section 2, Web Services seem to be the right means that achieve that goal. On the other hand, because it is new and maturing technology, we must wait and see if they can accomplish what other technologies failed to do in the past: realizing a truly and seamless distributed and interoperable environment (see: 4.2.1 below). The application of the Semantic Web is twofold: primarily as a format to annotate the narratives with information from the semantic processing and secondly, because it offers inference facilities, as an additional or complementary means for query resolution.

4.1. Semantic Web: applied XML

Looking for information on the Web is like searching for a needle in a haystack: several hundreds of hits with some ranking information for the simplest of queries, is quite common. This also holds for information about health and illness. Patients trying to find out about their specific illness, in their particular stage of a disease, after a certain amount of treatment etc. certainly need more focus than the Web nowadays can offer. In terms of IR: the recall is too high and the precision too low.

The World Wide Web Consortium started an initiative called „Semantic Web“ (W3C Semantic Web Activity, 2001) which aims at "an infrastructure for reasoning on the Web". It is based on several XML formats and technologies. These formats are available for different purposes: data exchange, data representation, data storage, data display etc. In the Narrator prototype XML formats are used to do annotations in the sense of the Semantic Web. In Figure 6 a layered structure of the Semantic Web is shown.

The red (lowest) layer specifies the basic building blocks of the Web: URI/IRI and the Unicode format (the HTTP protocol is taken for granted here). The orange layer defines the foundation for all XML technologies: the format specification itself (as an extension of HTML), Namespaces, XML Schema and XML Query. Specifications in these layers are in the stage of being or becoming official recommendations. The ones in the middle (grey) layers (RDF Model & Syntax and Ontology) are already a recommendation or they will be one very soon. The layers further up are more experimental

and in the stage of research. The addition of the vertical boxes (Signature and Encryption) along all the others makes the Semantic Web a secure environment that users can trust. For the Narrator prototype we focus on the grey layers: the RDF offers opportunities for useful annotating resources together with facilities to define relations between resources in an ontology that can be queried later on. These applications are explored below.

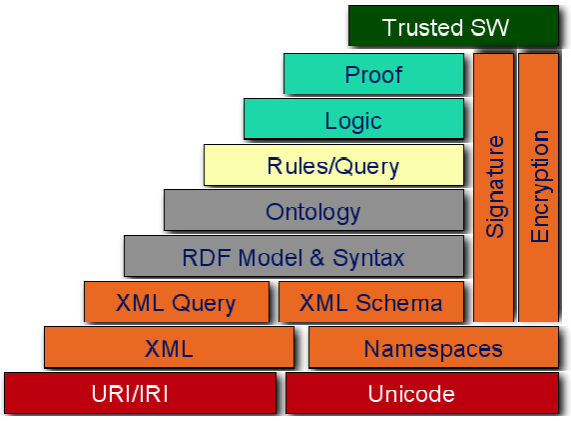

Fig. 6 The Semantic Web stack: how the W3C specifications relate to and build on one another.

In the healthcare community there is already some experience with linguistic tools in the area of medical terminology handling. For that purpose specific servers have been set up based on ontological principles to assist healthcare professionals. In the area of bio-medical research, where lots of different data-resources are interconnected and disclosed for information retrieval, there is a need for both integration and controlled vocabularies, in short: semantic interoperability. Some of the solutions there make use of techniques from the Semantic Web like RDF and OWL. Since the Narrator prototype deals with annotating narratives and defining relations between concepts for query resolution, it seems only natural to investigate the possibilities offered by the Semantic Web.

4.1.1. The Resource Description Framework: RDF

The specification of the Resource Description Framework (W3C RDF, 2005) paved the way for the Semantic Web: it both enabled resource annotation that nowadays Web search engines use already and stimulated research to further describe and annotate concepts from certain domains, eventually leading to the Web Ontology Language (W3C OWL, 2004).Graph based formalisms like RDF have been proposed earlier: in e.g. (Sowa, 2000) conceptual graphs are introduced that are very much related to predicate logic. However, these are most theoretical models that are hardly available for building applications. In the Narrator project we investigates which suitable implementations are available and RDF seems to be a good candidate. We use the Jena package from the HP Semantic Web lab (HP Labs, 2005), which is available via the open source community. This tool-kit contains implementations of the RDF and OWL specifications together with the RDQL query language.

4.1.2. Ontologies and the Semantic Web

There have been several initiatives in the area of ontologies: traditionally in the context of Linguistics and Artificial Intelligence. For the purpose of the Semantic Web an XML variant was defined, based on a restricted form of Description Logic. Together with initiatives to define query facilities for XML, this created new interest in inference engines and in the end, in logic and proof rules and -theory. Several implementations have been made based on the W3C specification. If and how these will be really useful for the Narrator prototype, is yet unknown.

4.2. Integration technology

4.2.1. Distributed systems technology and Web Services

The Internet was traditionally set up according to the client/server model, where multiple clients could connect to a single server (daemon) e.g. using the telnet or ftp protocols. In the late 80's when object oriented software development became popular, another kind of client/server model was introduced. In this model the client and server functions were regarded as a role: each server could also act as a client connecting to yet another server etc. This changed the initial 2-stage approach into a multiple stage one.

At the same time there were a lot of systems comprised of desktop (GUI) clients connecting to a central database server. In this 2-stage approach problems arose concerning the place where business specific software (data transformations, validation rules etc.) was to be deployed: on the client side or at the database server (by means of stored procedures). Together with the ideas from object oriented technology this led to the introduction of an intermediate layer dedicated to the business logic. This implied connecting to interfaces instead of directly operating on the data model; at the server side it enabled connections to several data sources.

An important question was how to connect from clients to servers in this distributed, networked environment. Important topics were: specification of an interface definition language, loosely coupled versus tightly coupled, connectionless versus connection oriented, platform and language (in)dependence etc. Implemented solutions were either consensus based (like OMG's CORBA, OSF's DCE) or proprietary ((D)COM, COM+ from Microsoft).

In spite of all the efforts each of these suffered from (technical) drawbacks. Recently, Web Services were introduced and they seem to be a candidate for standardization. For this reason and because they are rather well supported by development tools) and even complete frameworks like Microsoft's .NET are based on it), we take them into account for the Narrator prototype. The rest of this section is devoted to a well-known design pattern that has its origin in Smalltalk: Model, View, Controller (MVC).

4.2.2. The MVC design pattern

From the days of Smalltalk (Burbeck, 1987) this pattern was incorporated in the Rapid Application Development (RAD) approach of the late 80's where it was used to build rich, graphical user interfaces in a 2-tier environment. More recently, it became popular in the n-tier Java based enterprise applications. As such, it is usually presented as in Figure 7.

In essence, this application of the pattern resulted from the need to clearly separate presentation from content in web pages (the ones rendered by the browser) and also control flow management (dispatching) from data processing and - storage. The flow of control is as follows: a HTTP request is first sent (1) to the controlling servlet, which instantiates the necessary business logic components (2) that comprise the Model part and takes care of activating the presentation logic (3) in the Java Server Pages (JSP's) that make up the View part of the system. On the fly XSL transformations are performed as needed and the final response (5) is sent back to the requesting client browser.

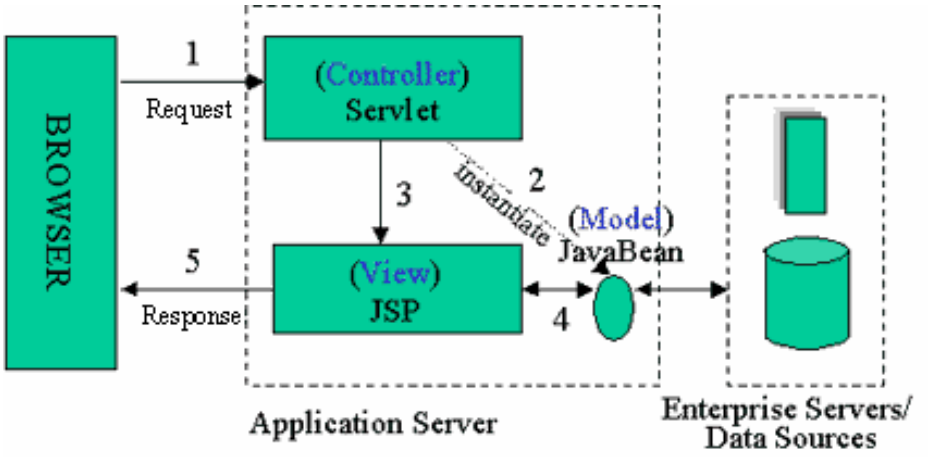

Fig. 7 The MVC pattern applied to Java Enterprise technology.

In the Struts framework (Apache Struts project, 2005) this pattern is generalized by making it highly configurable and by introducing components like Action, ActionForm, and ActionBean, which can be connected to one another using ActionMapping's. An Interactive Development Environment (IDE) like JBuilder supports this paradigm and it seems a promising framework for the project at hand.

During the last year the Struts framework has become the subject of serious debate. It's original inventor, Craig McClanahan, joined Sun and launched an alternative framework that has become known as JavaServer Faces (JSF). Although initially promoted as the preferred one, possibly in an effort to protect previous investments, both frameworks are now presented as opposite ends of a spectrum: Struts being procedure based and JSF component based. As part of technology assessment, both frameworks have been tried out. However, we have not been able to incorporate available component like Delilah in them. Web services seem to be preferable is this respect.

Besides that, new theories are developed in the area of component based software engineering. These are based on concepts like composition, coordination and associated languages for that purpose. This just goes to say that, as far as technology is concerned, we are never finished.

5. Conclusion

The Narrator project, as presented here, deals with an information retrieval problem faced with specific difficulties. First of all, the texts to be retrieved are highly heterogeneous, both in structure and content: they were constructed from incomplete sentences (fragments), written in varying styles (using different vocabulary) and by people with completely different backgrounds. Subtle semantic differences can determine the relevance of a specific text for a user, as was argued in section 3. This speaks in favour of symbolic and 'deep' semantic analysis, such as offered by Delilah. However, the incompleteness of the sentences encountered in the narratives favours a more robust analysis technique, such as LSA.

In this paper we have proposed an eclectic approach, which combines the linguistic analysis of Delilah with the document space analysis of LSA-like approaches. Our preliminary experiment, presented in section 3, has shown that clustering on the basis of concept-labelled texts appears to be more successful than clustering on plain texts. This could be due to the subtle semantic differences between the texts. However, only when suitable criteria can be defined beforehand and more datasets are taken into account, we can draw meaningful conclusions.

With the proposed design of the prototype we paved the way for algorithmic annotation of narrative documents leading to a semantic space that can be searched to resolve queries against. We do have to put it to the test though, since it all proved to be harder to realize than originally expected. The experimental nature of the prototype implies that all the proposed and developed solutions need to be tested, not only in terms of functionality but even more in the sense of producing meaningful results. If not, another solution must be found and implemented. In practice, this proved to be a time consuming process.

The adaptation of the semantic parser Delilah facilitated its integration and robustness. Further research directed at fine-tuning the set of concepts to be used by both Delilah (as the main semantic vocabulary) and LSA (in the clustering) is still necessary. These concepts will be validated, as stated, against real patient narratives and interview data from patients suffering from breast cancer. Since the project still runs for more than a year, we have ample time to conduct further experiments. The Narrator system can thus be rightfully characterized as a vehicle to test whether and which solutions actually work and then incorporate them.

6. References

Burbeck, S., 1987, "Programming in Smalltalk-80 (TM): How to use Model-View-Controller (MVC)," see: http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html

Cremers, C., Hijzelendoorn, M., 1996, "Filtering Left Dislocation Chains in Parsing Categorial Grammar," Proceedings of Seventh CLIN Meeting.

Cremers, C., Hijzelendoorn, M., 1997, "Pruning Search Space for Parsing Free Coordination in Categorial Grammar," Proceedings of International Workshop on Parsing Technology.

Davidson, D., 1967, "The logical form of action sentences," University of Pittsburgh Press, In the Series: The logical form of action sentences.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., Harshman, R., 1990, "Indexing by Latent Semantic Analysis," Journal of the American Society for Information Science, pp. 391-407.

Hewlett- Packard Development Company, 2005, "HP Labs Semantic Web Research," see: http://www.hpl.hp.com/semweb/

Hoenkamp, E., 2003, "Unitary operators on the document space," Journal of the American Society for Information Science and Technology, Vol. 54, No. 4.

Hoenkamp, E., van Dinther, G., 2005, "Live Visual Relevance Feedback for Query Formulation," Proceedings of 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York.

Hoenkamp, E., Overberg, R., 2006, "Computing Latent Taxonomies from Patients' Spontaneous Self-Disclosure to Form Compatible Support Groups," Presented to the MIE2006 (in press).

Nederlanse Organisatie voor Wetenschappelijk Onderzoek (NWO), 2000, "Toegankelijkheid en kennisontsluiting in Nederland (ToKeN2000)," (in Dutch) see: http://www.nwo.nl/subsidiewijzer.nsf/pages/NWOA_4XHBVY?Opendocument

Parsons, T., 1994, "Events in the semantics of English: a study in subatomic semantics," The MIT Press

Russell, S., Norvig, P., 2003, "Artificial Intelligence, A Modern Approach 2nd edn.," Prentice Hall

Sowa, J., 2000, "Knowledge Representation, Logical, Philosophical, and Computational Foundations," Brooks/Cole.

Sun Developer Network (SDN), 2005, "Java EE at a glance," see: http://java.sun.com/j2ee/index.jsp

The Apache Software Foundation, 2005, "The Apache Struts project," see: http://struts.apache.org/

Toussaint, P. J., Alpay, L. L., Zwetsloot-Schonk, J. H. M., 2002, "Communication Support: a challenge for ICT in health care," Proceedings of MIC 2002.

Van den Borne, B., Jonkers, R., Pruyn, J., Serail, T., (editors), 1987, "Over kanker geschreven: Dagboekfragmenten voor lotgenoten," De Toorts, Haarlem (in Dutch).

World Wide Web Consortium (W3C), 2001, " Semantic Web Activity ," see: http://www.w3.org/2001/sw/Activity

World Wide Web Consortium (W3C), 2002, "Web Services Activity," see: http://www.w3.org/2002/ws/

World Wide Web Consortium (W3C), 2005, "Resource Description Framework," see: http://www.w3.org/RDF/

World Wide Web Consortium (W3C), 2004, "Web Ontology Language (OWL)," see: http://www.w3.org/2004/OWL/