Organizational Memory: a Model Based on a Heterogeneous Network and an Automatic Information Integration Process

Jérémy Bascans[1,2], Max Chevalier[1], Patrice Gennero[2], Chantal Soulé-Dupuy[1]

[1] - University of Toulouse, IRIT (UMR 5505)
Toulouse, France
http://www.irit.fr
{First name.Last name}@irit.fr

[2] – Smart Kiwi
Ramonville-Saint-Agne, France
http://www.smartkiwi.net
{First name.Last name}@smartkiwi.net

***Abstract*— Organizational memory is a space where various information circulating in a company are capitalized. From the users' point of view, an organizational memory, which can be seen as an information system component, is very important since it stores the "shared knowledge" of the organization. But, at the same time, the cost of this knowledge is relatively high since users' participation, i.e. to integrate/maintain… the memory is important. The aim of our work is to model an organizational memory through a heterogeneous network on which is based an automatic information integration process to assist users in this task while limiting their effort. We developed a prototype and evaluated through an experiment its ability to integrate new information into an organizational memory based on the proposed model.**

Keywords*— *heterogeneous network; organizational memory; organizing information; information capitalization

I. INTRODUCTION

Today, information represents a significant capital for companies. Hence, Information Systems (IS), whose main role is to allow the collection, storage, processing and dissemination of information, have evolved with a new goal of capitalization and knowledge sharing. The concept of Organizational Memory (OM) was proposed as an answer to this need. Moreover, Organizational Memories become an important component of IS.

From end-users' point of view, building, populate and maintain an OM requires a strong cognitive and manual involvement. Furthermore [1] observed that systems based on community actions persist in time only when many people keep invested and active. In most cases, without such investment, OM are less and less used. In this context, our goal is to propose an OM that limits the investment expected from the users. First of all, we propose an OM model that is flexible and adaptable to many companies. This model is based more precisely on a heterogeneous network (i.e. a graph). Moreover, we define on this model an automatic information integration process. The users will just have to choose which information to capitalize and the system will "do the rest" in integrating this information into the OM.

In Section 2, we present organizational memory concept and heterogeneous networks. Section 3 presents the proposed OM model based on a heterogeneous network. Then, the definition of the automatic integration of information in the OM is proposed. We present in section 4 an implementation of our approach. Thanks to this prototype, we propose an experiment that evaluates its ability to automatically integrate and organize new incoming information into the OM. Finally, we present in the last section the different perspectives we identified to our work.

II. RELATED WORK

In this section, we introduce the definitions and issues surrounding organizational memories and their typologies. We identify then, in the related work, the challenges related to OM. Finally, we present the heterogeneous network model on which our approach is based.

A. Organizational Memory: definitions

In order to face the current companies' challenges, OMs aim at satisfying the needs of knowledge capitalization and perpetuation [2]. Knowledge can be defined ([3]) as the result of the interpretation of one or more information by a person. When a person wishes to share knowledge, he will generate and transmit information. This information will be interpreted by any other person to become his knowledge. However, for a correct interpretation, such information requires the addition of contextual elements, in order to ensure its commensurability (i.e. its evaluation "according to common measure units"). In this way and to maximize the knowledge sharing, an OM has to ensure that the interpretation of information contained in the memory remains as close as possible to the original knowledge.

In order to achieve these goals, organizational memories generally store, organize and share contextualized information that is used and conveyed in a company (that is seemingly heterogeneous and whose origin is not always known).

In order to take into account the companies' diversity, their business, their activities and the information diversity (formats, goals, etc.) several organizational memory types have been identified in the literature [4].

The typology that is commonly found in any organization is based on four memory types namely:

- *Business* (or technical) *memory* that stores all repositories, documents, tools and methods of a business domain. The volume of documents contained in this memory is very important because the result of a huge amount of experiences and researches on specific topics;
- *Company memory* that is strongly linked to the company itself: its activities, its products and partners (suppliers, customers, and so on). This is a collective memory because it is shared between all company employees;
- *Individual memory* that stores elements that are specific to an individual such as its status, skills, expertise and activities;
- *Project memory* that stores all information related to a project such as its definition, its activities, its history and its results. It represents all the experience acquired and implemented during a professional project.

B. Challenges & motivations

Several studies in the literature tackle organizational memories and the related implementation difficulties [5]. In these studies, we hold two main challenges linked to the complexity of implementing an OM in a company.

The first challenge we intend to address consists in providing a unique OM model that could correspond to any kind of OM (Business, Individual…). Indeed, every kind of memory is often very specialized. The corresponding implementation is generally based on a thorough knowledge of the company itself (structure, businesses, objectives…) [6]. To achieve this goal, the model should be sufficiently **generic and adaptable** to be transposable to most organizations and most information kinds. Moreover, an OM model must support **flexible update** of information contained in the OM (addition and removal). Lastly, an OM model must allow the **integration of contextualized information**.

As a solution, we define an OM model based on a *heterogeneous network* (cf. section II.C) that corresponds, to simplify, to a graph in which the information is organized. This organization relies on links that are semantically titled and characterized. This choice also offers to the model the required extensibility and flexibility.

The second challenge we intend to address consists in, on the one hand, **minimizing the users' efforts** (and involvement), and on the other hand, **minimizing the risk of bad information contextualization**. In fact, the use of an OM has a significant impact on the users' tasks and requires an excessive investment whose interest for them is not obvious at all. However, to ensure that OM is used (and so usable) and to ensure that users find an interest to use it in a collaborative way, the challenge is to provide an OM that is intelligible, effective, and at the same time non-intrusive [1].

As a solution, we propose to add in the OM different processes based on the model: (1) that limit the effort of users when integrating (and contextualizing) automatically an information in the OM and (2) that take into consideration the company's evolution over time. The OM should maintain a coherent view of information that can quickly evolve.

In this paper, we only focus on the first process that automatically integrates and contextualizes information in the OM. The second one remains as a perspective of our work

C. Heterogeneous Information Network

A heterogeneous information network is an information network composed of several objects types i.e. different kinds of nodes and links [7]. It is represented as a graph [8]. Such kind of network has widely been studied in recent years, especially in the Web and social networks fields [9], [10], [11]. A variety of related algorithms have also been proposed, particularly for the network structure mining [12]. Formally, these networks are well adapted for classification [13], clustering [14], ranking [15] and learning [16].

Hence, these networks offer us the opportunity to (re)organize various pieces of information, especially to make them intelligible (i.e. readable and understandable) owing to the various kinds of relationships that can exist between them.

This representation of heterogeneous and distributed information (through a heterogeneous information network) is considered in [17] as suitable to model an OM. Indeed, the heterogeneous information network provides the ability to represent information through different types of elements interconnected by different kinds of links.

Thus heterogeneous information network is really suitable to our goals since it can contain heterogeneous information, it supports many kinds of relationships (extensibility of the model) and offers a flexible structuring of information contained in the OM. Moreover such a structure can cover all OM types, and so, makes the memory adaptable to any company and memory type.

The following section presents the proposed OM model based on a heterogeneous information network.

III. Organizational Memory Model based on a Heterogeneous Information Network

As stated previously, our work aims at providing a consistent and sustainable organization of the information submitted by users into an OM. The proposed approach should be independent of the type of memory to implement (individual, business…). To do this, we defined the required features of an OM for a "generic" structuring of information.

We propose to use a heterogeneous network (i.e. various types of links can connect the same nodes or elements), which is navigable (the links are oriented), weighted and assigned.

A. Organizational memory: basic concepts

In this section, we present notations and definitions of this network and concepts allowing information organization in an OM. Finally, we present our OM model based on a heterogeneous network.

1) *Heterogeneous network definition:* The proposed formalization follows this notation: (1) sets are uppercase Greek symbols (e.g. ϑ or T); (2) values are lowercase Greek symbols (e.g. υ and τ), and (3) functions are lowercase Latin letters (e.g. w or n).

The graph φ is a heterogeneous network such as $\varphi = (\vartheta, \mathrm{E})$. $\vartheta = \{\upsilon_1, \ldots, \upsilon_v\}$ is the set of nodes, $\mathrm{E} = \{\varepsilon_1, \ldots, \varepsilon_e\}$ the set of links with $\varepsilon_i \in \upsilon \times \upsilon$. Each node has a type τ that belongs to $\mathrm{T} = \{\tau_1, \ldots, \tau_t\}$. The function $w : \upsilon \to \tau$ is used to return type τ of node υ. Each link has a type μ belonging to the set $\mathrm{M} = \{\mu_1, \ldots, \mu_u\}$. The $n : \varepsilon \to \mu$ function is used to return the μ type of a ε link.

Node and link types can be characterized by a set of attributes belonging to the set $X = \{\chi_1, \ldots, \chi_x\}$. All attributes of a node type τ can be retrieved with the function $y_\tau : \tau \to X_j$. All attributes of a link type μ can be retrieved using the function $y_\mu : \mu \to X_l$ where $X_j \subseteq X$ and $X_l \subseteq X$. An attribute value is recoverable from a node υ with the function $q_\upsilon : \upsilon \times \chi \to \alpha$ and from a link ε with the function $q_\varepsilon : \varepsilon \times \chi \to \beta$. α and β being the values of the attribute χ. These values are definable with the function $r_\upsilon : \chi \times \upsilon \times \alpha \to \upsilon$ for a node υ and $r_\varepsilon : \chi \times \varepsilon \times \beta \to \varepsilon$ for a link ε.

Note: all these definitions, features and functions are summarized in TABLE XI. in Annex section.

2) *Organizational memory concepts:* In order to use this heterogeneous network to implement an OM, we present "**Object Of Interest**", "**Category**" and "**Information**" concepts that constitute the core elements of the heterogeous network.

First of all, we propose to organize this information around the concept of "***Object Of Interest***" (OOI). This allows a coherent organization of the information collected in the OM and, in particular to contextualize this information. In other words, these OOI allow to bring together all information about, for example, a theme, a project, a person (who could be a specific contact in the company). Thanks to these OOIs, all OM users will access a shared representation of information.

The information explicitly introduced in the memory is represented by "***Information***" concept. We consider that each "*informational chunk*" at any granularity level (a text file, an email, a subsection of a document, a phrase etc.) is modelled through an *information*. The granularity level can be chosen by each company when building their own OM. The heterogeneous network allows this flexibility.

Then, "***Category***" concept, associated with any OOI by a specific relationship ("*belongsToCategory"*), describes which concept each OOI corresponds to. This "*Category*" concept is used to group "similar" objects from the company point of view. In order to characterize more precisely every OOI, a set of relationship types called "***Features***" are defined. These relationship types can be specific to each category. For instance, a category "*contact"* can be associated to specific relationship types: "*hasProfessionnalAddress*" or "*employeeOf*". Thanks to these features, an OOI that belongs to the category *contact* will have these specific relationships with information. Note that the category set and the associated relationship types will be defined by the company (t extensibility of the OM model).

As a simple illustration, in Fig. *1*, we find two OOI that represent two kinds of objects. *OOI#1* represents a *company* whereas *OOI#2* represents a *contact*. The two OOI are implicitly linked by one shared information (value: "*SmartK*"). Indeed, the information "*SmartK*" represents the name of *OOI#1* and corresponds as well to the company where *OOI#2* is an employee. This latter relationship comes, in our example, from the category *contact*. We also identify that the name of *OOI#2* is "*Jérémy B.*".

Thanks to the provided relationships, one can infer that *OOI#2* is linked to *OOI#1*. So implicit information and contextual information can be identified in such a model that allows users to see OOIs in their context (from the company point of view). This shows the high mining possibilities of the proposed model that will serve the OM users.

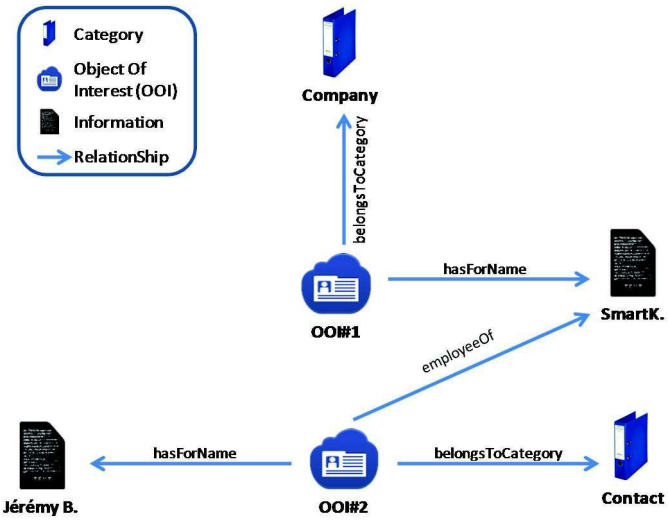

Fig. 1. OM core concepts

Based on these elements, each company will structure the information as it wishes. This model represents the angular stone of the proposed OM.

Organizational Memory complete Model: Based on the concepts previously defined (OOI, Category, etc.), we define the complete OM model based on the graph φ (cf. III.A). This network represents the information organization in the memory, where nodes have a unique type τ belonging to the set $T = \{ObjectOfInformation, Category, Information\}$. The nodes and links are characterized by a set of attributes (cf. III.A.1). More precisely in the proposed OM model, all links between two nodes, are characterized by a mandatory attribute named "*probability*" corresponding to the probability of existence of this link. Thus, information given by users has a probability value equals to 1 whereas all links that are identified by the system will have a probability lower or equals to 1. In addition every link type is characterized by an attribute named "*weight*" corresponding to the degree of

importance of such kind of link. This allows companies to identify among all link types those that are most important for them (e.g. according to their needs or their corporate culture).

Links and nodes can also have many other attributes (defined by the company) to express additional information about these nodes and links. For example, to more precisely characterize a link, an attribute called "*date of validity"* can be imagined. Indeed, this attribute could indicate the period during which the relationship is active. Such kind of information may be used to build an history that can be of interest for companies.

IV. Automatic Information Integration in an OM

In this section, we present notations and definitions used by algorithms aiming at automatically adding new nodes corresponding to a new information integrated in the OM.

Our goal is to automatically integrate new nodes into the heterogeneous network, and thus new information in memory. To do this, we propose algorithm 1. $\Gamma = \{\gamma_1, \dots, \gamma_u\}$ represents all algorithms to measure a correlation between two nodes and judge whether it is sufficient to establish links between them. The $d : \tau_a \times \tau_b \times \mu \rightarrow \Gamma_k$ function with $\Gamma_k \subseteq \Gamma$ allows to retrieve all algorithms that can enable the establishment of a link type between two node types τ_a, τ_b. For instance some algorithms can compute the similarity between textual content of two nodes. Some more complex algorithm can also be proposed. The $function\ g : \tau_e \times \tau_f \rightarrow \{(\mu_l, d(\tau_e, \tau_f, \mu_l)) \dots, (\mu_m, d(\tau_e, \tau_f, \mu_m))\}$ allows a set of links types and algorithms pairs. They determine whether a link can be established between the node types τ_e, τ_f. For the calculations, the $function\ c : \upsilon_s \times \upsilon_h \times \gamma \rightarrow [0,1]$ expresses in a range [0,1] the existence probability of γ link between nodes υ_s, υ_h. The $function\ p : \gamma \times \mu \rightarrow [0,1]$ allows to know the threshold beyond which one can consider that γ algorithm creates the link type μ. During the execution of algorithm 1, if function c returns a score higher than the threshold obtained by function p then the node will be attached to the network with the link type μ.

Thanks to the proposed model and the automatic integration process, a prototype named *Smart Kiwi* has been implemented. This prototype is detailed in the following section. This section also introduces the experiment we propose to evaluate the integration process quality.

V. Smart Kiwi Prototype and Evaluation

A. Smart Kiwi

Smart Kiwi is a prototype developed to implement a heterogeneous network and an automatic information integration algorithm corresponding to our OM model. The OM is thus as an instance of a heterogeneous network whose main nodes are objects of interests (OOI), categories (kinds of objects) and information granules (any kinds of documents at any kind of granularity). These nodes can be connected by different types of links that can depend on the type of node (cf. TABLE I.).

This prototype allows users to: (1) manage the OM by creating their interest topics (OOI); (2) automatically integrate new information and organize it in the OM (links computation). *Smart Kiwi* was developed so as to be generic with an objective of evolution and adaptation.

Algorithm 1 - Adding a new node υ_1 with type τ_1 into the network φ.

```
%Browse all graph nodes%
foreach node o ∈ ϑ (sorted by node type: information nodes first) do
  %possible pairs between the new node and network nodes%
  possibleLinks ← g(w(o), τ1)
  %Browse all pairs to extract%
  foreach pair (μ, Γo), link type and associated functions, belonging to possibleLinks do
    %algorithms for a possible link%
    foreach γ ∈ Γo do
      %Computing link probability thanks to selected algorithms and keep only relationships that have a probability higher than the threshold associated to this algorithm%
      if c(o, υ1, γ) ≥ p(γ, μ, w(o), τ1) do
        %if the node does not exist in the graph, add it%
        if υ1 ∉ ϑ do
          ϑ = ϑ ∪ {υ1}
        end if
        %The score is stored in the link l through the mandatory attribute%
        l ← rε("probability", mεn(μ, o, υ1), c(o, υ1, γ))
        %We add the link with the corresponding type between those nodes%
        E = E ∪ {l}
      end if
    end for
  end for
end for
```

Based on this prototype, we decided to evaluate our approach in a near real context. The goal of the experiment presented in this paper is to measure its ability to automatically integrate new information in the heterogeneous network (OM). For this purpose, we built a test dataset with information extracted from Wikipedia. The dataset and the protocol of evaluation are described in the following sections.

B. Dataset : Wikipedia

Wikipedia is the largest encyclopedia of knowledge [18] written and moderate collaboratively by several contributors and readers. It represents a large hypertext in which information and sources are organized into articles (some kinds of OOI).

In order to be close to a real use of the prototype, it was thus interesting for our experiments to select a subset of articles related to some specific topics, cinema in our case. Thus, we represent each article as an "OOI" belonging to a specific category.

We built the test dataset by extracting articles concerning the main protagonists of movies (e.g. actors, film producers, and film directors), events (mainly festivals) and products (mainly movies). To obtain a subset relatively homogeneous (cross references), we selected the most "popular" movies and festivals, different festival styles, with recognized awards like Oscars and Golden Globe.

The test dataset contains 1971 articles extracted from Wikipedia: 619 articles about protagonists, 1077 articles about movies and 376 articles about festivals. 1952 documents referenced by 875 of the 1971 articles were also extracted (uploaded).

C. *Memory creation: nodes and links instanciation*

Considering the information contained in a Wikipedia article (cf. Fig. *2*), we extracted several components: name, description, informational granules, features (from "Infobox" when this one is available) and references to external sources. Most of these external sources are text documents (PDF, Word, and so on) or Web pages. These documents have very heterogeneous formats. We have for example, PowerPoint presentations, Web pages, PDF files, etc. To take them into account in our OM, we developed the appropriate data readers that extract only the raw text of the documents.

The articles extracted from Wikipedia are integrated into the OM. They are integrated in the heterogeneous network as Objects of Interest "OOI" nodes. Then, the different information elements extracted from the article content are integrated as nodes which belong to pre-defined categories.

We defined three categories of OOI: (1) a "Person" category that will cluster together all protagonists, (2) a "Film" category that will cluster together all creations, and (3) an "Event" category that will cluster together all the festivals.

These categories allow us to define sets of links specific to each category. These links which connect information nodes to the OOI were identified from the different structures of the Wikipedia articles like names, descriptions and informational granules (cf. Fig. *1*Fig. *2*). Thus, we inferred the sets of links that are common to the same kind of articles, and thus which belong to the same category (TABLE I.). For example, movies articles belong to the category "Film" and each "OOI" node of this category "Film" will have the same kinds of links to other nodes. In this way, as shown in TABLE I. , each link "semantically" connects textual information to an OOI.

Once the categories and link types identified, we extracted from the set of selected articles the available information that corresponds to the appropriate link type. So, for each Wikipedia article, we create a new OOI in the memory. This OOI will be immediately connected to new nodes corresponding to the information elements extracted from the article content by specific link related to the category of the OOI.

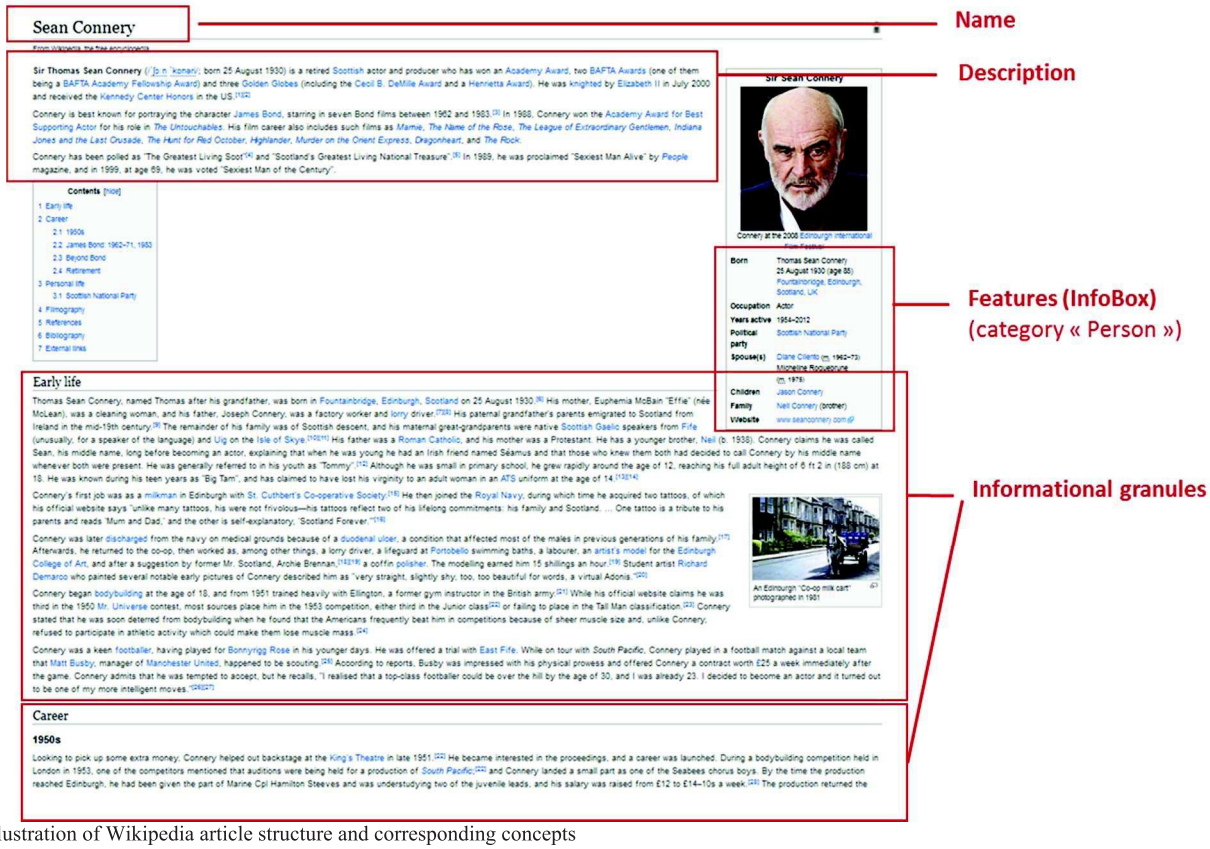

Fig. 2. Illustration of Wikipedia article structure and corresponding concepts

An example of instantiation is shown Fig. 3. The OOI#28 created from the Wikipedia article of Fig. 2 belongs to the category "Person" (specific link "*belongsToCategory*"). The information nodes correspond to the information elements extracted from the article content. They are connected to OOI#28 by 5 link types: "*hasForName*" (element Name), "*wasBorn*" and "*occupation*" (element InfoBox), "*isDescribedBy*" (element Description), "*hasForContent*" (informational granules identified by html tags).

TABLE I. AVAILABLE LINKS BY OOI AND CATEGORY

Elements	*Position in memory*	*Link types example*
Object Of Interest (OOI)	Common to all these OOI in memory	Name, Description, Referenced documents
"Person" category	Common to all OOI belonging to the "Person" category	Career, Privacy, Biography, Family Name, Filmography, Date of birth, Profession
"Film" category	Common to all OOI belonging to the "Project" category	Synopsis, Summary, Around the movie, Comment, Release Date
"Event" category	Common to all OOI belonging to the "Event" category	History, Historic, Creation date

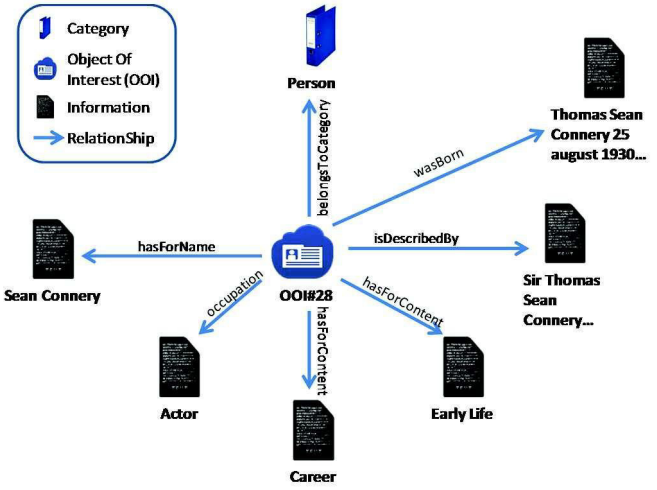

Fig. 3. Instantiation of OM model from the Wikipedia article of Fig. 2

D. Protocol algorithm settings

1) Set of algorithms: Since all the content extracted from Wikipedia corresponds to textual information we define the set of available algorithm Γ containing three algorithms (cf. TABLE II.). Two of them measure the similarity between two texts (i.e. two information type nodes) and one measures the proximity between an information type node and an OOI node. In this table, we show the main elements that are used in section IV. These measures are used to add a new information (i.e. a new information type node) into the OM and link it to existing nodes to "contextualize" this new information. The two first algorithms take into account text characteristics like length (cf. From/To columns). They measure relationships between an information type node, e.g. the new information node (*From*), and existing information type nodes (*To*). In our implementation, we consider that a text is short when it is composed of fewer than 20 words, and long in other cases. Note that the system is not limited to these measures since the algorithm set is extensible (for extensibility purpose). Moreover, thanks to algorithm 1 (cf. automatic information integration) only algorithms compatible with a pair of nodes will be applied according to their properties (i.e. long text and short text), each algorithm verifying if it is applicable to a pair of nodes.

The latest (third) algorithm aims at measuring the connexion strength between an information type node (e.g. the new information node) and an OOI. If the score resulting from each algorithm is greater or equal to the threshold (cf. algorithm 1 - section IV) a link between the two nodes is created and stored in the OM. The link type is given in the last column of the TABLE II. The link probability attribute value of this link is set to the score resulting from the algorithm.

So, in respect to algorithm 1 (automatic information integration) all scores resulting from the algorithms set Γ that are compatible with all the node type pairs are computed. After computing all relationships between a new information node and existing information nodes thanks to compatible algorithms (i.e. "Entities Search" algorithm or "Cosine" algorithm), connections between the new information node and all OOI nodes are evaluated (Connection algo.).

TABLE II. AVAILABLE ALGORITHMS (Γ) TO LINK TWO NODES

Algorithm Name	*From* υ_1	*To* $o \in \theta$	*Threshold* $p(\gamma,\mu,w(o),\tau_1)$	*Resulting Link Type* μ
Entities search	Information node	Information node **(short text)**	0	similarAs
Cosine	Information node	Information node **(long text)**	0.15	similarAs
Connection	Information node	OOI node	0	connectedTo

a) "Entity search" algorithm is suitable when one wishes to evaluate the presence of a relatively short textual value (*To)* in a text (possibly long) (*From)*. For example, it is used to search the existence of the OOI name value (e.g. the information node value linked to the OOI via the relationship *hasForName*) in the new information node value. We apply entities search to all words extracted from the *To* node value (e.g. "Sean Connery").

The score *S* related to the entity search in another text (*From*) is given by:

$$S = \begin{cases} 1 & \textit{if all words of the } \textbf{\textit{To}} \textit{ node value occurs in the value of } \textbf{\textit{From}} \textit{ node} \\ -1 & \textit{if at least one word of the } \textbf{\textit{To}} \textit{ node value } \textbf{\textit{does not occur}} \textit{ in the value of } \textbf{\textit{From}} \textit{ node} \end{cases} \quad (1)$$

b) "Cosine" algorithm is suitable to measure textual similarity between two long texts, e.g. the similarity between textual information describing an OOI (i.e. an information node linked to the OOI via the relationship *isDescribedBy*) and an information node value (e.g. a new textual document). This similarity measure, widely used in information retrieval systems, is based on a cosine measure between the weighted vectors representing these two text terms [19]. To extract these terms and weight them, we apply the standard tf-idf approach [20] used in information retrieval field.
This similarity between two texts is calculated in three steps:

- Computing the weighting *tf-idf* of each term in texts to compare. *Tf* ("term frequency") is the frequency of a term *t* in a text *d* such that:

$$tf_{t,d} = \frac{n_{t,d}}{\sum_{i=1}^{k} n_{i,d}} \quad (2)$$

Where $n_{t,d}$ is the number of the term *t* occurrences in text *d* and *k* the number of words in the text.

The *idf* ("inverse document frequency") is used to measure if a term *t* is common throughout the text corpus. It measures the word discriminating power. It is computed within all available texts in the set of texts D:

$$idf_t = \log \frac{|D|}{|\{d_i : t \in d_i\}|} \quad (3)$$

Where d_i represents a text *d* containing the term *t*.

Finally *tf-idf* weighting of a term *t* in the text *d* is obtained by multiplying $tf_{t,d}$ with idf_t,:

$$w_{t,d} = tf_{t,d} \times idf_t \quad (4)$$

- In order to measure textual similarity, each text is represented by a vector composed of weighted terms by measuring $w_{t,d}$ (formula (4)).
- The score *S* corresponding to the similarity between two texts *d* and *s* (i.e. between two vectors) is calculated as follows:

$$S = \cos(\vartheta_d, \vartheta_s) = \frac{\vartheta_d \times \vartheta_s}{\|\vartheta_d\| \times \|\vartheta_s\|} \quad (5)$$

Where ϑ_d and ϑ_s correspond to vectors characterizing each text *d* and *s*.

When considering relationships corresponding to *features* characterizing a *Category*, one may identify that it can exist multiple information node values linked to the same OOI via the same relationship. In this case we apply for each value the compatible algorithm and then calculate the average of the obtained scores to obtain the final score S.

c) "Connection" algorithm is suitable to measure the strength of a link between an OOI node, denoted *On,* and an information type node (e.g. a new information node), denoted *In*. In this paper we define the probability of a link between *On* and *In* as a mean of all probabilities of links between *In* and all information nodes connected to *On* (i.e. all information nodes shared between *On* and *In*). So we use both existing links (i.e. links that have a probability higher than the threshold) and rejected links (i.e. links that have a probability lower than the threshold that have not been kept in the OM). Existing links correspond to nodes that are shared between *In* and *On* nodes. These shared nodes are identified in the following by a set named $\boldsymbol{Shared_{in,on}}$. Note that the probability of an existing link is recoverable with the function $q_\varepsilon(\textit{"probability"}, \varepsilon)$. Since the proposed model does not store rejected links we store them in a local database (specific to this algorithm).

The computation of the strength of a link corresponds to an aggregation based on 3 steps:

- Since every algorithm of the set **Γ** can return scores that have different value domains $[x_i, y_i]$, a normalization phase is needed before aggregation. This normalization is achieved for every relationship *i* linking *In* with each information node *sn* contained in $Shared_{In,On}$ and in our local database. Such normalization is computed via the following rescaling method (6):

$$S_i' = \frac{S_i - A_i}{E_i(S_i) - A_i} \quad (6)$$

Where S_i' is the normalized value of the probability score of the link *i*, A_i is the threshold of the link *i* obtained by the function $p(\gamma, \mu, w(sn), w(In))$, S_i the initial value of the probability of the link *i* and $E_i(v)$ is defined by Formula (7):

$$E_i(v) = \begin{cases} x_i & if\ v \in [x_i, A_i[\\ y_i & if\ v \in [A_i, y_i] \end{cases} \quad (7)$$

- The previous step provides scores S_i' in the range [0,1] and mix normalized probabilities from existing or rejected links. Consequently, this range can no longer distinguish the scores expressing if a link exists or not. We reintegrate this information by using the formula (8) and obtain a score N_i.

$$N_i = \begin{cases} S_i' & if\ the\ link\ exists \\ -S_i' & if\ the\ link\ does\ not\ exist \end{cases} \quad (8)$$

- The importance of a relationship existing between an OOI and an information type node could be different from one relationship type to another one. We have to take this into consideration in the aggregation. So we weight each score N_i with $P_i = q_\varepsilon(\text{"weight"}, \varepsilon)$ where ε corresponds to the link established in the graph or saved in our database. The final score *S* (formula (9)) is a linear function based on a weighted average of scores *Ni* where *k* is the number of relationships linked to the OOI node.

$$S = \frac{\sum_{i=1}^{k} N_i \times P_i}{\sum_{i=1}^{k} P_i} \quad (9)$$

2) *Weighting scheme – Algorithms settings:* After conducting a few series of tests in this specific field, we identified that the different information nodes do not have the same importance in the information integration process. Indeed, structured information contents and attachments can be heterogeneous. Some of them may not be helpful in the information integration process but in some cases they provide additional lighting and can participate in the decision of associating a text with an OOI.

To obtain the final score *S* (Formula 9, cf. V.D.c)), we carried out some experiments to determine the weights played by each element linked to each OOI.

By taking the test dataset and the example of Fig. *3*, four element contents are considered: "Name" element and "Description" element that are common to all OOIs, and then a concatenation of "Information granules" elements, and an average of scores of "Features" element.

The resulting weights are synthesized in TABLE III. And the final score *S* (Formula 9) was thus computed as follows:

$$S = NName \times 0.60 + NDescription \times 0.15 + NInformationGranule \times 0.15 + NFeatures \times 0.10$$

TABLE III. WEIGHTING ELEMENT

Element	*Value type*	*Weight*
Name	"Short" value	0.60
Description	"Long" value	0.15
Information granule (concatenation of Synopsis, Summary, Around the movie, Comment, Release Date, History, Historic, Creation date)	"Long" value	0.15
Features (average of scores of Career, Privacy, Biography, Family Name, Filmography, Date of Birth, Profession)	"Short" value	0.10

E. Smart Kiwi prototype evaluation protocol

We want to evaluate the ability of our prototype to automatically integrate new information in a relevant way. For this purpose, we chose to evaluate the ability of our prototype to organize information in the same way as Wikipedia; i.e. to link all documents referenced in Wikipedia articles with respective and equivalent OOI in the OM. Thus, the evaluation concerns the automatic information matching process whose goal is to associate information with memory OOIs. The protocol established to integrate new information into the memory is based on the algorithm presented in section IV (Algorithm 1).

Then we have to compare the real documents-articles associations of Wikipedia (cf. Wikipedia dataset in V.B) with the documents-articles associations computed by Smart Kiwi (cf. Algorithm 1).

1) *Similarity / dissimilarity matrix computation:* The associations between referenced documents and Wikipedia articles or OOIs are summarized using a matrix where the lines are the referenced documents and the columns (*T1, T2, T3*) are the Wikipedia articles or OOIs. If a document is associated with a Wikipedia article or an OOI then the corresponding value in the matrix is 1, and the value is -1 in the opposite case (cf. Fig. 4). The algorithm providing results (scores S) in the range [-1, 1], we consider that a value in the range [-1, 0[is negative (value -1) and a value in the range [0, 1] is positive (value 1).

$$\begin{array}{c} \\ Doc\,1 \\ Doc\,2 \\ Doc\,3 \\ Doc\,4 \\ Doc\,5 \end{array} \begin{array}{c} \begin{array}{ccc} T1 & T2 & T3 \end{array} \\ \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & -1 & -1 \end{bmatrix} \end{array}$$

Fig. 4. OOI (or Wikipedia articles) / Document matrix example

Thus we establish a Wikipedia matrix corresponding to the extracted data, to serve as a "baseline", and a Smart Kiwi matrix corresponding to our prototype results. We propose to compare these two matrix to evaluate the prototype following the protocol presented in the next section.

2) *Similarity/ dissimilarity matrix comparison protocol:* This evaluation is made on the test base described in the previous section. First, we measure similarity and dissimilarity between the Smart Kiwi matrix with the Wikipedia "baseline" by determining the documents correctly or not associated and rejected (not associated) as shown in Table IV:

TABLE IV. MATRICES COMPARISON

Result	*Description*	*Interpretation*
True positive	If a value 1 of Smart Kiwi matrix corresponds to a value 1 of Wikipedia matrix	Correct association
True negative	If a value -1 of Smart Kiwi matrix corresponds to a value -1 of Wikipedia matrix	Correct rejection
False positive	If a value 1 of Smart Kiwi matrix corresponds to a value -1 of Wikipedia matrix	Incorrect association
False negative	If a value -1 of Smart Kiwi matrix corresponds to a value 1 of Wikipedia matrix	Incorrect rejection

From the observations of correct and incorrect associations, we found interesting to analyze in more detail the results and to conduct analysis by OOI (Wikipedia article) and by document (cf. TABLE VI.):

- *Analysis by OOI.* We propose to evaluate by OOI the proportion of documents correctly associated in comparison with the expected documents (i.e. documents originally referenced by the corresponding Wikipedia article). This proportion is used to evaluate the "correct associations" by OOI. Similarly, we evaluate the proportion of documents correctly rejected by OOI (i.e. documents not referenced by the corresponding Wikipedia article). This computation evaluates the proportion of "correct rejections" of documents by OOI.

Finally, we analyze the incorrect associations and incorrect rejections by considering: the number of documents wrongly associated by OOI ("incorrect associations") and the number of documents that should have been associated with the OOI but are not ("incorrect rejections").

- *Analysis by document.* Similarly, we propose to do the same by document to evaluate: the proportion of "correct associations" and "correct rejections", the number of "incorrect associations" and the number of "incorrect rejections" by document.

F. Results and analysis

Starting from 1971 articles and 1952 documents extracted from Wikipedia (cf. V.B, Wikipedia dataset), we wanted to evaluate all possible associations between documents and OOIs. Thus, the prototype has generated 3,847,392 associations, considering an association for each pair document/OOI.

According to the Wikipedia dataset, 1952 documents are associated 2012 times with articles. Among these 1952 documents, 1897 documents are associated with a single article (several different documents can be associated with a single article) and 55 documents are associated with 2 or 3 articles. These 2012 associations take the value 1 in the Wikipedia matrix. All other associations have value -1. In this evaluation, we attach an equal importance to the prototype reactions for these 2012 expected associations, and for the rejection of other associations.

1) *Overall results:* Condering the number of computed associations (3,847,392 associations), the proportion of expected associations are very low compared to the expected rejections (2012 correct associations and 3,845,380 correct rejections on the baseline). Thus, we propose to evaluate separately the percentage of correct associations (true positives, cf. TABLE IV.) and the percentage of correct rejections (false positives). Then we express the global effectiveness of the prototype by the average percentage of correct associations and rejections (cf. TABLE V.).

TABLE V. GLOBAL RESULTS (WITH A THRESHOLD = 0).

True positive	*True negative*	*False positive*	*False negative*	*Total correct*	*Total incorrect*
87.28%	99.47%	0.53%	12.72%	93.37%	6.63%

We get a great rate of correct rejections (99.47%) and a good rate of correct associations (87.28%) with a total score at 93.37%.

2) *Results by OOI and document:* To better understand the OOIs and documents role in the integration behavior, we proposed to analyze these results by document view and by OOI view (cf. evaluation protocol in V.E). The results of this evaluation are summarized in TABLE VI. .

TABLE VI. ASSOCIATIONS AND REJECTIONS

	Average	*Standard deviation*
Results by OOI		
Correct association	89.00 %	27.80 %
Correct rejection	99.30 %	2.50 %
Incorrect association	9.687 documents / OOI	45.929
Incorrect rejection	0.293 document / OOI	0.874
Results by document		
Correct association	87.30 %	33.20 %
Correct rejection	99.50 %	0.40 %
Incorrect association	9.781 OOIs / document	8.591
Incorrect rejection	0.141 OOI / document	2.221

Note: The averages between analysis by OOI and document are close and all standard deviations are high. However, the standard deviation of "incorrect associations" by OOI is significantly higher than that by document, and the proportion of incorrect rejections is twice as high by OOI than by document. This is explained by the fact that almost all documents reference a single OOI (1897 documents among 1952) and almost half of the OOIs reference between 2 and 50 documents.

3) *Prototype strong points:* The analysis of the results highlights two strong points. First, 1815 OOIs (92.1%) were correctly associated with all documents referenced by the corresponding Wikipedia articles. It should be noted that there were in the dataset 1096 Wikipedia articles without any references to document. On 875 OOIs for which at least one association was expected, 719 (82.1%) were correctly associated with all the documents referenced by the corresponding Wikipedia articles. Only 64 OOIs (7.3%) have no expected document and 92 OOIs (10.5%) have a part of the expected documents with between 1 and 8 missing associations (whereas between 2 and 50 associations were expected by OOI). Second, 1700 documents (87.1%) are correctly associated with the OOI corresponding to the Wikipedia article that references the document. Only 8 documents (0.4%) are associated with only a part of the expected OOIs. These good results show that the information integration prototype globally reproduces the expected results (baseline).

4) *Origin of incorrect behaviors:* The analysis shows that 1804 documents (92.4%) may have been incorrectly associated with OOIs, the main reason being that the majority of them have been associated with the corresponding OOI but also with other OOIs (from 1 to 111 additional associations for some OOIs). Among these 1804 documents, 244 documents (12.5%) have not been associated with the corresponding OOI. However, all the "incorrect rejections" are due to these 244 documents corresponding to false errors because, after manually analyzing a sample of them, they refer to "incorrect" documents. Indeed, we identify that these documents do not

correspond anymore to the Wikipedia article content. So, the prototype cannot find anything in the content of these documents for generating the expected "correct associations".

Moreover, the "incorrect associations" only concern 55.2% of OOIs (i.e. 1088 OOIs). Every of these OOIs has between 1 and 900 incorrect associations (up to 46% of documents).

In the following section we propose a deeper analyzis of these results, and more particularly we focus on the incorrect associations related to OOIs.

G. Additional analyzes

1) *"Incorrect associations" analysis:* A study of the distribution of incorrect associations among these 1088 OOIs (cf. Fig. 5), shows that 308 OOIs (28.31%) have more than 10 unexpected documents. 31 OOIs among these 308 OOIs have more than 100 unexpected documents. Such result could have been anticipated thanks to the collection characteristics and the way we compute the final score *S*.

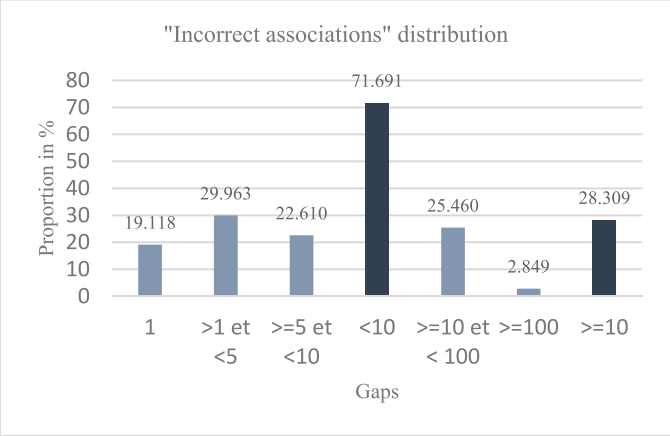

Fig. 5. Additional "incorrect associations" distribution by OOI

Indeed, the weight of "Name" element is 0.6 (of 1) in our *S* final score calculation (cf. V.D), it may correspond to the terms commonly used in the collection that create and explain these "incorrect associations". An analysis of these extreme cases confirms that these OOIs have names whose words have a high frequency in the collection. Indeed, some movie titles using common words in everyday language and not specific to the field (e.g. "Love", "Night", "Mission" or "Rebel"). Others designate common words in the collection as personalities whose names are often cited (e.g. "George Clooney" or "Woody Allen"), or several OOIs are thematically very similar (e.g. the "Hunger Game" saga where a Wikipedia article exists by movie).

In summary, the incorrect associations are resulting from the prototype behavior, and not from the collection (cf. V.F.4), i.e. the strong weighting of the OOI "Name" element in computation of score *S* of "connection" algorithm (cf. V.D.c). So, we study the sensibility of the automatic integration process to the weighting scheme used.

2) *Others weighting schemes in "connection" algorithm:* The proposed weight combination in the original "Connection" algorithm and its corresponding threshold was defined thanks to the results obtained in preliminary experiments. Moreover we identify in previous analyzis the impact of the high weight of "Name" element in the algorithm. We so wanted to determine and analyze optimum weight combinations specifically to the dataset. We denote *S'* and *S''*, two new weight combinations that:

- *S'*: maximize the percentage of "total correct" (cf. 0;
- S'': minimize the number of "total incorrect" (cf. TABLE VIII. .

To do this, we conducted a global analysis of all possible combinations of weights of Formula (9) in "Connection" algorithm. Every weigth varies from 0 to 1 by (step 0.05). At the same time, for each combination, we made the acceptance threshold varying from -1 to 1 (step 0.1).
Thanks to this study, we have identify the two optimal weighting schemes *S'* and *S''* as:

- $S' = NName \times 0.50 + NDescription \times 0.15 + NInformationGranule \times 0.35 + NFeatures \times 0$
- $S'' = NName \times 0.25 + NDescription \times 0.30 + NInformationGranule \times 0.40 + NFeatures \times 0.05$

Thanks to these two optimal weighting schemes we obtained results shown in 0 and TABLE VIII.

Concerning *S'*: we obtained a better percentage of correct (95.34%) with the threshold at -0.8 compared to 93.37% with weighting scheme *S* with the threshold at 0. This result confirms that "Name" element is important (weight=0.5) in "Connection" algorithm. In contrast, compared to the initial weighting of *S*, in *S'* "Information granule" element has a higher weight (0.35 in *S'* compared to 0.15 in *S*). We also underlined that "Features" element weight is totally ignored in S' (weight=0).

Concerning *S''*: we see a similar tendency for "Information granule" and "Features" elements in S'' as in S'. This can be explained by the fact that the documents used in the experiment are referenced by "Information granule" element and so contain main shared vocabulary. In counterpart the "Features" element is few present in Wikipedia articles or when present their content is not used in the documents we insert in the memory.

TABLE VII. S' WEIGHTING SCHEME IMPROVEMENT - THRESHOLD = -0.8

Weighting scheme	*Correct associations*	*Correct Rejections*	*% correct (avg)*
S	87.28%	99.47%	93.37%
S'	92.59%	98.09%	95.34% **+2.11%**

TABLE VIII. S'' WEIGHTING SCHEME IMPROVEMENT - THRESHOLD = 0

Weighting scheme	*Incorrect associations*	*Incorrect rejections*	*Total incorrect (sum)*
S	20491	256	20747
S''	208	1125	1333 **-93.575%**

In summary, we identified that the weight of "Name" and "Description" elements impact more the results than the other elements when applying "Connection" algorithm (connecting an information type node to an OOI node). So, when "Name" element of each OOI is not sufficiently deterministic the weighting scheme *S''* is more accurate than S'. In the contrary S' is more accurate than *S''* when "Name" element is deterministic. From a decisional point of view, we observed that *S'* weighting scheme is optimal (rather than *S''*) for our dataset since it maximizes the % of "total correct" while limiting the number of "total incorrect".

3) *New version of "Connection" algorithm:* The proposed "*Connection*" algorithm works with a local database keeping the scores of unkept relationships (see V.D.c). We identify that, in a concrete company, a high number of values (i.e. link probability) are stored to only compute the resulting value of the algorithm. For instance, in the previous version of the algorithm 62.819.264 link probability values are computed where:

- 4.587.559 links (7.30%) are really kept in the information network;
- 58.231.705 (92.70%) are stored in the local database.

In order to limit the required storage we propose an alternative of the "Connection" algorithm without any local database. The main evolution concerns the way *Ni* is computed in the algorithm. Thus, we modify the formula (8) of the algorithm by formula (10) as follows:

$$N_i = \begin{cases} S_i' & if\ the\ link\ exists\ in\ the\ network \\ -1 & if\ the\ link\ does\ not\ exist\ in\ the\ network \end{cases} \quad (10)$$

To evaluate this "stand-alone" version of the "*Connection*" algorithm, we conducted an analysis to identify optimal combinations of weights (see V.G.2) and compare it to S' and S" the optimal weight scheme for the original algorithm.

The optimal weighting scheme for the new version of the algorithm, depicted *S'''* (cf. (11)). Contrary to *S'* and *S''*, *S'''* is a single combination that, at the same time:

- maximize the percentage of "total correct" at threshold = -0.7;
- minimize the number of "total incorrect" at threshold = 0

$$S''' = NName \times 0.4 + NDescription \times 0.3 + NInformationGranule \times 0.3 + NFeatures \times 0 \quad (11)$$

The results of this combination are detailed in TABLE IX. and TABLE X.

TABLE IX. S''' WEIGHTING SCHEME IMPROVEMENT - THRESHOLD = -0.7

Weighting scheme	*Correct associations*	*Correct Rejections*	*% correct (avg)*
S	87.28%	99.47%	93.37%
S'''	88.47%	99.46%	93.97% **+0.642%**

TABLE X. S''' WEIGHTING SCHEME IMPROVEMENT - THRESHOLD = 0

Weighting scheme	*Incorrect associations*	*Incorrect rejections*	*Total incorrect (sum)*
S	20491	256	20747
S'''	71	1458	1529 **-92.63%**

Again, this result confirms that "Name" element is important (weight=0.4), "Features" element weight is ignored and "Information granule" and "Description" elements have higher weights than in *S* (formula (9)). Thus, the change in "*Connection*" algorithm provides a single combination whose results are below, but close to *S'* and *S''*. The main advantage of this new version of the algorithm is that it uses only 7.30% of the required information by the previous algorithm.

VI. CONCLUSION AND FUTURE WORK

In this article, we focus on the modeling and the automatic information integration in an organizational memory. This organization aims at being generic and adaptive. In addition, to meet the practices and uses of any company, it is imperative to be able to add an original process for automatic information integration. The proposed organizational memory model relies on a heterogeneous network that is associated to a set of generic concepts. An automatic integration process on the basis of our network has been proposed in order to take into account information heterogeneity. An implementation of our proposal is proposed. The latter can be extensible (link types, algorithms...). In order to verify if the proposed model and the associated integration process is accurate, we have set up an evaluation with a specific Wikipedia dataset. In this experiment we evaluate the obtained organizational memory and demonstrate that our prototype was able to correctly link the majority of documents with objects of interest. This shows that in a non-deterministic general framework, the use of the proposed model ensures the organization of Wikipedia data making them intelligible and allowing the automatic and correct integration of new information.

This paper describes the core principles for the definition of an organizational memory but many opportunities remain opened. We want to ensure that it meets all the challenges from the objectives set imposed by such kind of memory. More particularly we want to study more deeply the capability of the model to be adaptable and generic, through new experiments in various scenarios with several companies' partners of Smart Kiwi. Although the integration process is automatic we propose to include a feedback from users in order to adapt the way this process link new information in the OM. In addition, we plan to further analyze the "incorrect associations" proportion returned by the prototype (during the evaluation process). Indeed, much could be reduced by a new study based on a sample analyzed by several people (commensurability measure via for instance Kappa tests).

REFERENCES

[1] Ackerman M. S., McDonald D. W., "Answer Garden 2: merging organizational memory with collaborative help", Proceedings of the 1996 ACM conference on Computer supported cooperative work, 1996, p. 97-105.

[2] Basaruddin S., Haron H., Noordin, S. A., "Understanding Organizational Memory System for Managing Knowledge", International Conference on Advancements in Information Technology ICAIT 2011, IPCSIT vol. 20, 2011, Singapour, IACSIT Press.

[3] Nonaka, I., "A dynamic theory of organizational knowledge creation.", *Organization science*, vol. 5, no 1, 1994, p. 14-37.

[4] Dieng R., Corby O., Giboin A., Ribiere M., "Methods and tools for corporate knowledge management", International journal of human-computer studie, vol. 51, n° 3, 1999, p. 567-598.

[5] Atwood M.E., "Organizational memory systems: challenges for information technology." System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on. IEEE, 2002. p. 919-927.

[6] Hodge, G., "Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files", Digital Library Federation, Council on Library and Information Resources, 1755 Massachusetts Ave., NW, Suite 500, Washington, DC 20036, 2000.

[7] Flakes G. W., Lawrence S., Giles C. L., Coetzee F. M., "Self-organization of the web and identification of communities", IEEE Computer, 2002.

[8] Zhang J., Tang J., Liang B., Yang Z., Wang S., Zuo J., Li J., "Recommendation over a heterogeneous social network", Web-Age Information Management, 2008. WAIM'08. The Ninth International Conference on. IEEE, 2008, p. 309-316.

[9] Cai D., Shao Z., He X., Yan X., Han J., "Mining hidden community in heterogeneous social networks", Proceedings of the 3rd international workshop on Link discovery, ACM, 2005, p. 58-65.

[10] Mika P., "Flink: Semantic web technology for the extraction and analysis of social networks", Web Semantics: Science, Services and Agents on the World Wide Web, 2005, vol. 3, no 2, p. 211-223.

[11] Shen Z., Ma K. L., Eliassi-Rad, T., "Visual analysis of large heterogeneous social networks by semantic and structural abstraction", Visualization and Computer Graphics, IEEE Transactions on, 2006, vol. 12, no 6, p. 1427-1439.

[12] Schaeffer, S. E., "Graph clustering", Computer Science Review, 2007, vol. 1, no 1, p. 27-64.

[13] Kong X., Bokai C., Philip S. Y., "Multi-label classification by mining label and instance correlations from heterogeneous information networks.", Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2013.

[14] Papadopoulos A., George P., Marios D. D., "Identifying clusters with attribute homogeneity and similar connectivity in information networks.", Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on. Vol. 1. IEEE, 2013.

[15] Ming J., Han J., Danilevsky M., "Ranking-based classification of heterogeneous information networks.", Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011.

[16] Jones C. R., Ferreday D., Hodgson V., "Networked learning a relational approach: weak and strong ties", Journal of computer assisted learning, 2008, vol. 24, no 2, p. 90-102.

[17] Gandon F., "Engineering an ontology for a multi-agents corporate memory system." ISMICK 2001 Eighth International Symposium on the Management of Industrial and Corporate Knowledge. 2001.

[18] Denoyer L., Gallinari P., "The wikipedia xml corpus Comparative Evaluation of XML" Information Retrieval Systems. Springer Berlin Heidelberg, 2007. p. 12-19

[19] Gerard Salton, M. J. McGill, "Introduction to Modern Information Retrieval", 1983.

[20] Spark-Jones K., "A statistical interpretation of term specificity and its application in retrieval", Journal of Documentation, vol. 28, n°1, 1972, p. 11-20.

TABLE XI. MAIN NOTATIONS

Symbol	***Description***
$\vartheta = \{ \upsilon_1, \dots, \upsilon_v \}$	All nodes
$\mathrm{T} = \{ \tau_1, \dots, \tau_t \}$	All node types
$w: \upsilon \to \tau$	Function that returns to a υ node, its τ type
$\mathrm{E} = \{ \varepsilon_1, \dots, \varepsilon_e \}$	All links with $\varepsilon_i \in \upsilon \times \upsilon$
$\mathrm{M} = \{ \mu_1, \dots, \mu_u \}$	All links types
$n: \varepsilon \to \mu$	Function that returns for a ε link, its μ type
$X = \{ \chi_1, \dots, \chi_x \}$.	All attributes characterizing node and link types
$y_\tau: \tau \to X_j$	Function that returns for a node type μ, its set of attributes X_j
$y_\mu: \mu \to X_l$	Function that returns for a link type τ, its set of attributes X_j
$q_\upsilon: \upsilon \times \chi \to \alpha$	Function that return for a node υ and an attribute χ, the attribute value α
$q_\varepsilon: \varepsilon \times \chi \to \beta$	Function that return for a link ε and an attribute χ, the attribute value β
$r_\upsilon: \chi \times \upsilon \times \alpha \to \upsilon$	Function that update the value α of an attribute χ characterizing the node υ and returns the node υ updated
$r_\varepsilon: \chi \times \varepsilon \times \beta \to \varepsilon$	Function that update the value β of an attribute χ characterizing the link ε and returns the link ε updated
$\Gamma = \{ \gamma_1, \dots, \gamma_u \}$	All algorithms evaluating relationships between two nodes
$d: \tau_a \times \tau_b \times \mu \to \Gamma_k$	Function that returns for two node types τ_a, τ_b and a μ link type, a Γ_k algorithm set that are adapted to the node types, with $\Gamma_k \subseteq \Gamma$
$g: \tau_e \times \tau_f \to$ $\{ (\mu_l, d(\tau_e, \tau_f, \mu_l))$ $\dots, (\mu_m, d(\tau_e, \tau_f, \mu_m)) \}$	Function that returns for two nodes types τ_e, τ_f, a set of potential link types couples and algorithms to establish each connection between these two nodes types
$c: \upsilon_s \times \upsilon_h \times \gamma \to [0,1]$	Function that returns for two nodes υ_s, υ_h and an γ algorithm, the relationship probability between these two nodes
$p: \tau_e \times \tau_f \times \gamma \times \mu \to [0,1]$	Function that returns for two nodes types $\tau_e \times \tau_f$, an γ algorithm and a μ link type, the minimum threshold for the algorithm that valid this link between the two nodes types
$m_{\varepsilon_n}: \upsilon_p \times \upsilon_n \times \mu_l \to \varepsilon_n$	Function that returns for two nodes υ_p, υ_n and μ_l a link type, a ε_n link
$h: \tau_a \times \tau_b \times \mu \to [0,1]$	Function that returns for two node types τ_a, τ_b and a μ link type, a weight representing the importance of the relationships between the nodes
$f: \upsilon_s \times \upsilon_h \to \{ \varepsilon_1, \dots, \varepsilon_a \}$	Function that returns for two nodes υ_s, υ_h, all links $\{ \varepsilon_1, \dots, \varepsilon_a \}$ with $\{ \varepsilon_1, \dots, \varepsilon_a \} \subseteq \mathrm{E}$