1

Semantic Knowledge Representations for Soft Data Fusion

Claire Laudy
Thales
France

1. Introduction

Within the fusion community, studies have long been focused on the issue of fusing data that comes from physical sensors. This type of data is also called “hard data” as the data items represent physical measurements. Recently the issue of fusing information called “soft data” has come to the fore. Soft data is generated by humans and may be expressed as natural language or semi structured data. Soft data processing is a major issue for many support systems. The new intelligence support systems for instance, aim at taking the advantage of all types of data, and among them soft data available on the World Wide Web.

The aim of this chapter is to present the issue of soft data fusion and focus on one possible approach that allows taking into account the discrepancies that may be observed among different pieces of data.

The chapter is organized as follows. Section two gives an overview of what “soft data” is, as opposed to what is commonly named “data” or “hard data” within the fusion community. Relying on existing studies, we give an overview of soft data characteristics. We also emphasise on the need that has recently been identify, to take into account soft data within decision support systems.

Section three aims at giving the status and context of soft data fusion within the wide range of fusion approaches and studies. In order to explain the context of soft data fusion first, we present some of the different models that exist that aim at classifying fusion systems. Then, we focus on studies related to the fusion of graph structures, as they appear to be key structures for soft data representation. Given the exposed context, we then describe our proposition of framework for soft data fusion which is based on three main phases: the modeling of the application domain, an association phase and finally a multi-source fusion phase.

As we will see, soft data encompasses a high level of semantic. Therefore, semantic representation formalisms are needed for soft data representation and fusion. Section four is dedicated to the description of several semantic representation formalisms such as semantic nets, ontologies and conceptual graphs.

The fifth section is dedicated to the detailed description of our proposition for soft data fusion. The three phases defined before are detailed with a proposition of approach. We first describe a case study that will be used in order to illustrate our approach. It concerns TV program description fusion, within the context of a smart TV program recommendation system. Conceptual graphs are used for domain modeling. Therefore, we will focus on this semantic representation formalism and explain how we use it. We then describe the

association phase, which is carried out relying on several similarity measures and the fusion phase which relies on an existing operation on the conceptual graphs, the *maximal join*, extended thanks to domain knowledge.
Section six describes some of the experimentations that we conducted on TV program description, in order to demonstrate the validity and usefulness of our approach.

2. Soft data

2.1 From data to information: an abstraction process

In the domain of Artificial Intelligence (AI), data, information and knowledge are central notions. Many definitions exist for the words "data", "information" and "knowledge". In this paragraph, we give our definitions of these three concepts. They are inspired from the socio-economical domain. Data, information and knowledge are points in a continuum along which an abstraction process takes place.

Abstraction is the process of highlighting some of the aspects of a thing in order to grasp its characteristics. It is somehow a process of generalization. Abstracting an observable thing leads to a general representation of this reality, which is often called a *concept*.

Data items are unprocessed and uninterpreted symbols. They are elementary descriptions of measurable properties.

Information is what data items become once they have been interpreted and contextualized so to become useful within a specific objective and for a specific user. Having information is "knowing what is happening". The information answers to questions such as "who?", "what?", "where?" and "when?"

Knowledge is a combination of information with experience and judgment. It allows reasoning among information and interpreting data in order to create new data and information items. The knowledge answers to the question "How?".

In the specific case of fusion, the notions of data, information and knowledge are also linked one to another within the process of abstraction (see Figure 1). The aim of information and data fusion is to have a representation of an external situation. This representation can be built thanks to observations of the external situation that are acquired through sensors and reported to fusion systems. Sensors are either low-level physical sensors, that report about physical measurements such as temperature or speed, or human observers that report about (some parts of) complex situations. In the first case, the physical sensors give a set of data that must be interpreted. The human sensors, on the contrary, provide interpreted information. Combining all the information items in order to deduce new information and pieces of knowledge is the role of the information fusion systems.
Both data and information fusion systems use domain knowledge in order to interpret and combine the data and information items, according to a specific aim and within a specific context. Domain knowledge is also used in order to solve inconsistencies and discrepancies among the data and information items.

2.2 Soft data: a new challenge for decision support systems

"Soft data" items are observations that are generated by humans. They may be expressed as unconstrained natural language (see Sambhoos et al. (2008)), through textual data or speech signal, but can also be made of semi constrained data items such as xml files or data bases, which are keyed in by humans through forms for instance. As soft data is provided by

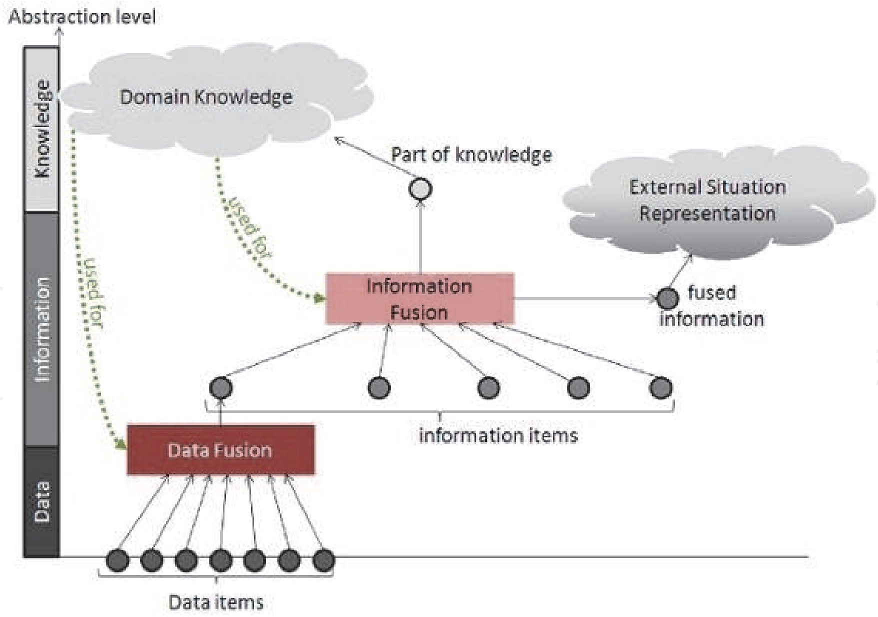


Fig. 1. Role of data, information and knowledge in the fusion process

humans, it is classified as information and knowledge according to the definitions given above.
The necessity of taking into account high-level information, which is also called "soft data" has recently been reported by the information fusion community. As stated in Pravia et al. (2008), under some circumstances, physics-based sensors are not as effective in the detection of people and complex activities for instance. In these situations, soft information sources are critical. The ability to express observed relationships is of importance for decision support systems, for inference purpose. However, most electronic sensors are feature based and do not generally provide information on relationships. Therefore, the study of soft data is of major importance.
Soft data items detections are typically qualitative, open to interpretation, and often outright inconsistent. These properties make the mathematical modeling of soft data very challenging. Studies such as Sambhoos et al. (2008), Petersen (2004) and Godbillon-Camus & Godlewski (2007) analyze the characteristics of soft and hard data, in order to clearly distinguish them. Three types of dimensions emerge from these studies:

Nature: hard information is quantitative - “numbers” (in finance these are balance sheet data, asset returns ...); soft information is qualitative - “words” (opinions, ideas, projects, comments ...); hard information is also rather “backward looking” (e.g. balance sheet data) as soft information is rather “forward looking” (e.g. business plan).

Collecting method: collection of hard data does not depend upon the context of its production, while collecting soft information includes its production context.

Cognitive factors: subjective judgment, opinions and perception are absent in hard information, whereas they are integral components of soft information.

Recent studies such as Buford et al. (2008), Sambhoos et al. (2008) and Laskey (2008) insist on the importance of such information for situation awareness and other decision support issues in general. They propose new approaches for information fusion, taking into account observations provided by humans.

3. Soft data fusion

3.1 Different levels of fusion

Many studies and systems exist that deal with data and information fusion. Each one of them focuses on a specific part of fusion. A detailed description of a large number of fusion models is proposed in Bosse et al. (2007). Within this section, we will focus on two of them. Our aim is to explain the purpose of semantic and soft data fusion, in the wide context of data and information fusion.

The North American Joint Directors of Laboratories (JDL) model for data fusion (see Hall & Llinas (2001)) is the most popular. It was first proposed in 1985 by the US Joint Directors of Laboratories Data Fusion group and revised over the years. The processing is divided into five levels described in Figure 2. The JDL model was initially proposed for the military applications but is now widely used in civil domains as well, such as business, medicine, etc.

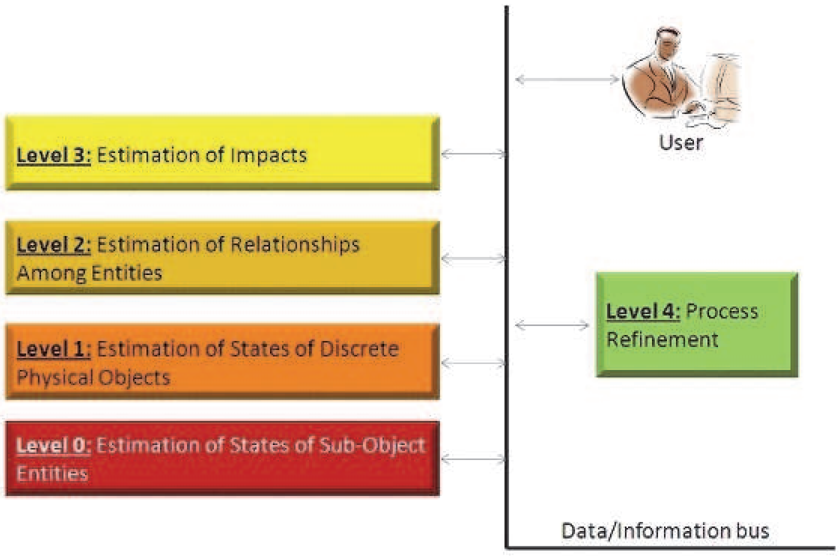

User
Level 3: Estimation of Impacts
Level 2: Estimation of Relationships Among Entities
Level 4: Process Refinement
Level 1: Estimation of States of Discrete Physical Objects
Level 0: Estimation of States of Sub-Object Entities
Data/Information bus

Fig. 2. The JDL data fusion process model (1999 revision)

Through its different levels, the JDL model divides the processes according to the different levels of abstraction of the data to be fused and the different problems for which data fusion is applicable. The initial levels are the following ones:

Level 0: Estimation of States of *Sub-Object Entities* (e.g. signals, features)

Level 1: Estimation of States of *Discrete Physical* Objects (e.g. vehicles, buildings)

Level 2: Estimation of *Relationships Among Entities* (e.g. aggregates, cuing, intent, acting on)

Level 3: Estimation of *Impacts* (e.g. consequences of threat activities on assets and goals)

Level 4: Process Refinement Level was initially recognized within the 1999 version of the JDL, but was then integrated to the Resource Management model levels and thus is not part of the fusion process itself.

Endsley (1995) models the data fusion process from a human perspective (i.e., Mental Model). The model has two main parts: the core Situation Awareness portion and the various factors affecting Situation Awareness. The core portion depicts three levels of mental representation: perception, comprehension and projection (see Figure 3):

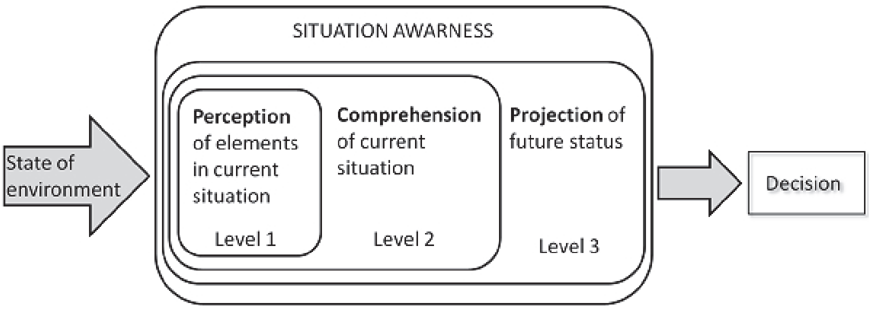

SITUATION AWARNESS
State of environment
Perception of elements in current situation
Level 1
Comprehension of current situation
Level 2
Projection of future status
Level 3
Decision

Fig. 3. Endsley's model of situation awareness (adapted from [Endsley, 1995])

Endsley's model illustrates three stages or steps of Situation Awareness formation: perception, comprehension, and projection.

Perception (Level 1): The first step in achieving Situation Awareness is to perceive the status, attributes, and dynamics of relevant elements in the environment. It also includes the classification of information into understood representations and provides the basic building blocks for comprehension and projection.

Comprehension (Level 2): Comprehension of the situation encompasses how people combine, interpret, store, and retain information. It includes the integration of multiple pieces of information and a determination of their relevance to the underlying goals. Comprehension involves a synthesis of disjointed Level 1 Situation Awareness elements through the processes of pattern recognition, interpretation, and evaluation. It includes developing a comprehensive picture of the world, or of that portion of the world of concern to the individual. Furthermore, as a dynamic process, comprehension must combine new information with already existing knowledge to produce a composite picture of the situation as it evolves.

Projection (Level 3): The third level involves the ability to project the future actions of the elements in the environment. Level 3 is achieved through knowledge of the status and dynamics of the elements and comprehension of the situation (Levels 1 and 2), and the ability to make predictions based on that knowledge.

3.2 Graphical methods for soft data fusion

Graph-based structures seam to be key structures for situation understanding and high-level information in general. Graph-based formalisms are easily readable and understandable by humans. Furthermore, graphs are a natural way to represent several ideas or objects interacting with each other. Therefore, information fusion based on graphs structures is a major stake

Sambhoos et al. (2008) relates about Inexact Graph Matching for real-time Fusion. Information items extracted from texts written in natural language are stored as RDF[1] triples. Each triple contains a subject, an object and a predicate (or relation) that exists between the subject and the object. The triples are considered as simple graphs.

Each text message may lead to the extraction of several RDF triples. These triples are then organized in more complex graph structures called "observation graphs". The larger graph is created by linking the commonly named nodes from the initial triples.

[1] Resource Description Framework (RDF) is a graph based model proposed by the World Wide Web Consortium (W3C), used to describe web resources.

Requests can then be processed on the set of observations, in order to determine whether a specific situation occurred or not. The study focuses on how to analyze the observation graph obtained after the fusion of the different information items. As for most of the studies dealing with graph homomorphism, the authors emphasize on the complexity of the underlying algorithms.
The authors propose an inexact graph matching technique using a similarity measure between the nodes and arcs. A model of a situation of interest is drawn, using a graph structure. The graph matching process then allows finding out whether this model is a sub graph of the observation graph.

3.3 Defining a high-level information fusion approach and framework

As said before, we focus on information that contains a higher-level of semantics. The approach that we proposed is optimized for information that has a high level of abstraction and that is structured. Regarding the JDL model, our approach is suitable for information fusion of levels 1 and 2 (Object Refinement and Impact Refinement). Level 2 - Comprehension - of Endsley's model for situation awareness corresponds well to our objectives as well: synthesis of perceived information items, determination of their relevance to the global objective of the user and (sub-)situation recognition through the matching with a sough-after situation.
We propose to use graphs representation formalism, and operations on graph structures. Representing information thanks to graph structures will allow us first to use existing operations and theoretical results on graphs. It will also enable to take the advantage of existing results regarding the optimization of graph algorithms. Our approach is close to the one proposed in Sambhoos et al. (2008). The aim is to fuse graphs.
However, we do not focus on the algorithmic issues of the problem, but on an other aspect: solving the conflicts that may appear in the data during the fusion process. When studying real soft data provided from operational systems, we observed that the different pieces of information, often contain conflicts. Indeed, as humans are providing the initial input data, there are very often typing mistakes or differences in the ways the same thing is reported. A simple example, is when one wants to refer to a person, a first human observer may use the person's name only, while another one will use name and surname. Therefore, the detection of conflicts among pieces of information, as well as the resolution of these conflicts within soft data fusion is of major importance.
We define hereafter the different stages that are necessary to achieve soft data fusion (Figure 4).

Situation & domain modeling is depicted by (1) and (2) on Figure 4. The situation modeling phase aims at providing a formal definition of the application domain as well as of the situations that are of interest for the user. The situations of interest are defined thanks to an empty pattern that describes their characteristics. The objective of the fusion system is to fill as much as possible this empty pattern with the observations acquired through the different sources of information.

Soft observations association is depicted by (3) on Figure 4. Observations coming from different information sources may relate to different objects. They may also relate to the same object but be reported with small differences. Therefore, it is necessary to determine whether two incoming observations rely to the same object before to fuse them.

Soft data fusion is depicted by (4) on Figure 4. When two observations are compatible and (at least partially) overlap, the multi-source information synthesis aims at building an unique view of the observed situation from them.

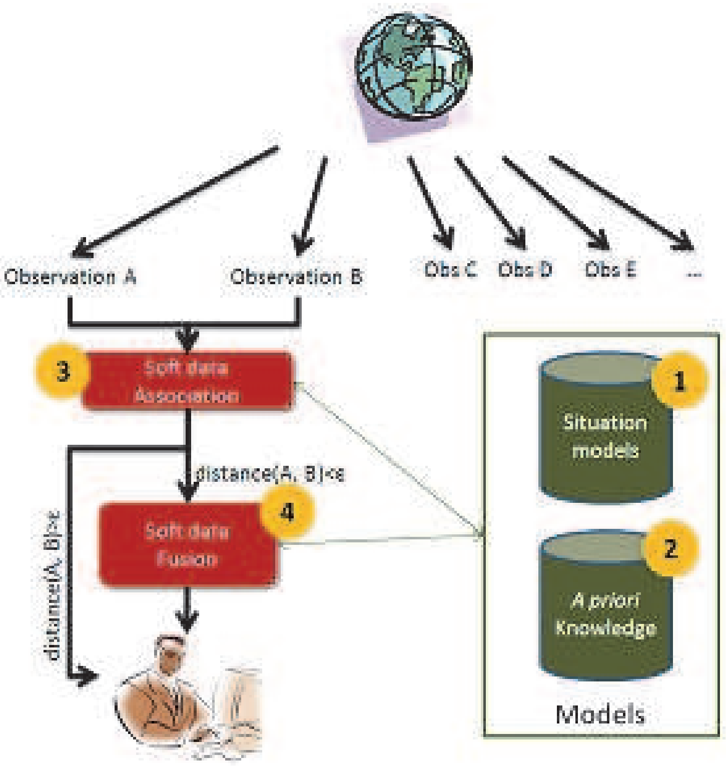


Fig. 4. General approach for situation recognition

In the remaining parts of this chapter, we emphasize on the modeling, association and fusion phases of soft observations that are not uniformly reported.

4. Domain knowledge and semantic representations

Domain knowledge has a major role within data and information fusion. Therefore, there is a need to express domain knowledge in a unique way, regardless of the different sources of information.

Furthermore, the data or information items acquired through the different sources are combined with this domain knowledge through the information process which produces new information items. This stresses the importance of having a unique formalism for knowledge representation that can also be used to represent and store the data and information that will be processed through fusion. The semantic used for representing the knowledge has to be shared between data and information as well.

4.1 Semantic networks

Within Artificial Intelligence, semantic representation formalisms were first developed in order to represent, store and automatically analyze the content of natural language.

Semantic nets (or Semantic networks) are graphical representations of interrelated concepts. A semantic network represents a taxonomy of concepts (or objects), denoted by the nodes of the network, and their properties, represented by the edges of the network. Two kinds of nodes

can be distinguished in a semantic network: concepts (or classes of objects) and individuals (or instances of the classes). The relations represented by the edges of the network are, among others, instanciation ("is a kind of") and composition ("has a").
The semantic networks were a first attempt to formalize semantic information and the reasoning process. However, as automatic processes attempted to get smarter, new semantic representations with more expressiveness and formalization were developed.

4.2 Ontologies

Ontology was initially a field of metaphysics, which aim is to study the kinds of things that exist and the relationships that can be observed among these things. Applied to the field of computer science, an ontology is a structured set of concepts that model a specific domain. Since the mid 70's, AI research scientists have had the need to capture knowledge, in order to provide "intelligent" capabilities to computers. Studies were achieved that aim at providing the ability to store general and domain specific knowledge, in a way that is understandable both by humans and computers.
An ontology captures the model of a specific domain as a catalog of categories of entities and the semantics associated with this domain. This allows making inferences and reasoning about the domain. The main components of an ontology are:

- individual entities,
- classes, which are sets or collections of entities
- attributes, which are properties of the entities represented by the different classes
- relations, which are relationships among different classes and
- events that change the state of some of the properties of entities or that modify relationships among entities.

Within the fusion community, works such as Matheus et al. (2003) insist on the importance of using ontologies to represent knowledge. For the military domain, several ontologies were developed, such as the JC3IEDM2 ontology (MIP (2005)) and the "SAW Ontology" described in Matheus et al. (2003).

4.3 Conceptual graphs

The conceptual graphs formalism is a model that encompasses a basic ontology (called *vocabulary*), graphs structures and operations on the graphs. The vocabulary defines the different types of concepts and relations that exist in the modeled application domain, while the graphs allow representing observations.
The conceptual graphs where proposed by John Sowa in Sowa (1976) as a graphical representation of logic, which was inspired by Peirce Peirce (1932). They allow representing knowledge in a easily readable manner for humans, experts of specific application domain, but non experts of knowledge representation formalisms. In our work, we use the conceptual graphs. The numerous studies achieved regarding graph algorithms and conceptual graphs in particular (Chein & Mugnier (2008)), lead us to use this model.

5. Using semantic representations for soft data fusion

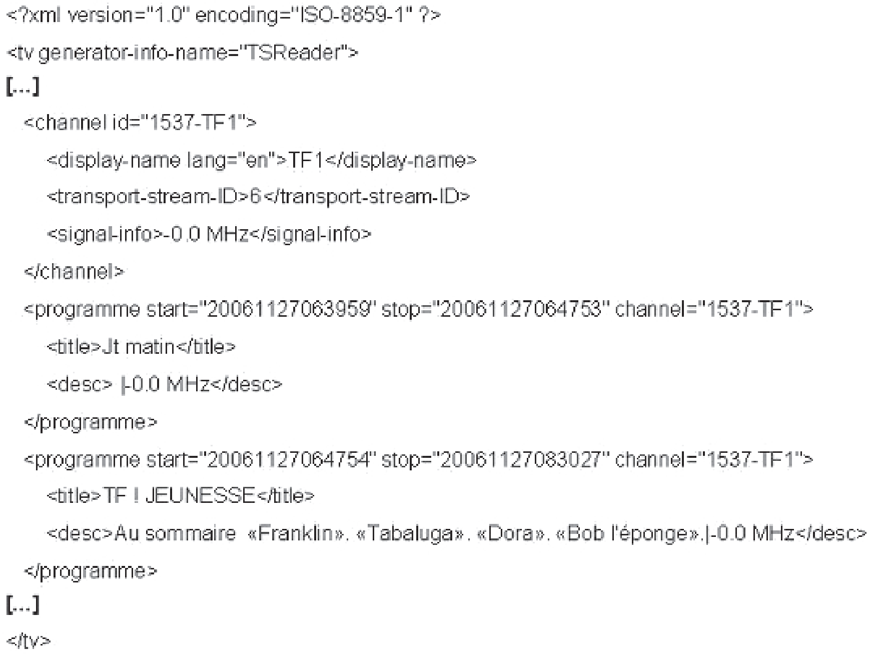
5.1 Case study

We applied our proposition for soft data fusion within a TV program recommendation system. While the number of channels that one can access increases very fast, the aim of the system is to help the users in choosing the programs that they would enjoy seeing. Based on background information and the description of a new program, the system evaluates whether the new TV program is of interest to a specific user. The recommendation system can be coupled with a Personal Video Recording system, so that interesting programs are recorded even when the users are not looking at TV.

The description on which the interest of a program is evaluated must therefore be very detailed concerning the content of the program itself. It should also be as precise as possible concerning the broadcast times so that the recording system can indicate the right slots of time.

Within the recommendation system, we used two sources of information. The first one, called DVB stream, is the live stream of metadata associated with the video stream on the TNT (Télévision Numérique Terrestre). Figure 5 shows an example of the information available on the DVB stream. The DVB stream gives descriptions of TV programs containing schedule and title information. It is very precise concerning the begin and end times of programs and delivers information about the technical characteristics of the audio and video streams. However, no description of the content of the program is given.

For each TV channel, it gives the descriptions of the currently playing program as well as the following one. The information on this source is constantly being updated. In particular, the scheduling times of the subsequent programs are adjusted.



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<tv generator-info-name="TSReader">
[...]
  <channel id="1537-TF1">
    <display-name lang="en">TF1</display-name>
    <transport-stream-ID>6</transport-stream-ID>
    <signal-info>-0.0 MHz</signal-info>
  </channel>
  <programme start="20061127063959" stop="20061127064753" channel="1537-TF1">
    <title>Jt matin</title>
    <desc> |-0.0 MHz</desc>
  </programme>
  <programme start="20061127064754" stop="20061127083027" channel="1537-TF1">
    <title>TF ! JEUNESSE</title>
    <desc>Au sommaire  «Franklin». «Tabaluga». «Dora». «Bob l'éponge».|-0.0 MHz</desc>
  </programme>
[...]
</tv>
```

Fig. 5. Example of an initial observation on TNT metadata

The second source of information is an online TV magazine. The descriptions contain information about the scheduling of the programs, their titles and the channels on which they are scheduled. They also contain details about the contents (summary of the program, category, actors, presenters etc). This source describes all the TV programs scheduled on all the TV channels during one week starting from the current day. The TV program descriptions may be updated once a day.

On the XML initial observations, we can see the information that we are going to fuse. For instance, the beginning time of the TV program appears inside the <programme> marker, as "start" attribute, the title is between the <title> markers, ...

5.2 Domain modeling / Representing knowledge with CG

In this section, we propose to use the Conceptual Graphs model in order to achieve the step preliminary to situation recognition: *domain and situation modeling*. We briefly describe the model, using the formalization proposed in Chein & Mugnier (1992) and Chein & Mugnier (2008). As said before, the conceptual graphs model was initially proposed in order to provide a logical system able to represent and process natural language. Therefore, it is particularly well adapted to the representation and processing of soft data.
The conceptual graphs model is essentially composed of an ontology called the *vocabulary* hereafter and the graphs themselves, containing concepts and relation nodes. We detail hereafter these general notions and their notations.

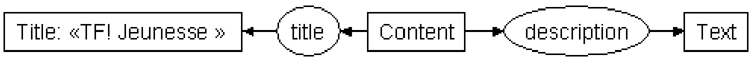


Fig. 6. Example of conceptual graph

5.2.1 Concepts, relations and vocabulary

The term "concept" is used to refer to a concept node. The concepts represent the "things" or entities that exist. A concept is labeled with two components: the concept's type and the individual marker. The conceptual type defines the category to which the entity belongs. For instance, in Figure 6 the concept [Title: "TF! Jeunesse"] is an instance of the category Title. Its concept type is "Title". The individual marker relates a concept to a specific object of the world. The object represented by [Title: "TF! Jeunesse"] has the name (or value) *"TF! Jeunesse"*. The first order logic form of the concept is $\text{Title}(``TF!Jeunesse'')$.
The individual markers may also be undefined. An undefined or generic individual marker is either blank or noted with a star *. It represents the existential quantifier. For instance, [Title] or [Title : *] stands for the following first order logic expression $\exists x, \text{Title}(x)$

The term "relation" is used to refer to a relation node. The relation nodes of a conceptual graph indicate the relationships that hold between the different entities of the situation that is represented. Each relation node is labeled with a relation type that points out the kind of relationship that is represented.
In this work, we consider binary relations. The arcs that link relations to concepts nodes are arrows, allowing distinguishing the source and target concept nodes.

The notion of vocabulary was defined in Chein & Mugnier (2008), as an extension of the *support* introduced in Chein & Mugnier (1992), which was itself based on Sowa's semantic network (Sowa (1984)). The concept types and the conceptual relation types, which are used to label the concept and relation nodes are organized in hierarchies. We restrict our approach to relation types that are unordered. Therefore, we manage only one hierarchy that contains the concept types.
The partial order that holds among the set of conceptual types is interpreted as a relation of specialization: $t_1 \leq t_2$ means that t_1 is a specialization of t_2, that is to say that any instance of the class denoted by t_1 is also an instance of the class denoted by t_2.

The ordered set of concept types is denoted T_C, the set of relation types is denoted T_R and the set of individual makers that are used to labeled the concept nodes is denoted $\mathcal{I}$.

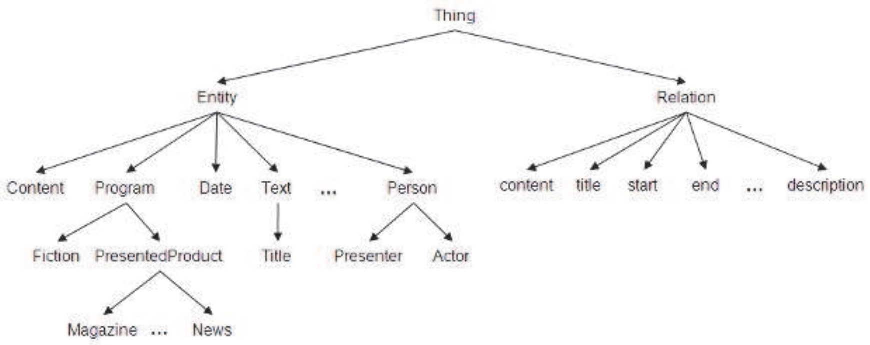


Fig. 7. Concept type hierarchy for TV programs

Figure 7 depicts an example of the hierarchy that contains the concept types used to describe TV programs.

5.2.2 Basic conceptual graphs

Several families of conceptual graphs (CG) exist, that allow describing different kinds of knowledge. Whithin this work, we focus on the basic graphs (defined in Chein & Mugnier (2008)). Basic conceptual graphs are bipartite graphs containing concept and relation nodes. Figure 6 gives an example of a basic graph. The rectangular boxes represent concept nodes and the ovals represent relation nodes.

Conceptual graphs allow to express logical formulas. Any conceptual graph can thus be translated into logic. The graph in Figure 6 for instance can be expressed in First Order Logic as follows:

$$\exists x, \exists y (\text{Content}(x) \wedge \text{Title}(''TF!Jeunesse'')$$
$$\wedge\, \text{title}(x, ''TF!Jeunesse'') \wedge \text{Text}(y) \wedge \text{description}(x, y))$$

A basic conceptual graph G is defined by a 4-uple over a vocabulary $V = (T_C, T_R, \mathcal{I})$: $G = (C_G, R_G, E_G, l_G)$ and is such that:

- (C_G, R_G, E_G) is a finite undirected and bipatite multigraph. C_G is the set of concept nodes. R_G is the set of relation nodes, and E_G is the set of edges.
- l_G is a naming function of the nodes and edges of the graph G which satisfies:
  1. A concept node c is labeled with a pair $(type(c), marker(c))$, where $type(c) \in T_C$ and $marker(c) \in \mathcal{I} \cup \{*\}$.
  2. A relation node r is labeled by $l(r) \in T_R$. $l(r)$ is also called the type of r.
  3. The degree of a relation node r is equal to the arity of r.
  4. Edges incident to a relation node r are totally ordered and labelled from 1 to the arity of r.

Given the order on T_C, the concept nodes that can be defined on a $T_C \times \{\mathcal{I} \cup \{*\}\}$ are partially ordered by the generality relationship. For instance, as the the concept type Text is greater (i.e. more general) than Title (see Figure 7) and the generic marker * is greater than any individual marker of $\mathcal{I}$, we have for instance:

[Text: $\star$] $\geq$ [Text : "News"] $\geq$ [Title : "News"], but [Text : "News"] and [Title: $\star$] are not comparable.

5.3 Soft data association

In this section, we focus on the *soft observation association* phase described earlier. The aim is to compare and analyze the different observations and decide which ones should be fused. More precisely, we detail here the comparison of two observations, taking into account the domain knowledge previously modeled as well as the values of the elements (i.e. nodes and edges) that make them up. This allows us checking their fusability.

We first describe here our proposition for a similarity measure between two conceptual graphs. Then we show how to use this measure in order to test the compatibility of two graphs in the association phase.

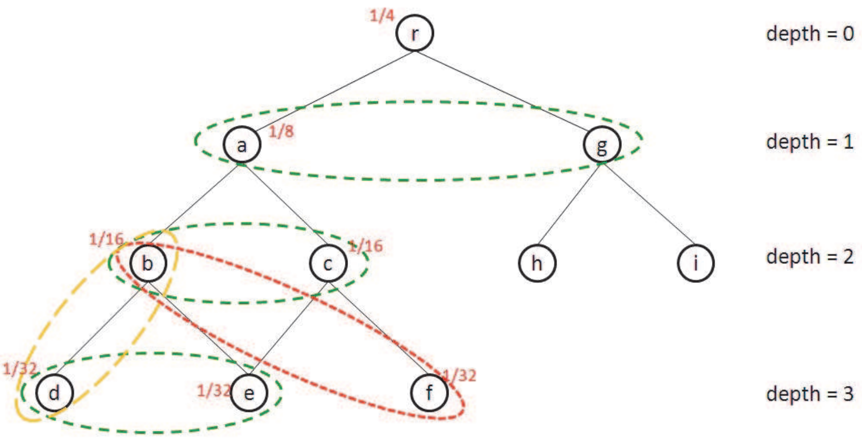
All the measures that we propose within this section are normalized. Extended proofs of this property are available in Laudy (2010)

5.3.1 Comparison of concepts

To measure the similarity between two concepts, we propose to compare their conceptual types, their values as well as their immediate neighborhood. The study of the neighborhood gives clue about the context in which a concept is used.

5.3.1.1 Comparison of conceptual types: $\text{diss}_{\text{type}}$

We first describe how to compare two concepts, regarding their difference, through dissimilarity processing. The dissimilarity between conceptual types is used to measure how much two situations are different. We adapt the distance between types proposed by Gandon et al. (2008), in order to obtain a normalized dissimilarity measure.



- $\text{diss}_{\text{Type}}(d,e) < \text{diss}_{\text{Type}}(b,c) < \text{diss}_{\text{Type}}(b,f) < \text{diss}_{\text{Type}}(a,g)$
- $\text{diss}_{\text{Type}}(d,e) > \text{diss}_{\text{Type}}(b,d)$

Fig. 8. Constraints on the dissimilarity over conceptual types

The main idea, is that the difference between two concepts is processed according to the number of edges that separate them from their nearest common parent. Furthermore, the deepest this common parent is in the lattice of types, the smallest the difference is between the two compared types.

The difference between two types with a nearest common parent of a depth `d` in the type lattice is always smaller than the difference between two types with a nearest parent of depth of `d-1`, whatever the number of edges between the types and their parents is.

As an illustration, looking at the figure 8, we want to have the following inequalities:

- $\mathrm{diss}_{\mathrm{type}}(d,e) < \mathrm{diss}_{\mathrm{type}}(b,c) < \mathrm{diss}_{\mathrm{type}}(b,f) < \mathrm{diss}_{\mathrm{type}}(a,g)$
- $\mathrm{diss}_{\mathrm{type}}(d,e) > \mathrm{diss}_{\mathrm{type}}(b,d)$

Definition 5.1. *The dissimilarity* $\mathrm{diss}_{\mathrm{type}} : T_C^2 \rightarrow [0,1]$*, where* T_C *is a set of conceptual types is defined as follows:*
$\forall(t_1,t_2) \in T_C \times T_C$,

$$\mathrm{diss}_{\mathrm{type}}(t_1,t_2) = \sum_{t_i \in \langle t,t_1 \rangle, t_i \neq t} 2^{-2-\mathrm{depth}(t_i)} + \sum_{t_i \in \langle t,t_2 \rangle, t_i \neq t} 2^{-2-\mathrm{depth}(t_i)}$$

with

- $t \in T_C$ *the nearest common parent of* t_1 *and* t_2
- $\langle t,t' \rangle$ *is the shortest path between t and* t'
- $\mathrm{depth}(t_i)$ *is the depth of* t_i *in the type hierarchy, with* $\mathrm{depth}(Entity) = 0$.

5.3.1.2 Similarity between two referents

The similarity between the values of two concepts depends on the application domain and the data type used to express the individual markers. Therefore, several similarity measures between referents are defined.
If, at least, one of the referents of the concepts is undefined, the value of the similarity is equal to 1.

Definition 5.2. $\forall(c_1,c_2) \in \mathcal{C}^2$, $c_1 = [t_1 : v_1]$, $c_2 = [t_2 : v_2]$ *and* $(t_1,t_2) \in T_C \times T_C$,

$$\mathrm{sim}_{\mathrm{ref}}(v_1,v_2) \begin{cases} = 1 \text{ if } v_1 \text{ or } v_2 \text{ is undefined.} \\ = \mathrm{sim}_{\mathrm{ref}strings}(v_1,v_2), \\ \text{or } \mathrm{sim}_{\mathrm{ref}num}(v_1,v_2), \\ \text{otherwise.} \end{cases}$$

With $\mathrm{sim}_{\mathrm{ref}strings}$ *and* $\mathrm{sim}_{\mathrm{ref}num}$ *two similarity measures for referents described hereafter.*

In the next sections, we define only the measures used within the case study. For a detailed definition and description of the other measures, see Laudy (2010).
Similarity of "String" referents
The idea of $\mathrm{sim}_{\mathrm{ref}string}$ is to say that, if one of the strings contains a large part of the other one, the two strings should be declared sufficiently similar in order to be fused. The measure relies on the proportion of substrings shared between the two referents, regarding the length of the smallest one.

Definition 5.3. $\mathrm{sim}_{\mathrm{ref}strings} : \mathbb{S}^2 \rightarrow [0,1]$ *is defined as follows:*
$\forall(s_1,s_2) \in \mathbb{S}^2$*, where* $\mathbb{S}$ *is the set of all strings,*

$$\mathrm{sim}_{\mathrm{ref}strings}(s_1,s_2) = \frac{\mathrm{lengthComSubString}(s_1,s_2)}{\min(\mathrm{length}(s_1),\mathrm{length}(s_2))}$$

where

- $\min$ *is such that*
  $\min : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$
  with $\forall (x, y) \in (\mathbb{N} \times \mathbb{N})$ *such that* $\min(x, y) = y$ *if and only if* $x \geq y$ *and* $\min(x, y) = x$ *in other cases.*
- length *is defined as follows:*
  $\text{length} : \mathbb{S} \to \mathbb{N}$ *such that* $\forall s \in \mathbb{S}, \text{length}(s) = x$*, with x the number of characters in s and where* $\mathbb{S}$ *is the set of all possible strings.*
- $\text{lengthComSubString}$ *is defined as follows:*
  $\text{lengthComSubString}$:
  $$\mathbb{S}^2 \to \mathbb{N}$$
  $$(s, s') \to \sum_i \text{length}(s_i^*)$$
  with $s_i^* \in S$ *such that the four following conditions are fulfill:*
  1. s_i^* *is a substring of both s and* s'
  2. s_i^* *contains at least two characters*
  3. s_i^* *is maximal (i.e. there is no other string that fulfill the conditions and is longer)*
  4. *the order in which the substrings appear in the two strings is preserved*

As an illustration of this second measure, let us consider two titles. Their similarity is the following one.
$\text{sim}_{\text{ref}_{strings}}(\text{“}The\ news\text{''}, \text{“}News\text{''})$

$$\begin{aligned} &= \frac{\text{lengthComSubString}(\text{“}The\ news\text{''},\text{“}News\text{''})}{\text{length}(\min(\text{“}The\ news\text{''},\text{“}News\text{''}))} \\ &= \frac{4}{4} \\ &= 1 \end{aligned}$$

Similarity of dates and numerical referents
To compare the numerical referents and dates given as numerical values, we rely on a distance and a threshold, that represents the tolerance with which two values may differ atmost.

Definition 5.4. *Let t be the threshold defined by the end user.* $t \in \mathbb{R}^{+*}$. $(v_1, v_2) \in \mathbb{R}^2$ *are two numerical values that must be compared.*
The function $\text{sim}_{\text{ref}_{num}} : \mathbb{R}^2 \to [0, 1]$ *is defined as follows:*

$$\text{sim}_{\text{ref}_{num}}(v_1, v_2) = \begin{cases} 0 \ \textit{if}\ |v_1 - v_2| \geq t \\ 1 - \frac{|v_1 - v_2|}{t} \ \textit{otherwise} \end{cases}$$

5.3.1.3 Similarity regarding the context of the concepts

In order to compare the context in which the two concepts are expressed, we propose to compare their immediate neighborhood. Intuitively, the similarity measure of two concepts regarding their context is processed by measuring the proportion of relations linked to the concepts and that have the same type and the proportion of relations that have different types.

Definition 5.5. *The similarity of a node* c_1 *of the graph* G_1 *and the node* c_2 *of the graph* G_2*, regarding their neighborhood is given by the function* $\text{sim}_{\text{context}} : \mathcal{C}^2 \to [0, 1]$ *defined as follows.*
Let R_1 *(respectively* R_2*) be the set of relations neighboring the concept node* c_1 *(respectively* c_2*). We define* $R_1^{\varnothing}$ *(respectively* $R_2^{\varnothing}$*), the union of the set* R_1 *(resp.* R_2*) and set containing the empty element noted* $\varnothing$*.*
Let $\mathcal{R}$ *be a symmetric relation between the elements of* $R_1^{\varnothing}$ *and* $R_2^{\varnothing}$ *such that*

- *The types of the relations are equals:* $x\mathcal{R}y \Leftrightarrow l(x) = l(y)$
- *one element of* R_1 *is related to at most one element of* R_2*:* $\forall x \in R_1, (\exists! y \in R_2,$ *such that* $x\mathcal{R}y) \vee x\mathcal{R}\varnothing$
- *one element of* R_2 *is related to at most one element of* R_1*:* $\forall y \in R_2, (\exists! x \in R_1,$ *such that* $x\mathcal{R}y) \vee \varnothing\mathcal{R}y$

To define the similarity measure regarding the context of two concepts, we use the two sets INTERSEC *and* COMPL*, defined as follows as follows.*
INTERSEC *is a set of couples of relations nodes of* R_1 *and* R_2 *that are related through the* $\mathcal{R}$ *relation.*
Let $\text{INTERSEC} = \{(x,y) \in R_1 \times R_2 | x\mathcal{R}y \textit{ with } \forall (x,y) \in R_1 \times R_2, (x',y') \in R_1 \times R_2, \textit{ if } x\mathcal{R}y \wedge x'\mathcal{R}y', x = x' \Leftrightarrow y = y'\}$.
COMPL *is the set of relations that could not be related through* $\mathcal{R}$
Let $\text{COMPL} = \{(x,y) \in R_1^{\varnothing} \times R_2^{\varnothing} | \nexists y' \in R_2 \textit{ such that } (x,y') \in \text{INTERSEC} \wedge \nexists x' \in R_1 \textit{ such that } (x,y') \in \text{INTERSEC} \wedge (x = \varnothing \oplus y = \varnothing)\}$
The similarity of the context of c_1 *and* c_2 *is then defined according to the cardinality of the two sets* INTERSEC *and* COMPL*:*

$$\text{sim}_{\text{context}}(c_1, c_2) = \frac{|\text{INTERSEC}|}{|\text{INTERSEC}| + |\text{COMPL}|}$$

5.3.1.4 Similarity of concepts

To compare two concepts, now, we use a similarity measure that combines all the measures described above.
The order of importance of the component of two concepts, when processing their similarity is the following one:

1. their concept types
2. their referents
3. the context in which they are used

To account for this hierarchy of importance, within the similarity measure $\text{sim}|_{\text{gene}}$, we apply different coefficients to the individual similarity (and dissimilarity) measures: a coefficient of 4 is applied to the part accounting for the similarity of the concept types, 2 to the part accounting for the referents and 1 for the contexts. In order to keep a normalized similarity, the the similarity score processed as described above is divided by 7.

Definition 5.6. *The similarity measure* $\text{sim}|_{\text{gene}} : \mathcal{C}^2 \to [0,1]$*, where* $\mathcal{C}$ *is a set of concepts defined on the same vocabulary, is expressed as follows:*
$\forall (c_1, c_2) \in \mathcal{C}^2$ *such that* $c_1 = [t_1 : v_1]$ *and* $c_2 = [t_2 : v_2]$*,*

- *If the most specific common parent of* t_1 *and* t_2 *is the root of the type hierarchy, we have* $\text{sim}|_{\text{gene}}(c_1, c_2) = 0$.
- *Otherwise, we have* $\text{sim}|_{\text{gene}}(c_1, c_2) = \frac{4(1 - \text{diss}_{\text{type}}(t_1,t_2)) + 2 * \text{sim}_{\text{ref}}(v_1,v_2) + \text{sim}_{\text{context}}(c_1,c_2)}{7}$

where $\text{diss}_{\text{type}}$*,* sim_{ref} *and* $\text{sim}_{\text{context}}$ *are the similarity and dissimilarity measures defined above.*

5.3.2 Graph association

On figure 9, we can see an example of the need for the association phase. Graphs $g1$ and $g2$ represent two TV program descriptions that we attempted to fuse. The result of the fusion is given by the graph $g3$, which depict a badly formed TV program. Indeed, this fused TV program has two begin and two end dates. Furthermore, looking at these two TV program descriptions, it is obvious that they are not compatible and should not be fused because they describe two different TV programs. Our aim is to provide a method that will enable discriminating the observations that can be fused from the ones that are obviously not compatible thanks to the *association* phase.

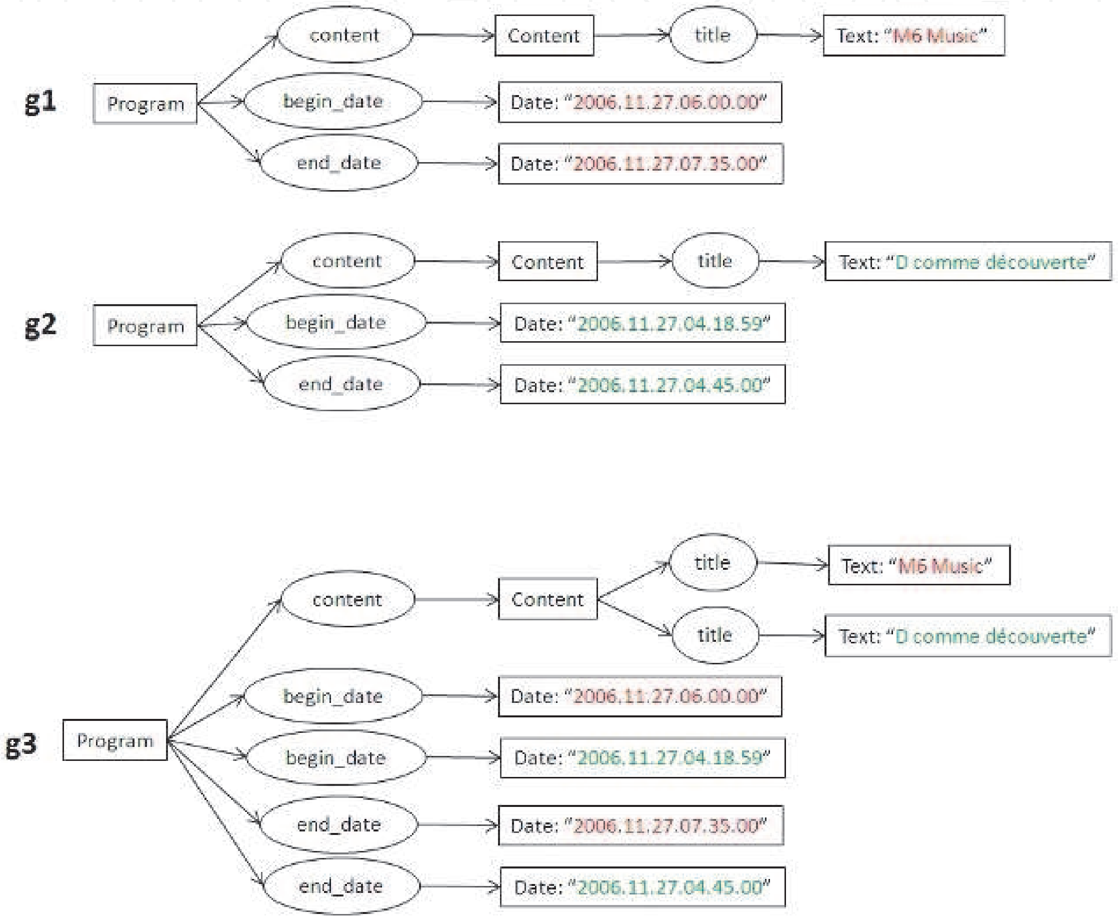

g1
Program
content
Content
title
Text: "M6 Music"
begin_date
Date: "2006.11.27.06.00.00"
end_date
Date: "2006.11.27.07.35.00"
g2
Program
content
Content
title
Text: "D comme découverte"
begin_date
Date: "2006.11.27.04.18.59"
end_date
Date: "2006.11.27.04.45.00"
g3
Program
content
Content
title
Text: "M6 Music"
title
Text: "D comme découverte"
begin_date
Date: "2006.11.27.06.00.00"
begin_date
Date: "2006.11.27.04.18.59"
end_date
Date: "2006.11.27.07.35.00"
end_date
Date: "2006.11.27.04.45.00"

Fig. 9. Incompatible graphs

Several similarity measures between general graphs and conceptual graphs have been proposed (for instance in Sorlin et al. (2003), Gandon et al. (2008) and de Chalendar et al. (2000)). Through our proposition, we focus on the local similarity of the different pairs of nodes. We propose to compute the similarity between two graphs with regards to the best matching of their nodes. Intuitively, we process the similarity of two graphs by maximizing the similarity of their couples of concepts.

Definition 5.7. $\text{sim}_{\text{Graph}} : \mathcal{G}^2 \to [0,1]$, *where $\mathcal{G}$ is a set of graphs defined on the same vocabulary, is the function defined as follows:*
Let G_1 and G_2 be two graphs to compare. C_1 (resp. C_2) is the set of concepts of G_1 (resp. G_2) and $|C_1|$ (resp. $|C_2|$) is the number of concepts in the graph G_1 (resp. G_2).
We rename G_1 and G_2 into G_1' and G_2' such that

- *if* $|C_1| \leq |C_2|$, $G'_1 = G_1$ *and* $G'_2 = G_2$
- *else* $G'_1 = G_2$ *and* $G'_2 = G_1$

C'_1 *(resp.* C'_2*) is the set of concepts of* G'_1 *(resp.* G'_2*).*
$\text{sim}_{\text{Graph}}(G_1, G_2) =$

$$\frac{\sum_{c_1 \in C'_1} (\max_{c_2 \in C'_2} p(t_{1,2}) * \text{sim}|_{\text{gene}}(c_1, c_2))}{\sum_{\substack{(c_1,c_2) \in C'_1 \times C'_2 | \\ \forall c_3 \in C'_2, c_3 \neq c_2 \wedge \\ \text{sim}|_{\text{gene}}(c_1,c_3) < \text{sim}|_{\text{gene}}(c_1,c_2)}} p(t_{1,2})}$$

where $p(t_{1,2})$ *is the weight associated with the conceptual type* $t_{1,2}$ *and that allows giving more or less importance to some of the concepts, according to the application domain;*

As a matter of example, let us consider the two graphs $g1$ and $g2$ depicted on figure 10. To compute their similarity, we process the similarities of the different possible pairs of concepts. The table on figure 11 shows the similarity scores, using $\text{sim}|_{\text{gene}}$, of all the pairs of concepts that can be matched between the two graphs.

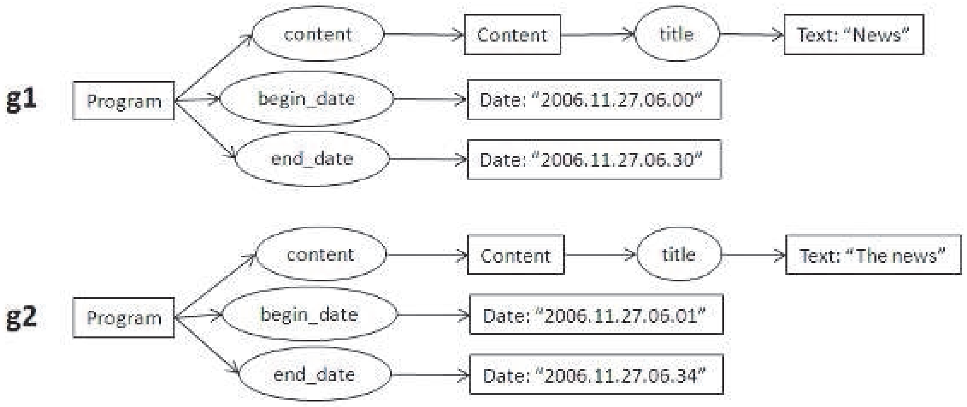


Fig. 10. Processing similarity - input graphs

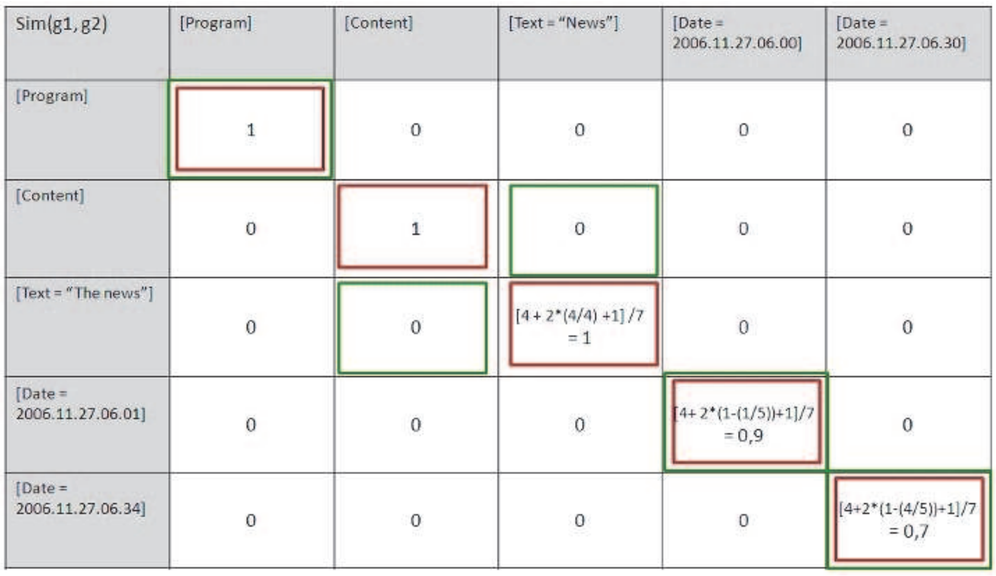
The matching surrounded in red continuous line has a similarity score of 0,92 and is the best that can be found.

5.4 Multi-source synthesis

The *multi-source information synthesis* phase relies on a fusion process that is an extension of the *maximal join* operation initially described by John Sowa (Sowa (1984)). The structures and contents of the two graphs to be fused are compared relying on homomorphism search. Redundant parts are fused (i.e. added only once) into the resulting fused graph and complementary elements are all added.

5.4.1 Projection based operations on conceptual graphs and maximal join

To fuse two graphs, we first have to find all the couples of nodes of the two graphs that represent the same parts of the TV program description. Doing so, one should look, not only at the identical nodes, but also at the ones that represent the same thing, potentially with different levels of precision. For instance, in Figure 12 the `[Program]` and `[Entity: P1]` concepts represent the same object of the world, (a TV program which referent is `P1`).



Sim(g1, g2)	[Program]	[Content]	[Text = "News"]	[Date = 2006.11.27.06.00]	[Date = 2006.11.27.06.30]
[Program]	1	0	0	0	0
[Content]	0	1	0	0	0
[Text = "The news"]	0	0	[4 + 2*(4/4) +1] /7 = 1	0	0
[Date = 2006.11.27.06.01]	0	0	0	[4+ 2*(1-(1/5))+1]/7 = 0,9	0
[Date = 2006.11.27.06.34]	0	0	0	0	[4+2*(1-(4/5))+1]/7 = 0,7

Fig. 11. Processing similarity - results

Matching the two graphs, according to these couples of nodes, should also keep the structure of the graphs. Arcs between nodes should not be deleted or modified. For instance, given that (`[Program]`, `[Entity: P1]`) and (`[Content]`, `[Content]`) are two couples of nodes that are compatible, the edge between `[Program]` and `[Content]` must have an equivalent between `[Entity: P1]` and `[Content]`, which is the case in our example. To do so, we use projection search between the two graphs.

Definition 5.8. *Let $u = (C_u, R_u, E_u, l_u)$ and $v = (C_v, R_v, E_v, l_v)$ be two basic conceptual graphs defined on the same vocabulary V. A projection of v in u is a function $P : V \times V \rightarrow (C_u \times C_v) \cup (R_u \times R_v)$ of the nodes such that the arcs and their labels are preserved and the labels of the nodes can be specialized.*

- $\forall (r_u, i, c_u) \in u, (P(r_u), i, P(c_u)) = (r_v, i, c_v) \in v$
- $\forall e \in C_u \cup R_u, l_v(P(e)) \leq l_u(e)$

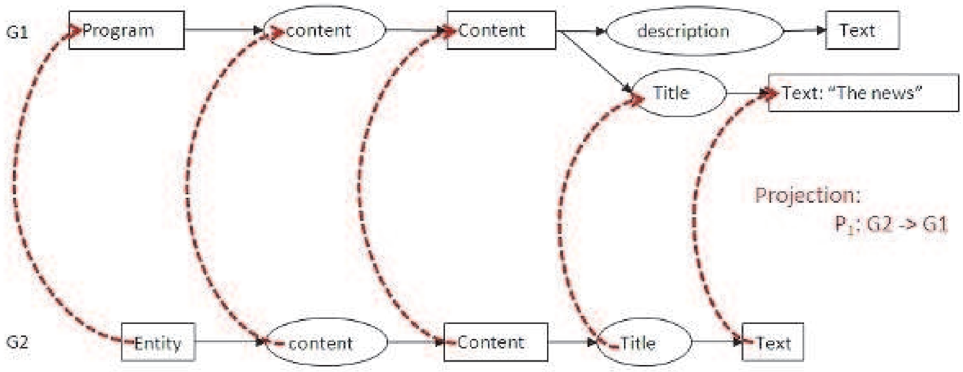


Fig. 12. Example projection between two graphs

Figures 12 depicts an example of projection. $G2$ can be projected in the graph $G1$ through the projections P_1 and $G1$ is more specific than $G2$. We use injective projections. Two different nodes of one graph have two different images in the other graph. Maximal join is a projection based operation defined on conceptual graphs.

The maximal join is composed of two components. First, it tests the compatibility of two elements of the graphs and then fuses them actually. To define the maximal join operation, Sowa defines several other operations that we detail hereafter.

Definition 5.9. *If a conceptual graph u is canonically derivable (see Sowa (2000)) from a conceptual graph v, then u is called a specialization of v and v is called a generalization of u.*

Definition 5.10. *Let two conceptual graphs $u1$ and $u2$ have a common generalization v with injective projections $P1 : v \rightarrow u1$ and $P2 : v \rightarrow u2$. $P1$ and $P2$ are* compatible projections *if, for each concept c in v, the following conditions are true:*

- *$P1(c)$ and $P2(c)$ have a common subtype,*
- *the referents of $P1(c)$ and $P2(c)$ are either equal or one of them is undefined.*

The definition of the maximal join of two graphs u1 and u2 given by Sowa in Sowa (1984) is the following one.

Definition 5.11. *Let v be the most specific common generalization of the graphs $u1$ and $u2$. There is no generalization $v2$ of $u1$ and $u2$ such as v is a sub-graph of $v2$.*
$P1$ and $P2$ are two compatible injective projections of v in $u1$ and $u2$. $P1$ and $P2$ are maximally extended ($P1$ and $P2$ are maximally extended if they have no extension).
A join on these projections is called a maximal join.

There may be several possibilities of fusion between two observations, according to which combinations of observed items are fused or not. This phenomenon is well manage by the maximal join operator. As there may exist several maximally extended compatible projections between two graphs, joining two graphs maximally may give several results, each one of them being a fusion hypothesis.
However, using the maximal join only is not sufficient in order to fuse information as it enables to fuse only strictly identical values. Figure 13 gives an example of such a case. Domain knowledge must used in order to extend the notion of compatibility between concepts, so that concepts with sufficiently similar referents can be be fused.

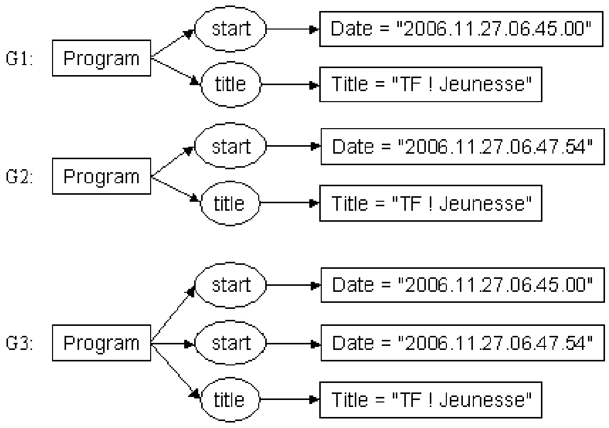


Fig. 13. Limitations of Maximal Join (1)

To do so, we use *Fusion Strategies* which are rules encoding domain knowledge and fusion heuristics.

5.4.2 Fusion strategies

Formally, the fusion strategies are expressed as rules that encompass two functions: a *compatibility testing function* that we call $\mathrm{f_{comp}}$, and a *fusion function* that we call $\mathrm{f_{fusion}}$.
For one application, we define a set of fusion strategies, each one associated with a (set of) conceptual type(s) for which the strategy is valid.

Definition 5.12. *Let c_1 and c_2 be concepts defined on the same vocabulary such that $c_1 \in G_1$ and $c_2 \in G_2$. A fusion strategy* $\mathrm{strategy_{fusion}}$ *is defined as follows:*

$$\begin{aligned} strategy_{fusion} = \;& if \;\; \mathrm{f_{comp}}(c_1, c_2) \\ & then \; \mathrm{f_{fusion}}(c_1, c_2) \\ & else \;\; \{c_1, c_2\} \end{aligned}$$

where $\mathrm{f_{comp}} : \mathcal{C} \times \mathcal{C} \rightarrow \{true, false\}$ *is a function testing the compatibility of two concept nodes, and* $\mathrm{f_{fusion}} : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$ *is a fusion function upon the concepts nodes of the graphs.*

The fusion strategies applied on two concept nodes result either in a fused concept node if the initial nodes are compatible, or in the initial nodes themselves if they are incompatible.

5.4.3 Definition of the compatibility function

A compatibility function $\mathrm{f_{comp}} : \mathcal{C}^2 \rightarrow \{true, false\}$ is defined regarding the similarity that exists between two concepts.

Definition 5.13. *Let $(c_1, c_2) \in \mathcal{C}^2$, with t_1 (respectively t_2), the conceptual type of c_1 (respectively c_2) and v_1 (respectively v_2) the referent of c_1 (respectively c_2).*
The compatibility function $\mathrm{f_{comp}}$ *is then defined as follows:*

$$\begin{aligned} \mathrm{f_{comp}}(c_1, c_2) &= \mathrm{sim_{fusion}}\, c_1, c_2 \geq t \\ &= \mathrm{sim_{type}}(t_1, t_2) * \mathrm{sim_{ref}}(v_1, v_2) \geq t \end{aligned}$$

Where $t \in [0, 1]$ is a threshold defined by the domain expert, $\mathrm{sim_{ref}}$ *is one of the similarity measures defined for referents earlier and* $\mathrm{sim_{type}}(t_1, t_2)$*defined as follows.*

To decide whether two concepts can be fused, regarding their conceptual types, one has to check whether the conceptual types have a common subtype. The fusion precises the observations. Regarding the fusion of conceptual types, this means that when fused, two conceptual types will result in their most general common subtype if this sub type exist.

Definition 5.14. *The similarity* $\mathrm{sim_{type}} : T_C^2 \rightarrow [0, 1]$ *is defined as follows:*
$\forall (t_1, t_2) \in T_C \times T_C$,

$$\mathrm{sim_{type}}(t_1, t_2) = \begin{cases} 1 \; if \; \exists t \in \mathcal{V} \; such \; that \; t \leq t_1 \; and \; t \leq t_2 \\ 0 \; otherwise. \end{cases}$$

As a matter of example, considering the fusion of TV program descriptions, let $c_1 = [t_1 : v_1]$ and $c_2 = [t_2 : v_2]$ where the most general common subtype of t_1 and t_2 is $t_{1,2}$.
The compatibility between two `Text` concepts is tested thanks to the following compatibility function:
$\mathrm{f}_{\mathrm{comp}_{\{``Text"\}}}(c_1, c_2) = \mathrm{sim_{type}}(t_1, t_2) * \mathrm{sim}_{\mathrm{ref}strings}(v_1, v_2) > 0,8$
With $\mathrm{sim_{type}}$ and $\mathrm{sim}_{\mathrm{ref}strings}$ defined above.

The compatibility between two `Date` concepts is given by the following function:
$fcomp_{\{``Date"\}}(c_1, c_2) =$
$\mathrm{sim}_{\mathrm{type}}(t_1, t_2) * \mathrm{sim}_{\mathrm{ref}_{num}}(v_1, v_2) > 0$
and

- $\mathrm{sim}_{\mathrm{ref}_{num}}(v_1, v_2) = 0$ if $|v_1 - v_2| \geq 5$
- $\mathrm{sim}_{\mathrm{ref}_{num}}(v_1, v_2) = 1 - \frac{|d_1 - d_2|}{5}$ otherwise

With $\mathrm{sim}_{\mathrm{type}}$ defined in 5.14, page 20 and $\mathrm{sim}_{\mathrm{ref}_{num}}$ defined above.
v_1 and v_2 are numeric values expressing the dates in numbers of minutes. For instance, the 27th of November 2006 at 6.45 am, written "2006.11.27.06.45.00" in the figures depicting the example graphs is expressed as: 200611270645.

5.4.4 Definition of the fusion function

The fusion function allows, for any couple of concept nodes, to process, if it exists, the concept node resulting from the fusion of the two initial nodes:

Definition 5.15. *The function* $\mathrm{f}_{\mathrm{fusion}} : \mathcal{C}^2 \to \mathcal{C}$ *is defined as follows:*

$$\mathrm{f}_{\mathrm{fusion}}(c_1, c_2) = c$$

where $c \in \mathcal{C}$ is the concept that results from the fusion of c_1 and c_2.

For instance, when fusing two "Text" concepts, we may choose to keep the longest of the two compatible string values.

5.4.5 Extension of the maximal join operation

The fusion strategies are used to extend the maximal join operation that was initially defined by Sowa. The notion of compatibility between two concept nodes is extended and the construction of the joint (i.e. fused) concepts is also modified, allowing to use the fusion function. We call this extension "*maximal join given a fusion strategy*".

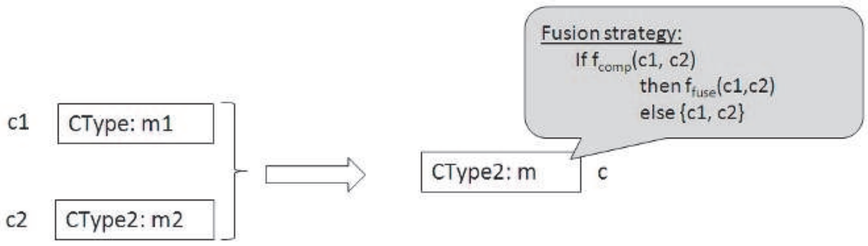


Fig. 14. Compatible concepts given a fusion strategy

Two concepts are compatible (Figure 14) if

- they have a most general common sub-type, and
- their values conform this most general common sub-type,
- they are compatible given the selected compatibility function.

The fusion of two concepts is a concept with their most general common sub-type as type, and the result of the fusion function applied to their values as value.
To process the compatibility of two relation nodes, we consider their types and the neighboring concepts. Their types must be identical, and the concepts must be compatible pair wise, respecting the labels of the edges that link them to the relations.

To compute the extended maximal join of two graphs, we have to find compatible sub-graphs of the two graphs that are maximally extended in terms of the number of their nodes. The compatibility of the two subgraphs is processed according to the compatibility of their concepts and relation nodes. Then, the compatible subgraphs are fused.

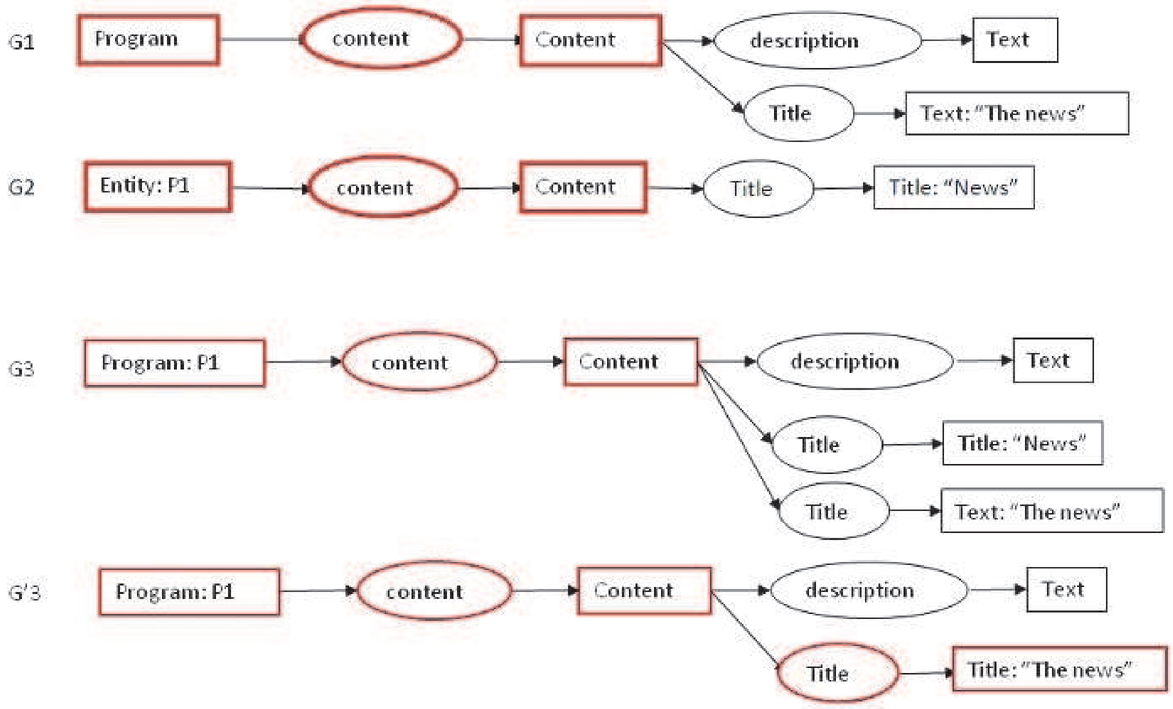


Fig. 15. Compatible relations (Extended maximal join)

On the example depicted on figure 15, we try to fuse the graph $G1$ and $G2$. We can can see that, according to the initial definition of compatibility between concepts and relation nodes, the subgraphs of $G1$ and $G2$ composed of `Program` and `Content` concepts, linked through a `content` relation on the one hand and `Entity` and `Content` concepts, linked through a `content` relation on the other hand are compatible. The result of the maximal join is thus the one depicted on graph $G3$.

When looking at the "News" and "The news" titles of the two programs depicted on figure 15 and given the remaining elements of the two descriptions, one would like to fuse the titles. Indeed, they "obviously" represent the same title and the descriptions are related to the same TV program. By including domain knowledge thanks to the compatibility testing function, we obtain as compatible subgraphs the two previous ones, with the titles in addition. The result of the fusion of $G1$ and $G2$ using maximal join extended with the fusion strategies defined in the examples above gives the graph $G'3$.

6. Experimentations

We describe here experiments that were conducted in order to validate the usefulness of both the fusion strategies and the association phase within soft data fusion. We used information acquired within the TV program case study described earlier. The fusion platform that was developed was also used within biological experimentation and intelligence applications.

6.1 Experimentation protocol

We fused TV program descriptions acquired on the DVB stream of meta data and an on-line TV magazine, from sixteen TV channels during 24 hours.

We used the TV program descriptions provided by the *INAthèque* as reference data to evaluate our fusion. The INA records the descriptions of all the programs broadcast on the french TV and radio. Thereby, we know whether a fused program corresponds to the program that was really played.

For our experimentation, we request every 5 minutes the two sources of information to give us the next program scheduled on one channel. The two provided TV program descriptions are then fused using one of the fusion strategies.

After fusion, we compare the fused TV program descriptions to the INA reference data. If the titles, subtitles, channels etc. are compatible, the fused program description is considered to be correctly found with regards to reality. The results that we obtained are detailed in the following sections.

6.2 Fusion strategies

The quality of the fusion that we obtained using different strategies was measured. To this aim, we launched our experimentations using the fusion platform first combined with no strategy and then with three different ones. The first experiment -no fusion strategy- is equivalent to using the initial maximal join operator for information fusion.

The strategies that encode domain knowledge are the following ones:

Strategy 1 extends dates compatibility. Two dates are compatible if the difference between the two is less than five minutes. If two dates are compatible but different, the fused date should be the earliest one if it is a "begin date" and the latest one otherwise.

Strategy 2 extends dates and titles compatibility. The dates compatibility is the same as for strategy 1. Two titles are compatible if one of them is contained in the other one. If two titles are compatible but different, the fused title should be the longest one.

Strategy 3 extends dates and titles compatibility. The dates compatibility is the same as for strategy 1. Two titles are compatible if the length of the common substrings exceeds a threshold. If two titles are compatible but different, the fused title should be the longest one.

6.3 On the usefulness of fusion strategies

As first interpretation, we compared the percentage of programs that were correctly found after fusion, to the reference data, and looked at the variations resulting of the use of the different strategies. Figure 16 shows the results that we obtained on a representative selection of TV channels. As expected, we can see that the fusion of observations using the maximal join operation only is not sufficient. Only the descriptions with strictly identical values are fused. Applying the three previously cited fusion strategies, we can see that the more the compatibility constraints between two values are relaxed, the better the results are. It is equivalent to inject more and more domain knowledge in the fusion process.

The different experimentations that we carried out showed that the quality of the fusion process is heterogeneous, according to several parameters. One of these parameters on which the fusion results can be dependent, is the period of the day and the specificity of the channel. For non-popular channels (BFM...) and at periods of low audience (early morning), we observed a lot of errors in the programs given by the TV magazine.

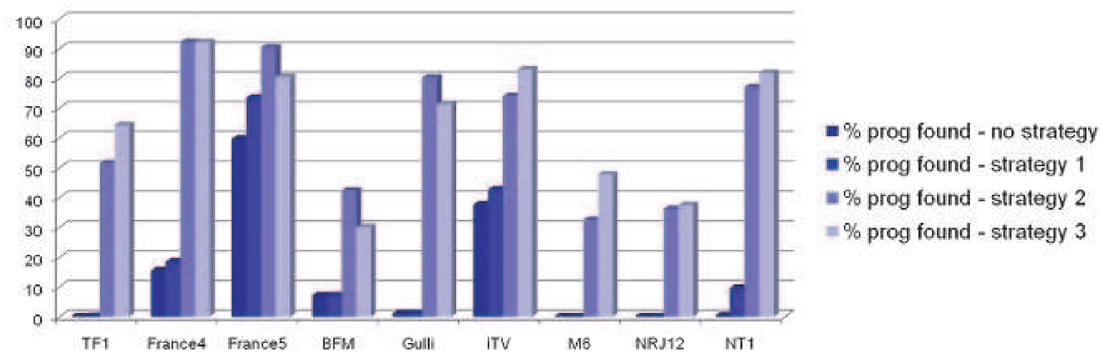


Fig. 16. Percentage of programs correctly fused and identified with different strategies

6.4 On the usefulness of association

The results of the fusion of TV programs that are scheduled on periods of low audience are very bad. Among other particularities, we observed that the TV magazine has "holes", especially for non-popular channels. During such periods, as next program to be broadcast, the magazine source of information gives a program that will actually be broadcast several hours after, whereas, the DVB gives the real next one.

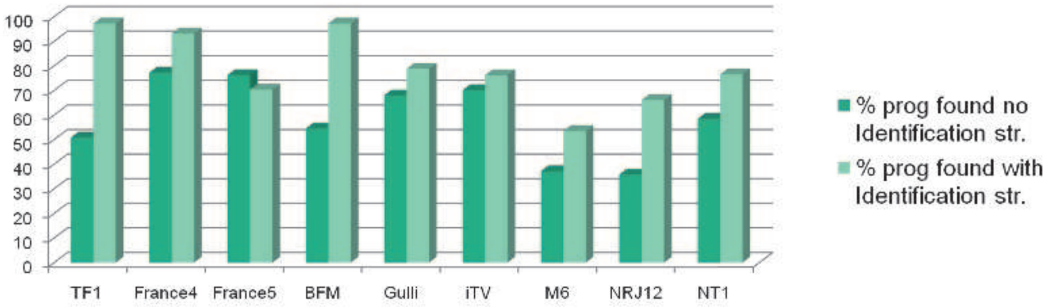


Fig. 17. compatibility testing

The two descriptions are then incompatible and the resulting fused program is not well formed (it has two different titles or begin dates for instance). To overcome such problems, we introduced the use of the association phase.

Figure 17 shows the different percentages of program correctly found, first without association, then using one based on title similarity and distance between begin and end times.

7. Conclusion and future work

We studied the issue of soft data fusion. The aim is to propose a generic approach and a generic framework for high level information fusion. The information items represent the descriptions of (part of) complex situations that themselves contain several actors or objects linked together through semantic relationships.

Besides, the proposed framework enables fusing informations items coming from several sources. Discrepancies among pieces of information are studied and we detect that two lightly different pieces of information concern the description of the same situation, and then choose what the fused description should look like.

We focused on three aspects regarding information fusion: the modeling of the situation of interest, the association phase, which aims at deciding whether two observations concern the same real world situation or not, and the information synthesis phase, where compatible observations of a single real situation are fused.

Regarding situation modeling, we showed that the conceptual graphs formalism could be used in order to represent situations of interest that have to be monitored.

The association phase relies on the use of similarity measures between graphs structures. Some parts of the measures are generic whatever the application domain is. Other components must be customized either by using specific similarity measures, or thanks to thresholds and weights. The measures we propose take into account the similarity of the values or referents of the concepts.

The information synthesis phase relies on the use of the maximal join operation defined on the conceptual graphs structures. We adapted this operation, that was initially proposed by John Sowa in Sowa (1984), by relaxing the constraints during the similarity testing of two concept nodes in the fusion process. Through this information synthesis step, we tackle the problem of soft data fusion and take into account the issue of discrepancies between the different information items. We use both a compatibility testing and a fusion functions inside the maximal join operation.

Finally, we show the usefulness of our proposition within a real application. We described how we propose to take the advantage of information fusion within a TV program recommendation system.

8. References

Bosse, E., Roy, J. and Wark, S. (2007). *Concepts, Models, and Tools for Information Fusion*, Artech House Publishers.

Buford, J., Lewis, L. and Jakobson, G. (2008). Insider threat detection using situation-aware mas, *In Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, pp. 212–219.

Chein, M. and Mugnier, M.-L. (1992). Conceptual graphs: fundamental notions, *Revue d'Intelligence Artificielle* 6(4): 365–406.

Chein, M. and Mugnier, M.-L. (2008). *Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs*, Advanced Information and Knowledge Processing, Springer.

de Chalendar, G., Grau, B. and Ferret, O. (2000). Conceptual graphs generalization, *RFIA 2000. 12ème Congrès Francophone AFRIT-AFIA*, Paris.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems, *Human Factors: The Journal of the Human Factors and Ergonomics Society* 37: 32–64(33).
URL: *http://www.ingentaconnect.com/content/hfes/hf/1995/00000037/00000001/art00004*

Gandon, F., Corby, O., Diop, I. and Lo, M. (2008). Distances semantiques dans des applications de gestion d'information utilisant le web semantique, *Proc. Workshop Mesures de similarites semantique, EGC*, INRIA Sophia Antipolis - Mediterranee.

Godbillon-Camus, B. and Godlewski, C. J. (2007). Credit risk management in banks: Hard information, soft information and manipulati, Munich Personal RePEc Archive.
URL: *http://mpra.ub.uni-muenchen.de/*

Hall, D. and Llinas, J. (2001). *Handbook of Multisensor Data Fusion*, CRC Press.

Laskey, K. (2008). Probabilistic ontologies for knowledge fusion, *In Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, pp. 1402–1409.

Laudy, C. (2010). *Introducing semantic knowledge in high level information fusion*, PhD thesis, Université Pierre et Marie Curie.

Matheus, C., Kokar, M. and Baclawski, K. (2003). A core ontology for situation awareness, *6th International Conference on Information Fusion*, Cairns, Queensland, Australia, pp. 545–552.

MIP (2005). Joint c3 information exchange data model (jc3iedm main), *Technical report*, Greding, Germany.

Peirce, C. S. (1932). *Collected papers of Chalres Sanders Peirce*, Vol. 2, C. Hartshone & P. Weiss (Eds.), Cambridge, MA.

Petersen, M. (2004). Information: Hard and soft.
URL: *www.kellogg.northwestern.edu/faculty/petersen/htm/papers/softhard.pdf*

Pravia, M., Prasanth, R., Arambel, P., Sidner, C. and Chong, C.-Y. (2008). Generation of a fundamental data set for hard/soft information fusion, *In Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, pp. 1621–1628.

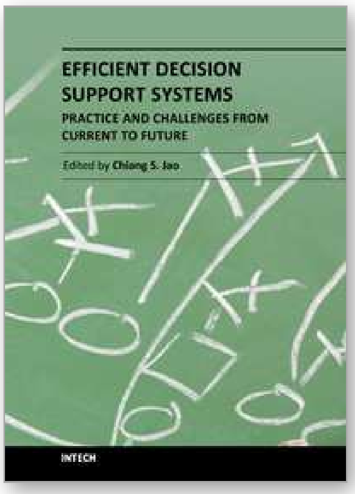
Sambhoos, K., Llinas, J. and Little, E. (2008). Graphical methods for real-time fusion and estimation with soft message data, *In Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, pp. 1621–1628.

Sorlin, S., Champin, P.-A. and Solnon, C. (2003). Mesurer la similarité de graphes étiquetés, *9èmes Journées Nationales sur la résolution pratique de problèmes NP-Complets*, pp. 325–339.

Sowa, J. (1976). Conceptual graphs for a data base interface, *IBM Journal of Research and Development* 4(20): 336-357.

Sowa, J. F. (1984). *Conceptual Structures. Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA.

Sowa, J. F. (2000). Ontology, metadata, and semiotics, *in* Springer-Verlag (ed.), *ICCS'2000: Conceptual Structures: Logical, Linguistic, and Computational Issues*, Vol. 1867/2000 of *LNAI*, Darmstadt, Germany, pp. 55–81.


EFFICIENT DECISION SUPPORT SYSTEMS
PRACTICE AND CHALLENGES FROM CURRENT TO FUTURE
Edited by Chiang S. Jao
INTECH

Efficient Decision Support Systems - Practice and Challenges From Current to Future
Edited by Prof. Chiang Jao

ISBN 978-953-307-326-2
Hard cover, 542 pages
Publisher InTech
Published online 09, September, 2011
Published in print edition September, 2011

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821