Modelling the travelling domain from a NLP description with TERMINAE

Nathalie Aussenac-Gilles (*), Brigitte Biébow (**) et Sylvie Szulman (**)

(*) IRIT, Université Toulouse 3, 118, route de Narbonne,
31062 TOULOUSE Cedex 4,
http://www.irit.fr, Nathalie.Aussenac-Gilles@irit.fr
(**) LIPN, Université Paris 13, Av. J.B Clément, 93430 VILLETANEUSE, http://www.lipn.univ-paris13.fr,
{Brigitte.Biebow, Sylvie.Szulman}@lipn.univ-paris13.fr

1. General TERMINAE method

First of all, TERMINAE proposes together a method and the tool supporting the method to build ontology from texts. The method relies on a linguistic analysis of the texts with the help of several natural language processing tools. We generally use two tools, one for term and relation identification, called SYNTEX [Bourigault,02], and another one for relation or role identification, called Caméléon [Ségéula, 99]. Both of these tools rely on the same linguistic hypothesis : the meaning of words and phrases is specific to a domain and can be inferred by observing the regularities of their use (in documents for instance).

The text here is too short (one page) to use these linguistic tools because they exploit repetition in the use of words or phrases. Nevertheless, we have used a term extractor (Syntex) that provides the list of all possible words and phrases available in the text, some relations between them (syntactical and grammatical dependencies), a direct access to all their occurrences as well as statistics such as their frequencies. Exploring the results of this tool is a complementary means to find out relevant concepts and knowledge.

A part of these results can be directly imported as an input in Terminae. These data are the input of the modeling process together with reading the original text. So identifying knowledge relies on two different main tasks that are carried out alternatively :

1) browsing the Syntex results to identify “important” knowledge or to decide how to represent some information according to the use of the words in the text ;
2) linear reading of the text to systematically extract as much knowledge as possible ;

Each piece of knowledge considered to be worth being integrated in the model is then represented. Terminae knowledge representation language relies on the following primitives : terminological file (for terms), generic concept (for classes), primitive concept (for instances) and role (for relations). The tool guides the various steps followed to define one of these item in the ontology.

The next stage in knowledge representation is normalization. The aim is to get to a well structured ontology, where each concept definition is justified through its relations with other concepts and comments. We suggest here to apply differentiation criteria that lead to make explicit the common and different properties of a concept with its father concept and brother concepts thanks to its roles.

The final stage is formalization in Terminae formal language, which is a kind of description logic. A classification function available in Terminae makes it possible to check the correctness of generic concept definitions. Concepts should be defined only once and have differentiating roles.

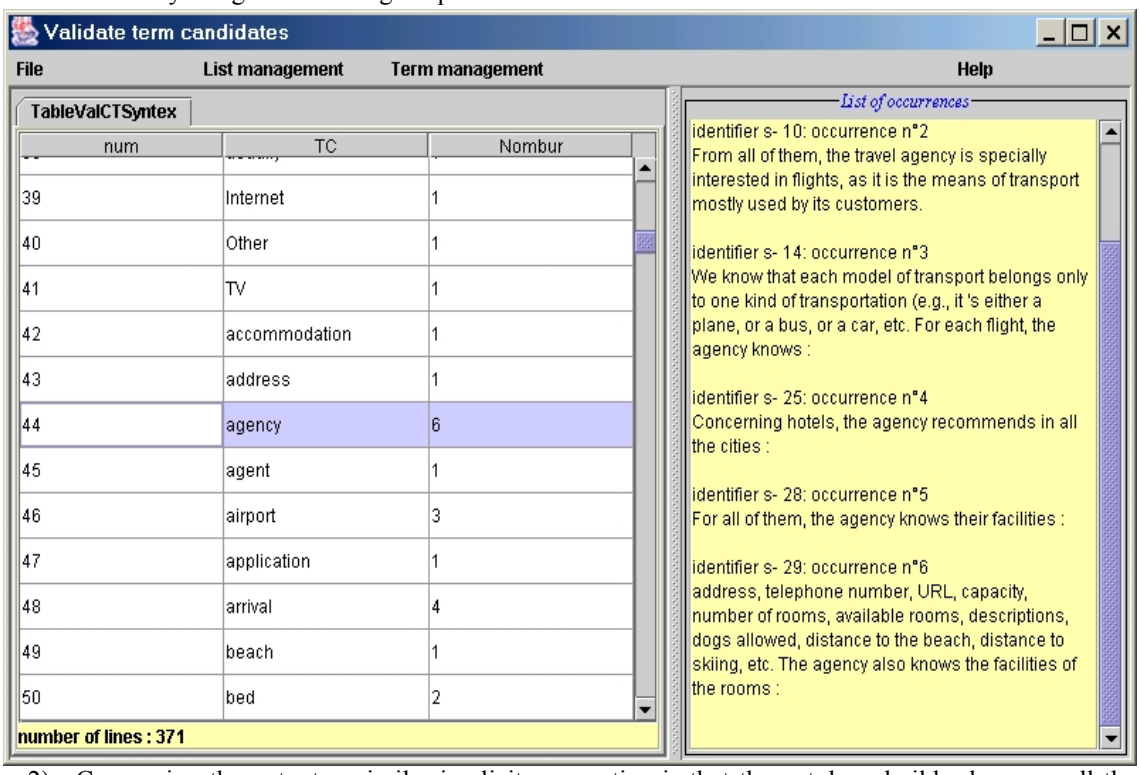
In the following, we describe how we proceed the two knowledge identification tasks, how we organize knowledge in the ontology and we illustrate the kind of consequences when applying the normalization rules. Then, we will list various modeling decisions that we took, whatever the way we identified the knowledge. TO end with, we will report some of the missing knowledge noticed by the classification function.

2. Knowledge identification tasks

2.1. Linear reading

Building concepts just from reading the texts assumes various facts :

1) The ontology builder knows enough domain knowledge to be able to decide which words (nouns, phrases, verbs or adjectives) are domain terms and possible concept or relation labels. In the particular case of this experiment, the domain is familiar to any one and common sense knowledge is almost enough to understand the text. In fact, we can suppose that it was one of the objectives of the writer, that every designer has enough expertise on the domain to model it.



2) Concerning the output, a similar implicit assumption is that the ontology builder knows well the way the ontology will be used, the task of the travel agent and how it could assisted with the help an ontology based system. This is much less obvious : in fact, we have tried to represented as much knowledge from the text as possible, without precise information about its relevance and use.

When we read the text linearly, we proceed in one the following ways :

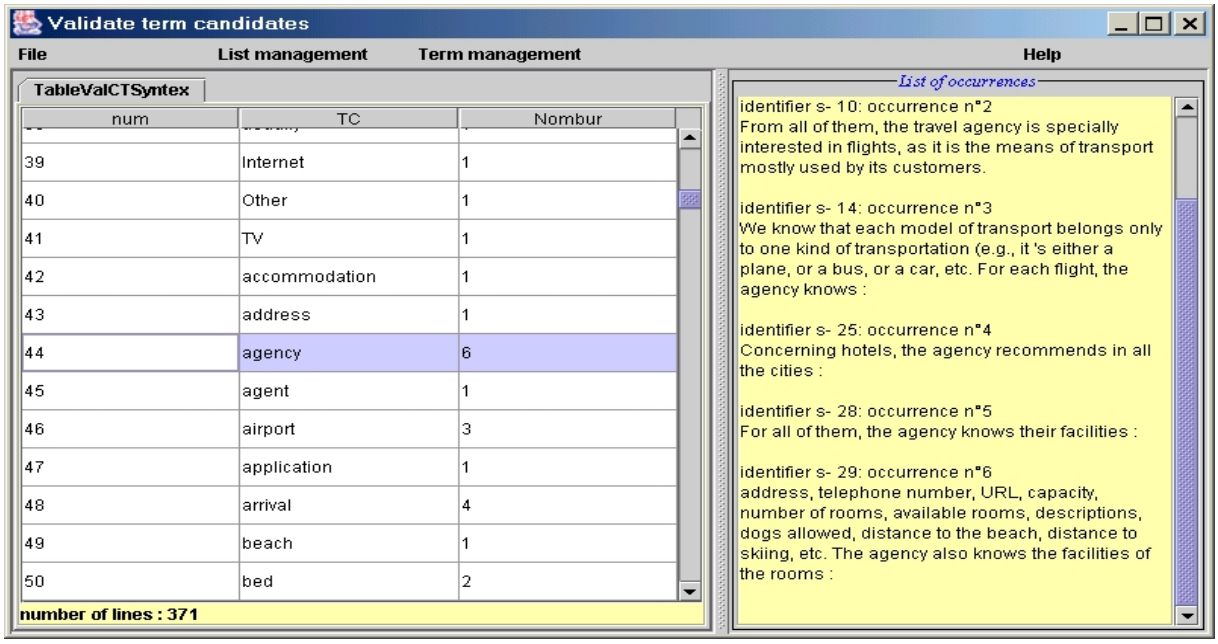
1) systematic inventory : from reading a sentence, we identify some concept names or role labels. It is frequent in this text because the writer prepares the ontology descriptions. For instance, in paragraph 2 we are suggested to define the "means of transport" concept as a class, and plane, car, ferries, trains, … as sub-classes of this concept. This leads to the definition of various concepts and IS-A relations in the ontology.
2) Structuring : some times, we use our domain knowledge to structure some information. We use it to make explicit with more abstract concepts some implicit knowledge in the texts. For instance, in the 5th paragraph about destinations, we are given a lot of examples (instances) and we are free to organize them into classes according to our mind. The same happens when deciding how to represent persons (we have two kinds of persons only : costumers (or clients) and the travel agent).

In both previous cases, the next step is knowledge representation in the ontology, with the definition of a concept (either generic or individual) or a role. See "Knowledge representation steps" below.

2.2. Browsing extracted candidate terms

Browsing Syntex results is generally much more efficient than reading when the domain is huge and the documents are numerous. For instance, if one or several books form the knowledge sources, Syntex criteria for identifying domain terms rapidly leads to find the main domain concepts and relations.

In the case of this project, the linguistic material is not prone to automatic processing. From Syntex we have obtained 372 terms (single words and phrases combing some of these words). Only 74 of them appear more than once, among 50 are relevant domain terms and some 5 or 6 refer to the domain of building ontologies. We can get read of irrelevant terms in a validation frame (see the screen dump below). For a given term, its occurrences are displayed to help decide whether to keep it or not. This work is a fastidious one, but rather fast in this case. It helps reduce the list of possible terms that will be browsed later on in the modeling process. We reduced the list down to 270 terms but many other terms could still be eliminated.



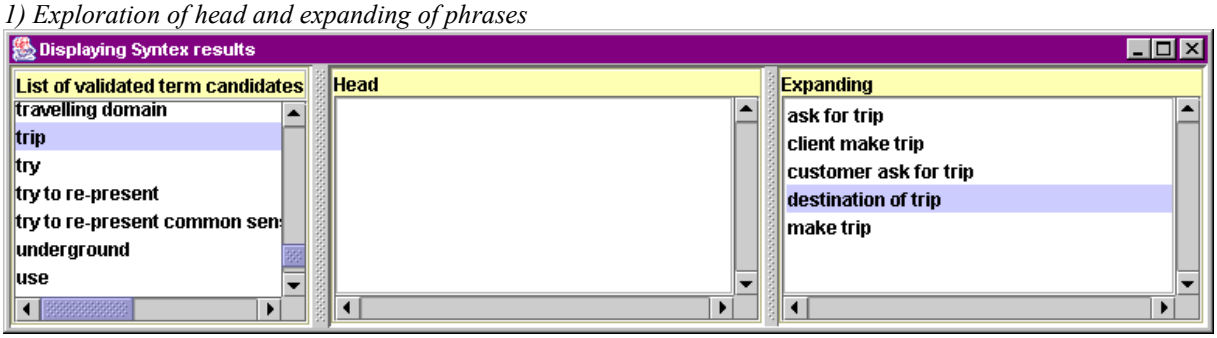
Although not completely adapted here, going from the list of possible (candidate) terms to the ontology is a good means :

1) to check the various use of a term
2) to identify synonyms (transport and transport means)
3) to automatically get comments that enrich the model and explain why some knowledge is represented in a certain way.

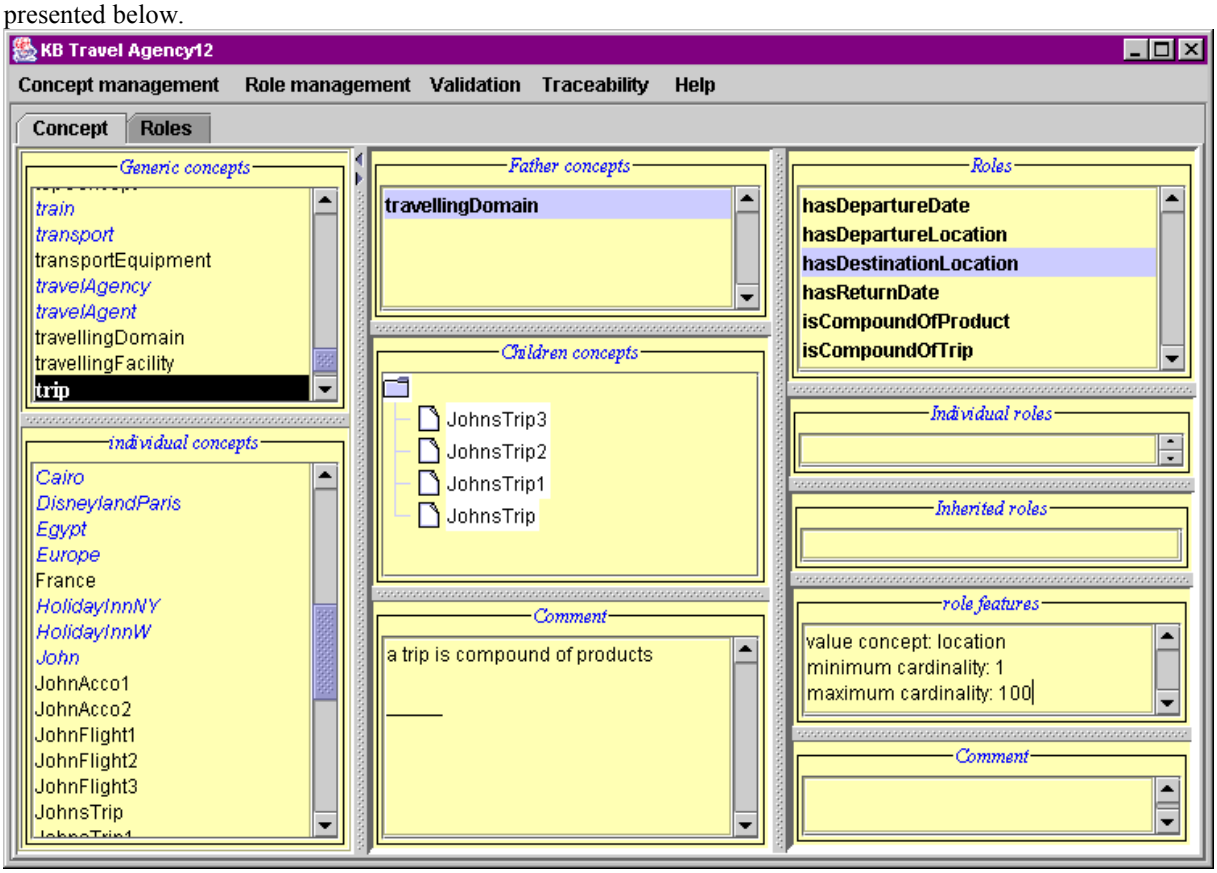
We can browse the list of possible terms according to their frequency, to the alphabetical order or (in the Syntex interface) to the grammatical category (verbs, nouns, noun phrases, verb phrases, adjectives or adverbs) and to the compositional relations (from phrases to their components, or from single words to the phrases there are used in). In this project, we did not use the Syntex interface. We carried out the following explorations :

1) looking for productive terms (that are part of many compound terms)
2) looking for the most frequent single terms: this often leads to major high level domain classes
3) looking for the most frequent noun and verb phrases : this leads to other main domain concepts and some domain relations
4) alphabetical exploration around the first identified terms.

1) Exploration of head and expanding of phrases

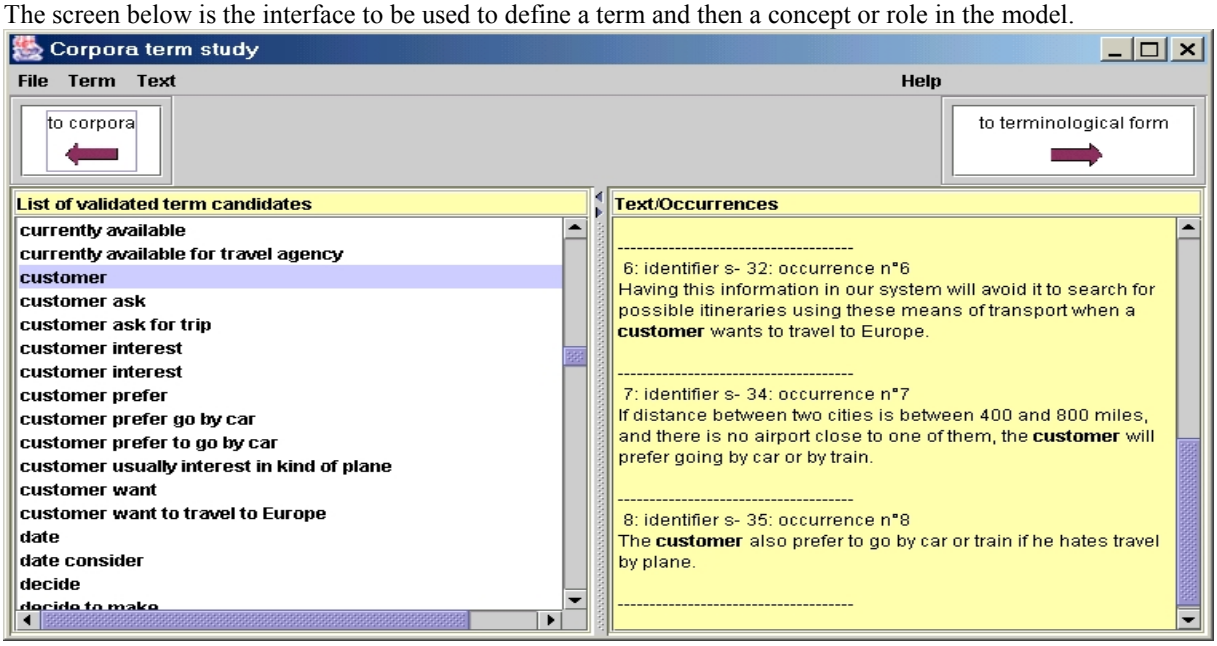


Exploring terms and their constitutive parts (head and expanding) helps to figure out their productivity. Key domain concepts are more likely to be labeled by terms that belong to various domain phrases, that is to say to productive terms. For instance, the visualization of the expanding of the "trip" term helps to define its roles as presented below.

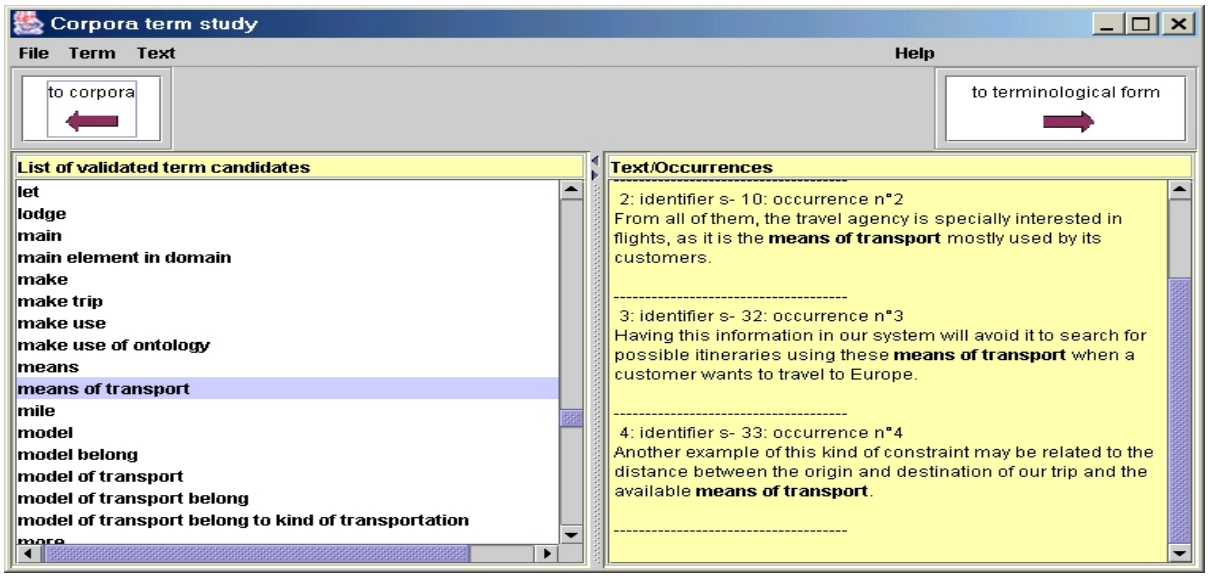


2) Exploration of the most frequent single terms

The screen below is the interface to be used to define a term and then a concept or role in the model.



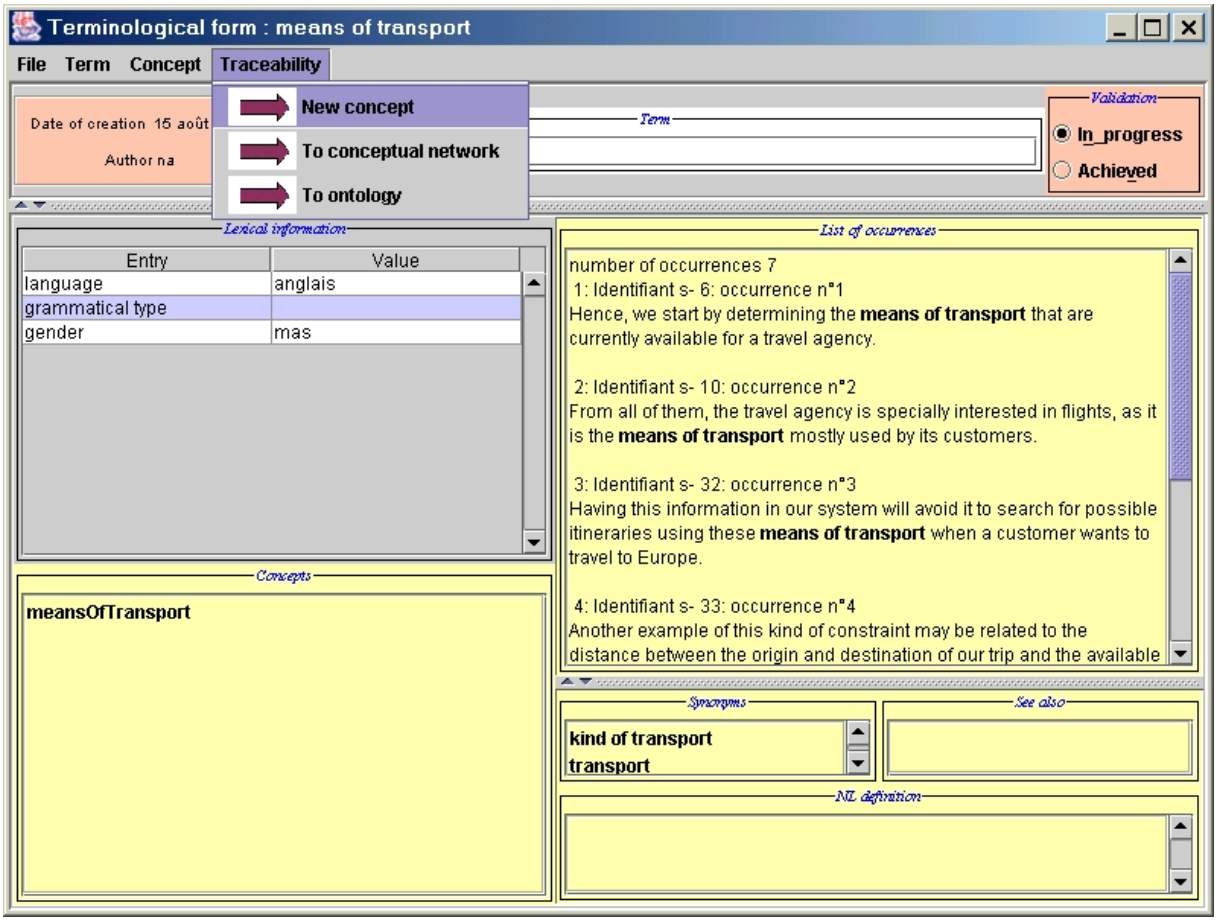
3) Exploration of the most frequent phrases



Defining a new term

When a terms in the list is considered a concept label, the ontology builder press on the "to terminological form" arrow, defines the corresponding term and then the corresponding concept. Here are the screens used for the definition a "means of transport" as a concept label.

The first screen is the terminological form where synonyms and new occurrences can be added, some linguistic information may be given. Here for instance, synonyms are "kind of transport" and even "transport". SO the occurrences of these words have been added to those of the phrase "means of transport".



From this screen, a concept can be defined in the model thanks to the option “define a concept” in the pop-up menu “Traceability”. At this time of the process, the ontology builder only knows that these words are important as domain knowledge labels, but he does not know yet how the corresponding knowledge should be represented in the ontology.

The next steps are described in the “representations steps” paragraph.

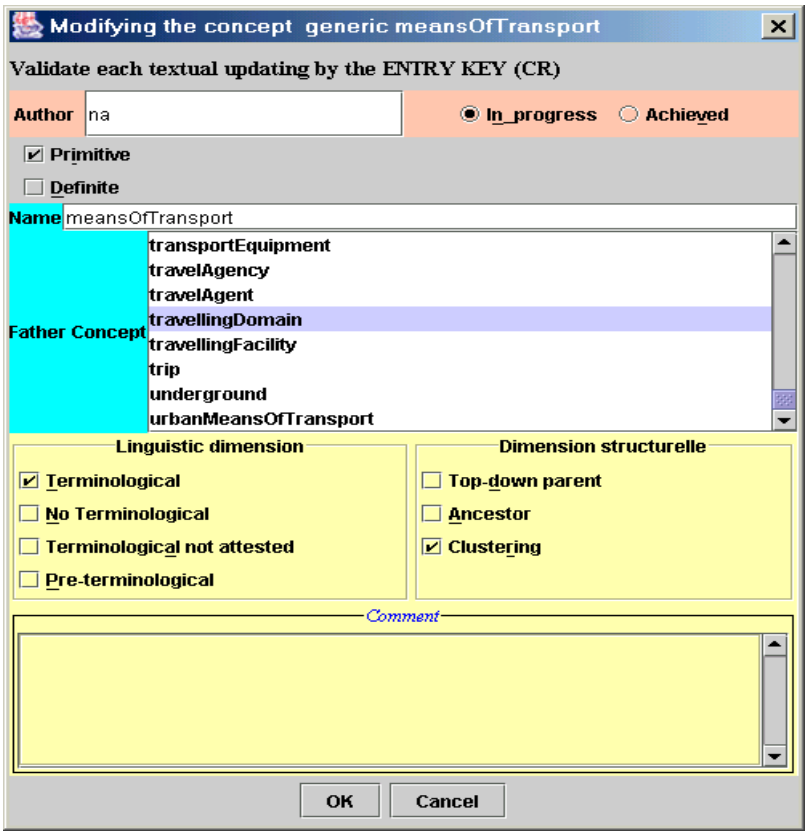
3. Knowledge representation steps

A concept may be created either from a terminological form or from the ontology editor (“create” option in the “concept” pop-up menu).

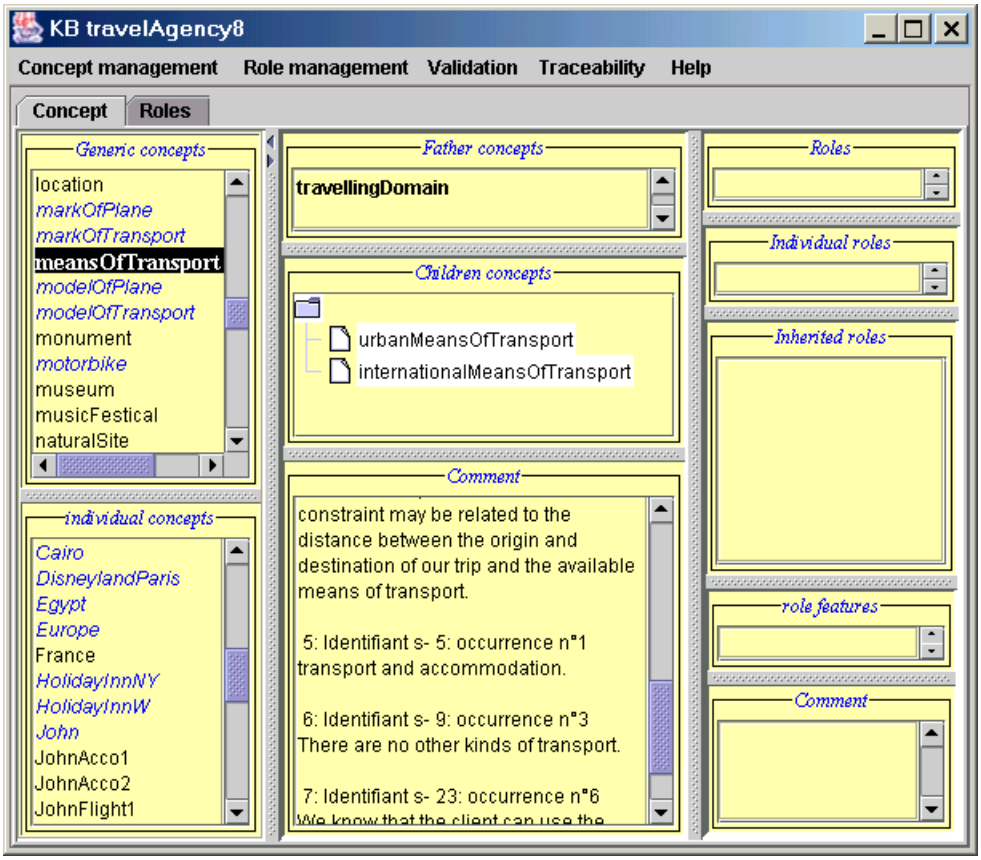
Each time a concept is created, a concept editor opens (screen bellow). The user must specify several properties :

1) The concept super class (father concept with the link isKindOf) in the hierarchy. The list of existing concepts is proposed. If the father concept has not been defined yet, the user can either enter its name and then define it or select TopConcept (the root of the hierarchy). For instance here, meansOfTransport is a class under travellingDomain. Terminae representation language allows multiple super classes and multiple inheritance of the roles. So several concepts can be selected as father in the proposed list.
2) whether the concept is terminological or not according whether it comes from the text or not. For instance, structuring concepts added from the builder own domain knowledge (e.g. country, urbanMeansOfTransport) are not terminological. This is a purely informative property that no impact on knowledge representation and formalization.
3) If the concept is built up from a terminological file, part of the occurrences may be cut and pasted to comment the concept. This help to easily store some design justification. Any other comment can be added too.
4) Whether the concept is primitive or defined. This refer to the formal representation with a description logic that is behind the Terminae interface. As long as formalization is no longer possible in Terminae, we did not check this property.

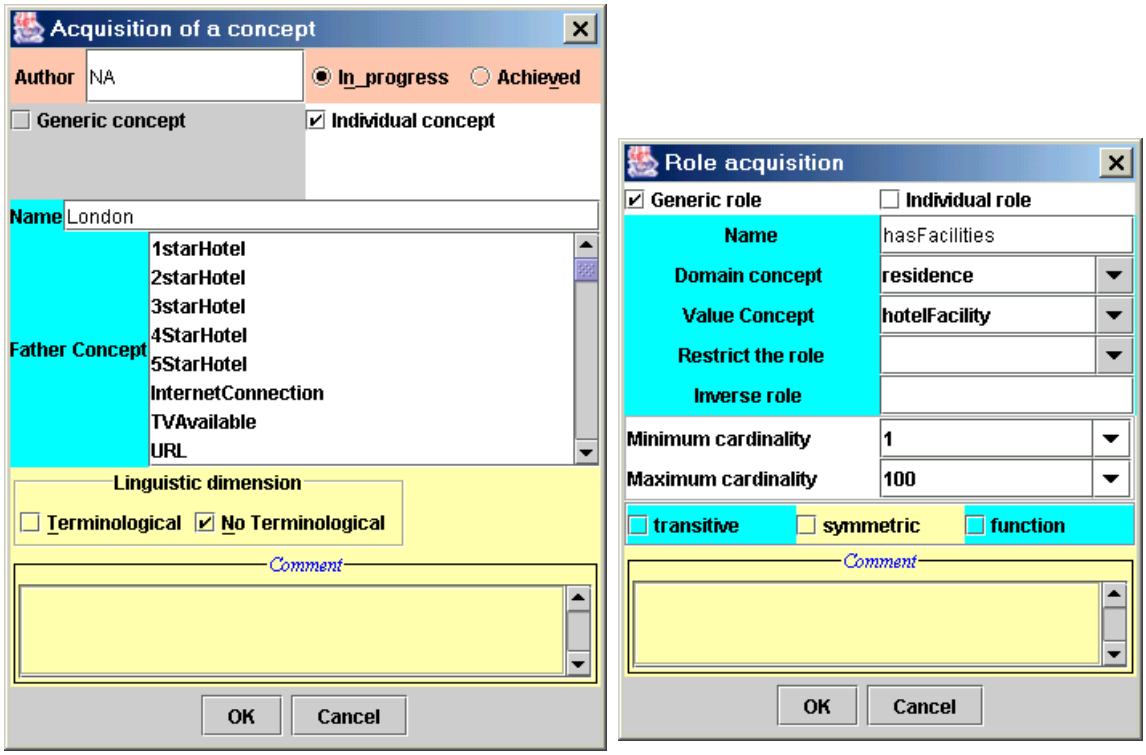
5) Whether the concept has been design following a bottom-up (ascendant) or top-down (descendant) process, or for structuring or gathering reasons. This property is also just for information. It keep tracks of one of the reasons that led to the concept definition : looking for a more generic class of various existing concept (bottom-up or gathering) or trying to list of the possible sub-classes of a given concept (top-down). For instance, the concept urbanMeansOfTransport has been characterized as a bottom-up one because it is the super class for a list of concepts (cityBus, taxi, etc.). NaturalSite is characterized as "top-down" because it is a way to list the sub classes of pointOfInterest.
6) When this first part of the definition is completed, the user can check the "OK" button. He will know later that this concept does not need to be checked again.



The concept is then inserted in the ontology and available from the ontology editor. The concept creation opens the ontology browser. The concept "meansOfTransport" is listed in the generic concept list of the ontology (left part of the screen bellow). Roles can be added later on.



Creating an individual concept (an instance) is done through a similar editor but it requires less information shown below. On the left side of the ontology editor here above, the two kinds of concepts are show in two different alphabetical lists : generic concepts are in the upper part whereas individual ones are in the bottom list.



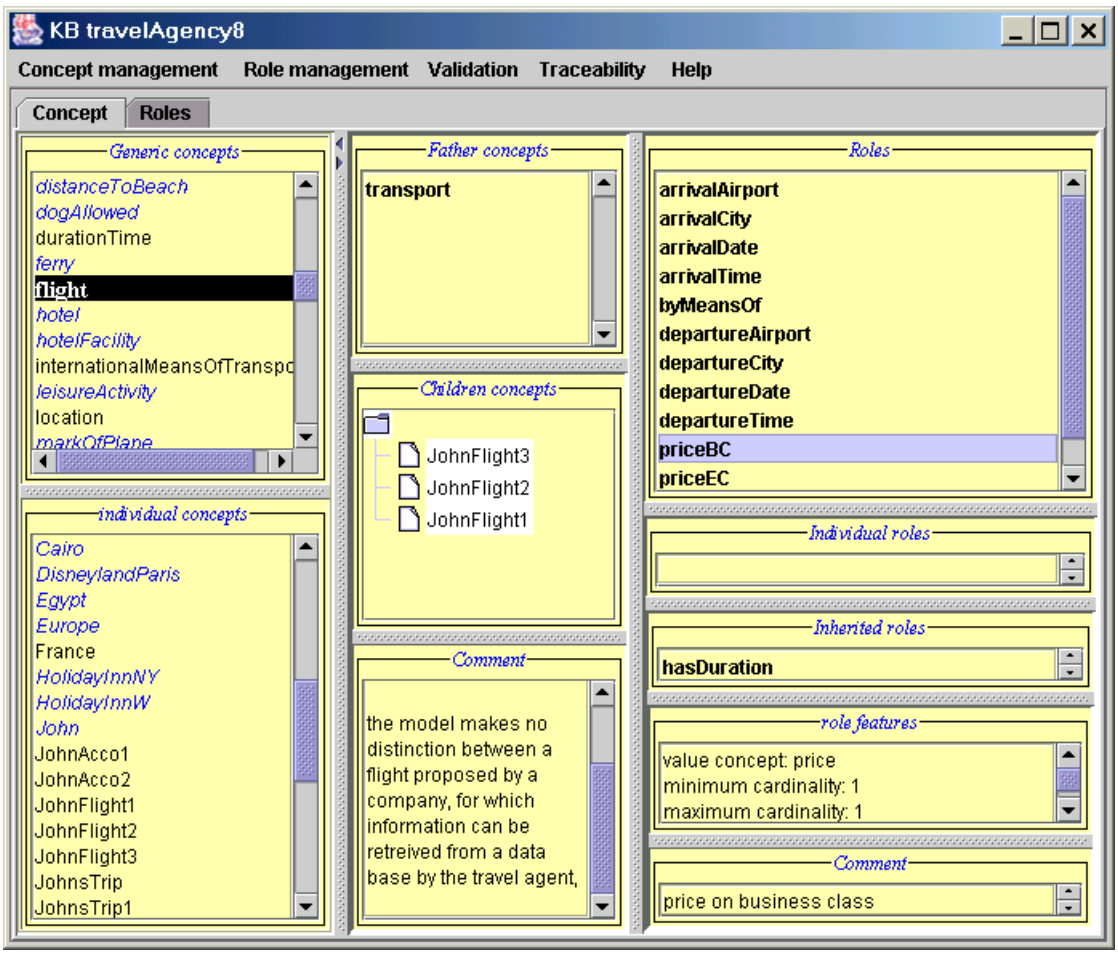
At any time, roles can be added that set relationships between concepts. The ontology builder may decide to add a role after having read the text (linear reading) or a term occurrence (when browsing terms). For instance,

reading the 6th paragraph leads to define the concept hotelFacilities and to a role that connects the concept hotel with hotelFacilities. A role can be either generic or individual. Generic roles are associated to a concept and inherited by its sub-concepts; their value concept (destination) is a generic concept. Individual roles are specific to a concept, they are no inherited and can be associated individual concepts; their value concept can be either generic or individual.

We show above the role editor. The editor proposes the list of existing concepts to select the concept value (value concept). A new concept can be defined from there if required. A role can restrict an inherited role of a more generic concept in the hierarchy. It means that the new role associated to the specific concept will have a more restricted concept as the value concept. The option « restricts role » proposes to select one of the inherited roles of the current concept. Then the value concept must be a subclass or an instance of the value concept of this inherited role.

Some of the information associated to roles (symmetry, transitivity, functional, inverse role) is not interpreted formally. The only relation between roles is the "Inverse role" relation. Cardinality indicates the minimal and maximum number of associated roles of this type that a concept may have. Cardinality is used to check that an individual concept of this class as at least zero or one or no more than one or many roles of this types towards other individual concepts.

After a concept has been assigned roles, its selection in the ontology editor makes it possible to see all of its roles and the related concepts, its comments and sub-concepts.



4. Limitations of Terminae representation language

Terminae representation language suffers from some limitations, mainly because constraints and relations between roles are not some of the primitives. Here are some of the other missing primitives :

1) operators such as OR, NOT to represent relations between concepts

2) existence operators like ONE-OF to represent a set of individuals as possible role values ; another solution would to enrich the possible types of role values : at the moment, it must be a concept, whereas in Protégé2000 for instance, it can be an instance, a value picked in a selected set or a generic type like String, Boolean, etc.
3) concrete types, as integer, string, …
4) more generally, axioms or relational expressions out of the language

For these reasons, we were not able to easily represent the kind of constraints defined in the 6th paragraph, about the way to go from America to Europe and so on. They could have been stored at least as comments.

Another limitation is that recent changes in the knowledge representation (like having individual concepts as role value even for generic concepts) make it now impossible to have a formal translation of an ontology in Terminae description logic.

5. Design decisions

In this section, we report our design decision according to the influence of the knowledge representation. Given the Terminae primitives, we still have the choice to represent an information in various ways. We motivate here some of our decisions. Some other decisions come from the application of differentiation rules. We will illustrate this normalization process in section 6.

5.1. Preliminary remarks

The text given to build the ontology intends clearly a target application which is not detailed. Although implicit, the objectives of the application lead to set some relations that would not be actually correct in a general ontology. They are acceptable because operational for the objectives. We clearly build a task ontology, not a generic one.
When representing a car as a means of transport, it is explicitly intended that it is a point of view no more detailed; if another point of view has to be taken into account, as the one of an automobile constructor, the modeling must be reconsidered.
Another illustration concerns rentalCar. It is a sub-concept of urbanMeansOfTransport and not a sub-class of car as it would be in a generic ontology. We could have used the multiple inheritance but this would have led to an inconcistency : a rentalCar would have been also an internationalMeansOfTransport. This is not false but not explicitly proposed in the text.

5.2. Choice between generic or individual concept

We report here two examples that led to two different decisions. In both cases, the starting knowledge is an enumeration of nouns. No significant syntactical (or linguistic) indication can be exploited. The choice comes from semantic and even pragmatic reasons.

1. Paragraph 3 "… the means of transport that are …. We will have in our ontology the following ones : planes, trains, cars, ferries, motorbikes and ships".
   We know that we have to represent in the ontology that "planes, ferries, …" are some means of transport (linguistic indicator : "the … that are available … are the following"). We have the choice between defining individual concepts or generic concepts related to the concept meansOfTransport. If we read the following of the page, we notice that we need to refer to several specific and real planes used for specific flights. So these real planes will be instances, whereas the notion of plane is considered here as a class, and requires to be a generic concept.

2. Paragraph 4 "For each flight, the agency knows : the arrival date, the departure date, the arrival city …"
   We have here another enumeration. From reading, we know that all the terms in this enumeration refer to some properties of a flight. In Terminae, properties are represented with roles, that connect the concept with another concept called the value concept. So we have to define as many roles and related concepts as there are properties in this list. Decisions must be made when defining the value concepts, that can be either generic or individuals.
   In this case, we decided to define the following roles and only generic concepts :

- the roles priceBC (price in Business Class), priceFC (price in first class) and priceEC (price in economy class) have the same value concept, price, which is a generic concept, and can be instanciated with specific price values;
- The roles departureTime and arrivalTime have the same value concept absoluteTime for similar reasons;
- The roles departureDate and arrivalDate also have the same concept value date;
- On the opposite, departureAirport and arrivalAirport have the same value concept airport;

In all those cases, we did not feel the need to differentiate the two kinds of airports, times, dates as classes because it has no meaning. These properties define roles. The same concrete object can play the two roles at different times. The decision is sometimes much more complex and hard to make, as described in the next section.

1) In the last paragraph, we had a similar problem with the various hotel classes. We define a role numberOfStar on hotel with value in starNumber. starNumber individuals are 1*, …5*. If we define each hotel class as an instance (individual concept), we can express that the Holiday Inn hotel in New York is a 4 star hotel thanks to the role numberOfStar. But then these individual concepts will never be used in the ontology as role value. Another solution is to define the various hotel classes as as many generic concepts (oneStarHotel, twoStarHotel, …) that are sub-concepts of hotel. Then The Holiday Inn hotel may be an instance of 4StarHotel. It is important to do so if we need to explicitly specify some of the facilities available in a 3 star hotel that make them different from a 4 star hotel for example. If the system does not need to do so because hotel classes and corresponding services are well known by the customers, it is no use defining concepts and the role numberOfStar of hotel is enough. We decided to define generic sub-concepts oneStarHotel, …, fiveStarHotel of hotel, with corresponding numberOfStar value.

5.3. Concepts or roles ?

Some knowledge may be represented either by concepts or roles. The choice may be difficult to make.

First example

2) For instance, if it is important for the customer to know the model of plane of the flight, it must be one of the characteristics of a flight, or accessible from a flight (by the plane); this is expressed by the role meansOfTransport with value plane, which is inherited from the concept transport. So meansOfTransport is both the label of a generic concept and a role of the concept flight.

Second example (paragraph 5)

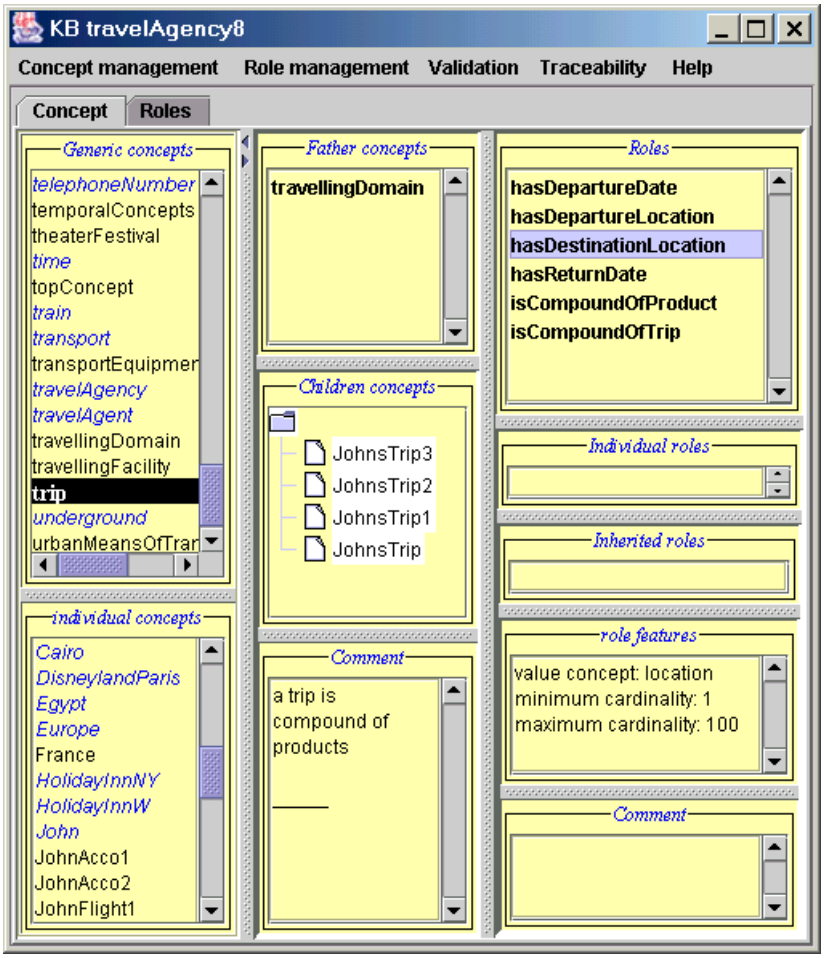
3) Destination are complex, including cities as points of interest located near a city with airport or even a continent. We first decided to call them all destination, to link this concept to trip. A destination may be a specific pointOfInterest as StatueOfLiberty, a city (and that implies a city to be a destination, that is not very correct), or otherDestination as Europe; each destination is linked to a correspondingCity. The problem is that otherDestination is a very general concept with many possible interpretations. It has to be made more precise.

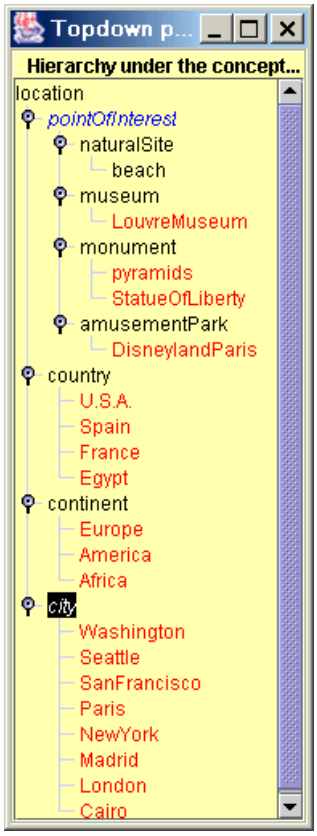
1) In fact, all destinations are locations and destination is rather the label of a role of the trip concept. So we ended by defining the location concept, with sub-concept such as city, country, continent, pointOfInterest. Such concepts have connecting roles to mean that a city belongs to a country, and that this country is part of a continent. This helps refine with a customer the destination associated to his trip: if he wants to go to Europe, the travel agent can suggest him various European countries or cities or points of interest located in Europe.

5.4. Instance definition leads to modification in generic concept definitions

4) For defining Paris as being a destination including the city and Disneyland, we define DisneyLand as an individual concept of pointOfInterest. In fact, more generally, we decide that any location can have various points of interest, and we feel the need to classify them into classes so that the customer whishes may be refined.

5) It appears also that a trip may be compound of several trips : a trip to Europe includes trips to London, to Paris (city or DisneyLand) and to Madrid. We need to express a recursive definition: a trip may be compound of trips.





5.5. One or several concepts ?

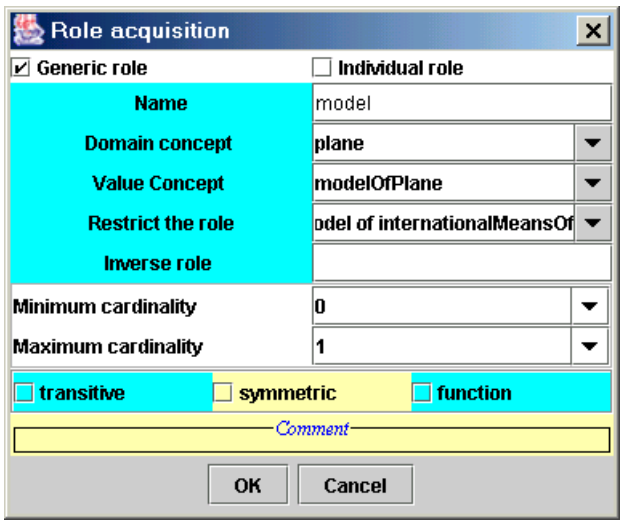
We illustrate this with paragraph 3 “customers are usually interested in the kind of plane they will fly on …”.

1. A first choice can be to identify mark and model as two different properties of the concept plane (planeModel and planeMark). As long as we cannot give “string” as the class value of these roles, we have to define 2 concepts : planeMark and planeModel.
2. Another solution would be not to differentiate mark and model (for instance, AirbusA320 would give the two information with a single individual concept). The notion of planeModel would include both the mark and the type of plane. We would have a single role and a single concept.
3. A third solution would be to consider that the mark is an information associated to a model, not to a plane, the model being the only information associated to a plane. Then the plane concept would have planeModel as a role, and planeModel would have planeMark as a role.
4. The reserve solution is also possible : consider the mark as the information associated to a plane, and the model as a role of a mark. Although selected in a first time, this solution is not very relevant: an invividual mark would have as roles all the possible plane models of this mark. So it is not easy to represent that a given plane (an individual concept) that has a given mark (individual role) is then related to a specific model.

So we choose the third solution because it seems to be more general and adequate. It allows to know the model of a plane in a first time, and to precise the mark if required. The relations between plane, markOfPlane, and modelOfPlane are delicate to establish. The difficulty is that the need of modeling mark and model of a plane appears when defining means of transport although this notion concerns a specific trip in plane, not plane as means of transport.

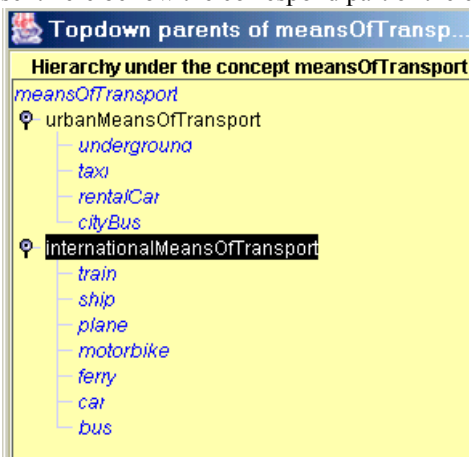
The same difficulty appears with other means of transport. For this reason, we decided to generalize the mark and model roles to any means of transport. This means that the roles were associated to the concept

meansOfTransport and modelOfTransport rather than plane and modelOfPlane. As a sub-concept, plane inherits of the modelOfTransport role. So we restricted it with the value modelOfPlane, which is a sub-concept of modelOftransport. The same happens for MarkOfTransport and markOfPlane.



5.6. Define structuring concepts or not ?

When defining a flight, it seems time to define some structuring concepts : an agency sells a trip which is compound of products, that are either transport or accommodation. Transport by means of plane is a flight. We added another concept to illustrate what other transports could be roadJourney.
Because the concepts that refer to means of transport are different to go from a location to a destination location and to move inside this destination location, we differentiate two classes of means of transports: internationalMeansOfTransport and urbanMeansOfTransport. This information is not explicitely in the text. So the labels of these structuring concepts may not be the best one as long as rental cars are not specifically urban.
We present here bellow the correspond part of the concept hierarchy.

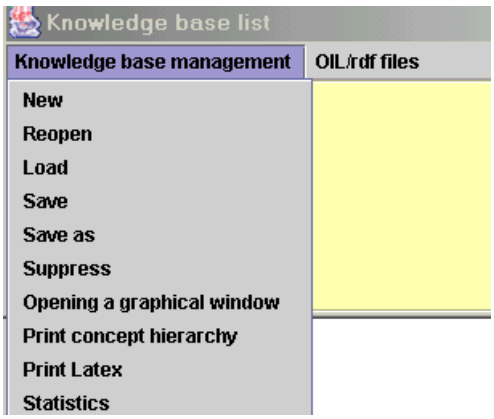


5.7. Checking the model : what's in the concepts ? what does the ontology look like ?

Terminae offers several means to have a more concise view on the ontology. The default of the concept and role editors is to give a split view.

- the ontology editor provides a rather concise view of each generic or individual concept : we can see its specific roles, its inherited roles, its sub-concept and its father in the hierarchy ; for each role, we can know its value ;
- several options available from the KN management pop-up menu of the ontology list manager help see the whole ontology (see screen copy bellow)
    - the whole hierarchy can be seen and printed thanks to "printing the concept hierarchy"

- a Latex file can be printed : it proposes a frame like presentation of all the generic and individual concepts with their associated roles and values. We present here bellow an extract of this file.




```
:topConcept :travellingDomain :trip
Concept primitif
T
TDS
*****rôles *******
isCompoundOfTrip trip
isCompoundOfProduct product
hasReturnDate absoluteDate
hasDestinationLocation location
hasDepartureLocation location
hasDepartureDate absoluteDate
**************************************************
:topConcept :travellingDomain :trip :JohnsTrip3
NT
**************************************************
:topConcept :travellingDomain :trip :JohnsTrip2
NT
****** rôles individuels*******
isCompoundOfProductJT22 JohnAcco2
isCompoundOfProductJT21 JohnFlight2
hasDestinationJT2 Washington
departure NewYork
**************************************************
:topConcept :travellingDomain :trip :JohnsTrip1
NT
****** rôles individuels*******
returnDateJT1 April112002
isCompoundOfProductJT12 JohnAcco1
isCompoundOfProductJT1 JohnFlight1
destinationLocationJT1 NewYork
departureLocationJT1 Madrid
departureDateJT1 April52002
**************************************************
:topConcept :travellingDomain :trip :JohnsTrip
NT
****** rôles individuels*******
returneDateJT April152002
isCompoundOfTrip3 JohnsTrip3
isCompoundOfTrip2 JohnsTrip2
isCompoundOfTrip1 JohnsTrip1
destinationLocationJT U.S.A.
departureDateJT April52002
departureCityJohnsTrip Madrid
```

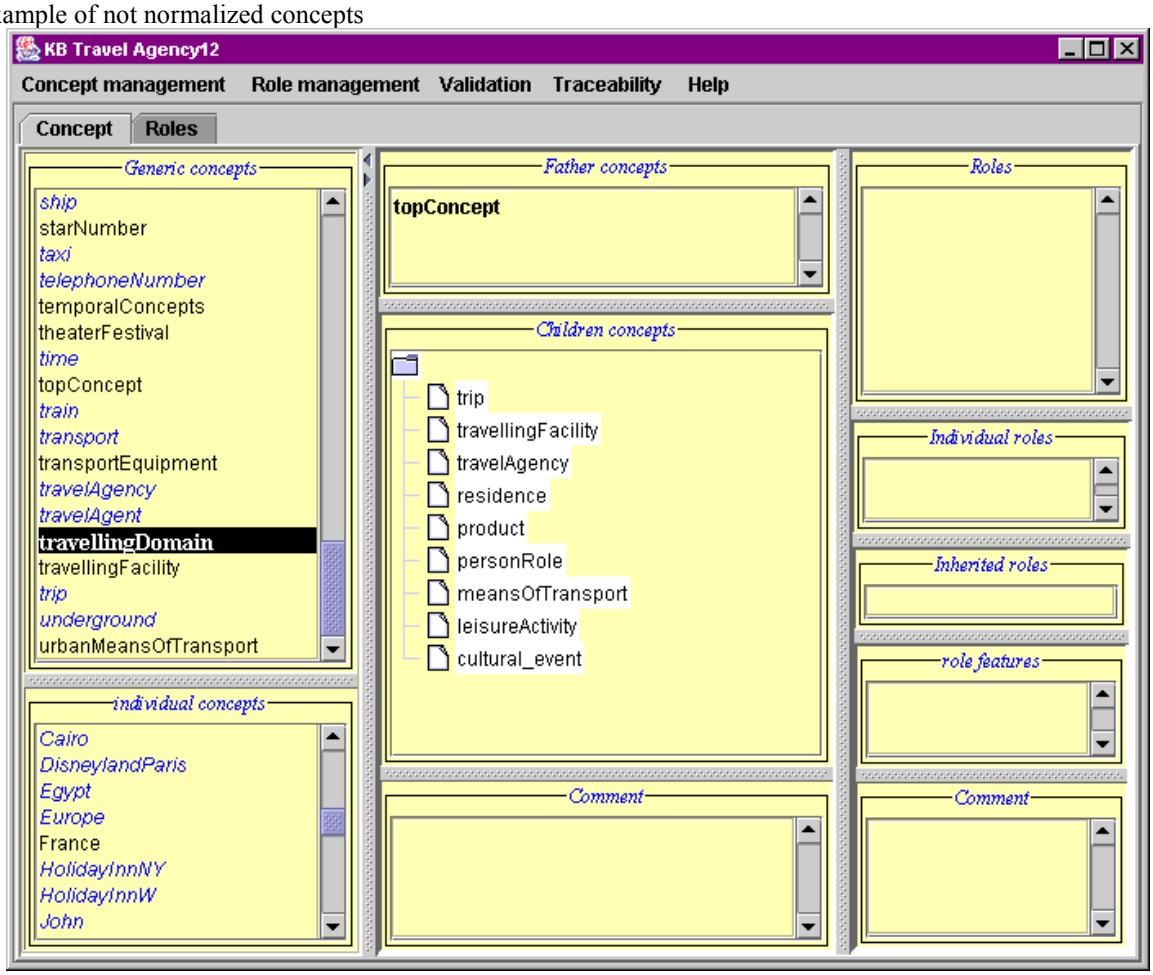
6. Normalization

Once most of the concepts found in the knowledge sources and required by the application needs have been added to the ontology, Terminae suggests to check the model according to differentiation rules. These rules lead to make explicit the modeling decisions. The knowledge engineer may require to look for additional knowledge back in the documents or from the expert. The differentiation rules require that for any given concept, the following information should be made explicit in the model (thanks to roles with Terminae representation language) :

- the concept must have at least one common role with its father concept (generally an inherited role);
- the concept must have at least one specific role that make it different from its father concept;
- the concept must have at least one share property (role) with its brother concepts (this role may be an inherited one);
- the concept have at least one specific property that make it different from its brother concept (this may be a specific role or value of an inherited role).

The application of these rules leads either to enrich the model or to eliminate some useless concepts or to reorganize the hierarchy with some intermediary concepts.

Example of not normalized concepts

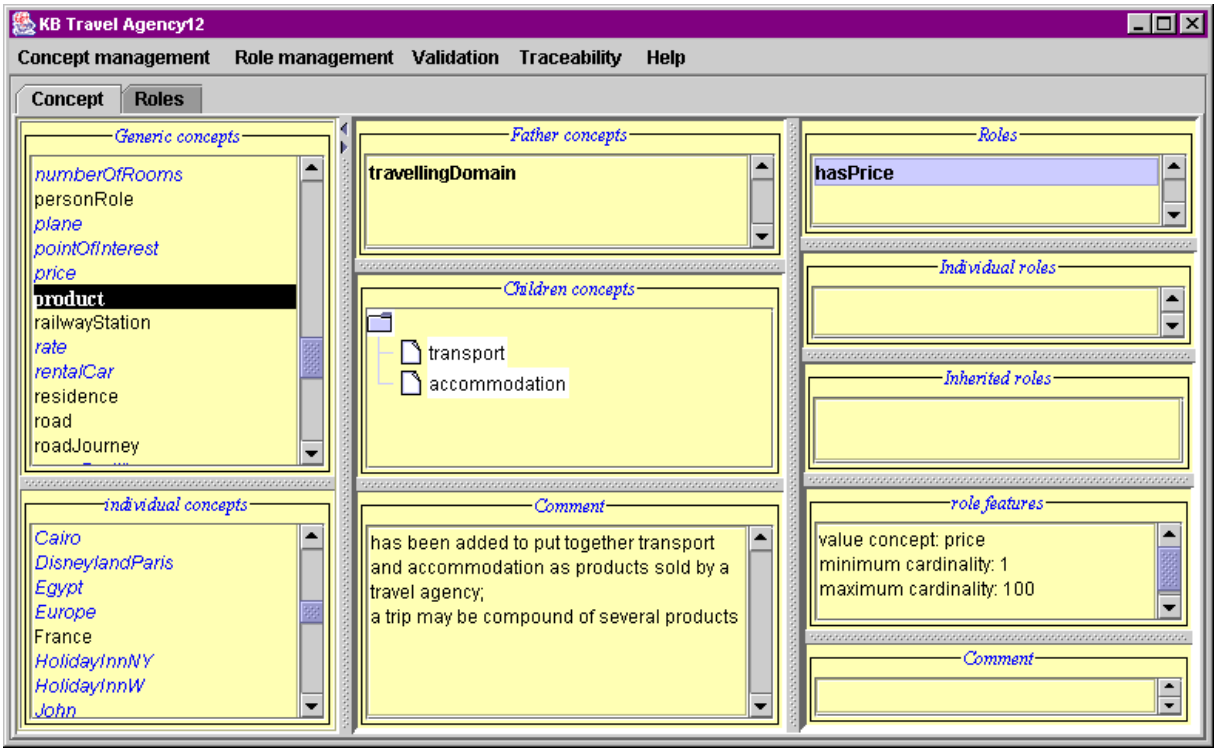


The concept travelingDomain has no specific role yet, that would make it different from the root topConcept. In fact, this concept is an artificial means to inform the reader that, from this concept and below, all the information is structured according to the point of view induced by the travel agency application.

Example of differentiated concepts of the hierarchy

In the example bellow, the role hasPrice has been added to the concept product in order to stress the commonalities between a product, an accommodation and a transport. This role also contributes to differentiate this concept from other children of the travelingDomain concept. The children concepts transport and

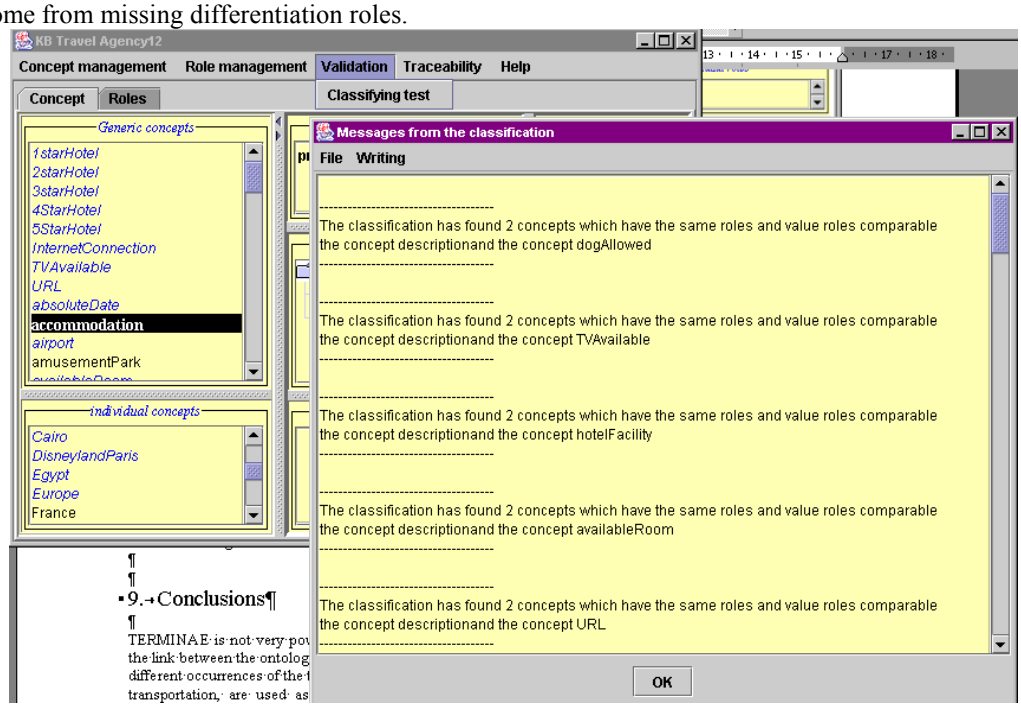
accommodation are different because they have their own roles (byMeansof -> meansOfTransport and hasDuration are specific roles to transport; firstNight and lastNight are specific roles of accommodation).



7. Concept classification

The validation option of the KB editor proposes a classification program. A report is then displayed to the user that can notice all the errors left in the model. The classifier expects each concept to be different from all the other, whether because it has a specific role or a specific value concept of a common role. Implicitly, the algorithm assumes that a concept is not worth being defined if it is not syntactically different from the other ones. Its label is not enough as a difference.

The screen copy bellow shows a report obtained before a systematic differentiation of our ontology. Most of the errors come from missing differentiation roles.



This control is optional and an ontology may be left with some errors according to these criteria. For instance, from the available document in our experiment, many knowledge is missing that would help to differentiate a hotel from a bed and breakfast, or to differentiate formally all the hotel facilities. May be these facilities should better be represented has individual concepts of the hotelFacility concept.

8. Output of TERMINAE

TERMINAE proposes various format for the output ontology and the terminological forms.
The ontology is stored by default in XML, and can also be exported in OIL or OIL-RDFs. It can printed as a LaTex file.
Each terminological form is stored in XML format .

9. Conclusions

TERMINAE is not very powerful as a representation language but rather as a guiding tool. Its main interest is the link between the ontology and the texts. For instance, the terminological form meansOnTransport gives the different occurrences of the term in the text, and it says that other terms in the text, kind of transport and kinds of transportation, are used as synonyms. A natural language definition can be given, which completes the conceptual definition. That helps the user to understand the underlying modeling of the ontology, and the modeling point of view.

We have shown the process from lists of terms to terminological forms and then to the ontology as an illustration of the guidance provided by the system. It would have been more powerful with a larger input set of texts. We are well aware of the need to have a formal model checking at the end.

10.References

AUSSENAC-GILLES N., BIEBOW B. & SZULMAN S., (2002), Modélisation du domaine par une méthode fondée sur l'analyse de CORPUS. In *Ingénierie des connaissances.* Paris : Eyrolles, à paraître.

BIÉBOW B. & SZULMAN S. (1999). TERMINAE: A linguistic-based tool for the building of a domain ontology, Proc. of the *11th European Workshop, Knowledge Acquisition, Modelling and Management (EKAW 99)*, Dagstuhl Castle (G), Springer Verlag, 49-66.

BIEBOW B. & SZULMAN S. (2000), Terminae : une approche terminologique pour la construction d'ontologies du domaine à partir de textes. *Actes de RFIA2000, Reconnaissances des Formes et Intelligence Artificielle*, Paris (F).

S. Le MOIGNO, J. CHARLET, D. BOURIGAULT, P. DEGOULET, M.-C. JAULENT (2002), Terminology Extraction from Text to Build an Ontology in Surgical Intensive Care. In *Proceedings of the ECAI2002 workshop on NLP and ML for Ontology Engineering*. Lyon (F). July 22-23, 2002.

SZULMAN S., BIEBOW B. & AUSSENAC-GILLES N. (2002), Structuration de Terminologies à l'aide d'outils d'analyse de textes avec TERMINAE, *TAL*, Paris : Hermès. Vol43,N°1. 2002.