CREATION AND MANAGEMENT OF A CONCEPTUAL KNOWLEDGE BASE IN AN INDUSTRIAL DOMAIN

Gian Piero Zarri
University Paris-Est/Paris12, LiSSi Laboratory, Vitry sur Seine, France

Keywords: Knowledge representation, Knowledge engineering, Ontologies, Inference, Gas/oil industry.

Abstract: This paper describes the implementation and test of an experimental knowledge base in the framework of a European project in the gas/oil domain. The base has been built up by using the NKRL (Narrative Knowledge Representation Language) conceptual tools to formalize and manage the information content of two different 'Storyboards' or 'Historians': these describe sequences of 'gas/oil' events like the detection of gas leakage alarms or the activation of a gas turbine.

1 INTRODUCTION

This paper supplies some information about the work done by the author at the Milan Polytechnic (Italy) – between June 1st, 2008 and January 31st, 2009 – to implement and test an experimental knowledge base in the framework of the EC VIRTHUALIS project ("New Production Processes and Devices" n. 515831-2). The base has been built up by using the NKRL conceptual meta-model/ environment, see (Zarri, 2005; 2009a) to *formalize and manage* the information content included in two VIRTHUALIS 'Storyboards', the "StatoilHydro Case Study Specifications" and the different versions of the "Sonatrach Case Study Specifications". 'Storyboard' is a term borrowed from filmmaking industry to *describe in written a series of interactive events*; a synonymous is 'historian'. The knowledge base is fully supplied in Appendix A of (Zarri, 2009b).

In the StatoilHydro case, the Control Room operators recognize a gas leakage alarm. They interact then with a Field operator who searches for the leakage position and tries to quantify its severity. She/he notifies her/his findings to the Control Room operators and, all together, they take decisions about the operations to be performed.

The Sonatrach storyboard illustrates the twelve sequences of operations needed for the activation of a gas turbine used to drive the compressor of a propane chilling section. For example, the Seq3 sequence concerns the start up of the turning gear, Seq5 describes the acceleration of the main turbine, Seq6 deals with the starting of the ignition operations, etc. This section of the knowledge base is completed by the description of an example of anomaly detected during the start-up procedures.

In the following, Section 2 will introduce the general context of the experiment. Section 3 describes some of the results obtained. Section 4 is a short "Conclusion".

2 GENERAL CONTEXT

2.1 'Static' and 'Dynamic' Information

A fundamental differentiation about the storyboard 'knowledge' concerns the separation between '*static*' and '*dynamic*' information.

'*Static*' information corresponds to notions to be considered, in a sense, as '*a-temporal*' and '*universal*'. This means that their formal definitions *are seldom subject to change, at least within the framework of a given application*; these notions define, typically, the *general context* of the application. Examples can concern the definition of the working functions of the personnel (control_room_operator), the description of the installations (plant_, valve_, alarm_), of the general environmental conditions (level_of_temperature), of the critical conditions for failure, etc.

A simple *'binary' approach* can then be used for their conceptual representation: in this approach, the 'properties' or 'attributes' that define a given concept are expressed as *binary* (i.e., linking only

Piero Zarri G. (2009).
CREATION AND MANAGEMENT OF A CONCEPTUAL KNOWLEDGE BASE IN AN INDUSTRIAL DOMAIN.
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, pages 214-219
DOI: 10.5220/0002300802140219

two arguments) *relationships of the 'property/value' type*, independently from the fact that these relationships are organised into frame format or take the form of a set of 'property' statements used to define a 'class' in a W3C language like OWL.

'*Dynamic*' information consists, on the contrary, of *structured, temporal sequences of (not predetermined) 'elementary events' that describe the active or passive 'behaviour' of given 'characters', 'actors' or 'personages'* (not necessarily human see, e.g., the 'behaviour' of a faulty valve or of a start-up turbine). Examples of dynamic information in a VIRTHUALIS context are 'elementary events' like "The Control Room operator presses the start-up button", "The oil extractor moves from the state 'idle' to the state 'running'", "The Field operator has heard the working noise of the oil extractor*",* "The field operator has visually checked the correct progression of ignition in chambers 1 and 4", etc.

The necessity of making use i) of '*conceptual predicates*' for specifying the basic type of state, action etc. described in each 'elementary event' included in the (dynamic) temporal sequence, and ii) of the *notion of 'role'* to denote the logical and semantic function of each of the 'characters' involved in the different events – in "The Control Room operator presses a button …", the 'individual' CONTROL_ROOM-OPERATOR_1 is the SUBJ(ect) of the action of 'pressing' and the individual BUTTON_1 the OBJ(ect) – *makes it impossible to make use of the common binary approach to represent correctly the dynamic knowledge*. In this last case, it is necessary to have recourse – to represent each one of the elementary events that make up the global dynamic situation – to the well-known '*n*-ary' schema denoted by Eq. 1:

$$(L_i\ (P_j\ (R_1\ a_1)\ (R_2\ a_2)\ \ldots\ (R_n\ a_n))) \qquad (1)$$

where L_i is the symbolic label identifying the particular *n*-ary structure (e.g., that corresponding to the representation of "The Control Room operator presses a button …", example), P_j is the conceptual predicate, R_k is the generic role and a_k the corresponding argument (e.g., CONTROL_ROOM-OPERATOR_1), see (Zarri, 2009a: 14-22).

To represent fully a given dynamic situation, it is also necessary to have a way of representing the '*coherence links*' that bring together its different, constitutive 'elementary events'. These are normally expressed through NL *syntactic constructions like causality, goal, indirect speech, co-ordination and subordination*, etc., see the example: "The control room operators push the reset button *in order to (GOAL)* verify the existence of an alarm situation". In this paper, we will use the terms '*connectivity phenomena*' to denote this sort of contextual clues.

2.2 Tools for the Gas/oil Industry

The W3C languages have been sometimes suggested – see, e.g., http://www.w3.org/2008/11/ogws-agenda.html#papers – as possible solutions for introducing new semantic/conceptual tools in the gas/oil industry world. This proposal is *questionable*, at least when, as in our case, the 'knowledge' to be used is largely based on the *'narration'* of *'sequences of events'*.

As well known in fact – see (Mizoguchi *et al.*, 2007; Zarri, 2009a), etc. – the *lack of expressiveness* linked with the 'binary' nature of the W3C languages *prevents them from representing correctly the 'dynamic' information*. When these languages must represent simple 'narratives' like "John has given a book to Mary" (or "The Control room operator notifies the situation to the Field operator" etc.), several difficulties arise. For example, "give" is an *n-ary (ternary) relationship* that, to be represented in a complete way, asks for the presence of a specific '*semantic predicate*' in the "give" or "transfer" style, where the '*arguments*' "John", "book" and "Mary" of the predicate must be labelled with '*conceptual roles*' such as, e.g., 'agent of give', 'object of give' and 'beneficiary of give' respectively. *An* n*-ary type of representation* in the style of Eq. 1 is then needed. Note that each of the (R_i a_i) cells of Eq. 1, taken individually, represents a binary relationship in the W3C (OWL, RDF…) languages style. The main point here is, however, that the conceptual structure represented by Eq. 1 can be fragmented for *practical purposes* like the concrete storing within a relational database, *but must be considered globally whenever significant querying/inferencing operations must be envisaged on the whole structure*, see (Zarri, 2009a: 14-33).

In a gas/oil industry context, an obvious candidate for the set up of conceptual descriptions is ISO 15926 ("Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities"). Because of the presence of temporal representational aspects, ISO 15926 is often defined as a '4D(imensions)', or 'space-time', model, holding that individuals are extended in time as well as space and dealing then with changes over time, see (Stell and West, 2004) in this context. In spite of this, the knowledge representation model of ISO 15926 is essentially '*binary*', as confirmed by its two-way, easy conversion into (W3C) OWL terms.

The existence, e.g., of an object labelled as 'M202' and classified as a 'lubrication pump' is described using two (RDF-like) binary relationships, Identification to link PHYSICAL_OBJECT_ to M202, and Classification to link PHYSICAL_OBJECT_1 to lubrication_pump. Also the 4D aspects seem to boil down, in practice, to the use of binary relationships see, e.g., the relationships hasBeginning and hasEnd that, once again in an RDF style, link a physical object to dates instances like dayIdentifier entity. A bridge with more evolved types of representation can be found in the ISO 15926 so-called *templates*. A template is a pattern for stating facts, formed essentially by a 'predicate' with its arguments. For example, a template like Parts-at-least(C, D, *i*) means "Any C has at least *i* D's as parts"; instantiated into, e.g., Parts-at-least(Car, Wheel, 3), will be then automatically converted by a set of 'expansion rules' proper to the template into the standard binary descriptions. The user can also deal directly with templates – to query or instantiate them – using some simple interfaces.

2.3 A Short Review of the NKRL System

We have then used NKRL to build up the VIRTHUALIS knowledge base in order to avoid the 'binary' limitations examined quickly in the previous Section. NKRL innovates with respect to the current ontological paradigms by adding to the usual '*ontologies of concepts*' an '*ontology of events*', *i.e., a new sort of hierarchical organization where the nodes correspond to n-ary structures called 'templates' that follow the format defined by Eq. 1 above*. This last hierarchy is called HTemp (hierarchy of *templates*). Templates are particularly concerned with the representation of the 'dynamic knowledge' aspects evoked above: they can be conceived, in fact, as the *formal representation of generic classes of elementary events* like "move a physical object", "produce a service", "send/receive a message", "make a change of state happen", etc.

Note that, in NKRL, an 'ontology of concepts' (according to the usual, 'binary' meaning of these terms) not only exists, but it represents an *essential component* for the correct functioning of the whole environment. This ontology is called HClass (hierarchy of classes), see (Zarri, 2009a: 103-137).

When a *particular elementary event* must be represented, the corresponding template is *instantiated* to produce what is called a '*predicative occurrence*'. To represent then an event like: "On October 16th, 2008, the production activities leader pushes the SEQ1_BUTTON in the context of a particular sequence of operations, SEQ1, associated with the start-up of the turbine", we must select first in HTemp the template corresponding to 'perform a task or an activity', represented in the upper part of Table 1. When creating a *predicative occurrence* like virt2.c32 (lower part of Table 1), the role fillers in this occurrence *must conform to the constraints associated with the variables of the father-template*. In virt2.c32, e.g., INDIVIDUAL_PERSON_102 is an 'individual', instance of the concept individual_person; this last is a specialization of human_being, specialization in turn of human_being_or_social_body, see the constraint on the variable (argument) *var1* associated with the SUBJ(ect) role in the template of Table 1.

What we have expounded until now concerns the representation of *elementary (simple) events.*

Table 1: Deriving an occurrence from a template.

name: Produce:PerformTask/Activity
father: Produce:
position: 6.3
natural language description: 'Execution of Intellectual or Industrial Procedures, etc.'

PRODUCE	SUBJ	*var1*: [*var2*]
	OBJ	*var3*
	[SOURCE	*var4*: [*var5*]]
	[BENF	*var6*: [*var7*]]
	[MODAL	*var8*]
	[TOPIC	*var9*]
	[CONTEXT	*var10*]
	{ [modulators], ≠abs }	

var1, var4, var6 = human_being_or_social_body
var3 = activity_, process_, temporal_development
var8 = activity_, artefact_, process_, etc.
var9 = pseudo_sortal_concept, sortal_concept
var10 = situation_, symbolic_label
var2, *var5*, *var7* = location_

virt2.c32) PRODUCE SUBJ INDIVIDUAL_PERSON_102: (GP1Z_MAIN_CONTROL_ROOM)
OBJ button_pushing
TOPIC SEQ1_BUTTON
CONTEXT (SPECIF SEQ1_GREASING_PUMP (SPECIF member_of F17_STARTUP_SEQUENCE))
date-1: 16/10/2008/08:26
date-2:

To deal with the *connectivity phenomena*, the basic knowledge representation tools have been complemented by more complex mechanisms that use *second order structures created through reification of the predicative occurrences' conceptual labels*, see (Zarri, 2009a: 86-98). For example, the '*binding occurrences*' are lists of labels (c_i in Eq. 1) of predicative occurrences; the lists are differentiated making use of specific binding operators like GOAL and CAUSE.

Reasoning in NKRL ranges from the *direct questioning* of a knowledge base – by means of *search patterns* (formal queries) p_i that unify information in the base using a *Filtering Unification Module* (*Fum*), see (Zarri, 2009a: 183-201) – to *high-level inference procedures*.

For example, the 'transformation' rules try to '*adapt*' a search pattern p_i that '*failed*' (that was unable to find an unification within the knowledge base) to the *real contents* of this base using a sort of '*analogical reasoning*'. Let us then suppose we ask: "Search for the evidence of the existence of an alarm situation in some industrial premises". In the impossibility of obtaining a direct answer, the corresponding search pattern can be *transformed* into the two logically linked patterns: p_1: "Search for information relating that the working staff is moving massively to a new location"; p_2: "Search for information confirming that the new location is outside the industrial premises", see Table 2. If the new patterns are able to unify some occurrences in the base, we can consider that *the information collected in this way is a sort of indirect answer to the query originally posited.*

Table 2: An example of 'transformation' rule.

```
t2: "working staff moving" transformation

antecedent:
EXIST    SUBJ    alarm_situation: (var1)
var1  =  oil/gas_processing_plant

first consequent schema (conseq1):
MOVE     SUBJ    var2: (var1)
         OBJ     var3: (var4)
var2, var3 = company_working_staff
var4  =  geographical_location
var3  =  var2;  var4 ≠ var3

second consequent schema (conseq2):
OWN      SUBJ    var4
         OBJ     property_
         TOPIC   (SPECIF var5 var1)
var5  =  outside_
```

To verify the existence of an alarm situation in some industrial premises try, along other things, to see i) whether we can find information concerning the fact that the working staff moves massively to a new location, and ii) whether the new location is outside the industrial premises.

With respect now to the *hypothesis rules*, these allow us to build up automatically a sort of '*causal explanation or context*' for some information (a predicative occurrence c_j) retrieved within an NKRL knowledge base. Let us suppose, e.g., we have directly retrieved, in a querying-answering mode, information like: "An operator has activated a piping segment isolation procedure in the context of an industrial accident" that corresponds then to c_j. Supposing we can found a hypothesis rule whose 'premise' corresponds to c_j, we should then be able to automatically construct, using this rule, a sort of '*causal explanation*' of the triggering event by retrieving in the knowledge base information in the style of: i) "someone has attempted to activate a (milder) corrective maintenance procedure" (c_1); ii) "this procedure has failed" (c_2) and iii) "the accident is considered as a serious one" (c_3). A detailed, formal representation of this rule is given in Table 3.

An interesting, recent development of NKRL concerns the possibility of *making use of the two above modalities of inference in an 'integrated' way*, see (Zarri, 2005) and Section 3.2 below.

Table 3: An example of 'hypothesis' rule.

```
h1: "isolation procedure" hypothesis
premise:
PRODUCE   SUBJ      var1
          OBJ       isolation_procedure
          CONTEXT   var2
var1  =  human_being
var2  =  industrial_accident
```

An individual has carried out an isolation procedure in the context of an industrial accident

```
first condition schema (cond1):
PRODUCE   SUBJ    var3
          OBJ     var4
          TOPIC   var2
var3 = human_being;  var3  ≠ var1;  var4 =
corrective_maintenance_procedure
```

A different individual had carried out a (milder) corrective maintenance procedure

```
second condition schema (cond2):
EXPERIENCE   SUBJ    var3
             OBJ     var5
             TOPIC   var4
var5 = failure_
```

This second individual has experienced a failure in this corrective maintenance context

```
third condition schema (cond3):
BEHAVE   SUBJ    var1
         MODAL   var6
var6 = control_room_operator
```

The first individual was a control room operator

```
fourth condition schema (cond4):
BEHAVE   SUBJ    var3
         MODAL   var7
var7 = field_operator
```

The second individual was a field operator

```
fifth condition schema (cond5):
OWN      SUBJ    var2
         OBJ     property_
         TOPIC   (SPECIF strength_ var8)
var8 = important_
```

The industrial accident is considered as a serious one

3 RESULTS OBTAINED

3.1 General Remarks

The NKRL modelling of the StatoilHydro storyboard/historian has given rise to:

- The insertion in the VIRTHUALIS KB of 86 NKRL conceptual structures:
  – 60 *predicative occurrences* (events);
  – 26 *binding occurrences* (representing *logical/semantic connections* among events).
- The addition of about 130 *new* 'static concepts' to the 'standard' HClass ontology.
- The addition of a *new* 'template' to HTemp, Produce:Choice/Decision, as direct specialization of the high-level Produce: template.

The added concepts pertain mainly to sub-branches of HClass like alarm_tool, use_of_systems_and_apparatus, industrial_accident, etc. The addition of Produce:Choice/Decision derives from the presence, in the StatoilHydro storyboard, of cyclic formulas like "… the operators decide to carry out a corrective procedure …".

For the Sonatrach case, the results are:

- The insertion in the KB of 278 new structures:
  – 222 predicative occurrences;
  – 73 binding occurrences.
- The addition of about 70 *new* 'static concepts' to the standard HClass ontology.

In this case, the new concepts pertain mainly to HClass sub-branches like industrial/technical_tool, measurement_unit, industrial/technical_procedure.

Several examples of query/answer operations are reproduced in (Zarri, 2009b: 22-27).

3.2 Some Inference Results

The simple transformation rule reproduced in Table 4 has been used to answer *indirectly* questions like: "Is the oil extractor running?", see Figures 1 and 2.

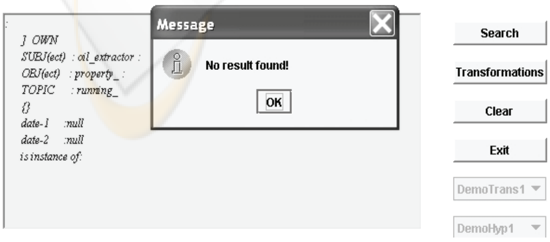

Figure 1: Failure of a formal query (search pattern).

Table 4: The 'working/noise' transformation rule.

***t5**: "working noise/condition" transformation*
antecedent:
OWN SUBJ *var1* OBJ property_ TOPIC running_
var1 = consumer_electronics, hardware_, technical/industrial_tool, etc.
first consequent schema (*conseq1*):
EXPERIENCE SUBJ *var2* OBJ evidence_ TOPIC (SPECIF *var3* *var1*)
var2 = individual_person *var3* = working_noise, working_condition
second consequent schema (*conseq2*):
BEHAVE SUBJ *var2* MODAL industrial_site_operator
Faced with the impossibility of proving directly that an industrial apparatus is running, the fact of, e.g., hearing its working noise, can be a proof of its running status.

The result of Figure 2 can be paraphrased as: "The system cannot assert that the oil extractor is running, but it can certify that the site leader has heard the working noise of this extractor".

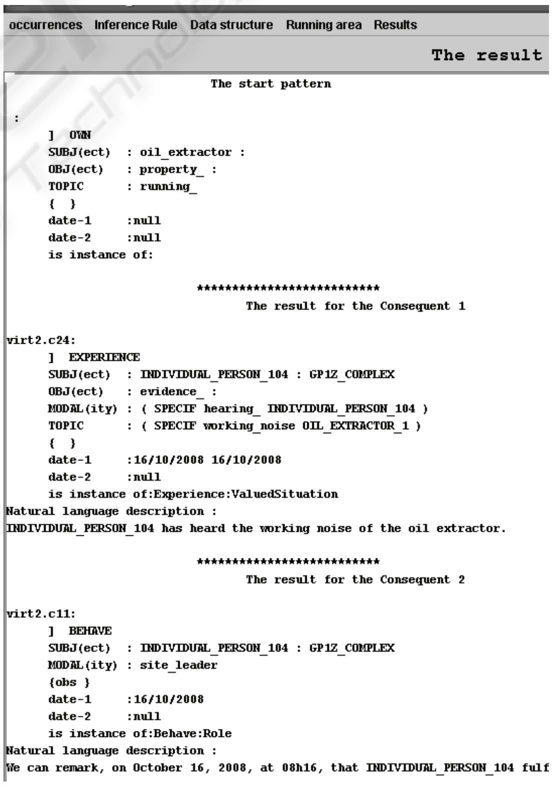

Figure 2: Indirect answer by transformation procedure.

With respect now to the hypothesis rules, and *to show the importance of being able to use transformations and hypotheses in an integrated way*, let us imagine to make use of a trivial hypothesis that – after having retrieved an event relating that someone has stopped a given technical/industrial procedure – tries to find out whether this stop is related to some accident affecting the tool concerned by this procedure.

When we go to see, however, if an hypothesis in this style (hypothesis *h3,* not reproduced here) can be activated starting from a predicative occurrence like virt3.c14 of Table 5, *we see that h3 cannot be used 'as it is'* because of the impossibility of *demonstrating directly that some sort of accident has concerned the GP1Z_TURBINE*.

Table 5: Stopping the start-up of the GPIZ_TURBINE.

```
virt3.c14)  PRODUCE SUBJ  INDIVIDUAL_PERSON_102:
                               (GP1Z_MAIN_CONTROL_ROOM)
                    OBJ    activity_stop
                    TOPIC (SPECIF turbine_startup
                                    GP1Z_TURBINE)
                    date-1:   1/11/2008/10:20
                    date-2:

  Produce:PerformTask/Activity (6.3)
```

A given person ends the start-up of the GP1Z_TURBINE.

Table 6: Transformation rule about 'related' accidents.

t8: *"part of, linked with" transformation*

antecedent:

```
PRODUCE  SUBJ   var1
         OBJ    detection_
         TOPIC  (SPECIF var2 (SPECIF var3 var4))
var1   =   individual_person
var2   =   industrial_accident
var3   =   relational_property, spatial_relationship
var4   =   technical/industrial_tool
```

first consequent schema (*conseq1*):

```
PRODUCE  SUBJ   var1
         OBJ    detection_
         TOPIC  (SPECIF var5 (SPECIF var6 var7))
var5   =   industrial_accident
var3   =   relational_property, spatial_relationship
var7   =   technical/industrial_tool;  var7 ≠ var4
```

second consequent schema (*conseq2*):

```
OWN      SUBJ   var7
         OBJ    property_
         TOPIC  (SPECIF var8 var4)
var8 = part/whole_relationship, binary_relational_property
```

If we are unable to detect an accident in the environment (var3) of an industrial tool (var4), we can try i) to see whether we can detect an accident involving another tool (var7), and ii) then prove that this second tool is (var8) either a component of the original tool or it is strictly connected with this last one.

In this case, the solution consists in making use, during the processing of hypothesis *h3*, of a rule like transformation ***t8*** in Table 6. We suppose then that detecting an accident involving a *component* of the tool concerned by a given procedure, or some other *associated device*, can be considered as *equivalent* to detecting an accident that concerns the tool itself.

Space prevents us from illustrating the different steps of the integrated hypothesis/transformation execution, see (Zarri, 2009b: 29-31) for the details. Very in short, trying to construct automatically a context/causal explanation for occurrence virt3.c14 of Table 5 implies necessarily to make use of transformation *t8* during the processing of hypothesis *h3*. In this way, the detection of an accident in the environment of GP1Z_TURBINE is reduced to i) the discovery of an oil leakage for the AUXILIARY_LUBRICATION_PUMP_M202, and ii) the verification of a *strict relationship* between this pump and GP1Z_TURBINE.

4 CONCLUSIONS

In spite of its short duration, we think that the "VIRTHUALIS Knowledge Base" experiment can be considered as a success. We can assume, in fact, that: i) the possibility of implementing an in-depth, conceptual modelling of the VIRTHUALIS' 'storyboards/historians' *with the minimum loss of the original meaning* is *largely proved*; ii) even if the existing 'rule base' is characterized, at the moment, by a very reduced size, its possible utility in a real industrial context can be easily inferred – once again, see (Zarri, 2009b) for the details and for (several) additional examples.

REFERENCES

Mizoguchi R., Sunagawa E., Kozaki K., and Kitamura Y. (2007). A Model of Roles within an Ontology Development Tool: Hozo. *Journal of Applied Ontology* **2**: 159-179.

Stell, J. G., and West, M. (2004). A 4-Dimensionalist Mereotopology. In: *Formal Ontology in Information Systems*. Amsterdam: IOS Press.

Zarri, G.P. (2005). Integrating the Two Main Inference Modes of NKRL, Transformations and Hypotheses. *JoDS* **4**: 304-340.

Zarri, G.P. (2009a). *Representing and Managing Narrative Information – Theoretical Principles and implementation;* London : Springer.

Zarri, G.P. (2009b). *Technical Report on the Set Up of a Knowledge Base for the VIRTHUALIS Project.* Milano: Politecnico di Milano, Virthualis Project.