DOI: https://doi.org/10.46354/i3m.2019.imaaca.010

AUTOMATION AND COGNITION: A METHODOLOGICAL PROCESS FOR HUMAN-CENTRED DESIGN TO MINIMIZE ERRORS

Jean-Marc Mercantini

Aix-Marseille Université, Université de Toulon, CNRS
Laboratoire d'Informatique et Systèmes (LIS - UMR 7020)
Domaine universitaire de Saint Jérôme
Avenue Escadrille Normandie-Niemen
13397 Marseille cedex 20

jean-marc.mercantini@lis-lab.fr

ABSTRACT

Disturbances undergone by a complex system can come as much from its external environment as from the internal elements which constitute it. Complex systems are understood in this study as composed of intelligent machines and humans (IMH), and being in charge to accomplish complex tasks in a collaborative way. Performances of these systems, in terms of robustness, adaptation and resilience, strongly depend on the behaviour of the IMH duo. The works that are the subject of this paper focus on the study of the IMH duo and propose a methodological process using jointly cognitive approaches with formal modelling and simulation to analyse, design and control complex systems. For those systems, human beings are necessarily implied in their global behaviour –including stability–, what crucially calls for a better understanding of their behaviour facing diverse complex situations: normal situations, risky situations, critical and accidental situations. Tools and methods proposed by cognitive Sciences, Cognitive Engineering and Knowledge Engineering allow to take into account the different mechanisms involved in human behaviour to enrich complex system models for a better design and control to munimize human errors.

Keywords: Automation, Cognitive Engineering, Formal modelling, Errors, Complex systems.

1. INTRODUCTION

It may seem curious, a priori, to associate the two terms or concepts: Automation and Cognition. But, from a historical perspective (Mercantini 2015), this association is very pertinent and, maybe even become a concept in its own right, refering to the evolution of the current technological systems (like Robotics and Artificial Intelligence). From the cognitive science dictionary (Tiberghien 2002), "cognition is a function allowing the knowledge realisation and examining the different activities relating to knowledge". Cognition may be also defined (Ganascia 1999) as the ability to integrate multimodal information for generating representations, building associations and elaborating generalizations. The ability to manipulate this knowledge allows the individual to develop a behaviour that depends not only on the environment or the immediate situation.

Originally, the sciences of cognition are based on the study of natural cognition for then evolving toward the study of artificial cognition mobilizing computers to reproduce the mental representations and the functions that allow their treatment. Cognition became an object of scientific study during the twentieth century. Its development is strongly linked to the development of computers used as tools to simulate the cognitive process models, but also used as a metaphor of the brain function where information is received, formatted, processed and stored in memory. This memory is then mobilized to elaborate reasoning and action plans.

In 1956, Cognitive science are emerging from the early development of the cybernetics which is defined by (Wiener 1948) as "the scientific study of control and communication in the animal and the machine". Cybernetics is founded on the key concept of the feedback loop, and its original goal was to provide a unified view of emerging areas of the automatic, the electronic and the mathematical theory of information (Wikipedia 2019).

With cognitive science, the understanding of the outside world changes its viewpoint. It is not external objects that attract attention, but the tool with which they are observed. Cognitive science is concerned with the processes of perception, reasoning, pattern recognition, concept formation, understanding, interpretation, problem solving, control, planning and action. Cognitive engineering and knowledge engineering will propose formal methods, guidelines and norms to design systems in which cognition has a central position.

From the Oxford Handbook of Cognitive Engineering (Lee 2013), Cognitive Engineering is an interdisciplinary approach to the analysis, modelling, and design of engineered systems or workplaces, especially those in which humans and automation jointly operate to achieve system goals. Cognitive engineering characterizes an area of activity (scientific and technical) that is concerned by integrated human-technology systems. It combines knowledge and experience from Cognitive Science, Human Factors, Human-Computer Interaction Design and Systems Engineering (Gersh et al. 2005).

Cognitive Engineering emerged in the early 1980s in response to transformation in the workplace by two major sources (Gersh et al. 2005): (i) computer systems were escaping from the confines of machine rooms and thus design principles were needed to ensure than ordinary people would be able to use them and, (ii) Safety Critical Systems were becoming more complex and increasingly computer controlled; design principles were needed to ensure that teams of skilled technicians could operate them safely and efficiently. Otherwise, this emergence is also linked to the maturation of cognitive science into a discipline whose theories, models and methods are capable of guiding application.

This brief historical review show that Automatics, Automation, Cognition, Cognitive engineering, safety and risk engineering (Mercantini 2015) are closely linked for the design of artefacts that have to be associated with human beings. The combination of Automation with Cognition (and cognitive engineering) leads almost "naturally" to the idea of building new intelligent systems where human beings and artefacts can work together in a coherent organization to face complex tasks and problems. It implies new approaches and new tools to model, to analyse, to control, to predict, to prevent and to protect. The joint consideration of automation and cognition might lead to address automation issues with a more comprehensive and coherent vision, which should lead to the design of new tools marked of consistency.

From a methodological perspective, this paper shows the importance of ontologies to jointly considere automation and cognition with the purpose to minimize human errors within piloting activities of complex systems. Ontologies constitute fundamental tools (i) for structuring a domain (at the conceptual level) as perceived by its actors and (ii) for building computer tools dedicated to assist human actors in solving complex problems in that domain. The Knowledge Oriented Design method (KOD) (Vogel 1988), originally designed to develop Knowledge Based Systems, has been used to elaborate domain or application ontologies.

After describing problems due to complex system piloting, a methodological process is proposed to tackle them with a cognitive perspective, by the use of the KOD method. Results obtained by applying this methodological process to a chosen case is presented and discussed. Finally, we conclude on the suitability of the methodological process proposed to take in account cognition in automation design to minimize errors.

2. ANALYSIS OF PROBLEMS DUE TO COMPLEX SYSTEM PILOTING

2.1. Analysis of the complexity

The generic functional representation of a dynamic system is conventionally represented by a feedback loop (Figure 1). These dynamic systems will be qualified as to be complex because composed of Humans interacting with Intelligent Machines (HIM), and being in charge to accomplish complex tasks in a collaborative way. Performances of these systems, in terms of stability, robustness, adaptation and resilience, strongly depend on the behaviour of the HIM duo. The objectives of these systems can be declined in terms of productivity, reliability, availability, security, quality, but also protection of the environment, risk, or any other objectives more specific to the nature of the piloted process, which can itself be partially or fully automated.

The piloting systems, depending on the nature of the process and the expected performances, can be classified according to different levels of complexity (Table 1). Level 0 and 1 correspond to "classical" commands of the analog or digital type without taking into account the human factor. The levels from 2 to 5 correspond to piloting systems where human supervisors are cooperatively associated to intelligent systems for process control and monitoring and problem solving assistance (CCM or DCCM, in table 1). Human supervisors constitute a homogeneous team (HoHST) when they are trained to work together to perform complex tasks related to the process. They constitute a heterogeneous team (HeHST) when they are coming from diverse origins (cultural, professional, social, academic, etc.) and have not been trained to work together. They may even have opposite objectives and opposite decisions to pilot the process, like it is often the case in crisis situation. In both cases (HoHST et HeHST), human errors have to be taken in account.

Controlled processes may also be classified according to their level of complexity (Table 2). Levels 0, 1 and 2 correspond to processes consisting of more or less complex artificial machines, from a simple machine to an automated industrial plant, without taking in account human operator teams. From level 3 to 5, human operator teams are considered within the automated industrial plant to form a complex system. Level 5 corresponds to complex large-scale systems, that is to say a complete territory that may consist of several industrial systems, an ecosystem, a population and intervention teams. The human component may correspond to operator teams who work in contact with the machines (HoHOT) or to a heterogeneous set of operators in the case of co-activities or dysfunctional or accidental situations, including the intervention teams (HeHOT = HoHOT + external agents). In all cases, the controlled process may be in a "normal functioning state" or in an "abnormal functioning state" that is to say, it may be a faulty process or within a risky or accidental situation.

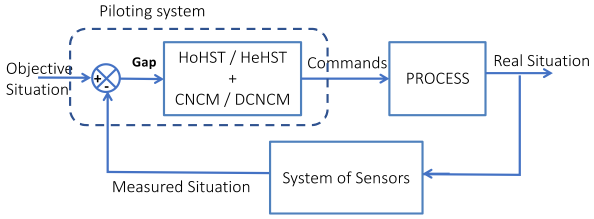


Figure 1: functional representation of dynamic systems

Table 1: Complexities of the piloting systems

Piloting Systems
Complexity Level : Nature
Level 0 : Analog Control Level 1 : Computer Numerical Control Level 2 : CCM + HoHST Level 3 : CCM + HeHST Level 4 : DCCM + HoHST Level 5 : DCCM + HeHST
CCM : Computer Control and Monitoring DCNCM : Distributed Computer Control and Monitoring HoHST : Homogeneous Human Supervisor Team HeHST : Heterogeneous Human Supervisor Team

Table 2: Complexities of the controlled processes

The Controlled Process
Complexity Level : Nature
Level 0 : Electromechanical Machine Level 1 : Automated Machine Level 2 : Industrial Plant (automated system) Level 3 : Industrial Plant + HoHOT Level 4 : Industrial Plant + HeHOT Level 5 : Territory + HeHOT
HoHOT : Homogeneous Human Operator Team HeHOT : Heterogeneous Human Operator Team

2.2 The Human-Machine Cooperation

At the level of the piloting system, Human-Machine (H-M) cooperation has been the subject of numerous research studies since the 1980s (Millot 1999)(Millot 2012)(Aguiar 2015)(Benloucif 2018), questioning the automation and optimization of the distribution of supervisory tasks, the ergonomics of the H-M relationship and the behaviour of human operators and supervisors facing diverse work situations.

According to (Millot 1999), H-M cooperation can take two structural forms: the vertical and horizontal structures. With the vertical structure (or hierarchical structure), the human operator / supervisor is responsible for generating all orders. It can use a computer tool for decision support or problem solving support. With the horizontal structure (or heterarchical structure), the decision-support or problem-solving support computer tool is also connected to the control inputs of the process. It becomes an agent at the same hierarchical level as the human operator / supervisor. The problem that arises then is the dynamic distribution of tasks between man and machine.

2.3 The Human Errors

Whether at the level of the controlled process or the control system, the human component regularly and inevitably produces errors that can be interpreted as the result of dysfunctions of cognitive functions such as perception, recognition, comprehension, interpretation, planning. , action, etc. Many authors have studied this problem of human error since the 1980s. Among those that have strongly influenced scientific advances in this area are (Amalberti 1996, 1999) (Hollnagel 1998) (Rasmussen 1982) (Reason 1990) (Vanderhaegen 2003). The results obtained make it possible to better understand their classification, their genesis, their causes, their consequences or their statistics. These human errors are naturally superimposed on the problems of H-M cooperation and those of the complexity of dynamic systems, making the control of work situations more and more complex.

If this complexity can be controlled and mastered in "normal" situations, it can become a real source of danger in critical situations where decisions must be taken and executed under high stress. In this context, the design of new software tools to support piloting tasks must take into account the experience and vision of implied actors according to the issues raised by the complexity of critical situations. Errors and their uncontrolled propagation can call into question the stability of the system or aggravate its state according to whether it is in a normal functionning state or an abnormal functionning state. In both cases, there is the problem of governability, accident avoidance or piloting within accidental situation.

The treatment of errors, with a view to minimizing their occurrence, propagation and consequences, is based on a set of measures that can be combined:

- the training of operators / supervisors on simulator,
- the development of decision support tools that can be integrated in a vertical or horizontal structure,
- the design of these help tools as well as those dedicated to the control and monitoring according to a cognitive logic similar to that of their users (cognitive ergonomics),
- the development of automatic error detection functions and filtering,
- the experience feedback to improve training, procedures, tools and process.

3. METHODOLOGICAL APPROACH

3.1. The methodological process

The proposed methodological approach is based on the assumption that reducing the occurrence and severity of the consequences of pilot errors, despite the increasing complexity of work situations, requires the coherence of conceptual representations of each agent, whether human or artificial, as well as their communication languages. Ontologies and works currently developed by the community of cognitive and knowledge engineers can provide relevant answers to problems raised in the previous paragraph.

The term *ontology* is often associated to the knowledge related to objects of a delimited universe and their relations. Ontology refers to a conceptual language used for the description of this delimited universe (domain). A domain ontology is an example of knowledge level model (Ushold 1998). The emergence of this notion in Knowledge Base System (KBS) engineering comes from the fact that the way to observe the world and its

interpretation are directly dependent of the observer culture, his (her) means to observe it as well as to his (her) intentions. One of the objectives of ontologies is to facilitate the exchange of knowledge between humans, between humans and machines as well as humans via machines (Ushold 1996). In this sense, it becomes necessary to resolve the difficulties caused by observation, representation and interpretation of (normal or critic) situations to facilitate problem solving (intent).
Ontologies can also be defined according to their level of genericity as proposed by Guarino in (Guarino 1998) (Figure 2). The so-called top-level ontologies describe very generic concepts independent of any particular problem or area. They must be "reusable from one domain to another and are designed to reduce inconsistencies in terms defined downstream" (Vandecasteele, 2013). Domain ontologies and task ontologies respectively describe the concepts of a generic domain (such as medicine, production, accidentology, etc.) or the concepts of a generic task (or problem) (such as diagnosis, prognosis, planning, simulation, etc.). They specialize terms introduced by high-level ontologies. Application ontologies (the most specific) describe concepts related to a task (or problem) occurring in a particular field (such as medical diagnosis, road traffic accident diagnosis, industrial planning, etc.) . They are both a union and a specialization of ontologies of tasks and domains (Maedche and Staab, 2001).

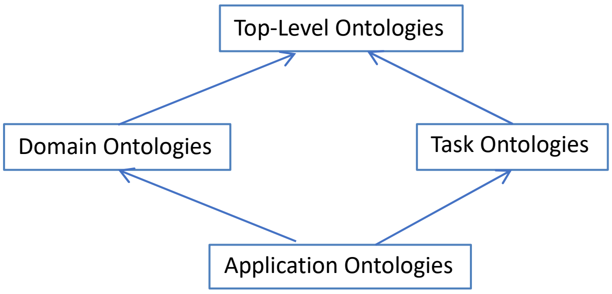


Figure 2 : Classification of ontologies according to their genericity. Arrows represent specialization relationships. From (Guarino 1998).

The proposed methodological process (Figure 3) consists in adopting approaches and methods from Knowledge Engineering (KE) combined with formal modelling. KE approach consists in developing application ontologies aiming to model in a unified way the triplet Td = <Domain, Problem, Method>. In this sense, the ontology structures the Domain according to the Problem to be solved and taking into account the Problem Solving Methods. Tools so built are carrying knowledge shared by actors of a domain, what makes them more effective to accomplish complex tasks in a collaborative way within normal or critical situations.
The inductive process is based on a corpus of documents describing each element of the Td triplet: the Domain corpus, the Problem corpus and the Method corpus. The corpus constitution is really a fundamental step of the process because it has to content an exhaustive knowledge. To illustrate our discourse, previous works can be cited as examples: traffic road accident (Mercantini et al. 2003), aircraft piloting errors (Sadok et al. 2006), industrial plant piloting errors (Mercantini et al. 2004), accidental seaside pollution (Mercantini 2015) or simulation of supply chain vulnerability (Sakli et al 2018).
The Domain corpus must encompass the set of knowledge defining the limits and a deep description of “what is the Domain”. It gives a pertinent vision of the cultural dimension of the Domain actors and the different ways the domain can be perceived.
The Problem corpus must encompass a set of representative (pertinent) practical cases of the studied problem. The aim is to get a complete vision of what could happen and the different forms they are taking on. The Problem corpus give a pertinent vision of the wrong behaviours of the Domain actors.
The Method corpus must encompass a set of representative practical cases of the implemented methods to solve the studied problem (practical technics, good practices, formal procedures, quality procedures, etc.). The Method corpus give a pertinent vision of the actors “Know How” of the Domain.
On the second step of the process, the ontology elaboration is based on the "Knowledge Oriented Design” (KOD) method (Vogel 1988). KOD was designed to guide the knowledge engineer in its task of developing knowledge based systems. This method was designed to introduce an explicit model between the formulation of a problem in natural language and its representation in the chosen formal language. The inductive process of KOD is based on the analysis of a corpus of documents, speeches and comments from domain experts, in such a way to express an explicit cognitive model (also called conceptual model).
Depending on the type of result desired, the third step of the process is to use the application ontology to perform one or a combination of the following operations: writing specifications, formal modelling, software modelling. The dashed arrows symbolize this choice or combination.
The final fourth step is the production of the tool. It can be a software tool (computer tool for decision support, problem solving support or simulation), a methodological tool (not necessary computerized), a formal model, a mathematical tool.

3.2. The KOD method

KOD is based on an inductive approach to explicitly express a cognitive model (or conceptual model) based on a corpus of documents, comments and experts’ statements. The main features of this method are based on linguistics and anthropological principles. Its linguistics basis makes it well suited for the acquisition of knowledge expressed in natural language. Thus, it proposes a methodological framework to guide the collection of terms and to organize them based on a terminological analysis (linguistic capacity). Through its anthropological basis, KOD provides a methodological framework, facilitating the semantic analysis of the terminology used to produce a cognitive model (conceptualisation capacity). It guides the work of the knowledge engineer from the extraction of knowledge to the development of the conceptual model.

The implementation of the KOD method is based on the development of three successive models: the practical models, the cognitive model and the software model (Table 1). Each of these models is developed according to the three paradigms: <Representation, Action, Interpretation / Intention>.

The Representation paradigm gives the KOD method the ability to model the universe such as experts / actors represent it. This universe is made of concrete or abstract objects in relation. The KOD method provides methodological tools to develop the structure of this universe of knowledge according to this paradigm. The Action paradigm gives the KOD method the ability to model the behaviour of active objects that activate procedures upon receipt of messages. The Interpretation / Intention paradigm gives the KOD method the capability to model reasoning used by experts / actors to interpret situations and elaborate action plans related to their intentions (reasoning capacity).

The practical models are the representation of speeches or documents expressed in the terms of the domain, by means of “taxemes” (static representation of objects – French word), “actemes” (dynamic representation of objects – French word) and inferences (base of the cognitive reasoning pattern). A “taxeme” is a minimum grammatical feature; it is the verbalisation of an object or a class of objects. An “acteme” is the verbalisation of an act or a transformation, a unit of behaviour. An inference is the act or process of deriving logical conclusions from premises known or assumed to be true. The cognitive model is obtained by abstracting the practical models. The cognitive model is composed of taxonomies, actinomies and reasoning patterns. The software model results from the formalization of the cognitive model expressed in a formal language independently of any programming language.

3.3. The ontology building process using KOD

Research work in Ontology Engineering has highlighted five main steps for building ontologies (Dahlgren 1995; Uschold 1996; Aussenac-Gilles 2000; Gandon 2002):

1. *Ontology Specification*. The purpose of this step is to provide a description of the problem as well as the method to solve it. This step allows one to describe the objectives, scope and granularity of the ontology to be developped.
2. *Corpus Definition*. It consists to select among available information sources, those that will allow the objectives of the study to be attained.
3. *Linguistic Study of the Corpus*. It consists in a terminological analysis to extract the relevant terms and their relations. Linguistics is specially concerned to the extent that available data for ontology building are often expressed as linguistic expressions. The characterization of the sense of these linguistic expressions leads to determine contextual meanings.
4. *Conceptualization*. The candidate terms and their relations resulting from the linguistic study are analyzed. The relevant terms are transformed into concepts and their lexical relations are transformed in semantic relations. The result of this step is a conceptual model.
5. *Formalization*. The step consists in expressing the conceptual model by means of a formal language.

The projection of the KOD method on the general approach for developing ontology shows that KOD guides the corpus constitution and provides the tools to meet the operational steps 3 (linguistic study) and 4 (conceptualization) (Table 2).

4. CASE STUDY

4.1. The CLARA 2 project

The purpose of the CLARA 2 (Calculations Relating to Accidental Releases in the Mediterranean) project is to design a problem solving software to assist stakeholders from crisis centres to plan fight actions against marine pollutions (hydrocarbon and chemical products) in Mediterranean area. Stakeholders usually implied in an crisis centre for managing maritime accidents are: the Navy, the National Administrations, the local administrations, the National Meteorology and expert institutes like the French Research Institute for Exploitation of the Sea (IFREMER) or the Centre of Documentation, Research and Experimentation on Accidental Water Pollution (CEDRE). Managing such accidents generates complex and critical work situations. According to table 1 and 2, the complexities of the piloting system and of the controlled process are at level 5, and the structural form of the H-M cooperation is vertical. The potential users of the tool are experts from CEDRE.

Decisions and actions undertaken by crisis center Stakeholders need to mobilize a large number of information from various sources and under high time pressure. These information need to be integrated in a coherent way prior to be interpreted and finally to be the base of any decision and action. Among the main activities carried out by operational center actors it can be cited: situation acquiring, situation analysis, determining fight strategies, choosing the right fight

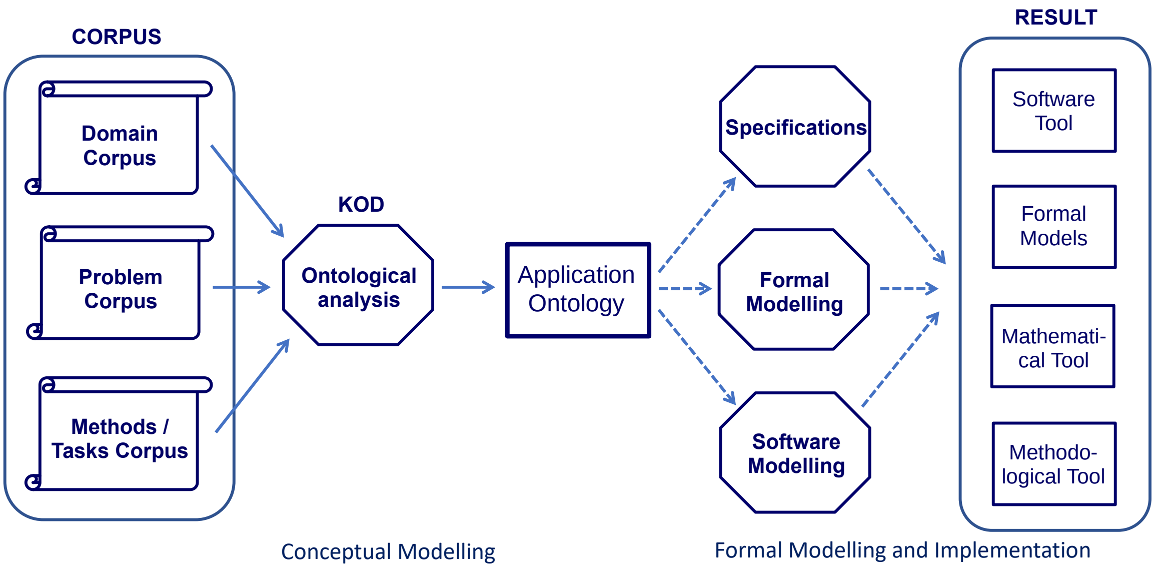


Figure 3: The generic methodological process based on application ontologies to produce specific tools

Table 1. KOD, the three modelling levels according to the three paradigms.

Paradigms Models	Representation	Action	Interpretation
Practical	Taxeme: object static representation	Acteme: dynamic representation of active objects	Inferences
Cognitive	Taxonomy: object static organization according to theirs properties	Actinomy: dynamic object organization	Reasoning Pattern
Software	Classes	Methods	Rules

Table 2. Integration of the KOD method into the elaboration process of ontology.

Elaboration process of Ontology	KOD process	Elaboration process of ontology with KOD
1. Specification 2. Corpus definition 3. Linguistic study 4. Conceptualisation 5. Formalisation	1. Practical Models 2. Cognitive Model 3. Software Model	1. Specification 2. Corpus definition 3. Practical Models 4. Cognitive Model 5. Formalisation 6. Software Model

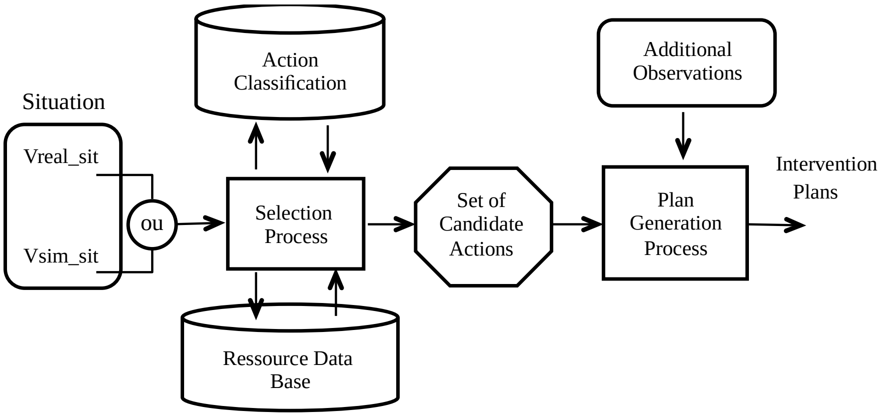


Figure 4: Data flow diagram of the GENEPI module from (Mercantini 2015b)

strategies, choosing the right fight actions, elaborating fight action plans and anticipating future situations.

In this paper, the focus is on the implementation of the generic process (Figure 3) for the study of the GENEPI module (the Generation Module of Intervention Plans – Figure 4) integrated into the CLARA 2 project. See (Mercantini 2015b) for a wider and deeper presentation.

4.2. Elaboration of the Application Ontology

4.2.1 Ontology specification

The domain is that of maritime accidents with the release of pollutant products (hydrocarbon or chemical) and causing a marine pollution. The problem is to assist crisis management teams to elaborate action plan to fight the pollution. The problem solving method consists in the elaboration of a cooperative software tool, which implement the generation process of fight actions against marine pollutions.

4.2.2 Corpus Definition

Documents to be collected must be both representative of the triplet <Domain, Problem, Method> and meet the criteria of suitability required by the three paradigms <Representation, Action, Interpretation / Intention>. The corpus has been established on the basis of documents from CEDRE and REMPEC (the REgional Marine Pollution Emergency Response Centre for the Mediterranean Sea). The types of documents that make up this corpus are the following:

- Documents relating to the evaluation of each fight technique or method,
- Documents about the general organization of emergency plans (plan ORSEC: Organization of the Civil Security Response),
- Return on experience documents about the major maritime disasters such as that of the Erika, Prestige, etc..
- Return on experience documents about maritime accidents of lower magnitudes.
- Quality procedures (from CEDRE) for crisis or accidental event management.

4.2.3 The Practical models

This phase consists in extracting from each document of the corpus, all the elements (objects, actions, and inferences) that are relevant to accident representation and fight action implementation.

Taxeme Modelling

The linguistic analysis is performed in two steps: verbalization and modelling into taxems. Verbalization consists in paraphrasing corpus documents in order to obtain simple sentences allowing to qualify the employed terms. Modelling consists in organizing terms representing objects and concepts of the triplet Td by means of binary predicates such as <Object, attribute, value>. Attribute defines a relationship between the object and a value. Five kinds of predicative relationships are defined: Classifying (is-a, type-of), Identifying (is), Descriptive (position, failure mode, error mode, cause…), Structural (composed-of) and Situational (is-in, is-below, …).

The following example is an extract from the “Prestige” oil tanker accident.

“... On November 13th, 2002, the Prestige oil tanker flying the Bahamian flag, sends an emergency message from the Finisterre Cape ...”

Paraphrases

1. The Prestige is a oil tanker
2. The Prestige flies the flag of the Bahamas
3. On November 13, The Prestige is located at the Finisterre Cape
4. On November 13, the Prestige sends an emergency message

Taxems

1. <Prestige, IS A, oil tanker>
2. <Prestige, FLAG, Bahamas>
3. <Prestige, LOCATION, Finisterre Cape>
4. <Prestige, DATE, November 13th>

The last paraphrase is related to an action, it will be modelled as an actem. The extent of this analysis at the whole Corpus, has produced to the set of taxems needed for the representation of the universe. An object of the real world is modelled by the sum of related taxemes.

Acteme Modelling

Obtaining actemes consists in identifying verbs of the corpus documents that represent activities (or tasks) carried out by human or artificial operators. An activity is performed by an action manager, by means of instruments, to modify the state of the addressee. The following example illustrates how to extract actemes from the Corpus.

“... the Prestige sends an emergency message...”

The activity is “SENDING an emergency message” and it is translated into a 7-tuple (the acteme):

<Action Manager, Action, Addressee, Properties, State1, State2, Instruments>

Where: Action Manager performs the action; Action causes the change; Addressee undergoes the action; Properties represent the way the action is performed; State 1 is the state of the addressee before the change; State 2 is the state of the addressee after the change; Instruments, are means used to cause the change.

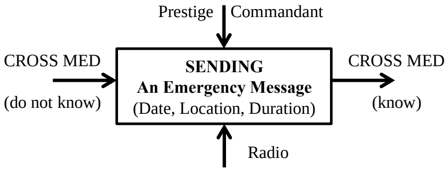
The actem “SENDING an emergency message” is represented as following:

<Prestige Commandant, SENDING an emergency message, CROSS MED, (date, location, duration), CROSS MED (do not know), CROSS MED (know), Radio>.

CROSS MED (Centre Régional Opérationnel de Secours et de Sauvetage en Méditerranée), is the French organism that receives emergency messages from ships. Figure 5 illustrates this acteme and the case of a fight action where the formalism has been extended to take in account suitability criteria:

<Action Manager, Action, Addressee, Properties, Suitability Criteria, State1, State2, Instruments>

Each element of the 7-tuple (or 8-tuple for fight actions) must be previously defined as a taxeme.



Action: FLUSHING	
Components	**Values**
Action Manager: Operator	{Human Means}
Addressee: Substratum	{Sand, Stone, Concrete, Rock, etc.}
Addressee State1	{Polluted, Cleaned}
Addressee State2	{Polluted, Cleaned}
Instruments	{Pump + Water_Hose + Recovery_Means}
Properties	Efficiency
Suitability Criteria	Viscosity Pollutant, Pollution level, Kind Of Substratum

Figure 5: Two examples of actemes. One is in a datagram form (SENDING An Emergency Message) and the other (FLUSHING) in a the table form.

Inferences Modelling

Inferences modelling consists in representing the elements of the corpus that characterize the cognitive activities of humans or machines.

Inferences are the basic elements of the Interpretation / Intention paradigm.

In this study, the Interpretation addresses pollution situations and the Intention concerns fight action planning. Premise propositions are resulting from the interpretation of the situation observed. The conclusion is related to choose (or not) actions (planning).

Let us consider the following example:

"*... dispersants should not be used in areas where water circulation is not good, close to spawning, coral reefs, shell deposits, wetlands and industrial water intakes...* " where the following inferences have been produced:

IF *spawning areas close* THEN *do not use dispersants*
IF *coral reefs close* THEN *do not use dispersants*
IF *shell deposits close* THEN *do not use dispersants*
IF *industrial water intakes close* THEN *do not use dispersants*

Where *spawning areas close*, *coral reefs close*, *shell deposits close* and *industrial water intakes close* are premise propositions. The observation and interpretation will give them the value True or False. To use dispersants, all the values have to be True. The suitability criteria associated to each fight action are the result of inference analyses.

4.2.4 The cognitive model (conceptualisation)

It consists in developing the cognitive model by abstraction of the practical models. The abstraction from practical models into a cognitive model is based on the operation of classification to produce taxonomies, actinomies and patterns of reasoning.

Taxonomy Building

The first step consists in solving problems induced by homonym and synonym terms, with the objective to build a coherent and common terminology.

The second step consists in analysing the nature of attributes (or relationships) that characterize each object. From the nature of these attributes will depend the building of taxonomies (relationships "kind-of" or "is-a") or others kinds of tree structures (relationships "is-composed-of", “is-on", etc.).

As an example, the term “Skimmer” is meaningful and thus it deserves the concept status. It is significant of a set of recovery devices (previously modelled by means of taxems). As a result of the analysis of the knowledge related to “Skimmer”, the taxonomy of the figure 6 has been built and the “Skimmer” concept is defined through his attributes as follow:

Skimmer attritutes:

<Type, Flow, Quantity, Storage Location, City, Dimension, Weight, Performance Limit, Selectivity, Recovery Rate>

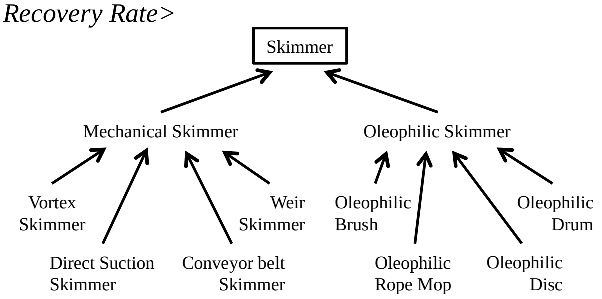


Figure 6: The Skimmer taxonomy (“kind-of” relation)

All the taxems of the corpus are organized in taxonomies and each concept is defined by a set of attributes.

Actems abstraction

One result of the actem analysis is that actems can be organized into five main action categories:

- Actions related to pollutant behaviour,
- Actions related to stricken ship behaviour,
- Actions related to reasoning patterns,
- Actions related to CLARA 2 services,
- Actions related to operations against pollution.

Amongst actions related to pollutant behaviour it can be cited: Evaporation, Dissolution, Drift, Emulsion, etc.

Amongst actions related to stricken ship behaviour, it can be cited: Listing to starboard, Sinking, Sending an emergency message, Requesting evacuation, etc.

The actions related to reasoning patterns such as « Choosing the shoreline clean-up methods » are used to select or to plan fight actions. To be performed, they use the suitability criteria associated to each actem.

The actions of the CLARA 2 services category are implemented to improve the GENEPI functionalities. As examples: Coastal Mapping, Evaluating the Pollution Movement, Evaluating the Pollution Impact.

The actions of the last category are fight actions. They are divided into two main classes: the shoreline clean-up methods and the clean-up methods on the sea. The set of actems from this category has been structured by means of a Taxonomy. Figure 7 is an extract of this taxonomy. Some of these actems can be organized in a structural and temporal way to form actinomies. The interest of this kind of structure is that actions are already planned.

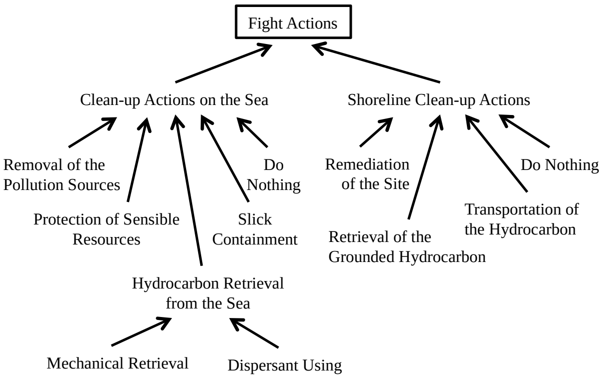

Fight Actions
Clean-up Actions on the Sea
Shoreline Clean-up Actions
Removal of the Pollution Sources
Do Nothing
Remediation of the Site
Do Nothing
Protection of Sensible Resources
Slick Containment
Retrieval of the Grounded Hydrocarbon
Transportation of the Hydrocarbon
Hydrocarbon Retrieval from the Sea
Mechanical Retrieval
Dispersant Using

Figure 7: Extract of the Fight Action Taxonomy

4.3. Architecture of the GENEPI module

The architecture of the GENEPI module (Figure 7) has been designed around the ontology enriched with the instances of the concrete classes to constitute the knowledge base (Maedche 2002). For the formal representation of the GENEPI ontology, the frame-based language of the Protégé platform has been used.

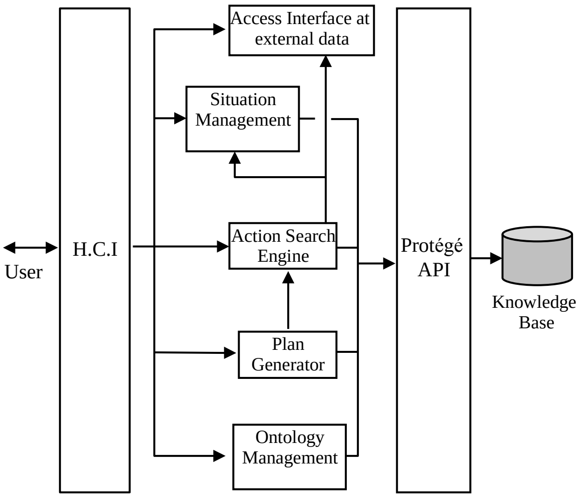


Figure 7: Architecture of the GENEPI module

4.3.1 The Situation Management

Each accident has its own characteristics and for a particular accident, circumstances and context change from one moment to another. To take this into consideration, the notion of Situation is defined. A Situation consists of a set of attributes (S) that characterizes accident and its context. The set of these attributes is a superset of the set of suitability criteria (Ca) associated to fight actions. Thus, attributes common to Ca and S have the same types. Instances of the Situation are obtained from data delivered by the access interface to external data (coming from others CLARA2 modules), and from data supplied by the user (Figure 7).

4.3.2 The Action Search Engine

The search engine receives as input the Situation. As results, it provides four sets of fight actions:

- The set A, which contains the actions where all criteria are verified,
- The set B, which contains the actions where at least one of the criteria could not be assessed by lack of information in the situation,
- The set C, which contains the shares of which at least one criterion was not satisfied,
- The set D, which contains the actions of the set B enriched by criteria not assessed.

Rules to select fight actions are based on the suitability criteria and values taken by the corresponding attributes of the situation. Rules are of the form:

$$c1 \wedge c2 \wedge ... \wedge cn \rightarrow True / False$$

With c1, c2, ... cn, the criteria associated to a fight action. The conclusion of the rule is about the possibility whether or not to select the action. A criterion is satisfied if the value taken by the corresponding attribute of the situation is compatible the criterion constraints.

Upon the receipt of the Situation, the action-selecting algorithm analyzes actems. From each actem, it extracts the criteria and it applies the selection rules previously presented. According to the results obtained, the actem is placed in the corresponding set (A, B, C or D).

After running the algorithm, if the user is not satisfied with the result, it can enrich the initial situation to assess the criteria that have not been. This new running should reduce the size of the B set, by transferring actions in the set A or in the set C. The algorithm is independent of changes in the ontology.

4.3.3 The Plan Generator

Fight action plans are the result of a collaborative work between GENEPI and the user. From the set A (set of actions where all criteria are satisfied), the user selects actions to constitute the Plan. Once the actions are selected, the Plan Generator produces a document where every action is completely defined: a detailed description of the fight action, a detailed description of human and material means required for its implementation, a detailed description of precautions and safety measures to be followed for its implementation, a reminder of the suitability criteria.

4.3.4 The ontology management module

This module provides users with the functions needed for maintenance (updating, adding and deleting classes, attributes and instances) and consultation (searching knowledge) of the ontology.

3. Conclusion

The aim of this paper was to show that cognitive approaches offer powerful engineering environments to tackle the issues raised by complex system piloting. The responses proposed concerns the design of intelligent machines to assist operators and supervisors in their tasks of problem-solving and decision-making with the purpose to minimize piloting errors.

The methodological process proposed is based on the elaboration of an application ontology combined with the use of formal languages. The purpose of that ontology is to structure the domain according to the problem to solve

and to the problem solving method (the conceptual model). The ontology is obtained by means of a cognitive approach, which consists in applying the KOD method, which has proven to be adequate. The choice of the formal language depends of the final resulting tool.
To illustrate the process implementation, the case study of the CLARA 2 project has been presented.

REFERENCES

Aguiar Y. P.C., Vieira M. F. Q., Galy-Mari E., Santoni C., 2015. Analysis of the User Behaviour When Interacting with Systems During Critical Situations. In: Mercantini JM., Faucher C. (Editors), *Risk and Cognition*, Springer-Verlag, pp 129-154. Intelligent Systems Reference Library, Vol. 80, January 2015.

Amalberti R., 1996. *La conduite des systèmes à risques*. Le travail humain, Editions PUF.

Amalberti R., 1999. Dysfonctionnement des systèmes et dysfonctionnement de la cognition. In J.-G. Ganascia (Coordonnateur), *Sécurité et cognition* (pp. 185-201). Éditions HERMES, Paris.

Aussenac-Gilles N., Biébow B. and Szulman S., 2000. Revisiting Ontology Design: a method based on corpus analysis. EKAW-2000, Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management, R. Dieng et O. Corby (éd.), LNAI 1937, Springer, pp. 172-188.

Benloucif M. A., 2018. Coopération homme-machine multi-niveau entre le conducteur et un système d'automatisation de la conduite. Interface homme-machine [cs.HC]. Université de Valenciennes et du Hainaut-Cambresis, 2018. Français. NNT : 2018VALE0012.

Ganascia J.G., 1999. *Sécurité et cognition*. Hermes, Paris.

Gandon F., 2002. Ontology engineering: a survey and a return on experience. Research Report n° 4396. INRIA Sophia-Antipolis, mars 2002.

Gersh J.R., McKneely J.A., Remington R.W., 2005. Cognitive Engineering: Understanding Human Interaction with Complex Systems. *Johns Hopkins APL Technical Digest*, *26*(4), 377-382.

Gruber T.R., 1993. A translation approach to portable ontology specifications. *Knowledge acquisition*, pp 199-220.

Hollnagel E., 1998. *Cognitive reliability and Error Analysis Method (CREAM)*. Elsevier Sciences Ltd, Oxford.

Lee J.D., Kirlik A., 2013. Cognitive Engineering: History and Foundations. In Lee, J.D., Kirlik, A. (Ed.), *The Oxford Handbook of Cognitive Engineering*. Oxford, New York: Oxford University Press.

Mercantini J.-M., Capus L., Chouraqui E., Tourigny N., 2003. Knowledge Engineering contributions in traffic road accident analysis. In: Ravi K. Jain, Ajith Abraham, Colette Faucher, Berend Jan Van der Zwaag, eds. Innovations in Knowledge Engineering. pp 211-244.

Mercantini J.-M., Turnell M.F.Q.V, Guerrero C.V.S, Chouraqui E., Vieira F.A.Q et Pereira M.R.B, 2004. Human centred modelling of incident scenarios. IEEE SMC 2004, Proceedings of the International Conference on Systems, Man & Cybernetics, pp. 893-898. October 10-13, The Hague, The Netherlands.

Mercantini J.-M., 2015a. Introduction. In: Mercantini JM. and Faucher C. (Editors), *Risk and Cognition*, Springer-Verlag, pp 1-20. Intelligent Systems Reference Library, Vol. 80, January 2015.

Mercantini J.-M., 2015b. Building a domain ontology to design a decision support software to plan fight actions against marine pollutions. In: Mercantini JM. and Faucher C. (Editors), *Risk and Cognition*, Springer-Verlag, pp 197-227. Intelligent Systems Reference Library, Vol. 80, January 2015.

Millot P., 1999. La supervision et la coopération homme-machine dans les grands systèmes industriels ou de transport. In J.-G. Ganascia (Coordonnateur), *Sécurité et cognition* (pp. 125-145). Éditions HERMES, Paris.

Millot P. and Boy G. A., 2012. Human-machine cooperation : a solution for life-critical systems ?. *Work*, vol. 41, no. Supplement 1, pp. 4552–4559.

Rasmussen J., 1982. Human Errors: a Taxonomy for Describing Human Malfunctions in Industrial Installations. *Journal of Occupational Accidents*, v.4, p.311-35.

Reason J., 1990. *L'Erreur Humaine*. Paris, PUF.

Sadok S., Chouraqui E., Mercantini J.-M., 2006. Modeling the deviation to improve safety aircraft piloting activities. In: *Proceeding of The 1st IEE International Conference on System Safety*, 6-8 June 2006, London.

Sakli L., Mercantini J.-M., Hennet J.-C., 2018. Study of Supply Chain Vulnerability Based on Cognitive Engineering and ARIMA Formal Models. In: *Proceeding of the 11th International Conference on Integrated Modelling and Analysis in Applied Control and Automation*, 17-19 September 2018, Budapest.

Tiberghien G., 2002. *Dictionnaire des sciences cognitives*. Colin, A., Paris.

Uschold M., 1998. Knowledge level modelling: concepts and terminology. *The Knowledge Engineering Review*, vol. 13, n° 1, 5-29.

Uschold M. and Grüninger M., 1996. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, volume 11, n° 2, pp. 93-136.

Vanderhaegen F., 2003. *Analyse et contrôle de l'Erreur Humaine*. Lavoisier Hermes Science publications, Paris. ISBN 2-7462-0722-2

Vogel C., 1988. *Génie cognitive*. Paris, Masson (Sciences cognitives).

Wiener N., 1948. *Cybernetics: Or Control and Communication in the Animal and the Machine*. Cambridge, Massachusetts: MIT Press.

Wikipedia 2019. https://fr.wikipedia.org/wiki/Cybernétique