Proposal of an ODX Data Exchange System Applied to Automotive Diagnosis

A. Cremmel[*,**], A. Azarian[*,**], A. Siadat[**] *(IEEE Member)*
* SIEMENS AG A&D AS AP SE 3 IC, Siemensallee 84, 76187Karlsruhe, GERMANY
**LGIPM-ENSAM, 4 Rue Augustin Fresnel, 57078 Metz, FRANCE
email: cardiagnosis@aureliencremmel.com ;armin.azarian@siemens.com; ali.siadat@ensam.fr

***Abstract* — One challenge today in the automotive industry is to handle the increasing multitude of product variants. The after sales network has to manage an increasing amount of data in particular data related to the communication with ECUs (Electronic Control Units) described in ODX (Open Data eXchange) files. The problem of ODX files is that they contain rough communication data that is specific to each ECU, so that knowledge is not well abstracted. In order to simplify the tasks of the authors and avoid data redundancies we decided to develop an intelligent import module for the car-diagnosis software SIDIS-Entreprise developed by Siemens AG. This module is based on a multi-agent system that analyses the content of the ODX files related to the vehicle communication and classifies the ECU jobs on a abstracted functional level.**

***Keywords* — Car diagnosis, ODX (Open Data eXchange), ECU Communication, Automotive, Multi-Agent Systems, Data-Mining**

I. INTRODUCTION

The distinguishing of products across the product line to meet the demands of customers is made more and more difficult because of the ever increasing multitude of product variants, options and niche models. In this context, about 90% of innovations are in the field of automotive electronics, which represents 30% of the car's value, as opposed to 0.5% in the 1980's; and the trend is increasing. The consequence is that diagnosis systems of the after-sales network have to manage this increasing amount of data. SIDIS-Enterprise is a tool for the diagnosis of car breakdowns which is developed by the Siemens AG to meet this requirement and is used by technicians in after-sales network of car manufacturers. The localization of breakdowns is based on the acquisition of defaults codes from ECUs (Electronic Control Units) and perceived symptoms. The communication with ECUs is a fundamental aspect at diagnostic session because they send the default's codes and the related environment description data which permits to suspect components. Regarding the increasing number of vehicle variants and different features of the ECUs, the authors of diagnosis programs need assistance in the editing process to manage the increasing amount of data.

A. Structure of SIDIS-Enterprise

SIDIS-Enterprise consists in two main parts: the first is the Authoring system which allows to edit the knowledge bases and the second is the workshop system which is used in the after-sale network of the car's manufacturer to perform diagnosis session. The data is organized in trees where child nodes inherit the properties and attributes of parent nodes. The knowledge related to the diagnosis is organized in following structures (cf. Figure 1):

- **The diagnosis object tree** which represents a hierarchical model of the car. A service program is associated to each node in order to test the corresponding component.
- **The fault code tree** which contains the fault memory models of the ECUs (or default's code)
- **The ECU Jobs tree** which contains a description of ECU-jobs (diagnostic communication objects) of the ECUS, including request and response parameters
- **The ECU function tree** which represents the ECU-jobs on an abstracted and functionalized level
- **The perceived symptom tree** which contains descriptions of the client's symptom
- **The characteristic tree** is composed of the different characteristics and equipments of vehicles
- **The suspicion links** are links between fault codes or symptoms and diagnosis objects.

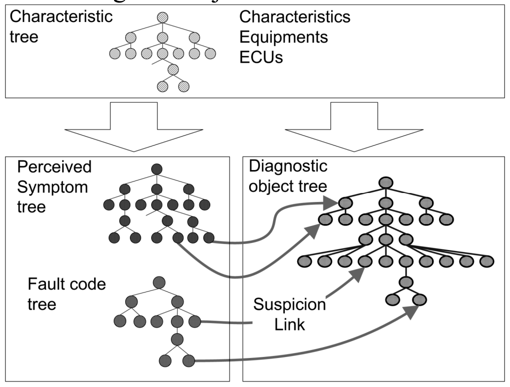


Figure 1: Diagnostic related knowledge structures

Links between nodes in the characteristic tree and nodes in the diagnostic objects tree make it possible to determine the diagnostic objects automatically according to the characteristics chosen by the technician at the beginning of the diagnostic session. This happens while considering only the diagnostic objects corresponding to the characteristics of the car (model range, construction year, country, etc.) and ignoring the optional equipment that is not part of the vehicle. The selection of the characteristics of the vehicle can be realized manually or automatically by reading the VIN (Vehicle Identification Number) using the vehicle

communication. Once the vehicle is identified, the technician initializes the diagnosis session by selecting perceived symptoms like: “Air conditioning is not working” and the acquisition of fault codes from the ECUs of the car. The suspicion links edited by the authors will determine a failure probability of the nodes in the diagnostic object tree. By successive tests called “service programs” linked to these nodes, the algorithm will incriminate or discriminate suspect components. In order to facilitate the writing of these service programs, we wish that the author refers to abstract fully functionalized objects describing the ECU communication elements. The use of this method implies that the amount of work can be reduced for him and that tests can be written in a generic way, which would increase their robustness. The ECU variant can be resolved during the identification step of a diagnostic session. The crucial issue in this method is to reduce the complexity of the information to administrate for the authors due to the increasing number of vehicle variants.
Despite the complexity, this paper is concerned only with the ODX (Open Data eXchange) files parts that describe the ECU communication objects. The next sections give an overview of the ODX Data model and describe a use-case of the selected approach with the developed import module.

B. ODX Data Model

ODX (Open Data Exchange) is an international standard which has been established by the ASAM e.V. in order to define the requirements for ECU diagnostic and programming data. It covers the needs of the entire vehicle lifecycle from system engineering to the service shop. The standard [1] consists in a description of diagnostic data and communication interfaces of vehicle ECUs and includes the UML data model for ECU diagnostic and programming data as well as its XML implementation. ODX files consist in five main parts [2]:

• **Diagnostic protocols** (DIAG-LAYER-CONTAINER), where diagnostic communication objects, requests and responses of ECUs are defined.
• **Communication parameters** (COMPARAM-SPEC), where the parameters for communication (e.g. timings) are set.
• **Jobs for more than one ECU** (MULTIPLE-ECU-JOB-SPEC), where Java job code for several ECUs making use of the diagnostic communication objects is stored.
• **Vehicle information** (VEHICLE-INFO-SPEC), where topological information of vehicles, physical links and logical links to ECUs are described.
• **Flash data** (FLASH), where memory layout and ECU-programming data are to be found.

The DIAG-LAYER-CONTAINER and the MULTIPLE-ECU-JOB-SPEC are especially relevant for the diagnosis as they describe the ECU functionalities. Further on, their structure is also very interesting as a value inheritance mechanism is implemented to reduce considerably the amount of data. In order to provide a maximum of flexibility, two different inheritance mechanisms can be used: the Short-Name resolution principle and the ODX-Links. From the most general to the most specific diagnostic layer, the DIAG-LAYER-CONTAINER contains following information layers:

1. **ECU-Shared Data**: this layer provides a pool of information which is available for each ECU.
2. **Protocol Layer**: contains data which is standardized through the diagnostic protocol used.
3. **Functional-Group Layer**: contains data of ECUs having similar functionalities but dealing with different physical components (e.g. trunk doors and side doors)
4. **Base-Variant Layer**: this layer describes a group of ECUs which have similarities (for instance because they are produced by the same manufacturer)
5. **ECU-Variant Layer**: this layer contains the description of a real physical variant of an ECU and hence is the most specific one.

The principal information in the Diagnostic Layer is the DIAG-COMM object, which is a diagnostic communication element. The ODX model specifies two types of DIAG-COMMs, the DIAG-SERVICE, which defines the basic communication element and the SINGLE-ECU-JOBS, which consist in Java code and build on one or several DIAG-SERVICES to return interpreted output parameters. A DIAG-SERVICE refers to a request and to one or several responses (positive or negative), which are defined independently from the services. Each request or response is built with parameters, which are to be interpreted with independently defined DOPs (Data Object Properties).

Besides this information, the VEHICLE-INFO-SPEC data also plays a central role during the diagnosis session to identify the in vehicle ECUs using the vehicle identification data and vehicle topology description [3].

II. IMPORT OF ODX FILES

A. Difficulties to face

Both the ODX information structures and the actual Authoring System from Siemens AG have been developed in order to manage the great amount of data in the best possible way. In both cases, two main strategies have been identified to do so:

- Development of concepts to reduce information redundancy
- Development of concepts to make a functional-oriented description of data

The data reduction possibilities have been especially developed in the ODX data model with complex inheritance mechanisms, which imply that the first challenge to be faced when importing ODX files is to find a way to avoid duplicating information unnecessarily in the Authoring System. In the Authoring System, the tree structures form an object-oriented database which also offers possibilities to reduce data redundancy. The second challenge when trying to reduce data redundancy concerns the functional description of data. The ODX model is set up with objects such as functional classes which make it possible to link diagnosis data (e.g.

diagnostic services) with their functional description but the SIDIS-Enterprise model has to go much further because the signification of each object has to be determined to perform the car diagnosis. Hence, a special effort has to be made when importing the ODX information in order to understand the meaning of the data. The following points lead to special difficulties when wanting to perform an automatic import of ODX files in the Authoring System:

- The functional information is not always separated clearly from ECU-specific information, which means that the knowledge is not always described on an abstracted level
- For some elements, there is a lack of abstracted data
- Data redundancy

B. Possible approaches

As only selected data has to be imported in the Authoring System, an automated import of ODX data means defining which information has to be imported and how it has to be functionalized. A filtering of the ODX-data files using predefined rules should pre-select the interesting data for the diagnosis. After that a detailed analysis in a functional layer can be run in order to fully functionalize the considered ODX-Data. Two possibilities have been examined considering the role of the functional layer:

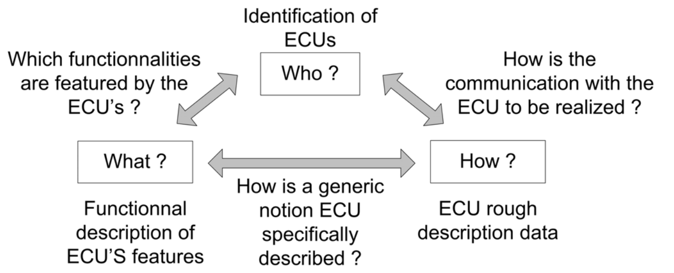
1. The first one consists in **not changing the data structures in the Authoring System**. This means that the data analysis would have to feature knowledge discovery methods [4] [5], which rely on interpretation of natural language [6] and possibly link mining techniques [7] or that the functional layer would need a data structure in which the knowledge from the author can be represented, for instance by an ontology. The strength of ontologies consists in their good capacity for knowledge representation but the time spent to formalize the knowledge has to be taken into account because it rests on cost-benefits analysis, on stability of the knowledge and on the question if knowledge can reasonably be formalized [8]. In order to improve the results, ontologies and data mining can be combined using expert knowledge in the data mining process [9] [10].
2. The second possibility consists in **changing the information structures in the Authoring System** which permits to make a reliable functionalization of the data. A possible approach for the realization is to use an approach based on the principle of the ontological based databases [11].

One constraint to be considered is that whatever the model for data in the authoring system is, it shall fit into tree structures as they are the basic information structures in the authoring system. The solution which is going to be developed bases on a generic and ECU-independent representation of the ECUs' functionalities using groups of features directly linked to one characteristic. It develops the concept of ontological based database presented in [11], where data as well as ontological models are supported in two distinct parts of the same database.

C. Proposed approach

The model uses a meta-schema to administrate data by associating each data object to the ontological concept which defines its signification. Using this idea, it is possible to represent the ECU functionalities (what the ECU can do) using a generic model (**functional data**). As the concrete (ECU-specific) job and parameter names cannot be ignored for vehicle communication, **how** the ECU realizes the generic jobs must also be defined (**informational data**).

This leads to the three part model presented in Figure 2, which identifies three distinct parts for functional vehicle communication : who, what and how.



Identification of ECUs
Who ?
Which functionnalities are featured by the ECU's ?
How is the communication with the ECU to be realized ?
What ?
How ?
How is a generic notion ECU specifically described ?
Functionnal description of ECU'S features
ECU rough description data

Figure 2: ECUs Diagnostic related knowledge structures

This model has the clear advantage to separate functional information from ECU-specific information and thus makes it possible to have an internal and unified access to ECUs.

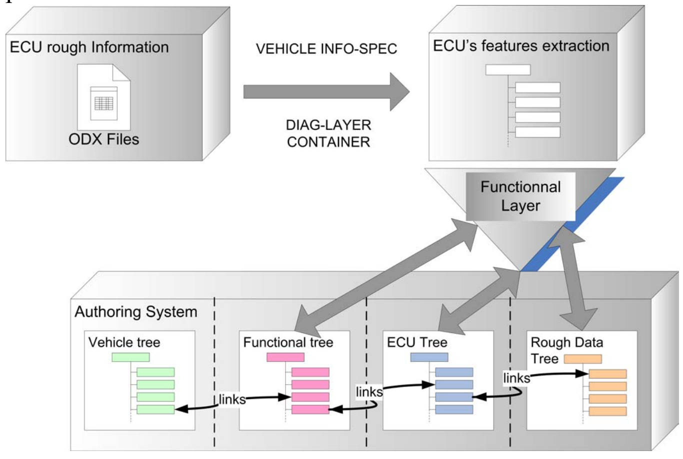


Figure 3: Information exchange in the Authoring System

By the same time, the ECU-specific part benefits from a maximum flexibility to adjust to different ECU description formats such as ODX. This is fundamental in that sense that the structures developed on the ODX side can take the maximum advantage of the ODX structures to reduce information redundancy. Moreover, this structure is interesting because it gives the possibility to:

- Write generic diagnostic program in relationship only with functional information
- Perform diagnostic session using an unified language
- Enable automated information import

At the **Functional Layer**'s level, a module having artificial intelligence capabilities enables to compare the abstracted concepts from the functional trees with the new ODX ECU-description data so as to make links propositions between ECU-specific objects and functional objects.

So as to automate the import of ODX data, the Functional Layer should be able to write new nodes in the Rough-data trees and to create the links to the functional trees. However, its first role is not to set new nodes in the functional trees, so that a read-only access to these trees is modeled in Figure 3. Of course, when the Functional Layer is not able to link all the objects contained in the ODX-files, it can make new functional nodes propositions to the author.

D. *Information model*

In order to eliminate completely the data redundancy which appears in tree structures, the proposed information model makes a wide use of Virtual Hierarchy Links (VHLs) in the proposed tree structures. VHLs allow reusing a node or a sub-tree defined in another tree and thus leads to the definition of virtual parent nodes. So as to be defined, one child node has to have exactly one real parent but can be reused by defining several virtual parents. Using this VHL concept, the following functional trees are introduced and strongly linked together:

- **The request parameters tree** which contains the definition of request parameters and their possible values
- **The response parameters tree**: this tree contains the definition of response parameters and possibly their expected values
- **The Fixed Functions tree**: this tree contains the definitions of Fixed Functions
- **The Fixed Functions Lists tree**: this tree contains nodes (Fixed Function Lists) grouping a set of Fixed Functions together
- **Fault Texts tree**: this tree contains functionalized Fault Texts in different languages, which give a description for ECU-specific default's codes.

The functional information contained in these functional trees is defined independently from a specific ECU in order not to be duplicate one abstracted element when new ECUs are introduced.

The **ODX rough data tree** contains a partial copy from the information contained in the DIAG-LAYER-CONTAINER, and thus is made of an information structure which makes widely use of VHLs. This is for instance useful to represent the ODX-Fields, which are defined independently from diagnostic services.

Once the functional data and the ODX data is present in the Authoring System, it is possible to establish links between them using the **ECU description tree**, making the ODX nodes turn into "cases of" functional nodes. As the ECU node is the root node of the ECU description tree, it plays a central role in this model. As the functional part has to remain generic, i.e. independent from specific units, a master data tree (**unit tree**) containing a description of unit groups is introduced to convert ODX parameter values into functional parameter values of different unit.

As well as the parameter values have to have a concrete meaning for the ECU, they have to make sense for the technician or the running service program. To ensure an abstract meaning to the values, a conversion has to be possible between a value sent (or received) to the ECU and a value used in the service program. For instance, a French ECU sending back the value "Allumé" has to be interpreted into "On". Considering this aim, the principle of value linking has been developed, where different value types can be defined in the functional trees as well as in the ODX-rough data trees. The value type is defined according to the use case of the value (transmit a random value or convert a value into another one) and its data type (string, numerical, byte field etc.).

In the considered information model, the default's codes have also to be considered as functional information due to their central role in the diagnosis process. However, only the signification of the code is interesting and not the code itself so that the default's code in the considered model consists in:

- A functional fault text, which is the basis for the diagnosis.
- A node containing the ODX data default's code value and its associated ODX text value.

III. MULTI-AGENT BASED APPROACH

A. *Presentation and goals*

The data structures which have been introduced make it possible to formalize the authors' knowledge on the one side and to model the ECU description on the other side. Our intelligent import module takes fully advantage of these structures when importing new ODX data to:

- Select the relevant data to be imported in the Authoring System
- Propose a functional interpretation of the objects to be imported
- Identify new ECU functionalities which do not have a generic model in the Authoring System yet

Performing the analysis of a new ODX file is a complicated task which is difficult to execute with a monolithic processing system because of the complexity and the irregularity of the information content. Consequently, a solution using a distributed problem solving method based on a multi-agent system is proposed in this section for the functional layer module. According to the information structure presented in Figure 2, the role of the multi-agent system is to determine which functionalities are featured by the ECU and to justify its proposal by showing how functionality is mapped on the ODX side. As the data structure introduced in the previous section describes a case base, the cognitive skills of our agents is adapted from a CBR (Case Based Reasoning). Considering the review of case retrieval methods given in Figure 3, our agents use the nearest neighbour method formula where the similarity between a case C and a query case Q is given in (1).

$$Sim(Q,C) = \frac{1}{n_f} \sum_{f=1}^{n_f} w_f \sigma(q_f, c_f) \quad (1)$$

With σ a comparator (cf. III.C) and w_f the local weight of the attributes [12].

B. *Global system architecture*

The multi-agent system which has been developed bases on three agent types:

- **A global supervisor agent,** whose role is to organize the work of the other agents considering the author's knowledge contained in the functional trees
- Agents responsible for **finding information** in databases (ODX DataMiner and AS DataMiner)
- Agents capable of case based reasoning on specific objects

The last category of agents is built considering the structures to analyse by following a top down logic: some agent is responsible for determining the match of Fixed Function Lists (FFL Matchers), whereby their conclusion rely on the analysis results from the agent FF Matcher (matches the Fixed Functions). Analysing a Fixed Function also means finding a match for the parameters contained into it and their values, which is the role played by the agents FF-Parameter Matcher and Value Matcher.

In our approach, some agents are to be considered as auxiliary agents, which provide a clever access to the data, which is the case from ODX Data Miner and AS Data Miner. As the information to be compared is only textual information, a further agent String Comparator is introduced to compare two string expressions and to give a match concerning their functionality. Several difficulties have to be faced by this last agent, including:

- Grammatical and word spelling difficulties
- Semantic difficulties
- Multilinguism

Several techniques are available today to face such problems. The classical one consists in comparing "bags of words", where the results can be improved using options like a stop list, a stemming process and synonym lists [13] but other techniques like inexact string matching have also been experienced, especially in spam filtering [14].

C. *Agent FF-Parameter Matcher*

The role of the agent FF-Parameter Matcher is to determine how the parameters of a new ODX DiagComm match with functional parameters stored in the database. To do so, it receives a new ODX DiagComm and the Fixed Function that is suspected to map the DiagComm functionally from the agent FF Matcher. The first step when trying to find a functional interpretation for ODX parameters is to look up to the information already contained in the Authoring System and in the ODX file, so that an analysis order is sent to both AS DataMiner and ODX DataMiner to get back a precise description of the parameters. The case representations sent back by these agents consist in several information types:

- Parameter identification data (Short Name, Long Name etc.)
- Parameter description data (Parameter type)
- Functional data (OID, Semantic)

As all the features to be compared are strings, the agent FF-Parameter Matcher requires the services of the agent StringComparator to estimate the value $\sigma(q_f, c_f)$.

Once each case similarity is obtained, the agent FF-Parameter Matcher has to decide which parameters seem to match best with the functional case in order to let the values be analysed by the agent Value Matcher. Filtering the data between each analysis step is crucial in order to reduce the processing time through not pulling the analysis forward where it is not necessary. In order to improve the quality of the decision capability of the agent FF-Parameter Matcher, it has access to the belief of the agent FF Matcher concerning the match of the Fixed Function with the DiagComm. This enlargement of the environment perception augments the chances that the direction of the searching process takes the optimal direction [15].

IV. MOTIVATIONAL EXAMPLE

In this section, we present a simple example which illustrates the capabilities of our system. This example bases on the import of the description of a new ECU (CDC_1DIN) in a database where the description of the ECU XHUAGW is already linked with the related functional information. In our test base, a series of nine DIAG-COMM objects are to be interpreted and five default's codes are to be put into relationship with their functional counterpart.

In our experimental case, some objects belonging to the new ECU are strictly similar to those of the existing one. However, in order to demonstrate the intelligent capabilities of our system and to simulate a real situation, some have been slightly modified. The example in Figure 4 illustrates that the functional job *Read sensor list* is going to be executed by the DiagComm *JobGetSensorList* when communicating with the ECU XHUAGW. The response is made of an ODX table and a row containing the three parameters *service*, *securityLevel* and *sensorName*. When analyzing the new ECU CDC_1DIN, our multi-agent system suggested that the unknown DiagComm *JobReadSensorList* should correspond to the function *Read sensor list* and that its unknown parameter *Security* might correspond to the functional parameter *Security Level*. Even the returned values are interpreted in a functional way, so that the signification of a result 7/8 is automatically found when reading the database. Note that the value 7 returned by the other ECU will have exactly the same signification on the functional level. Once these suggestions have been accepted by the author, the description of the new ECU can be imported in the Authoring System and so complete the case base. Once it is done, a functional communication with the ECU CDC_1DIN can be started using the functional description and the ECU-specific information for the vehicle communication. The completed case base is also available to the multi-agent system for further ECU imports.

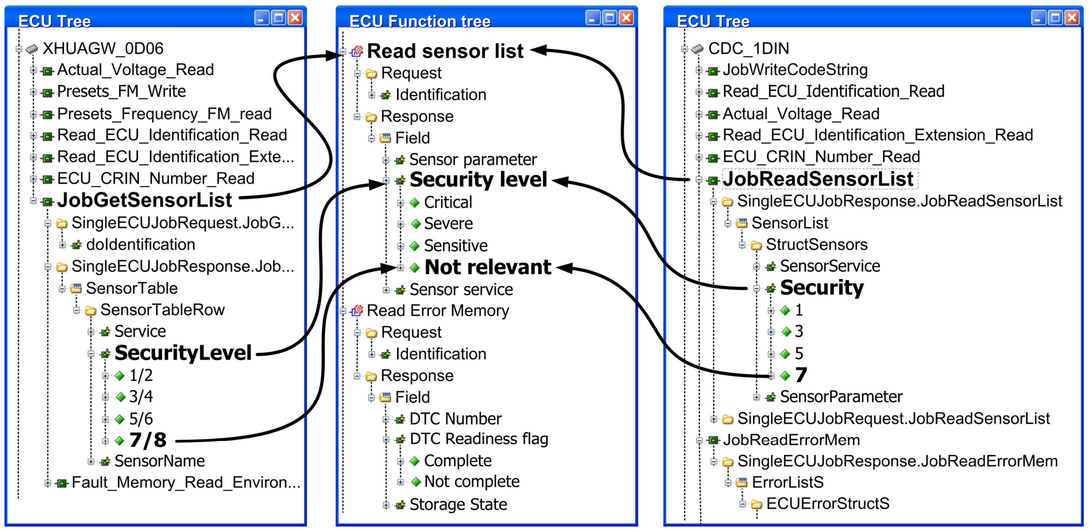


Figure 4: The tree extracts from the Authoring System show the ODX description of the ECUs XHUAGW and CDC_1DIN and the functional job "Read sensor list" in the middle

V. CONCLUSION

This paper has presented an overview of the data structures in the actual Authoring System from Siemens. The preliminary study of these structures have lead to the conclusion that an author realizes a selection and a functional interpretation of the ECU-description data when importing it into the Authoring System and introduces new knowledge-related data for each ECU to be described in the Authoring System. This procedural method does not help for the automated import of ODX data because it means that the first step before processing the import is to get to "know what the author knows", outgoing from widespread knowledge. Thus, techniques making it possible to formalize knowledge and retrieve it have been presented. In order to take advantage of the data reduction possibilities offered by the ODX data model, new information structures have been developed using the concept of value inheritance and of Virtual Hierarchy Links. With these data structures for functional vehicle communication, the author's knowledge is represented independently from the physical ECU description in one unique database, which makes the development of distributed problem solving method basing on an analysis of the ECU capabilities robust and easy. The multi-agent system which has been developed uses a case-based approach to analyse the new ODX data and to make automatically import propositions.

Three main ideas which are being developed at the present time concern the introduction of reinforcement learning techniques, the improvement of the pro-activeness of agents and the possibility to give them alternative cognitive capacities.

REFERENCES

[1] Augustin, Backmeister, Beiter, Dogan, Hallermayer, Hecker, Hümpfner, Köhler, Kricke, Kolbe, Michard, Öhlenschläger, Ramrath, Schleicher, Wallschläger, Watzal, Wolter, Zweigler, „ASAM MCD-2D (ODX) Version 2.1.0 - Data Model Specification" in *ASAM specification*, 2006

[2] C. Kricke, "ODX, der neue Standard für Diagnose-Daten im Kraftfahrzeug" in *Diagnose von E/E-Systemen im Automobil, Freising bei München (Germany)*, 2005, pp. 1-27

[3] W. Zimmermann and R. Schmidgall, "Bussysteme in der Fahrzeugtechnik", ISBN-13-978-3-8348-0166-6, *Friedr. Vieweg & Sohn Verlag*, 2006

[4] P. Alpar and J. Niedereichholz, „Data mining im praktischen Einsatz", ISBN 3-528-05748-3, Vieweg Verlag, 2000

[5] E. Lefevre, J. Manata and D. Jolly, "Classification par la théorie de l'évidence pour la gestion de tournées de véhicules" in *Proceedings of R.F.I.A., Toulouse (France)*, 2004, pp. 1-9

[6] H. Karanikas, C. Tjotjis and B. Theodoulidis, "An approach of text mining using information extraction" in *Proceedings of principles and practice of knowledge discovery in databases, Lyon (France)*, 2000

[7] L. Getoor and C.P. Diehl, "Link Mining: a survey" in *SIGKKD Explorations 7.2*, 2005, pp. 3-12

[8] M. Liao, A. Abecker, A. Bernardi, K. Hinkelmann and M. Sintek, „Ontologies for knowledge retrieval in organisational memories" in *Proceedings of the Learning Software Organizations(LSO'99) workshop, Kaiserslautern (Germany)*, 1999, 19-26

[9] L. Brisson and M. Collard, "Une expérience d'intégration des connaissances dans un processus de fouille de données" in *Rapport de recherche ISRN I3S/RR-2007-07-FR*, 2007

[10] L. Brisson, M. Collard and N. Pasquier, "Ontologies et base de connaissances pour le pré-traitement et le post-traitement en fouille de données" in *Fouille de données complexes workshop in EGC conference, Lille (France)*, 2006

[11] G. Pierra, H. Dehainsala, Y.A. Ameur, L. Bellatreche, J. Chochon and M. E.-H. Mimoune, "Base de données à base ontologique : le modèle OntoDB" in *Proceedings of Bases de données avancées 2004, Montpellier (France)*, 2004

[12] Z. Wen, J. Crossman, J. Cardillo and Y.L. Murphey, "Case base reasoning in vehicle fault diagnosis" in *IEEE proceedings of the International Joint Conference on Neural Networks (4), Portland (USA)*, 2003, pp. 2679-2684

[13] C. De Loupy, "L'apport de connaissances linguistiques en recherche documentaire" in *Proceedings of the 8th TALN conference, Tours (France)*, 2001

[14] D. Sculley, G.M. Wachman and C.E. Brodley, "Spam filtering using inexact string matching in explicit feature space with on-line linear classifiers" in *Proceedings of the 15th Text Retrieval Conference, Gaitehrsburg (USA)*, 2006

[15] M. Woolridge and N.R. Jennings, "Intelligent agents: theory and practice" in *Knowledge engineering review10.2*, 1995, pp.115-152