

MULTI-AGENT DATA AND INFORMATION FUSION

Architecture, Methodology, Technology and Software Tool

Vladimir Gorodetsky, Oleg Karsaev, Vladimir Samoilov

St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences

Phone: +7-812-2323570, fax: +7-812-3280685, E-mail: {gor, ok, samovl}@mail.ias.spb.su,

<http://space.ias.spb.su/ai/gorodetski/gorodetski.jsp>

Abstract: The paper introduces the present-day understanding of the data and information fusion problem and describes some aspects of the methodology, technology and software toolkit developed by the authors for the design, implementation and deployment of a class of multi-agent information fusion-related applications. The distinctive feature of the presented technology supported by the software toolkit is that it is distributed and agent-mediated, i.e. it assumes a distributed fashion of designers' activity mediated by agents performing most of the routine engineering work and supporting coordination of collaborative designers' activities. The above methodology, technology and software toolkit are implemented and were practically used for prototyping of several applications from the data and information fusion scope.

Key words: multi-agent system, information fusion, distributed data mining, distributed decision making, decision combining, methodology and technology of information fusion, agent mediated technology, multi-agent software tool kit

1. INTRODUCTION

The authors of [32] define Information Fusion (IF) as " the process of combining data to refine state estimates and prediction" [32]. It is assumed that these data specify either a complex system comprising a set of semi-autonomous objects operating according to a joint goal (intent) or a natural phenomenon evolving in space and time, which is not directed by any intent.

More extended description of information fusion was given by B. Dasarathy [21]: "Information Fusion ... encompasses the theory, techniques

and tools conceived and employed for exploiting the synergy in the information acquired from multiple sources (sensor, databases, information gathered by humans, etc.) such that the resulting decision or action is in some sense better (qualitatively or quantitatively, in terms of accuracy, robustness and etc.) than would be possible if any of these sources were used individually without such synergy exploitation."

IF is also an important component of a more general and integrated problem referred to as *situational awareness*. Objective of situational awareness problem is to provide a comprehension of "what is going on so I can figure out what to do" [1] and how to make a prediction of this comprehension in the nearest future [7]. As a rule, situational awareness task is composed of many particular subtasks and among them IF is one of the most important.

The most commonly accepted model of IF referred to as JDL model was proposed in [32]. It considers five level structure of data and information processing (Fig. 1) intended to provide users with comprehension of the situation associated with a mission and assessment of the results of effects on it of certain planned actions.

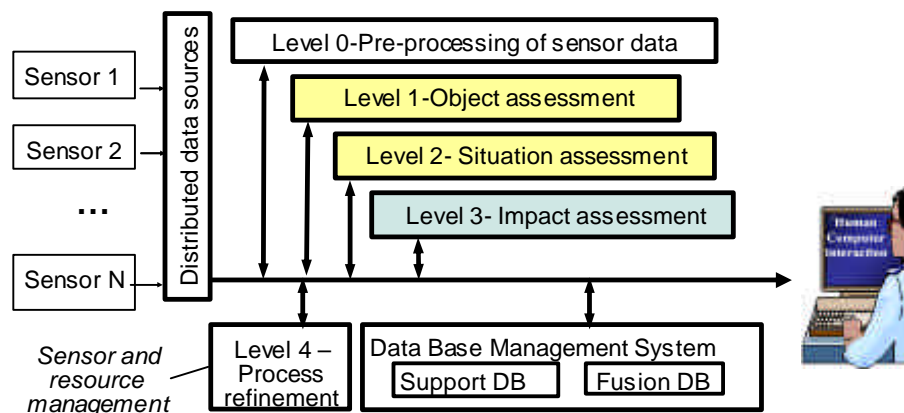


Figure 1. The JDL model of data and information fusion

This paper is relevant to the tasks associated with two of five levels, called in Fig. 1 as "*Object assessment*" and "*Situation assessment*." Object assessment is a task of classification of the states of particular objects constituting a complex system of interest or phenomenon of the natural environment. This task is also often referred to as *Data Fusion* although the boundary between data and information fusion tasks is too vague.

Any particular situation is specified in terms of the states and/or activities of objects constituting the situation as well as meaningful *relationships* between them. *Situation assessment* is a classification task aiming at estimation and updating of the state of the situation. The situation is developing in time, i.e. this notion is of a dynamic nature. As a rule situation assessment is referred to as *Information Fusion*.

The objective of this paper is to describe basic multi-agent IF system components, architecture and technology for the design, implementation and deployment of IF applications in a computer network, and also to introduce software tools supporting the above technology. Accordingly, section 2 introduces the problem statement and assumptions and describes two examples of IF applications. Section 3 motivates the use of a multi-agent architecture for IF systems intended for object and situation assessment. Section 4 presents the detailed description of the methodology of data and information fusion that is compiled of the well known principles. The methodology determines IF system components, their functionalities and interactions, and also presents main algorithms used in data and information fusion. Section 5 introduces the generic multi-agent architecture of an multi-agent IF system and describes functionalities of its basic components. Section 6 outlines the developed technology for multi-agent IF system engineering and supporting software tools. One of them, MASDK, supports engineering of reusable components of IF Multi-Agent System (IF MAS) while the second one, IF Design Toolkit, supports engineering of the problem and application-specific components of IF MAS. Section 7 describes MASDK and the IF Design Toolkit in more detail. The conclusion summarizes the main ideas presented in the paper concerning multi-agent information fusion systems' analysis, design and implementation.

2. PROBLEM STATEMENT

The main objectives of the paper are to describe the methodology of information fusion for object and situation assessment as well as technology and software tools for IF MAS design, implementation and deployment. To make it clear what kind of IF applications are the subjects of the paper, let us first describe a couple of examples of such applications.

2.1 Intrusion detection in computer networks [2]

At present, coordinated distributed attacks performed by a team of malefactors from spatially distributed hosts constitute the primary threats for computer networks and information. "Traces" of an attack are manifested in

various data perceived of or generated by a computer network assurance system in different hosts of a computer network. For example (see Fig. 2, taken from [14] and slightly reworked), the traces of malefactors' attacks are displayed in *tcpdump* file containing data resulting from input traffic preprocessing, in an audit data trail, in sequences of system calls made by the operating system, in short-term and long-term statistical data resulting from monitoring of application servers, in queries to databases and directories, in data specifying user profiles, etc. To detect a broad variety of attacks against computer networks including distributed attacks it is necessary to make use of all the available data and information sources [2]. Formally, intrusion detection is a classification task that utilizes a combination of alerts produced via analysis of data and information obtained by particular sources mentioned above. The assessment of the computer network security status presents a typical example of IF problem.

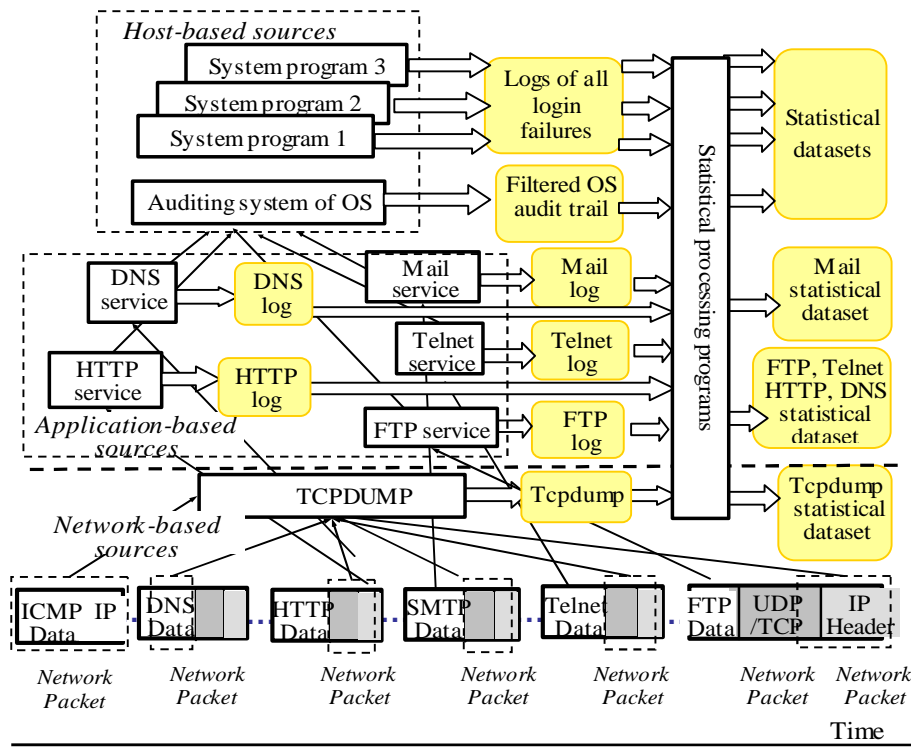


Figure 2. Multiplicity of data and information sources (given in grey color) available for assessment of the security status of a host (the content of the figure is taken from [14])

2.2 Analysis and prognosis of natural and man-made disaster evolving

Many different kinds of potentially dangerous situations often emerge in the different regions of many countries. They can emerge due to natural disasters (earthquakes, floods, etc.), man-caused emergencies (chemical, nuclear, etc.) and so on. The specific features of such phenomena are a rapid and weakly predictable development in time and space, and a strong dependence on weather conditions, landscape, building infrastructures and so on. To assess the situation as a whole in order to be able to predict its development and prevent undesirable or catastrophic consequences, it is necessary to make use of data and information collected from many different sources. A shortened example of data sources pertinent to a particular case of disaster that is a flood is presented in Fig. 3 and 4. In this example, the sources of data and information and processing procedures are presented in two levels. The first level (Fig. 3) corresponds to the data and information specifying flood situation parameters in particular regions, provided by their own flood monitoring and data processing subsystems. In the second level

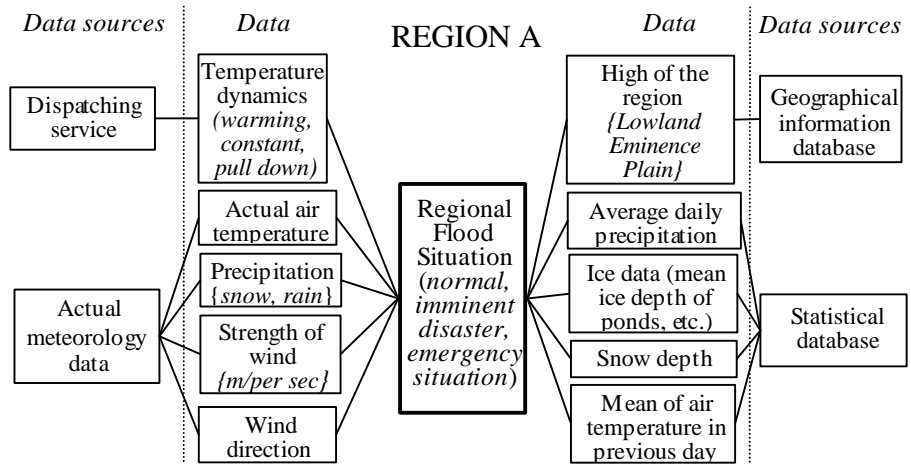


Figure 3. Data sources used in flood monitoring, prediction and related management

information characterizing situations in different regions is collected. It is here processed together with additional more aggregated information received, for example, from airborne equipment. Figure 4 demonstrates the second level of IF and also a closed loop of situation assessment, impact assessment and process refinement for flood situation monitoring and management, considered within the JDL model of information fusion (see Fig.1).

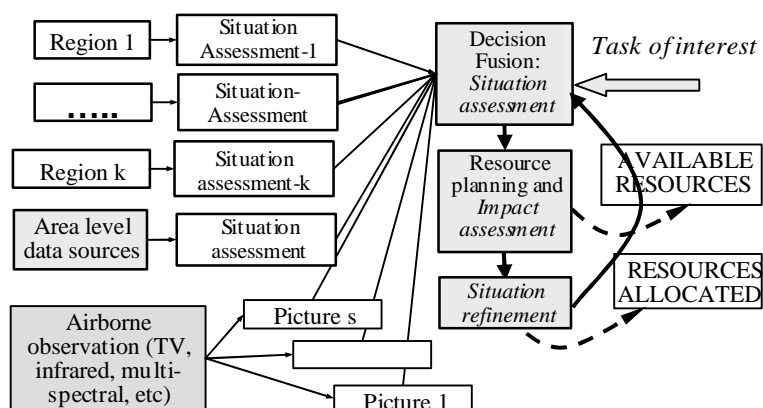


Figure 4. Closed loop of situation management: Situation assessment → Impact assessment → Situation refinement → ...

2.3. Paper Objective and Main Assumptions

Thus, the subjects of this paper are design and implementation issues of multi-agent IF systems for applications similar to ones described above. It is assumed that the central task of the design is classification aiming at the assessment of the states of a complex object or situations constituted by semi-autonomous objects possibly operating according to a shared goal (intent). It is also assumed that the classification is produced on the basis of fusion of heterogeneous data and information obtained from multiple distributed sources. Data and information of different sources can be presented in different measurement scales and can be of various accuracies. It is also assumed that data can be incomplete, i.e. some sources may contain missing values of attributes and the total dimensionality of data can be large.. An important assumption is also that some data can be unavailable for centralized processing. For example, data of particular sources can be private or classified. The set of possible states of the object of interest or situations to be assessed is known and finite. Thus, it is assumed that the IF system solves a *distributed classification task*.

An important assumption is that the design and implementation of classification mechanism is performed on the basis of data mining and knowledge discovery technology, using interpreted historical datasets. The last task can be the responsibility of a separate system, the IF Learning System. The latter can also be designed as a particular component of the IF system as a whole, thus providing the latter with offline learning capabilities

utilizing the IF system positive and negative experience from using the currently installed decision making mechanism.

Other assumptions are of less importance and explained when necessary.

3. WHY MULTI-AGENT INFORMATION FUSION? WHY MULTI-AGENT TECHNOLOGY

The Multi-agent system (MAS) view represents an advantageous paradigm for the analysis, design and implementation of complex software systems. It proposes powerful metaphors for information system conceptualization, a range of new architectures, techniques and technologies specifically destined for large scale distributed intelligent systems [35, 37].

IF systems definitely fall into the class of potential MAS applications. Indeed, each IF application is naturally distributed: data sources are spatially distributed; data processing is performed in a distributed manner, the systems and/or users interested in the results of an IF system operation are distributed. If data from different sources are private or classified (military data, commercial data, etc.) then such data is not available for a centralized processing, in particular, the data holders do not render the accumulated datasets for the learning of classification. At the same time, they can make this data available for situated agents in order to process the private data locally, without revealing its content.

In most cases IF systems are of a large scale, and it is exactly these kinds of applications that take the most of advantages provided by MAS architectures. In particular, MAS architectures are especially convenient for software implementation of such decomposable large scale applications.

A number of advantages can also be exploited if a software tool supporting the engineering and implementation of IF systems is also built as a multi-agent system. The most important advantage originates from the fact that in many cases IF systems are being developed by several spatially distributed designers. In such a case, agents of the software tool can take the roles of mediators among different designers by supporting their collaboration according to predefined protocols that provide the design process with both flexibility and automated support of the technology discipline. It is demonstrated below that most of the technological activities performed in a distributed manner can be coordinated with more ease if they are mediated by agents operating according to predefined protocols.

Recent research practices show that the popularity of multi-agent paradigms as applied to IF systems is continuously expanding and thus it is gaining a reputation as a very promising technology.

4. BASIC PRINCIPLES OF THE INFORMATION FUSION METHODOLOGY FOR OBJECT AND SITUATION ASSESSMENT

IF design process is determined by a number of basic conceptual solutions that have to be made in regard to the following engineering aspects:

1. Allocation of data and information processing functions to the data source-based level and to meta-level responsible for generalization of the source-based decisions;
2. Structure of decision making and decision combining in an IF system called hereinafter *Decision fusion meta-model*, or DF meta-model;
3. Structure and representation of the IF system knowledge base (KB) and ontology;
4. Particular techniques used for the engineering of IF system KB and respective decision making mechanisms;
5. Particular techniques used for information fusion.

Let us analyze the above aspects in order to justify the adopted solutions.

4.1 Allocation of information fusion function

There exist several variants of the allocation of functions to source-based and centralized levels of data and information processing proposed for IF [8]. Let us outline and evaluate them to justify the selection.

4.1.1 Centralized data and information processing

This variant assumes the straightforward transmission of data from data sources into a central database for subsequent centralized data and information fusion. By default both classification and learning are here also to be performed in the centralized mode. This approach possesses very obvious drawbacks: (1) inefficiency in the case of the very high dimensionality of the entire data representation space;(2) very high communication overhead and data duplication; (3) inability to preserve data privacy both in learning and classification procedures. If various data structures are used by different sources then this methodology becomes practically infeasible.

4.1.2 Combining knowledge bases of data sources

Knowledge bases designed on the basis of particular data sources can be simply combined within a single KB, which afterwards is used as the KB of a centralized classification system. Although an obvious drawback of this model is that it supposes the use of centralized classification, it preserves source data privacy, because the agents but not designers have the access to the private data. This model is applicable in case if all local knowledge bases are represented in a common structure, e.g. all of them are rule-based. An example of a relevant application is fraud credit card detection, if several banks agreed to use for this purpose a common protection system. Unfortunately, such an approach is not applicable if source data are of large dimensionality and knowledge representation in different sources differs greatly. Thus, such an approach is not the best choice in most cases for information fusion applications.

4.1.3 Fusion of decisions produced on the basis of data of particular sources

In this model the decisions produced by local classification mechanisms (they are called base classifiers [28]) are combined in the meta-level. This model is advantageous in many applications, in particular, if:

- There are many data sources;
- There are representative interpreted datasets sufficient for training and testing of both base and meta-level classifiers;

The advantages of this data and information fusion model are as follows:

- It provides considerable decrease of communication overhead;
- It is applicable in the case, in which the structures of data representation of various sources are very different. In this case independently of the data structures of particular sources only decisions produced by source based classifiers are forwarded to the meta level and these decisions are represented in either binary or categorical measurement scales;
- There exist several effective and efficient methods of combining such decisions in the upper level to obtain the combined decision;
- It preserves source data privacy.

This model of data and information fusion outperforms the models described in subsections 4.1.1, 4.1.2 in many respects. It is used below as a component of the methodology of data and information fusion.

It should be noted that combinations of the fusion models described in subsections 4.1.1-4.1.3 can also be used in some specific cases. For example, in some cases it is preferable to forward to the upper level both the decisions produced by base classifiers and part of the data of some sources.

4.2 Decision fusion meta-model

The *meta-model of decision fusion (DF meta-model)* specifies structure of classifiers and their interaction in decision making and decision fusion. It comprises three types of hierarchically ordered particular structures: *source-based structure of decision making*, *structure of decision combining*, and structure of classification called hereinafter for brevity *classification tree*. Let us consider these structures and their composition in the *DF meta-model*.

4.2.1 Source based decision making structure

In the simple case when the dimensionality of the vector of data source attributes is small (about 20–25) and data representation structures are more or less homogeneous within the source, e.g. they are measured either in numerical or in discrete scales, then, for this source, one source-based base classifier can be used. In such a case the *data source decision making structure* consists of a single base classifier (Fig. 5). In more complicated

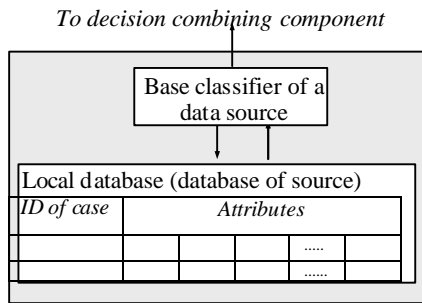


Figure 5. Data source decision making model: Variant 1

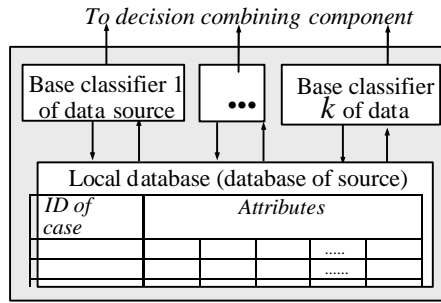


Figure 6. Data source decision making model: Variant 2

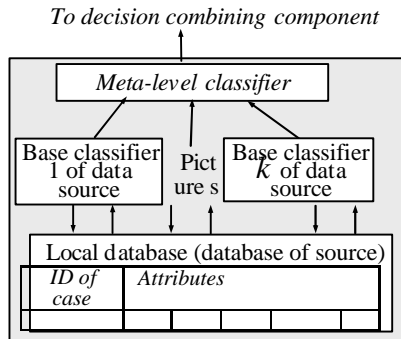


Figure 7. Data source decision making model: Variant 3

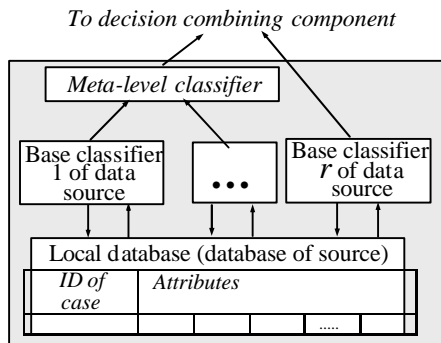


Figure Data source decision making model: General case

cases, if the dimensionality of the attribute vector is high enough and/or source data are too heterogeneous (measured in different scales, are of different accuracies and reliabilities, have missing values of attributes, etc.), then it is reasonable to provide source data with several base classifiers. Each such classifier has to produce classifications using different sets of attributes or/and it has to be trained on the basis of different training and testing datasets. In this case the *data source decision making structure* consists of several base classifiers. The decisions produced by these classifiers can be forwarded to the meta-level for combining with decisions produced by base classifiers of other sources (Fig. 6). An alternative is to combine these decisions and forward the combined decision to the upper level (Fig. 7, 8). It is reasonable to use this model in the case that the number of source-based classifiers is too large. Example is given by an intelligent sensor network containing hundreds of multi-attribute sensors, whose outputs are collected within a single data source.

4.2.2 Structure of decision combining

The *structure of decision combining* specifies how decisions produced by base classifiers could be combined in the meta-level as collective decision [29, 8]. The variants of these structures are demonstrated in Fig. 5 through Fig. 8. Fig. 5 and 6 present the cases when no intermediate level of decision combining is used. Fig. 7 present an example in which decisions of base classifiers, before passing information to the meta-level, are previously combined. Fig. 8 presents a more general case when part of base classifiers decisions are combined, and the rest of them is directly passed to the meta-level.

It is important to note that, if no uncertainty measures are assigned to the decisions of base classifiers, then all decisions passed to meta-level are represented either in binary or in categorical scale. In the technology developed, classification task for multiple classes is reduced to a number of binary (pair wise) classification tasks. Hence the input data of a meta-classifier are represented as binary vectors. For brevity, the structure of decision combining is hereafter called *decision making tree*.

4.2.3 Structure of classification ("Classification tree")

The *structure of classification* ("*classification tree*") is a component of the *DF meta-model*, that is used to reduce multiple classification tasks to a number of binary ("*pair wise*") ones. The nodes of the *classification tree* that are not leaves are called hereinafter *meta-classes*. An example of the classification tree for the task with 4 classes is presented in Fig. 9, on the

left. In this tree, the root node corresponds to the *meta-class* representing all possible solutions. The classification procedure in this node discriminates the situations of classes 1 and 3 from those of classes 2 and 4, i.e. discriminates instances of meta-class 1 from instances of meta-class 2. In the second step, the decision should correspond to a leaf of the classification tree indicating the final solution.

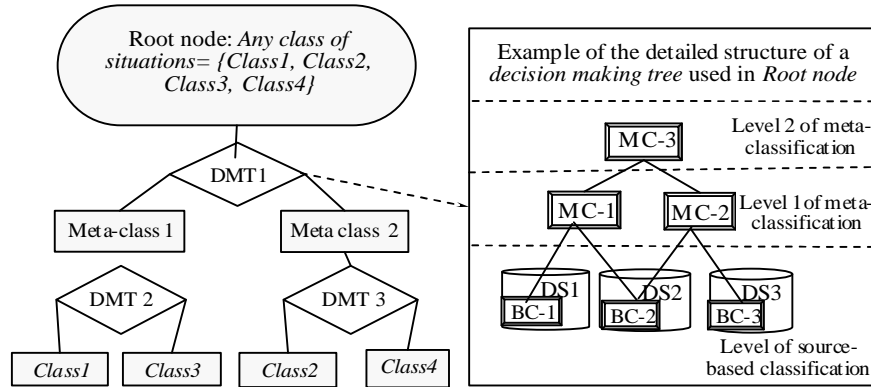


Figure 9. An example of the Decision Fusion meta-model composed of classification and decision making trees. *DMT*: Decision making tree; *DS*: Data source; *BC*: Base classifier; *MC*: meta-classifier

4.2.4 Decision fusion meta model

On each node of the *classification tree*, a decision making task is mapped that involves in the decision making procedure all the base classifiers and meta-classifiers of the *decision making tree*. Thus, on each node of the classification tree a *decision making tree* is mapped that is composed of a *structure of decision combining* and *data source decision making structures* associated with data sources. Such a *decision making tree* can be different for different nodes of the *classification tree*.

An example of a *decision making tree* corresponding to the meta-class (node) *MC1* is presented in the right side of Fig. 9. This figure also explains the interconnection of the notions introduced in this subsection, i.e. the notions of *classification tree* (it specifies upper level structure of data and information fusion), *decision making tree* (in the lower level), *meta-class*, and also introduces the notions of *base level of classification* and *higher level meta-classification*.

Thus, the Meta-model of Decision Fusion, DF meta-model, is composed of *decision making trees*, structured according to the *classification tree*.

4.3 Structure of an IF system knowledge base. Ontology

The knowledge base of an IF system is composed of knowledge bases of particular classifiers, meta-knowledge base and ontology. A KB peculiarity is that it is *distributed* over hosts, in which the data of particular sources are stored, and particular fragments of the KB are situated in the hosts, in which the combining of source-based decisions is performed. As a rule, the distributed components of KB are *heterogeneous*.

The distributed character and heterogeneity of a KB considerably affect the methodology of its design. Additional peculiarities of KB design originate in that case when data sources are private or classified and thus unavailable for centralized processing. Through the above peculiarities several new problems have to be resolved in design process that influence on the methodology of IF KB [8].

The *first problem* is the necessity to provide the system with a *shared thesaurus* required for *monosemantic understanding of the terminology* used by distributed components (in our case, agents) of the IF system. This problem arises due to the fact that specifications of data belonging to different sources can be developed independently by data holders. The latter can denote different attributes and domain notions by the same terms, and vice versa, they can denote the same notions by different terms, which obviously results in the inconsistency of knowledge representation and therefore to the agents' misunderstanding.

The problem called *non-coherency of data measurement* [8] results from the fact that different sources can contain overlapping sets of attributes and the same attributes can be measured in different units in different sources. In IF procedures they must be used in the same units. Thus, the problem is how to provide KB with the capability to deal with such data consistently.

The third problem is the so-called "*entity instance identification problem*" [8]. Fig. 10 demonstrates the essence of this problem. Indeed, due to multiplicity of sources of information about situation, an instance of a situation snapshot is represented by its fragments in several data sources. For example, the instance of the situation # 4 in Fig. 10 is represented in data sources **A**, **B** and **C**. To detect that these fragments represent the information about the same situation instance, it is necessary to have an identification mechanism. In turn, the complete specification of situation instances is necessary for further use them for training of the meta-classifier performing combining of the decisions produced on the basis of particular data sources. The aforementioned problems are effectively resolved on the basis of ontology-centered distributed knowledge representation [16]. The focus of this approach is to explicitly represent all the notions (terminology) used within the system and the relationships between them as top level knowledge

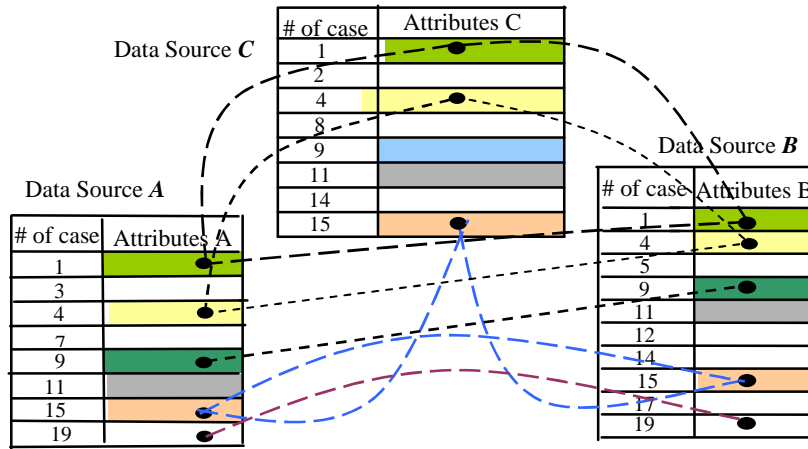


Figure 10. Illustration of the essence of the data identification problem

shared by all components of the IF system. This component of knowledge representation is called *shared ontology*. The typical structure of ontology is explained in Fig. 11. In it, the upper level corresponds to the problem ontology that represents the common part of the ontology pertaining to all applications of the problem in question, in our case, the IF problem. The middle level represents a component of the ontology specifying notions and relationships peculiar to the application. This component of the ontology is called *shared part of application ontology*. This is exactly the level of ontology that must be developed so as to make it possible to resolve the aforementioned problems (see section 7.3 describing how these problems are resolved in the developed technology). The lowest level of the ontology, private ontology, specifies notions used only by particular source-based components of the IF system.

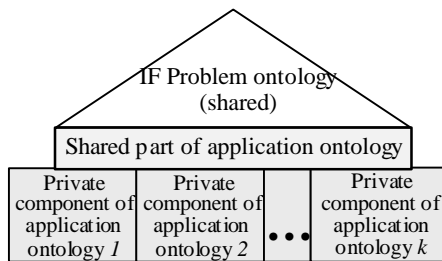


Figure 11. Tower of IF ontology components

Thus, the structure of the IF system KB consists of ontology and distributed KBs of particular base and meta-classifiers that are structured

according to the *DF meta-model* consisting of a *classification tree* (in the upper level) and the set of *decision trees* mapped to each node of the former.

5. TECHNIQUES USED FOR CLASSIFICATION AND COMBINING OF DECISIONS

5.1 Techniques Used for Source-Based classifier Learning

The development of techniques for training and testing of base classifiers, i.e. data mining and knowledge discovery techniques, is out of the main IF scope. Let us just briefly indicate what particular methods and techniques are implemented as components of the IF MAS Design Toolkit supporting the IF technology.

In the IF MAS Design Toolkit, three different techniques are used that are destined for extraction of production and association rules from databases comprising numerical, binary and categorical types of attributes. Let us indicate these techniques.

Visual Analytical Mining (VAM). This technique is destined for the extraction of production rules and/or decision trees from datasets containing attributes of numerical and linear ordered measurement scales [11, 12]. VAM allows to extract production rules specified in terms of the first order quantifier-free logic. *GK2*. This technique is destined for the extraction of production rules from data represented in discrete scales, i.e. binary, categorical, integer and liner-ordered [9, 13]. Conceptually, it is similar to the well known AQ technique [25], but uses different algorithms for the extraction of minimal rules..

The *FP-grows* algorithm of association rule mining is well recognized within the data mining and knowledge discovery community for its efficiency. Its formal and informal description can be found in [15].

The *VAM*, *GK*, and *FP-grows* techniques are implemented as classes of the *Library of training and testing methods* of the IF MAS Design Toolkit.

5.2 Classification mechanisms of base classifiers

The techniques, described above, allow the semi-automatic engineering of the knowledge necessary for decision making by base classifiers, in particular rule-based knowledge. Let us consider how base classifiers use the rules for producing decisions in respect to newly incoming data.

It should be reminded that according to the adopted IF methodology, in each node of the *classification tree*, and accordingly, in each node of *decision making tree*, the task of binary classification is solved (see section 4.2). Let us denote the classes (meta-classes) assigned to any node of the classification tree as Q and \bar{Q} . The production rule mining techniques *VAM* and *GK2* described in subsection 5.1 generate two sets of rules $\{R_1^+, R_2^+, \dots, R_k^+\}$ and $\{R_1^-, R_2^-, \dots, R_l^-\}$. The first set is such that it “argues in favor” of class Q , i.e. contains rules of the type $R_i^+ = F_i^+ \supset Q$, $i=1,2,\dots,k$, (F_i^+ is the rule premise represented by a conjunction of propositions), whereas the second set of rules “argues in favor” of class \bar{Q} , i.e. contains rules of the type $R_i^- = F_i^- \supset \bar{Q}$, $i=1,2,\dots,l$. (F_i^- is the rule premise represented by a conjunction of propositions). For brevity, these rules will be called *arguments* hereinafter. While taking into account application-dependent requirements, a metric for argument quality assessment can be introduced and computed for each of the extracted rules. Such metrics are based on use of a *confusion matrix* comprising probabilities of correct classification and true and false positives.

Several decision making variants can be used if rules of base classifiers are presented in the above form. For example, the simplest variant consists in counting the weights of the “positive” and “negative” arguments in favor a particular decision, and the conclusion is made in favor of this class whose arguments are “stronger.” In fact, this variant of decision making corresponds to a well-known “*weighted voting*” approach (see, e.g. [29], although this method was practically used decades years ago). This method is quite widely used these days, and works relatively well. Other variants of the above arguments combining are considered below in the next section.

In the working version of the IF software tool, the implemented classification mechanism can be characterized as argumentation based on the notions drawn from the *Inferential Theory of Learning* [24]. This theory equates learning to knowledge mining through *knowledge space transformation*. From such a viewpoint, each hypothesis generated by an inductive learning procedure can be considered as twofold. On the one hand, such an hypothesis can be considered as a new generalized attribute specifying a new dimension of the data specification, and a set of such hypotheses, in turn, can be considered as a new representation space determined by the primary set of attributes. On the other hand, a new hypothesis (for example, a rule) may represent a decision procedure intended to discriminate the situation of a category from that of another. Thus, in the latter case, the set of hypotheses is viewed as a decision structure [23].

In the implemented model of decision making, the rules (arguments) extracted through the use of *VAM* and/or *GK2* techniques are considered as new coordinates of the representation space computed via a transformation

of the primary space. All of the newly computed coordinates are binary. Accordingly, the initial training and testing datasets, after their transformation into the new space, are subsequently used for the training and testing of base classifiers. In most cases, this representation space transformation and subsequent rule mining results in the extraction of more "strong" "pro" and "contra" arguments of the class Q . Experience proved that this process, possibly repeated more than once, results in a situation, in which the decision making procedure consists of computing the truth value of a single rule given over the truth values of lower level rules and/or attributes.

It should be noted that this technique is close to the meta-classification method, which was developed concurrently as an independent project. The latter is described in the next section.

5.3 Techniques for combining decisions in data fusion systems

According to the methodology accepted here, the strategy of decision combining consists in the use of a hierarchy of multiple classifiers producing decisions on the basis of particular data sources followed by combing these decisions in the meta-level (see section 4.2 above). Let us briefly review the existing decision combining approaches and techniques and justify our preferences.

To date, several techniques and approaches to decision combining are proposed. They can be grouped as follows:

1. *Voting* algorithms;
2. *Probabilistic* and *fuzzy* algorithms;
3. *Meta-learning* algorithms based on *stacked generalization*;
4. *Meta-learning* algorithms based on *classifier competence evaluation*.

Voting methods have been in use for at least four decades while nevertheless preserving popularity [29]. The main drawback of these types of algorithms is that they actually have no theoretical ground and there is no guarantee that the chosen variant of voting will work perfectly in a particular application for all possible input data. The advantage is mainly in the simplicity of such techniques.

Methods of the *second group* are based on the use of different uncertainty models. Among them, Bayesian "a posteriori" probability-based models (for example, different kinds of Bayesian networks), possibility theory models, Dempster-Shafer theory of evidences, and fuzzy set-based models are most frequently used (for details see, for example [8], although the aforementioned uncertainty models are in use during decades). These "classical" models have a long history and are well published. However, they

are practically applicable to low scale applications provided that enough expert information or/and historical data are available for empirical assessment of the respective uncertainty measures with satisfactory accuracy and reliability..

Recently two new approaches to combining decision have been proposed. The first of them is based on "*stacked generalization*" idea whereas the second one consists in *assessment of base classifiers competence*.

The idea of *stacked generalization* [36] is very simple, but it gave birth to several particular techniques of decisions combining. Its most advantageous variant, "*meta-classification*", was proposed in the early 1990s [28].

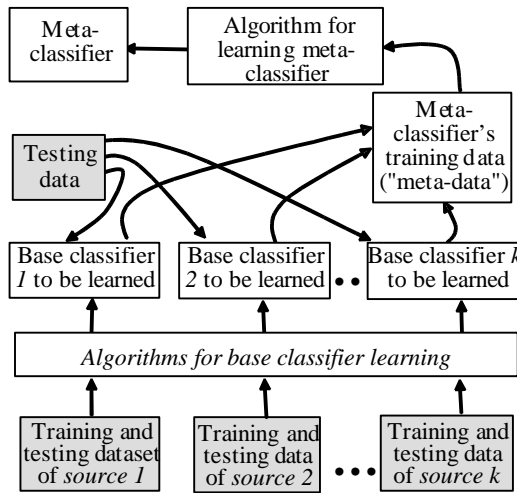


Figure 12. Meta-classification scheme

A generalized structure of decision combining based on "*meta-classification*" is illustrated in Fig. 12. Meta-classifier is a conventional rule-based classifier, which input is constituted by the row of outputs of the source-based classifiers. For training of the meta-classifier the two-step technique is used. In the *first step* so-called *meta-data sample* is computed. It is done based on testing of the source-based classifiers. For this purpose a special testing data sample is used. Let us remind that a complete specification of a situation instance comprises fragments corresponding to the data of particular data sources. The main requirement to the above testing sample is that it has to be complete, i.e. each instance of this sample has to have no aforementioned fragments missed.

For each instance of the testing sample the result of testing of base classifiers is represented as a row of classification labels produced by them.

Some of these labels can be correct, other ones—incorrect. This row is extended by the correct classification label and the resulting row in the next step is considered as an instance of the *meta-data* to be used for training and testing of meta-classifier. The *second step* of meta-classification algorithm design consists in conventional training and testing performed based on the above meta-data sample.

In general, *stacked generalization*-based techniques of decision combining are effective and still being actively researched. A drawback of this group of techniques is their inability to preserve unchanged the already existing classifiers if a new classifier is inserted into the classification system. In contrast, the considered below competence-based group of techniques is free of this drawback..

The basic idea behind the competence-based group of techniques exploits the natural assumption that each classification algorithm is the most "competent" while dealing with a particular region of the representation space. Therefore *evaluations of classifiers' competences* with regard to each particular record of input data is the central procedure of such algorithms. This idea was firstly proposed in [33] and [22].

The focus of this approach is the use of a special procedure called "*referee*" (see Fig. 13) associated with each particular base classifier, with the responsibility of assessing the competence of "own" classifier with regard to input data [26]. To provide the referee with the ability to do so, a learning procedure can be exploited. Referee learning is a common learning task, which is solved on the basis of the same learning dataset that is used for the learning of its respective classifier. The only distinction is that while testing a base classifier, it is necessary to assign a label indicating whether the classification of this particular example is true or erroneous to each example of the testing data set. As the result of the testing, the examples of the testing data set are divided into two classes. One of them corresponds to the class of the examples classified correctly by the base classifier (the region where the classifier is "competent") and the second corresponds to the class of the examples classified incorrectly (the area of the classifier's "incompetence"). This partition of this testing data set is further used for training referee of the respective classifier. Certain measures may be used by referees for assessment of the classifier's competence within particular regions.

Thus, in competence-based techniques, decision combining procedure consists of two steps: (1) detection of the most competent classifier and (2) selection of the classification produced by the most competent classifier.

Important improvements of this method were proposed in [32] and [34]. The main advantages of competence-based technique are higher accuracy (as compared with both voting and stacked generalization-based techniques) and

also the capability of preserving in already existing set of classifiers unchanged if a new classifier is inserted in the decision combining model.

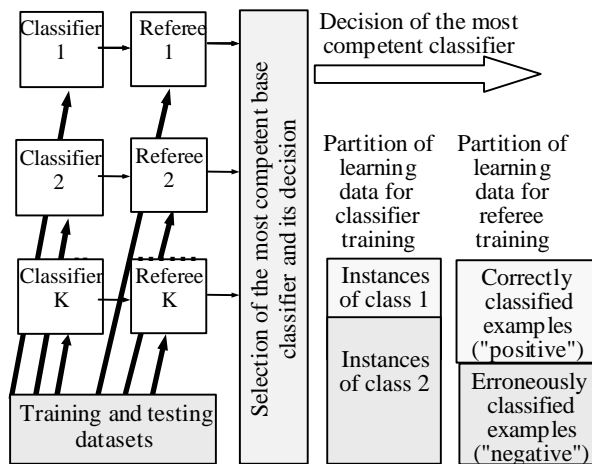


Figure 13. Explanation of the idea of a competence-based approach to combining decisions of multiple classifiers

Two types of methods discussed in this section, namely meta-classification and competence-based methods, are so far used in the IF software tool as decision combining techniques.

6. MULTI-AGENT ARCHITECTURE OF INFORMATION FUSION SYSTEM

Multi-agent architecture is used for both learning and decision making components of IF MAS of the class considered in the paper. Both components comprise two groups of software components (Fig.14): (1) those handling the source-based data; and (2) those manipulating meta-data generated on the basis of source-based data. As a rule, the components of the first group are situated in the same hosts as databases of data sources.

In Fig. 14, the learning components of IF MAS are denoted as *Training and Testing agents*, or *TT-agents*. Let us note that the learning components can be absent in the target IF MAS if they are not intended to further use for improvement of its performance.

A more detailed architecture of IF MAS is presented in Fig. 15. In this figure, the learning components (both source- and meta-levels) are depicted on the left and the components performing decision making functions (also

both source- and meta-levels) are depicted on the right. Let us outline the structure and functionalities of the IF MAS architecture components, most of which are agents of different classes having particular roles in the system.

The source-based components (Fig. 15, lower part) are common to each source. They are as follows:

Data source managing agent

- Participates in design of the shared and private parts of the application ontology;
- Collaborates with meta- agents in management of training and testing datasets, participates in base classifiers learning and in computing of meta-data for meta-learning;
- Supports a gateway to its database via transformation of queries from the language used in ontology into SQL language;

KDD data source agent

- Trains and tests the base classifiers constituting the source-based classification agent and assesses the quality of their performance. In training and testing, it uses the library of learning methods, ontology, and source-based training and testing datasets.

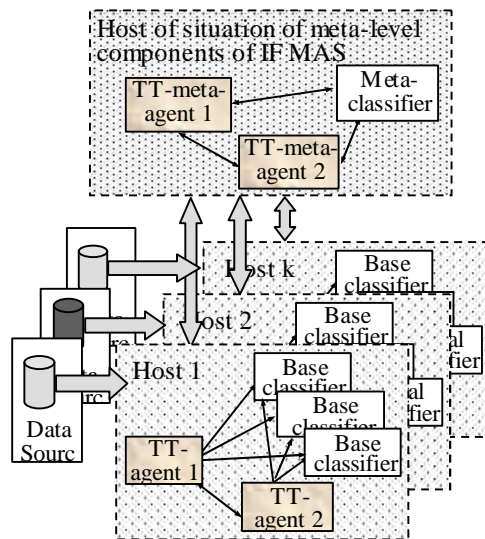


Figure 14. General view of IF MAS architecture .

Source-based classification agent

- Produces decisions using source input. It is the subject of learning performed by the KDD data source agent. This agent is composed of

several base classifiers structured according to decision making model.

Server (library) of training and testing (learning) methods

This component is not an agent. It comprises a multitude of software classes implementing particular KDD methods, metrics, etc.

The Meta-level components of IF MAS (Fig.15, upper) are, as follows:

Agent Master of meta-learning (“KDD Master”)

- Manages the distributed design of the shared application ontology;
- Computes the training and testing meta-data samples;
- Manages design of the DF meta-model (decision and classification trees).

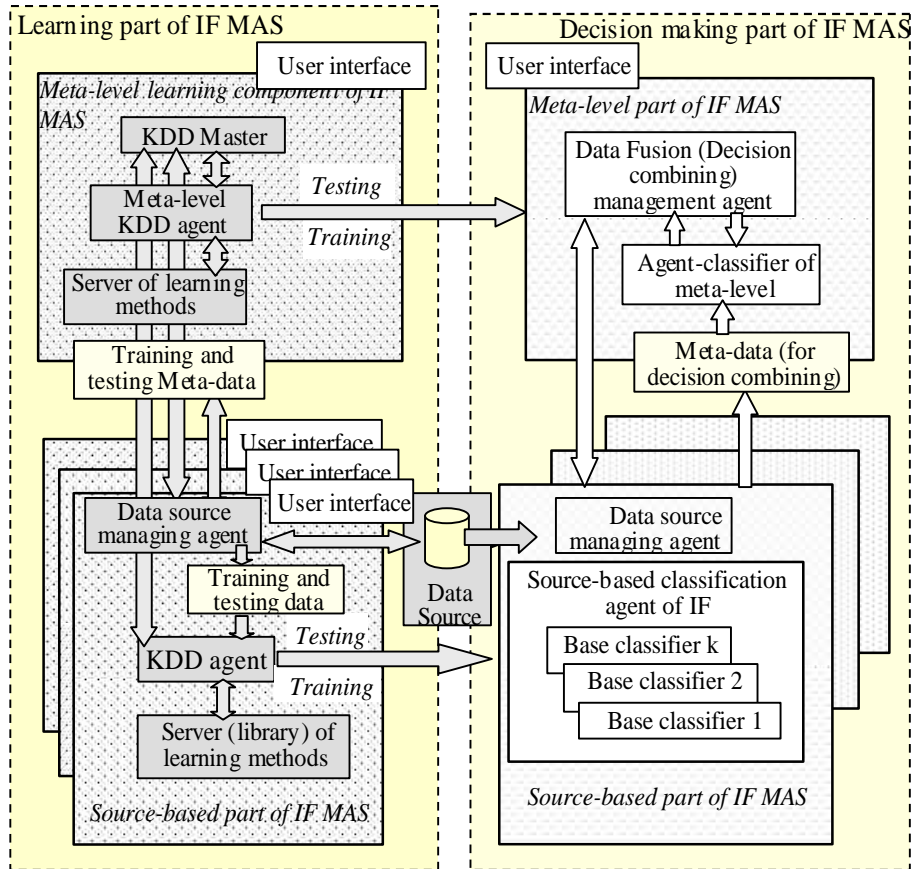


Figure 15. Architecture and interaction of IF MAS learning (left) and IF MAS decision making (right) components

Meta-level KDD agent

- Trains and tests meta-level classification agent and assesses its quality.

Agent-classifier of meta-level

- Produces decision combining using meta-level information. It is the subject of learning performed by the Meta-level KDD agent of IF MAS.

Decision combining management agent

- Coordinates behavior of the meta-level agent-classifier and Meta-level KDD agent both in learning and decision combining modes.

Communication environment

In the development of the agents' communication environment KQML language is used as message content wrapper, while the content itself is specified through the use of XML representing content within an application ontology. The transport level of message wrapper also corresponds to the standard protocol that is TCP/IP.

A conceptual view of the structure of the IF MAS agent communication environment is depicted in Fig. 16. Each communication act is supported by three intermediate components that are:

- Portal of the computer in which an agent sending a message is situated;
- Portal of the computer in which an agent-addressee is situated;
- Communication meta-agent (agent-facilitator) of the IF MAS supporting message addressing.

In IF MAS, these components provide the complete transport services.

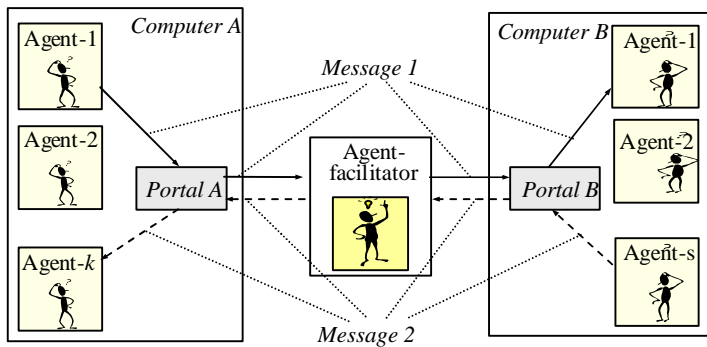


Figure 16. Architecture of the communication environment: Message exchange routing

Protocols supporting agent communication are divided into three groups:

1. Protocols that support agent *message exchange* in accordance with the commonly accepted three-level scheme in MAS: "message *transport* protocol" (message envelope) – "message *syntax* specification" – "message *content* specification".
2. Protocols managing *semantically interconnected dialogs* (conversations) of agents that take place if agents need cooperation in task solving. These protocols is of meta-level in regard to the first group protocols.

3. Protocols supporting agent interaction in cooperative design, learning and decision making procedures. They are considered in further detail in the subsequent section. These protocols plays the role of meta-protocols in regard to the protocols of the first and second groups.

7. INFORMATION FUSION MAS TECHNOLOGY: DESIGN AND IMPLEMENTATION ISSUES

7.1 General View of the IF MAS Technology

The developed IF MAS technology is based on the methodology presented in the section 4. It consists of two basic phases supported by two types of software toolkits, respectively. Fig. 17 explains the hierarchy of design phases comprising the technology.

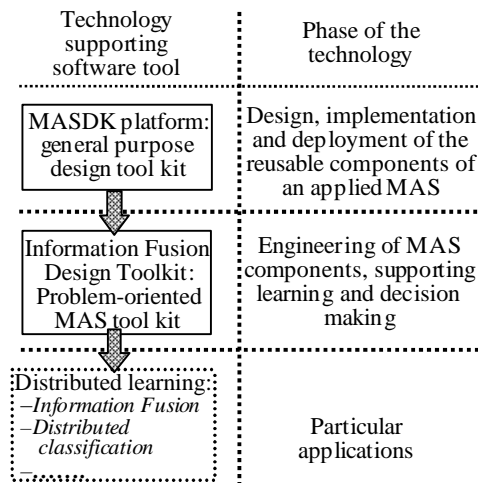


Figure 17. Explanation of the IF MAS engineering technology. DDM–Distributed Data Mining; DM–Decision Making

The first phase mainly² aims at design, implementation and deployment of so-called *Generic IF MAS*, which comprises a part of *application ontology*, *agent classes* and *agent instances* deployed in the network, and the *communication environment*. In fact, the first phase of applied IF MAS technology is mostly destined for engineering of its general-purpose components and also some problem-oriented components. Respectively, in *Generic IF MAS* the agent classes and their instances are provided with a

part of IF problem-oriented functionalities and problem ontology. This phase of IF MAS design is mainly supported by a software toolkit called *Multi-agent System Development Kit*, MASDK [10], which is a general purpose MAS software tool. The "technical" aspects of its practical use are sketched later in this section.

The second phase is mainly destined for *specialization* of the *Generic IF MAS*, developed in the first phase into an application of interest. In this phase, the technology is supported by the so-called *Information Fusion Design Toolkit*, *IFDT*, developed by authors. It is an IF problem-oriented software toolkit, which, together with MASDK provides complete support for the development and deployment of IF MAS applications.

The components of IFDT are presented in Fig. 18. They can be divided into three groups: (1) protocols supporting collaborative engineering of applied IF MAS and also the cooperative operation of its agents; (2) library of training and testing methods, and (3) user interfaces supporting designer activity. It is should be noted that the *Information Fusion Design Toolkit* actually implements a new kind of agent-oriented software engineering that could reasonably be called as "*Agent-mediated software engineering.*"

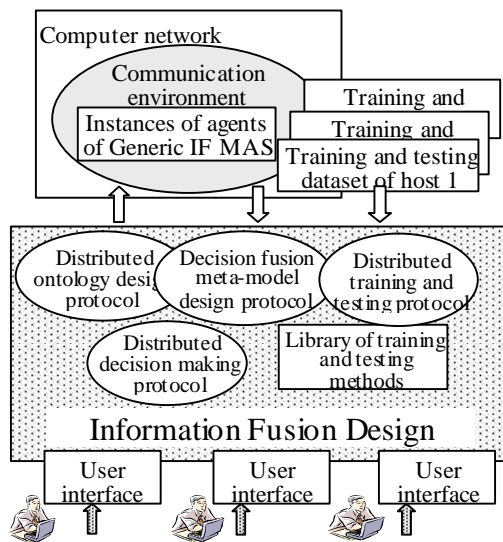


Figure 18. IF Design Toolkit

As compared with the existing variants of the agent-oriented software engineering, the new features specific to the technology supported by IFDT are (1) *Distributed design* of an applied IF MAS that in some cases (private or classified training and testing datasets) is the only admissible one, and (2)

use of *agents as mediators* of designers in engineering procedures. A high-level scenario (protocol) of applied IF MAS design supported by IFDT is presented in terms of the IDEF0 diagram in Fig. 19.

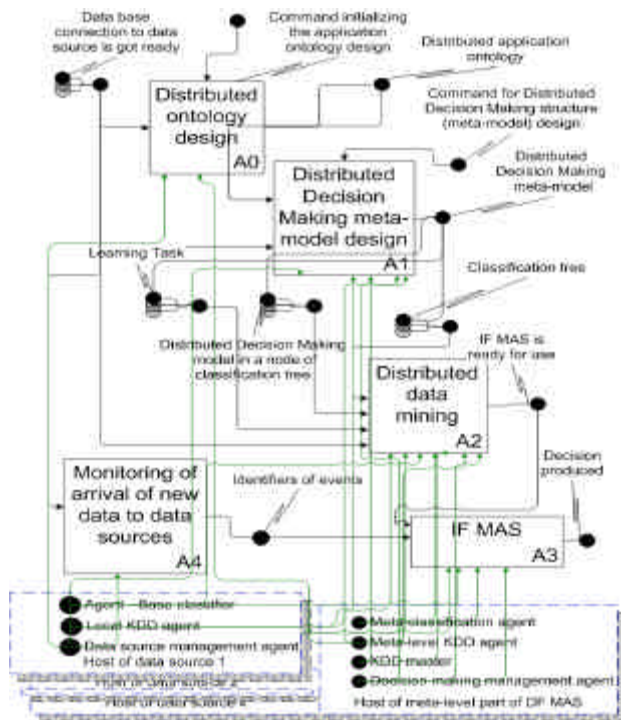


Figure 19. High-level protocol of the distributed design of IF MAS

The main components of the above scenario are as follows:

- A0. Distributed engineering of shared and private parts of application ontology.
- A1. DF meta-model design, i.e. design of decision making and classification trees.
- A2. Distributed data mining for engineering of the IF system distributed KB used in decision making processes according to DF meta-model.
- A3. Design IF MAS operation scenario processing new input data, i.e. scenario of distributed decision making.
- A4. Monitoring of arrival of new data to data sources.

Fig. 19 specifies the interaction of agents in design procedures, input, intermediate and final results and the order of activities processed. The core sub-protocols of the above technology are those providing IF MAS with

learning capabilities, i.e. A0, A1 and A2 protocols. The sub-protocol A3 supports operations of IF MAS intended for classification of new input data specifying a situation. The above sub-protocols are described in section 7.3.

7.2 IF MAS technology support: Design and implementation of reusable components

Until now, many multi-agent software tools and platforms (more than 70) have been developed. Among them, the most popular are JADE [4], Zeus [6], Bee-geat [3], MadKit [17], FIPA-OS [27], etc. However, despite the diversity in existing tools, most are still in the research stage and, as a rule, possess limited capabilities.

Most of the existing MAS software tools exploits the idea of reusability assuming that, in different MAS, there exist many common functionalities and data structures that are practically independent of applications [31] and thus can be used as generic classes and structures. This idea is also the basic one behind the MASDK [10] used to support design, implementation and deployment of the reusable components in the IF MAS.

MASDK comprises two parts destined for different purposes. *The first* of them, the so-called *Generic agent*, is considered as an upper level (the most general) class of agents that possesses *reusable* classes and generic data structures of MAS. *The second* part is composed of a number of particular editors destined for partial specialization of the *generic agent* according to the application of interest.

Generic agent is a nucleus that is "bootstrapped" by a designer using user-friendly editors into agent classes provided with specialized data structures. In the next step, agent classes are replicated and specialized to instances of particular agents. MASDK also supports design and implementation of the communication environment (see Fig. 16) and the deployment of IF MAS within a computer network. Further specialization of agent instances and IF MAS as a whole is supported by IFDT. Let us consider in a little more detail how IF MAS technology is supported by MASDK.

As it was noted above, there are two types of components comprise each agent instance designed with MASDK. The first one corresponds to the reusable components of the "generic agent," while the second one is designed by means of MASDK editors supporting a specialization of generic agent. Fig. 20 presents the composition of these types of components. The *reusable* components of an agent class are presented in the left part of this figure while the specialized (*applied*) ones are presented in its right part. Let us note that part of the applied components given in Fig. 20 is further specialized or designed by the use of IFDT (see next subsection). Let us

briefly describe this figure with particular focus on the functionalities of the particular agent components.

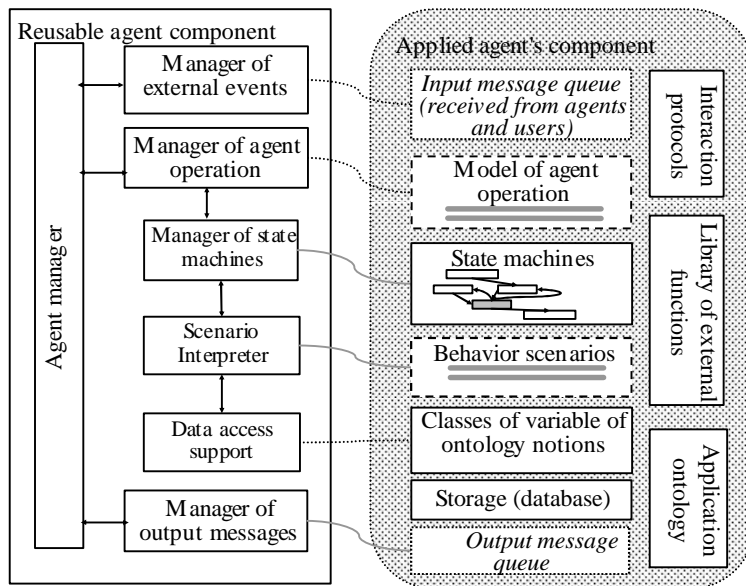


Figure 20. Model of reusable and applied agent's components

The meta-scenario of an agent's behavior is managed by the *Agent manager*. It allocates CPU time for processing of the three main execution threads that are:

- Primary processing of the events incoming to the agent from outside is performed by the *Manager of external events*.
- Analysis of the agent's current state and management of the agent operation according to its state and incoming events is performed by the *Manager of agent operation*.
- Sending of output agent messages produced according to its behavior scenarios and to an interaction protocol under execution. This is done by the *Manager of output messages*.

The first execution thread, *Manager of external events*, identifies the model of the external world (thus forming the agent's belief), which is further analyzed by the *Manager of agent operation*. The latter determines jobs to be allocated to the *Manager of state machines*, which, in turn, initiates particular agent behavior.

Scenarios of agent behavior are represented in terms of *state machines* [5]. A scenario selected for execution is processed by the *Manager of state machines*. It forms a sequence of sub-scenarios that are further allocated for

execution to particular *state machines*. The execution of the scenarios is entrusted to the *Scenario Interpreter*, while access to the agent database is provided when necessary by a component *Data access support*.

Applied agent components are presented in the right part of Fig. 20. These components and their functions are as follows:

- An *Application ontology* representing the classes of notions of the application domain and the relationships between them. This ontology is shared by all agents. The agents use ontology notions in a twofold mode: (1) as a shared terminology providing unambiguous interpretation of messages, which agents exchange with; and (2) as a terminological basis for knowledge representation. It is appropriate to mention here that the agent's knowledge is specified in terms of (1) *preconditions* and *transition rules* designated to every state (node) of *state machines*, (2) behavior scenarios assigned to the states (nodes) of *state machines*, and (3) behavior scenarios performed in transition of a *state machine* from its current state (node) to a new state in accordance with the transition rules.
- *Interaction protocols* specifying coordination of agents' interaction and behavior in the joint execution of certain distributed algorithms;
- *Library of external functions* containing names of methods represented as executable code used for solving specific sub-tasks. External functions are invoked by state machines representing agent behavior scenarios;
- *Input message queue* ordering messages in temporary storage;
- *Model of agent operation* containing upper-level specification of an agent's behavior represented in terms of particular state machines. It specifies rules determining functions to be executed depending on the current agent's state;
- *State machines* specifying agent meta-scenarios. A state machine specifies the states and transition rules to be selected for execution according to both the preconditions designated to state machine nodes and input data;
- *Behavior scripts* specifying agent behavior corresponding to its particular states and transitions;

Specialization of an agent's reusable components is supported by a number of specialized editors provided with user-friendly interfaces. It should be noted that design operations in which designers use MASDK are alternated with the operations supported by IFDT described below.

7.3 Information fusion design toolkit

The Information Fusion Design Toolkit (IFDT) is used together with MASDK. It primarily supports design of the application-specific IF MAS

components and protocols for IF MAS agent interactions. The peculiarities of the technology supported by it are as follows:

1. It supports a distributed mode of IF MAS design mediated by agents. In the design process, designers interact each other according to a number of hierarchically specified protocols monitoring and managing the designers' activity. The toolkit is primarily oriented to engineering spatially distributed IF applications with focus on agent interaction protocols. It is also suitable for engineering IF MAS in the case that data sources contain private data unavailable for centralized processing.
2. This toolkit provides support for such sophisticated IF MAS design activities as maintenance of the consistency of shared and private components of application ontology, design of the DF meta-model, learning of distributed classification and also engineering of the decision fusion procedure itself.

General explanation of the technology supported by IFDT is given in subsection 7.1. The collaboration of participating designers and interaction of agents mediating designers' activities are explained in Fig. 21. Although design activity initiative mostly belongs to designers, they act under the monitoring and management of the design protocol (see Fig. 19), which enforces the designers to follow a predefined design discipline.

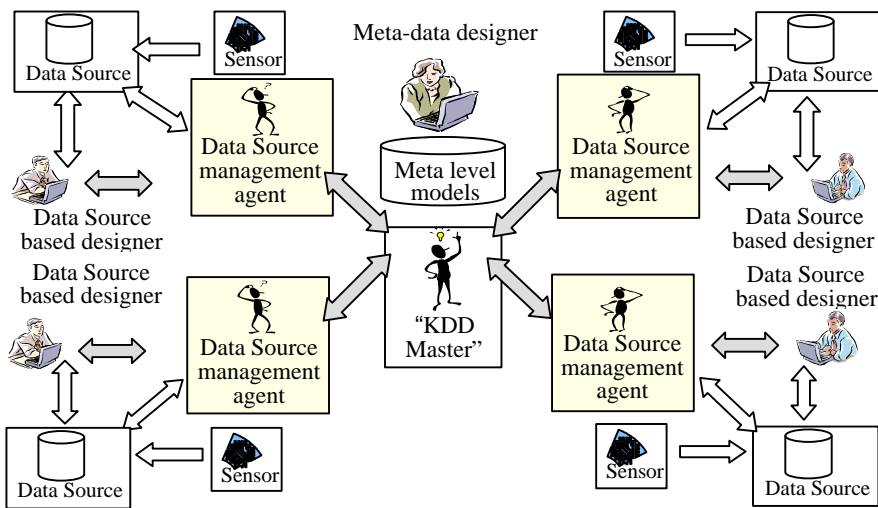


Figure 21. Explanation of the agent-mediated technology support provided by Information Fusion Design Toolkit

Let us consider the steps of the protocol given in Fig. 19 in a little more detail. The processes comprising this protocol are below described

superficially, although each of them is represented in several levels of detail up to the sub-processes that do not suppose distributed execution.

7.3.1 Distributed ontology design

Design of application ontology is supported by both MASDK and IFDT toolkits. High level ordering of the design activities supported by IFDT, represented in terms of standard IDEF0 diagrams, is given in Fig. 22. It also indicates agents participating in the execution of particular activities. The particular design tasks to be solved by the designers operating according to this protocol aim at providing IF MAS application ontology with the properties that were discussed in subsection 4.3. The main steps of the ontology design procedure are briefly explained below.

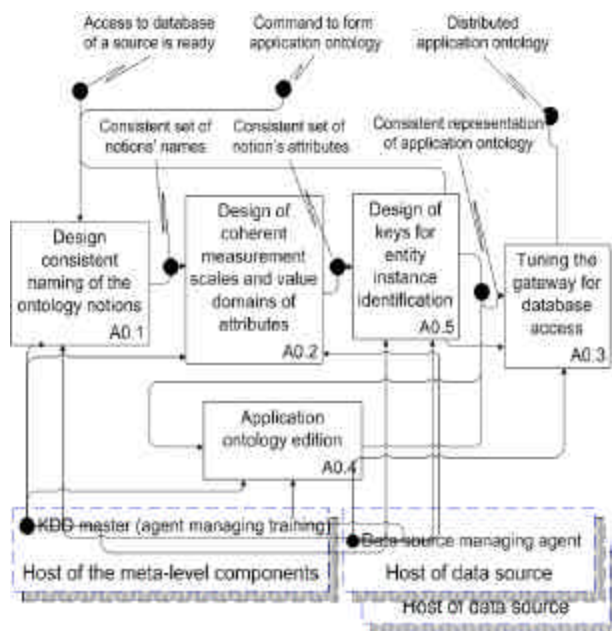


Figure 22. Protocol ordering ontology design activities supported by IFDT

A0.1. Design of consistent naming of the ontology notions

This activity is also performed according to a sub-protocol (we omit its specification due to limited paper space) specifying the A0.1 protocol in the next level of detail. Analysis and maintenance of the consistent naming of ontology notions is led by the meta-level designer (see Fig. 21) mediated by

the *KDD master agent* (see Fig. 15, 21), which interacts with source-based designers via mediation of *Data source management agents*.

A0.2. Design of coherent measurement scales of the ontology notion attributes

The motivation of this task and its objective were outlined in subsection 4.3. Let us describe how this task is solved. Let us denote an attribute measured differently in different sources as *X*. In the shared application ontology the type and the unit of the attribute *X* measurement are determined by the meta-level designer. At the next step, in all the sources in which this attribute is presented, expressions are determined for it through which the attribute can further be converted into the same scale in all the sources. This allows the use of the values of attributes in the meta-level regardless of their measurement in the particular data sources from which they originated. Agreement on the units to be commonly used for attribute *X* is reached via negotiations according to a protocol. Such a protocol is an IFDT component.

A0.3. Design of keys for entity instance identification

The entity instance identification task (see subsection 4.3) is solved in the following way. In the application ontology, for each entity the notion of entity identifier ("*ID entity*") is introduced. This ID entity plays the role of the entity primary key (in analogy with the primary key of a database table). For each such identifier, a rule given over the attributes of the notions of application ontology is defined. For example, in some cases, when data fragments specifying a situation snapshot (instance) are time stamped, a time interval can be used as attribute of the predicate determining the aforementioned rule. This rule is further used to calculate the value of this key. A specific rule is defined for each particular data source that allows to uniquely connect the entity identifier and the local primary key in this source. In a special case, it may be a list of pairs "value of entity identifier" – "value of local key." When such rules are determined for each source it is possible to form the list of all the entity instances in the meta-level. This list identifies all fragments of the same instances presented in different data sources.

In current research specifications of the applied IF MAS, ontology is so far written in terms of the XML language. In the next version of MASDK, RDF, DAML+OIL languages [20] are considered for use.

Use of XML or other pertinent languages to represent ontology leads to an additional IF specific problem. Indeed, instances of the ontology notions are stored in databases and to access them the query must be specified in terms of standard SQL language. Since databases "do not understand" queries represented in terms of XML or its derivatives, it is necessary to provide ontology-based systems with a special "gateway". The function of

this gateway is to transform ontology language-based queries into SQL-queries. The sub protocol A0.3 supports the design of such a gateway.

7.3.2 Decision fusion meta-model design

According to the high level protocol (Fig.19), design of the DF meta-model is carried out at the second step of technology supported by IFDT. In section 4.2, the role and general structure of the DF meta-model in learning and decision fusion are described. Let us remind that the upper level of the DF meta-model is constituted by a classification tree (see Fig. 9). To each node of this tree which corresponds either to a meta-class or class of situation, a decision tree is mapped. The latter can also be composed of several levels (Fig. 9).

The design procedures, their ordering and agents participating in the particular design procedures of the DF meta-model are depicted in Fig. 23. The main steps of this phase of the technology are as follows:

1. Specification of the decision making task (A1.1).
2. Design of the structure of decision making (decision making tree) (A1.2).

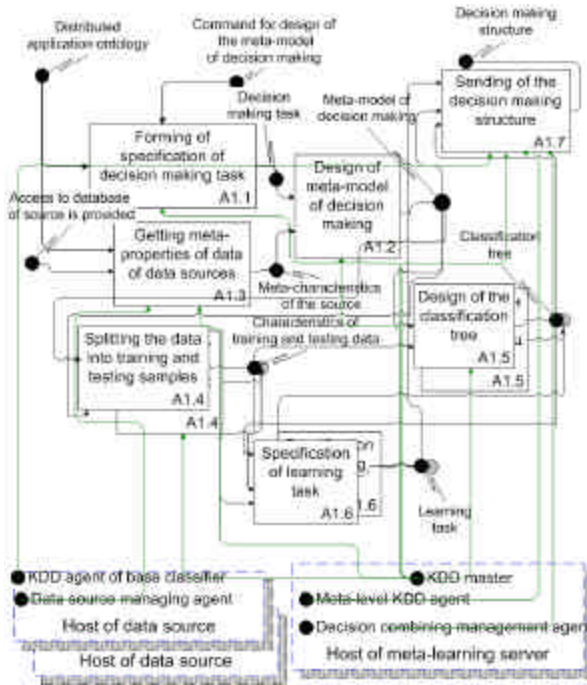


Figure 23. Protocol of DF meta-model design

3. Getting meta-properties of the data of data sources (A1.3).
4. Splitting the data into training and testing samples (A1.4).
5. Design of the classification tree (A1.5).
6. Specification of the learning task (A1.6).
7. Forwarding the decision making structure (A1.7).

These processes are mediated by the *KDD master*, *Meta-level KDD agent*, and *Information Fusion management agent*, which are meta-level agents, and also by the *Data source managing agents* and *KDD agents of base classifiers* situated in the same hosts as data sources.

7.3.3 Distributed data mining protocol

This protocol is the core of IF system technology because it supports the basic functionalities of IF MAS design. The latter are training and testing of particular classifiers and design of the decision combining procedures. An IDEF0 diagram of this protocol is presented in Fig. 24. It involves the interaction of all the agents of the IF MAS. Part of these agents play the roles of design mediators; the rest of the agents are designed with support of the above mediators.

The basic processes of the distributed data mining protocol are:

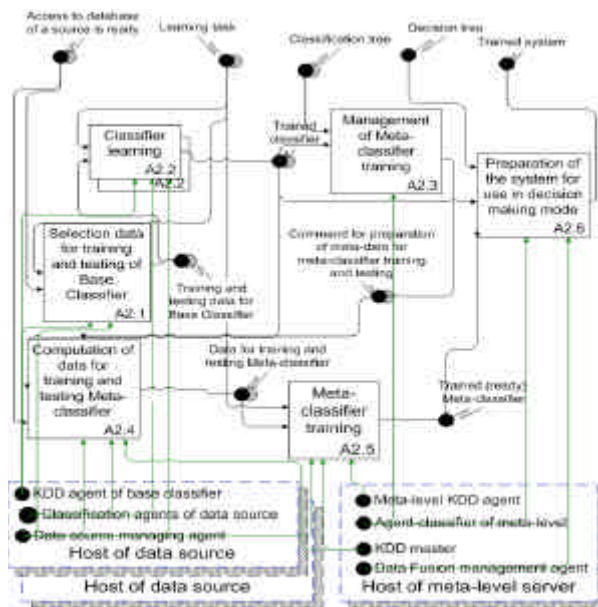


Figure 24. Distributed data mining protocol

1. Selection of data for training and testing of the base classifiers (A2.1).
2. Training and testing of classifiers (A2.2).
3. Management of meta-classifier training (A2.3).
4. Computation of data for training and testing meta-classifier (A2.4).
5. Training and testing of meta-classifier (A2.5).
6. Preparation of the IF system for use (A2.6).

The sub-processes (sub-protocols) of the distributed data mining composing the protocol A2 are specified in the same style in several levels of detail up to the level where the processes are executed by particular agents.

7.3.4 Protocols for distributed decision making (decision fusion)

The components of this protocol, that in turn is specified in the lower levels in more details, are as follows (Fig. 25):

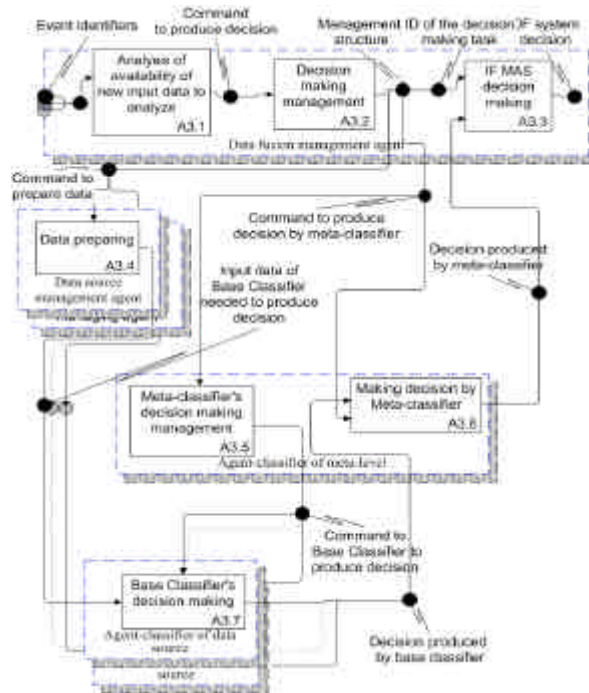


Figure 25. Distributed decision making protocol

1. Analysis of the availability of new input data to process (A3.1).
2. Decision making management. This sub-protocol manages the production of decisions according to the *Decision fusion meta-model* (A3.2).

3. Preparing of data (A3.4).
4. Decision making by base classifiers (A3.7).
5. Management of meta-classifier decision making (A3.5).
6. Decision making by Meta-classifier (A3.6).
7. IF MAS decision making (A3.3).

Let us note that the above sub-protocols are listed in the order corresponding to the order of their execution in decision making procedures.

8. CONCLUSION

This paper presents a methodology of data and information fusion, a multi-agent architecture of the respective software system, the technology destined for the design, implementation and deployment of applied multi-agent data and information fusion systems, and also outlines software tools supporting the developed technology.

An application area of the above methodology and technology concerns the tasks of object and situation assessment corresponding to levels 1 and 2 of the JDL model of information fusion [30]. In turn, data and information fusion are the core tasks in providing for situational awareness.

In the presented methodology, technology and software tool the multi-agent paradigm is used. It provides designers with powerful tools for conceptual modeling of the data and information fusion problems, adequate architectures, and provides with techniques for implementing cooperation of distributed software components. This tool is specifically destined for decomposable large scale intelligent systems for data and information fusion.

It is important to note that in design of Information Fusion MAS we distinguish two aspects of utilization of the multi-agent paradigm:

- Utilization of the *multi-agent architecture* for the software implementation of data and information fusion systems (*Multi-agent data and information fusion system, IF MAS*); and
- Utilization of the *agent-mediated software engineering* for IF MAS technology, in which agents play the roles of mediators between distributed designers supporting a predefined "design discipline" specified formally in terms of interaction protocols, and perform a lot of routine computations.

Both of these aspects are the subjects of this paper.

The developed methodology provides designers with a number of ready solutions concerning such important methodological aspects of IS MAS design as:

1. How to allocate functions of data and information processing to data source-based level and meta-level of fusion;
2. How to structure decision making and decision combining components of the IF system, i.e. how the DF meta-model should be organized;
3. How to structure an IF system distributed knowledge base, how to provide and maintain consistency of its distributed components and how it could interact with the IF MAS ontology;
4. What kind of data mining techniques to use for learning of the IF MAS decision making and decision combining components.

The developed two phase technology, i.e. (1) design of the IF MAS with use of MASDK that is software tool of general purposes (see section 7.2) and (2) design of the application-specific IF MAS components and protocols for IF MAS agent interactions (see section 7.3) provides designers with a number of flexible and powerful software means supporting engineering, implementation and deployment of applied IF MAS. In particular, engineering processes performed by a distributed team of designers, support such sophisticated design procedures as the design of decision fusion meta-models. These processes also support engineering of distributed IF MAS knowledge base, whose upper level is constituted application ontology, and also design of the protocol fusion of decisions produced by source-based distributed classifiers.

The developed methodology, technology and software tools were validated on the basis of two well-known case studies that are Multi-spectral image classification (Landsat Scanner image dataset, GRSS_DFC_0010, [18]) and KDDCup99 [19].

Currently the described technology and both aforementioned software tools are being further developed. Current efforts are directed also to IF technology validation and accumulation of the experience in IF MAS design via design and implementation of various applications from IF scope.

ACKNOWLEDGEMENTS

This work is supported by the European Office of Aerospace Research and Development (EOARD, department of US AFRL/IF) and Russian Foundation for Basic Research (Project #04-01-00494a).

NOTES

1. Actually, meta-classification approach can be also considered as a special case of one proposed in "Inferential Theory of Learning" [24] mentioned in subsection 5.2.

2. Practically, the described below "first" and "second" phases of an applied IF MAS design overlapping and some design operations called here as "first phase operations" can be carried out after some operations referred to as second phase operations.

REFERENCES

1. E.Adam. Fighter Cockpits of the Future. Proceedings of 12-th IEEE/AIAA Conference on Digital Avionics Systems, 1993, 318-323.
2. T.Bass. Intrusion Detection and Multisensor Information Fusion: Creating Cyberspace Situational Awareness. *Communication of the ACM*, vol.43(4), 2000, 99-105.
3. Bee-gent Multi-Agent Framework. *Toshiba Corporation Systems and Software Research Laboratories*. 2000, <http://www2.toshiba.co.jp/beegent/index.htm>.
4. F.Bellifemine, A.Poggi, and G.Rimassa. JADE – A FIPA-compliant agent framework. *Proceedings of PAAM'99*, London, UK, 1999, <http://sharon.cselt.it/projects/jade>.
5. G.Booch, J.Rumbaugh, and A.Jacobson. The unified modeling language user guide. *Adison Wesley Longman*, 2000.
6. J.Collis, D.Ndumu: The Zeus Agent Bilding Toolkit. ZEUS Technical Manual. *Intelligent Systems Research Group*, BT Labs. Release 1.0. 1999. <http://193.113.209.147/projects/agents/index.htm>.
7. M.R.Endsley and D. G. Garland (Eds.). Situation awareness analysis and measurement. *Mahwah, NJ: Lawrence Erlbaum*, 2000.
8. I.Goodman, R.Mahler, and H.Nguen. Mathematics of Data Fusion. *Kluwer Academic Publishers*, 1997.
9. V.Gorodetski, O.Karsayev. Mining of Data with Missed Values: A Lattice-based Approach. *International Workshop on the Foundation of Data Mining and Discovery in the 2002 IEEE International Conference on Data Mining*, Japan, 2002, pp.151-156.
10. V.Gorodetski, O.Karsaev, I.Kotenko. *Software Development Kit for Multi-agent Systems Design and Implementation*. In B.Dunin-Keplicz and E.Nawareski (Eds.) "From Theory to Practice in Multi-agent Systems". *LNAI*, vol. 2296, Springer Verlag, 2002, 121-130.
11. V.Gorodetski, V.Skormin, L.Popyack. *Data Mining Technology for Failure Prognostics of Avionics*, *IEEE Transactions on Aerospace and Electronic Systems*. Volume 38 (2), 2002, 388-403.
12. V.Gorodetski, V.Skormin, L.Popyack, and O.Karsayev. Distributed Learning in a Data Fusion System. *Proceedings of the Conference of the World Computer Congress (WCC-2000) "Intelligent Information Processing" (IIP2000)*, 2000, 147-154.
13. V.Gorodetski and O.Karsayev. Algorithm of Rule Extraction from Learning Data. *Proceedings of the 8th International Conference "Expert Systems & Artificial Intelligence" (EXPERTSYS-96)*, 1996, 133-138.
14. V.Gorodetski, I.Kotenko and O.Karsaev. Multi-agent technologies for computer network security: Attack simulation, intrusion detection and intrusion detection learning. *International Journal of Computer Systems Science & Engineering*. Vol.18, No. 4, July 2003, pp. 191-200.
15. J.Han and M.Kamber. Data Mining. Concept and Techniques. *Morgan Kaufman Publishers*, 2000.
16. <http://www.semanticweb.org>
17. <http://www.madkit.org>
18. <http://www.dfc-grss.org>

Multi-Agent Data and Information Fusion

19. <http://kdd.ics.uci.edu/databases/kddcup99/>.
20. <http://www.w3.org/DesignIssues/Semantic.html>
21. International Society of Information Fusion. The Fusion Problem. <http://www.inforfusion.org/mission.htm>
22. C.J.Merz. Combining classifiers using correspondence analysis. In *Advances in Neural Information Processing*, 1997.
23. R.Michalski and A.Kaufman. Data Mining and Knowledge Discovery: A Review of Issues and Multistrategy Approach. *Machine learning and Data Mining: Methods and Applications* (Eds. R.Michalski, I.Bratko and M.Kubat) John Wiley&Sones Ltd., 1997.
24. R.Michalski. The Inferential Theory of Learning: Developing of Foundations of Multistrategy Learning. *Machine Learning: A Multistrategy Approach, volume 4* (Eds. R.Michalski and G.Tecuci), *Morgan Kaufman Publishers*, 1993.
25. R.Michalski. A Theory and Methodology of Inductive Learning. *Machine Learning*, vol.1, Eds. J.G.Carbonel, R.S.Michalski and T.M.Mitchel, Palo Alto, pp.83-134, 1983.
26. J.Ortega, M.Coppel, and S.Argamon. Arbitrating Among Competing Classifiers Using Learned Referees. *Knowledge and Information Systems*, 4, 2001, 470-490.
27. S.Poslad, S.Buckle, and R.Hadingham: The FIPA-OS agent platform: Open Source for Open Standards. *Proceedings of PAAM 2000*, Manchester UK, 2000. <http://fipa-os.sourceforge.net/>
28. A. Prodromidis, P. Chan, and S. Stolfo. Meta-Learning in Distributed Data Mining Systems: Issues and Approaches, *Advances in Distributed Data Mining*, AAAI Press, Kargupta and Chan (eds.). 1999. <http://www.cs.columbia.edu/~sal/hpapers/DDMBOOK.ps.gz>
29. L.Rastrigin and R.Erenstein. Methods of collective pattern recognition. Moscow, *Energoizdat Publishers*, 1981. (In Russian.)
30. J.Salerno. Information Fusion: A High-level Architecture Overview. CD *Proceedings of the Fusion-2002*, Annapolis, MD, 2002, 680-686.
31. A.Sloman: What's an AI Toolkit For? *Proceedings of the AAAI-98 Workshop on Software Tools for Developing Agents*. Madison, Wisconsin, 1998.
32. A.N.Steinberg,C.L.Bowman, and F.E.White. Revision of the JDL Data Fusion Model. *Proceedings of the SPIE Sensor Fusion: Architectures, Algorithms and Applications III*, 1999, 430-441.
33. K.Ting. The characterization of predictive accuracy and decision combination. *Proceedings of 13th International Conference on Machine Learning*, Morgan Kaufman, 1996, 498-506.
34. L.Todorovski and S.Dzeroski. Combining classifiers with meta-decision trees. D.A.Zighen, J.Komarowski and J.Zitkov (Eds.) *Proceedings of 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD-00)*, France, Springer Verlag, 2000, 54-64.
35. G.Weiss (Ed.) *Multiagent Systems. A modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.
36. D.Wolpert. Stacked generalization. *Neural Network*, 5(2), 1992, 241-260.
37. M.Wooldridge. An Introduction to Multi-agent Systems. *John Wilie&Sons*, 2001.