The Technology for Development of Decision-making Support Services With Components Reuse

V. V. Gribova, A. S. Kleschev, P. M. Moskalenko, V. A. Timchenko, E. A. Shalfeeva

The Institute of Automation and Control Processes
IACP FEB RAS
Vladivostok, Russia

***Abstract* —** The article represents methodology and technology of development and assembling of cloudy decision-making support systems which use knowledge bases in network representation. The system analysis of problem area and identification of known subtasks allows to reuse tasks ontologies and domain ontologies, to find similar problem solving method for these subtasks. The IACPaaS technology allows to reuse software components which implement certain solving methods in development process of solvers of intelligent tasks. It also allows to reuse information components, integrating them with the solvers at assembling of cloud decision-making support services.

Keywords—intelligent task, knowledge base; decision-making support; cloudy service; cloudy software system; component

I. INTRODUCTION

An automation of daily intelligent activity and its quality management at the present stage demands updating of technologies of intelligent systems development, focused on assembly by reused semantic compatible components, using of the known problem solver methods (task solution methods) and its software realizations. The modern platforms provide developers with sets of tools and services for different problem and task solution, including the scientific problems and focusing on big data processing (the CLAVIRE platform) [1,2]. The majority of the known cloud platforms for applications development support relational submission of databases [2,3]. There are cloudy editors of knowledge bases and ontologies, supporting object-oriented submission of information (Protege) [4,5].

The IACPaaS platform (Intelligent Applications, Control and Platform as a Service) which developed in IACP FEB RUS focused on working with knowledge bases and/or data in network representation and intellectual services creation [6]. Rather independent development of knowledge bases and software parts of such services is supported. The platform has the convenient editor for formation of knowledge bases, databases and another stored information in terms of their meta-information [6]. Nowdays there are supporting tools for three technologies of cloudy services development. Each technology promotes a reuse of program (software) and information components created earlier.

The paper purpose - the description of decision-making support cloudy software systems development technology allowing to reuse information and program components on the IACPaaS platform.

II. Decision-Making Problems and Used Knowledge

A system analysis is the important stage for decisions of intelligent activity support and automatization. All intelligent *task*s solved by specialists, correctness of knowledge using are discussed. We will call *intelligent task* the task which statement (formal description of task) includes *knowledge base* (KB) and *situation model* (its mathematical abstraction - algebraic system AS) as input or output. KB and situation model must be consensual with domain ontology.

At the system analysis stage analysts seek to reveal in intelligent activity those intellectual tasks which statements are known (or to decompose to known subtasks). The different classes of intellectual tasks (called briefly further by *tasks*) are known, the formal statements are proposed for many of them [7,8]. Allocation of famous (known) subtasks allows to reuse problem ontologies, structure of processed information, methods of this information processing and reuse ready components implementing such processing and some methods these subtasks decisions (problem solving methods).

It is shown in the offered multilevel tasks classification [8] that "genetic linkage" takes place between some tasks. The "genetic linkage" is based on the principle of complication of domain properties (fig. 1). The complication of domain properties is meant there are new kinds of data which should be considered during tasks solution in "more difficult" domain (for example, time, space, classes, cause-and-effect relations, events, internal processes in system, not inherent to it), and additional restrictions.

In many domains there are problems considering classifications of situations, and KB include assertions about properties of situations of each class.

Some tasks are related to ontologies which contain time-dependent functional compliances (for representation of time-dependent attributes of dynamic system or a situation, for example, the state).

In many domains there are problems considering the actions (ordered at least partially), leading\conducting to some purpose.

So in multilevel classification of tasks each complication of domain is considered as "a layer of domains with similar features" where a new tasks or expansed statements of a basic tasks are expected. We can see the example of "genetic relation" between task of search of the *hypotheses explaining findings, recognition task* and task of diagnostics (fig. 1).

In the *task of search of the hypotheses explaining findings* it is required to find all hypotheses corresponding to findings and KB, if KB and system' findings are set.

The *recognition task* is considered as specification of a *task* of search of the hypotheses explaining findings where it is required to find all hypotheses of a class of the situation described by the findings.

The major components of *recognition task* can be summarized as follows:

Givens:

- R (system' case findings and characteristics);
- KB (knowledge base).

Goals:

- to fit case into possible classes: $class_{i1}$, … $class_{i\,k}$;
- to find all explanation models AS (R, $class_{i1}$), AS (R, $class_{i1}$).

Constraints:

- all assertions of KB are concordent to findings and characteristics;
- R coordinated with domain ontology;

In *diagnostics task* it is required to define possible cause-and-effect models of system by findings, characteristics of system and events, which were taken place. At that the diagnosis is set of some internal processes of this system which aren't inherent to it.

The *diagnosis task* can be summarized as follows:

Givens:

- R (system' case findings, characteristics and events);
- KB.

Goals:

- to fit case into possible diagnoses Δ_{i1} , … Δ_{ik}; the diagnosis is the subset of system internal processes;

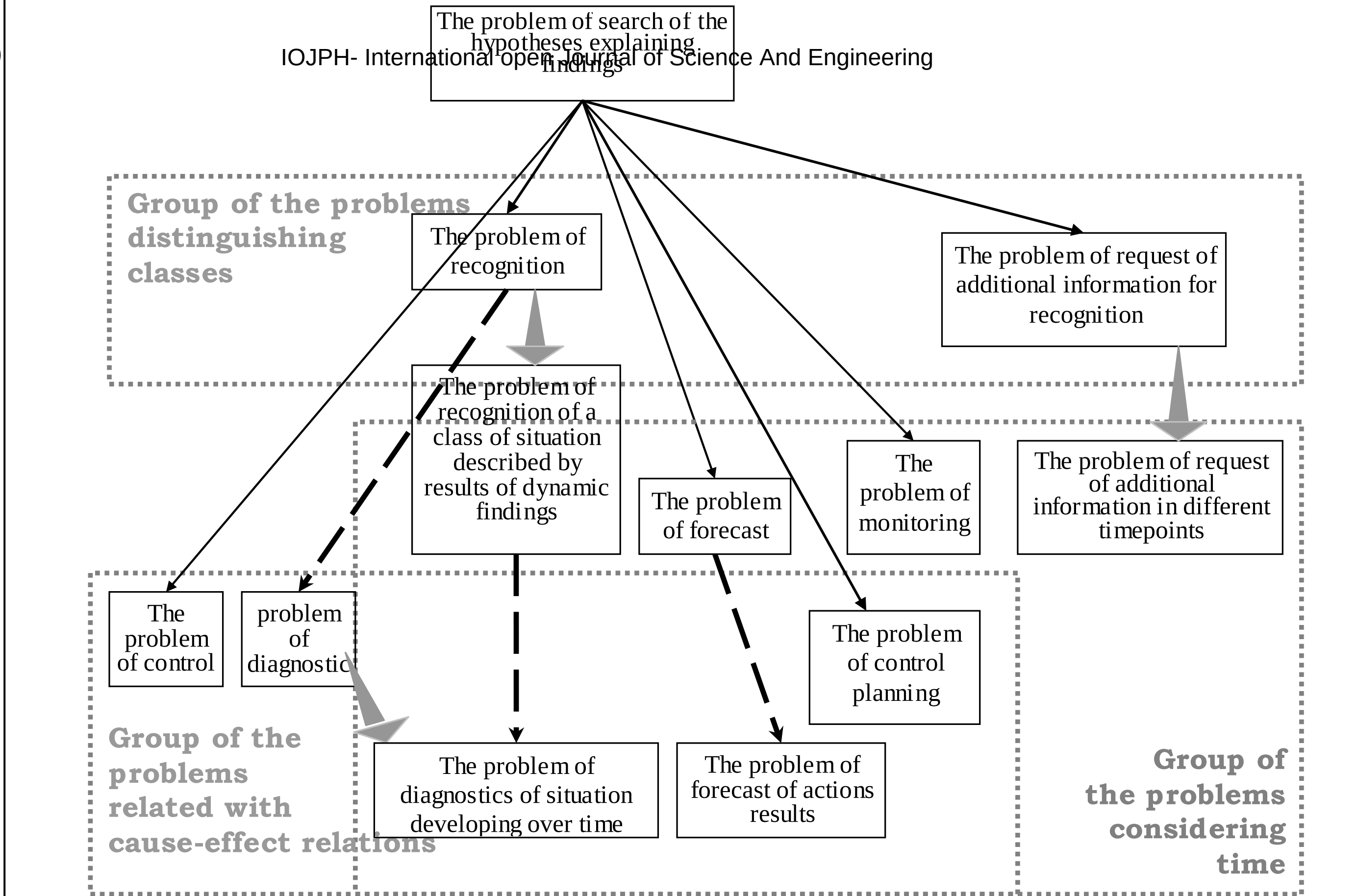

Fig. 1. The fragment of multilevel task classification [8].

- to find all possible cause-and-effect models AS (R, Δ_{i1}), AS (R, $\Delta_{i\,k}$) for explanation of diagnoses Δ_{i1}, … $\Delta_{i\,k}$.

Constraints:

- all assertions of KB are concordent to findings, characteristics and events;
- all cause-and-effect models are concordent to KB.

Classification of tasks with above-mentioned genetic relations gives the chance to choose suitable tasks, to consider their mathematical statements and formulate statements for the tasks of intelligent activity being automated, concretizing concepts and abstractions in the chosen statements..

III. IACPaaS Services Development Technologies with Reuse of Knowledge and Other Components

Just as the majority of cloud platforms the IACPaaS has development tools for all components of applied services. IACPaaS has means of declarative submission of knowledge bases and other stored information in domain terms in the form of hierarchical semantic networks. The development of services processing declarative knowledge, means creation (programming) of specialized solvers.

Each service is being designed with by a multi-component specialized solver and information stored in Fund. The technology is focused not so much on creation of services from scratch how on assembly of services from components - new, ready or modified copies of the existing components.

One of technologies is "basic" or universal technology. Creation of service demands creation of information and the Integrated Solver. If all information, necessary for service, already exists in Fund, then only new solver is required and vice versa. More often in Fund of a platform there can be not all, but some information components, (e.g., KB) and some program (software) components of a solver. The found components need to be integrated into new service.

The main stages of service's development (by universal IACPaaS technology) are:

- creation of input information resources - databases and knowledge bases,
- creation of tasks solver as sets of interacting program units (PU).

To specify and construct the input information resource (IR) it is necessary to take care of the description of its structure (meta-information). If the structure was described earlier, it being reused, otherwise it must be formed explicitly by meta-language. Only if the structure (meta-information) of processed IR is available it is possible to start designing of PU which processes the IR.

It is necessary to notice that structure of output information as rule is unique for every intelligent service. That is why the creation of structure of output information resources is important stage of software service's designing (in universal IACPaaS technology).

Software components of specialized solvers have an obligatory declarative presentation as IR. Each such IR is used by the IACPaaS solver processor in the course of running of service for dynamic connection of components. The common meta-information was formed for constructing these IR-declarations. This meta-information is always reused.

The main stage of creation of the integrated solver is compound "work", it includes creation of software logic components, creation of GUI and creation of meta-information for input and output IR. This work can be executed in two ways: (1) passing consistently all main stages of life cycle, beginning from requirements model, or (2) starting at once to creation of solver design.

The first way assumes "the planning work", including creation of model of requirements. Declarative representation (declaring) of a solver in the form of information resources provides a possibility of the structural description of model of requirements. This model will be projected onto program units being included in solver architecture. In the second way a solver be created without these models. The first way facilitates maintenance process, the second – accelerates obtaining the first version of service.

Declarative representation of a solver contains formal parameters setting formats of the processed information resources (for example, meta-information of used knowledge bases). Declarative representation of a solver contains name of the root agent and can specify set of all other agents who have to realize logic of the solution of a task. There are two levels of architecture designing of a solver. One can specify a set of all agents at one level or specify "control flow graph" which register concrete communications of agents with each other. So the declarative representation of a solver is its design model.

The development requires also: to declare a solver, to declare every PU (an «agent»), to execute design of every PU, to create their communication model, to execute coding, to execute loading and testing of PU.

The design model of each agent contains his interfaces' specification – templates of messages for communication with other agents. If the Fund of a platform doesn't contain suitable (adequate) templates, the developer has to define and execute coding of templates of the messages.

The role of root agent – to receive focus of control, to receive references to IRs being processed in a developed service and to transfer message to delegate to other agents necessary processing of the IR and other work. Service with the user interface will require development of a homepage for solver interface - web page. In this case the root agent is Interface controller which allows service to communicate with the end user through page web. So a root agent is standardized component being generated by tools of IACPaaS.

The basic technology provides the greatest opportunities of a reuse of elements of developed services. The services elements are placed in the Fund or in a private «office» of the developer. The reuse is carried out at the different levels:

- at assembling of service – use of available information resources (input usually) and ready solvers,
- at creation of a solver – use of ready agents solving typical subtasks of processing,
- at creation of agents – use of ready classes of specialized operations of access to information resources,
- at creation of information resources - use of meta-information as language for forming a resource,
- at creation of specialized operations of access to information resources – IACPaaS platform API.

By now a number of intellectual systems of different types is developed on the platform, using the technologies described above. Among them there are an expert system for diagnosis of acute diseases, service for the automated designing of proofs of theorems, service for modeling of transport streams in the developing city infrastructure and some others.

IV. DEVELOPMENT OF MEDICAL SOFTWARE SERVICES WITH REUSE OF KNOWLEDGE BASES AND SOLVERS

The medicine remains subject domain where decision-making support by software services still isn't provided up-to-the-mark. Some reasons of this situation are labor input of creation of the real knowledge bases taking into account a development of processes with time and a technical possibility of collective improvement of knowledge bases. The ontology (structure) of this knowledge, first of all, has to reflect representation of experts and be presented in their terms. The IACPaaS platform aspires to providing tools of representation of such structure of information and functions of access to it which allow to overcome these problems.

One of medical IACPaaS services being created is intended for support of decisions on drug treatment. We will consider the possibilities of a reuse at creation of a medical service.

One of medical IACPaaS services being created is intended for support of decisions on drug treatment. We will consider the possibilities of a reuse at creation of a medical service. This medical service is being created in conditions when other medical services are already created and placed on a platform. Therefore the reuse of components is expected. First of all, the structure (meta-information) of case history (CH) will be reused. CH already created for service of medical diagnostics.

It is required in *the task of drug treatment* (planning of impacts on system) to define such set of events and their timepoints at which findings (as functions of time in cause-and-effect models of system) will reach desirable values, if characteristics, diagnosis and desirable findings of system «patient» are known.

The *task of drug treatment of the patient with the established diagnosis* (*task of management of the patient*) can be summarized as follows:

Givens:

- R_O (patient's findings, observed symptoms and characteristics);
- Δ (diagnosis);
- R_{cond} - (desirable findings),
- KB (knowledge base).

Goals:

- R_{ev} at which new findings R'_{ex} will meet R_{cond};
- explanation AS ($R_{ex} \cup R_O \cup R_{ev}$, Δ, $R_{cond} \subseteq R'_{ex}$);

Constraints:

- R'_{ex} are functions of time which concordent to KB.

Since in practice doctors need consultation, but not the solution of a task instead of them, it is necessary to find in a task full explanation for R_{ev}.

The services is constructed from two KB (the treatment scheme of concrete disease and The Index of medicaments - description of pharmacological means), a database of patients case histories and a solver generating an explanation. Such composition of service provides (at the current stage of development) fast assembly of similar services for different medical profiles. In practice the analysis of contraindications of tens or hundreds of known medicaments (drugs) with the necessary pharmacological influence is connected with essential expenses of time. The doctor, as rule, needs information on what concrete drug shouldn't be appointed for this patient with his "features". The ontology (i.e. structure, a meta-information) explanations of a solver of treatment is formed, taking into consideration these circumstances.

KB of medicaments is common and reusable, it is developed and is followed for all domain. There are a number of KB of schemes of treatment; every KB is created for a separate medical profile according to the standards approved by the Ministry of Health (for example, the standard of "medical care by the patient with acute pancreatitis" is dated November, 2007).

The input data for service is one IB or the Set of IB from which the next IB for decision-making will be chosen. Such Set of IB is being formed not so for a separate medical profile how for "place" of application of service (for the doctor or concrete medical establishment).

Let us consider the possibilities of a reuse at creation of software logic components of the service. The nearest task from the point of view of formal description and problem solving method (decision method) (from classification of tasks [8]) is the control planning task.

It is required in *the task* (planning of control) to define such set of events and their timepoints at which factors (as functions of time in cause-and-effect models of system) will reach desirable values, if characteristics, diagnosis and conditions on factors' values are known.

The *task of planning of control* can be summarized as follows:

Givens:

- R_O (system' characteristics, properties or factors);
- Δ (diagnosis);
- Cond - (conditions on values of characteristics),
- KB (knowledge base).

Goals:

- R_{ev} (set of events and timepoints) at which new characteristics or factors R'_{ex} will meet Cond;

Constraints:

- R_{ev} are functions of in cause-and-effect models of system, concordent to KB.

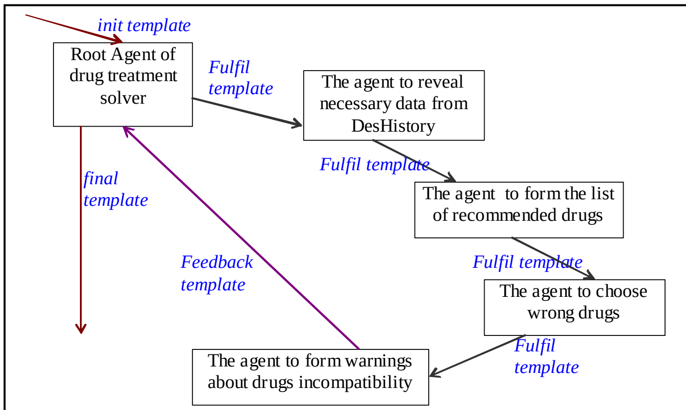

Fig. 2. Example of treatment solver architecture.

In the presence of a ready solver of *control planning task* together with ontology of the processed information in the Fund the reuse of this ontology and of method of decision-making for a solver of *task of drug treatment of the patient with the established diagnosis* is possible. Ontology of the processed information is the meta-information which sets terminology and assosiations between concepts. It has to be detailed: characteristics and attributes of system are concretized to the patient' characteristics.

The ontology of knowledge for planning of control presents such entities and assosiations as impacts on a system and reactions to impacts. It has to be detailed for medicine domain. As a result the ontology for task of planning of control in medicine concretizes such assosiations: therapeutic impacts (actions, drug intakes), reactions to therapeutic impacts.

The solver components which implemented algorithm of information processing and decision-making for control planning task can be adapted (through detailing) for medicine domain. The solving process consists of some steps. The first step consists in obtaining a medicaments list recommended to patients with the specified diagnosis. Further all medicaments which aren't suitable to specific patient (with specified age and other "features" of an organism) are being eliminated from this list. It is supposed that a doctor will want to choose several medicaments from the final list of potentially suitable ones. The scheme of application is usually standard, if necessary it can be proposed by solver. In case of a joint disease (polypathology) the solving process analyses a set of the recommended medicaments in terms of their compatibility in simultaneous reception.

Declarative representation of the solver contains reference to format of the treatment scheme of concrete disease, reference to format of The Index of medicaments (description of pharmacological means), reference to format of stored set of CH, a name of root agent. Besides, depending on way of designing of a solver, the Declaration can contain or a set of names of other program units or a model of messages transfer for those PU. These PU (agents) have to implement a logic of solution of a task (PSM); model of messages transfer have to register concrete communications of PU-agents with each other (fig. 2). A program units of a solver have an obligatory declarative representation in the form of IR too. One of versions of the architecture is shown in fig. 2 (in graphical view).

In the course of the service development the following components have been reused:

- knowledge bases of the scheme of treatment, KB of pharmacological means,
- meta-information of input information resources (clinical records),
- specialized operations of access to clinical records and pharmacological means,
- and also standard IACPaaS API of access to concepts (knots) of networks (i.e. IR).

This solver constructed on the IACPaaS platform becomes a reusable "software module" for a number of services of therapeutic treatment for several profiles of medicine. A fast assembling of such service for any medical profile is supposed at the current stage of development of IACPaaS. The solver of the first version receives control via the standardized IACPaaS interface. The additional program interface for the solver (created as the additional production of the root agent) turns it into the component being included in other medical services by analogy with including of PU or choreography of web-services.

Conclusion and Acknowledgment

The paper contains the methodology IACPaaS for intellectual services development and their knowledge bases formation. The functionality of the IACPaaS cloud platform facilitates development of the intellectual services having knowledge bases in network representation and specialized solvers processing knowledge bases and other stored information in the form of hierarchical semantic networks. The technology of development of services supporting use of earlier made decisions and the realized components is presented. The platform has tools for support of cloud services development by means of various technologies and the tools for network knowledge bases and databases. The support for development, integration and maintenance of all components of services are necessary to receive the viable intellectual services using knowledge bases. The work is performed with partial financial support of the RFBR (projects 16-07-00340 and 15-07-03193) and FEB RUS program "Far East" (project 15-I-4-029).

References

[1] A.V. Bukhanovsky etc. “CLAVIRE: perspective technology of cloud computing of the second generation,” Journal of Instrument Engineering (Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie). 2011. No. 10, pp. 7-14. (in Russian).

[2] Le Sun, H. Dong, F.Khadeer Hussain, O. Khadeer Hussain, E. Chang “Cloud service selection: State-of-the-art and future research directions,” Journal of Network and Computer Applications, 2014, Vol. 45, pp. 134–150.

[3] Qi Zhang, Lu Cheng, R. Boutaba. "Cloud computing: state-of-the-art and research challenges,” J. Internet Serv App., 2010. №1, pp. 7–18.

[4] T. Tudorache, C. Nyulas, N.F. Noy, M.A. Musen WebProtégé: A Collaborative Ontology Editor and Knowledge Acquisition Tool for the Web,” Semantic Web Journal (SWJ), 4, Number 1, 2013, pp. 89 – 99.

[5] Zagorulko Yu.A. Semantic technology for development of intellectual systems oriented to ex-perts in subject domain,” Ontology of Designing, 2015, № 2 (15), pp. 30-46. (in Russian).

[6] Gribova V.V., Kleshchev A.S., Krylov D.A., Moskalenko F.M., Timchenko V.A., Shalfeeva E.A. Basic technology of development of intellectual services on the cloudy IACPaaS platform. Part 1. Development of the knowledge base and solver of tasks,”Program engineering. 2015. No. 12. Pp. 3-11. (in Russian).

[7] F. Puppe, Knowledge Reuse among Diagnostic Problem Solving. Methods in the Shell-Kit D3,” International Journal of Human-Computer Studies, 1998. Volume 49, Issue 4, P. 627–649

[8] A.S. Kleschev, E.A. Shalfeeva, An ontology of intellectual activity tasks,” Ontology of Designing, 2015, № 2(16), pp. 179-205. (In Russian).