Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 9 (2012) 1694 – 1703

Procedia Computer Science

International Conference on Computational Science, ICCS 2012

High-Level Knowledge-Based Structures for Simulation within Urgent Computing Tasks

Sergey V. Kovalchuk[a,]*, Alexander V. Boukhanovsky[a]

[a]*National Research University of Information Technologies, Mechanics and Optics, Birzhevaya line, 4, 199034, Saint-Petersburg, Russia*

Abstract

The paper presents conceptual and technological approach to describe high-level domain-specific structure of simulation and decision making process. The structures concerned as an extension to regular workflow conception which can be defined using knowledge base for particular problem domain and intelligent support procedures. The structures are concerned as prototype of user interface allowing constructing them in easy drag-and-drop way supported by intelligent system. Presenting approach is based on iPSE concepts and are planned to be implemented as extension for distributed computational environment CLAVIRE.

Keywords: composite application; workflow; expert knowledge processing; decision support.

1. Introduction

Contemporary urgent computing tasks [1] often require active involvement of computer simulation procedures which can be performed within Simulation-Driven Approach (SDA). In this case decision support processes should be considered as service toolkit which needs to be managed in a composition with simulation tasks. This approach usually require complex interaction of different resources: hardware – as a platform for running high-performance computations; composite software performing complex systems simulation procedures; data sources, containing static and dynamic information about the urgent situation and real-world objects involved into the situation; organization and human resources including decision makers, experts etc. All these resources should be integrated within the framework of the composite process, which is oriented to provide decision makers with the support for challenging management issues. Traditionally hardware, software and data resources are integrated within workflows [2], including calling of software services with input and output data management. But urgent computing requires concepts of workflow to be extended with providing of high-level management abilities for resources of different types including (as one of the most important parts) human and organization resource management.

Moreover in case of complex resource integration scheme the system which provide decision support should include special procedures for automatic composition and intelligent user support during composition process. This capability allows operator of the DSS easily develop and execute complex composite solutions for decision support depending on particular issues to be solved.

* Corresponding author. Tel.: +7-962-687-0460; fax: +7-812-232-9704.
E-mail address: sergey.v.kovalchuk@gmail.com.

doi:10.1016/j.procs.2012.04.187

This paper presents a conceptual and technological approach which offers integration of resources of different kind within a composite workflow-like structure. This approach, combined with user's knowledge-based support, offers an opportunity to solve complex issues of systems analysis and problem management. Considered in a framework of urgent computing this approach can be effectively applied for building decision support systems as a part of solutions for crucial problems, which originally causes a need of urgent computing.

2. Decision Workflow Concept

Decision workflow (DWF) is a high-level semantic structure, which allows to integrate different types of resources within a form of oriented graph, defining interaction between the resources. Extending basic workflow concept [2] with this concept allows high-level operation managing, representing usage of various resources, in a way which is defined by decision process. This leads to appearance of the following features within DWF usage process:

1. Explicit management of data sources and flows within the DWF allowing exploration of data using DWF as a basic concept of user interface.
2. Comparative and explorative capabilities within the user interface built upon DWF concept, which gives high-level analyzing abilities to the user.
3. Forecasting available as a running regular workflows built using constructed DWF gives user ability to investigate future states of the concerning system.
4. Decision blocks combined with forecasting capability allows construction of early warning systems, ready to run continuously with incoming data flows processing (e.g. coming from sensors or measuring devices).
5. General requirements of decision support, solving the tasks of urgent computing, and process of early warning systems' development states a set of issues related to these processes. E.g. time-crucial computations and decisions in frame of urgent computing tasks should be solved within defined period of time. This and resembling requirements should be processed on every level from highest (as meta-parameters) to lowest (as parameters of scheduling algorithm).

DWF concepts allows user to compose a high-level workflow using abstract blocks which are then mapped onto resource calling. This mapping process produces regular workflow, which defines one of possible interpretations available for current DWF.

DWF construction process can be performed with intelligent support based on domain-specific knowledge about computational simulation process, which is stored within knowledge base of the system. Knowledge-based user support during composite application development and running is one of basic principles of the iPSE concept (Intelligent Problem Solving Environment) [3, 4], which forms a conceptual basis for the set of projects performed by National Research University ITMO within last years. During these projects the infrastructure platform for building and executing of composite application for e-Science was developed. The CLAVIRE (CLoud Applications VIRtual Environment) platform [5] allows building of composite applications using set of domain specific software available within service-oriented distributed computational environment. Thus DWF concept can be concerned as a high-level extension of iPSE concept allowing constructing, running and interacting with the system within a framework of decision process.

2.1. Conceptual Structure of Used Knowledge

Considering general process of computational simulation a set of common concepts can be defined as a core of that process. Further concepts, defined here, are used to arrange domain knowledge in the framework of simulation and decision support processes. We use following conceptual hierarchy to describe simulation process:

1. Simulated *object*, which represent the main entity, which is explored during simulation. The object can be concerned as a composite entity, or system of objects. In this case simulation process might be defined for whole composition or for separate objects, with explicit definition of theirs interaction. The explored system can be concerned as composition of the objects.
2. Simulated *model*, which describe set of static and dynamic characteristics of the object and can be used to explore it. The model can be defined as *static* if it describes object's structure within a fixed moment of time. Models of

this type can be used for structural analysis of the system. In case the model describes evolution of the system within the time domain it is defined as *dynamic*.

3. *Method* can be defined as imperative description of the model usage process. Method allows calculating output model parameters having input parameter set. Methods are implemented in simulation software as algorithms for solving some particular domain *problem*. Considering this hierarchy the *problem* can be defined as a composition of object description and model.
4. *Software packages* are used as implementations of defined methods in form of as algorithm. Usually this kind of software is developed by domain specialists. Often there are huge amount of software with different variants of the same method implementation.
5. *Service* within distributed computational environment (in case we are using SOA [6] approach) can be considered as software deployed on computational resource (hardware or virtual machine). Services are low-level elements of regular workflows.

This conceptual hierarchy defines five abstract levels of problem solving. Within CLAVIRE computational environment user can define abstract workflows, using concepts of levels 3 and 4 by calling methods of registered software packages within domain-specific language EasyFlow, which was originally developed under iPSE concept. Mapping abstract workflow onto the 5th level – set of available services (including tuning implicit technical parameters) is performed by CLAVIRE implicitly (in hidden way). Furthermore applying intelligent support extension [4] user is provided with knowledge-based workflow composition guidance, which leads him/her through the levels 1 to 4 (considering levels 1 and 2 as a single concept *problem*). In this research we are trying to develop conceptual and technological implementation of interaction with the user on the domain-specific levels (1 to 3) with detailed elaboration of system's semantic structure. Within this paper DWF is presented using two knowledge-based structures:

1. ***S-DWF*** – System-based DWF. This structure is based on definition of semantic structure of simulated system as a set of interacting objects. This structure is described in Section 3.
2. ***P-DWF*** – Procedure-based DWF. This structure defines structure more resembling regular workflow. It can be considered as a procedure containing calling of different resources. This structure is described in Section 4.

These structures can be concerned separately or sequentially as P-DWF can be constructed on S-DWF basis or without it. This approach, being extended with special capabilities for decision support can be used for building decision workflows, presenting powerful tool for knowledge-based interaction with user. Finally, constructed DWF can be translated into regular abstract workflow, ready to be run using CLAVIRE computational environment. This procedure is described in Sections 3-5.

2.2. Resource Integration

Considering different kind of resources we define four basic classes of the resources available for integration within a framework of DWF:

1. *Hardware*. Computational resources belong to different organizations. Differ by architecture, access protocols, policies, management systems etc.
2. *Software.* Packages for computing-intensive simulation. Differ by language, supported platform, usage procedure, technologies, running type etc.
3. *Data sources.* Storages, archives, databases, measurement devices of different type. Differ by access protocol, available data sets, data access capability etc.
4. *Human resources.* Users with different roles within decision process: experts, decision makers, support officers, composite software engineers etc.

Hardware, software and data resources can be considered within a framework of traditional workflow structure [2], while human resources (HR) need to be concerned separately. HR can be considered as users available for the system's requests. Usually there are a lot of users involved into decision process. E.g. following list presents possible user roles often mentioned in complex software systems with roles corresponding to them in framework of DWF usage:

- Operators. This role defines the user, who tunes DWFs, runs it and investigates static and dynamic characteristics of concerning system or object.
- Composite software developers. This role defines the user, who develops DWFs and AWFs (abstract workflows) for operator's tasks solving. Often the combination of this and operator role is performed by one person.
- Domain experts. These users support the process with own knowledge. Expert's knowledge may be stored within knowledge bases or be available through the direct interaction process within consulting mode.
- Decision makers. Special kind of experts that have rights to decide which solution is appropriate in case of emergent situation arises.
- Administrators. Supports the system and infrastructure in stable state. This role supposes implicit participation within the process of execution by managing bases of resources of different kind available for the system.
- Virtual professional society. It operates as a spectator of decision support process. But society is one of the most important sources of knowledge as persons within the virtual professional society usually have great collaborative experience of running different kind of workflows. Also some of society members can be concerned as experts who have ability to make advices or collective judging (e.g. by voting).

Generally speaking HR can be concerned within regular service model with respect to specific of its request procedure. I.e. we need to take into account following features of HR while we are trying to integrate them within a single composite solution:

- We need to deal with human nature specifics: response latency according to speed of human reaction and processing time; working schedule of a person, which can be involved into the solution on permanent position or on request-response scheme; psychological and sociological factors etc.
- Specific access policies to this kind of resources also need to be considered: salary instead of computational time cost, privacy along with personal responsibility etc.
- Decision and solution specific: in most cases responses of HR should be considered as opinions estimated with special metrics (trust level, status of the expert etc.)/
- Interaction with HR should be performed using special technological and organizational solutions. E.g. for interaction with experts special GUI-software, running and configured by expert themselves, can be used as a frontend.
- HR base should be used as special integration solution for unified interaction with users of different kind, history and provenance storing.
- Typical interaction with experts should include scenario with multiple opinion collection and comparison with composition or choosing the best one.

2.3. Knowledge Base

Today semantic knowledge is often expressed using ontology approach [7] that is popular within such technologies as Semantic Web [8] or (more relevant for considering issues) Semantic Grid [9]. Concerning semantic software description mentioned above we can define ontology structure which integrates all the parts of knowledge.

Fig. 1 shows an example of ontology part describing available software using proposed structure. This part of ontology (simplified for illustration) defines five concepts: package (which represents particular software), method (implemented within the software), cluster (as a subclass of resource), value (domain-specific) and format of the value used by software. Here concepts *package* and *method* are related to high-level concepts introduced in Section 2.1. Important part of the knowledge is mentioned as attributes of individuals and relationships between the individuals. E.g. we can define (in full version of ontology it is defined by attributes of algorithm individuals) that ORCA package implements DFT method with particular quality and performance. It can be defined in different ways: as constant quality value (reliability: 0.95), as function with set of parameters (e.g. performance model), as table with profile values measured for particular parameters of application and environment.

This approach allows supporting of dynamic composition of software and hardware resources within complex application given by set of requirements. For instance it is possible to compose an application that requires shortest time to execute or gives the most precise solution using available parameter dictionary within defined time.

One of the important issues related to knowledge base is knowledge acquisition. In some cases the system is so complex that the single human or even group of experts is unable to manage such systems. Today there are studies devoted to exploring and management of soft systems but typically they concern local systems and human-centered management. On the other hand there are huge amounts of knowledge distributed across the world which can be used to solve the problem of complex systems' exploration and management. As amount of knowledge becomes more and more available via global networks this type of processor become useful tool for solving wide range of problem related to complex systems. Concerning knowledge intelligent processing several sources of knowledge should be taking into account:

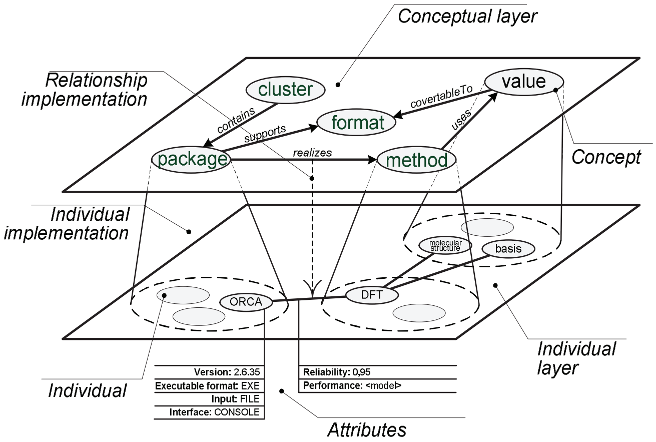

Fig. 1. Basic ontology structure for software semantic description.

- Semantic knowledge bases. This type of knowledge sources is mainly developed to be used like this. Containing description of different pieces of knowledge it can be integrated as a description of many and many objects of the world. Models of related entities and processes can be acquired from such semantic descriptions as well. Moreover the metadata, stored in such storages may contain links to other knowledge sources. Special type of knowledge bases is semantically supported data storages. If such storages have rich semantic description they can be concerned not only as sources of data, but as sources of knowledge.
- Software. Today there is a great diversity of software available for solving different tasks of science, technology or everyday life. Generally these software pieces can be also considered as representations of knowledge (acquired from domain experts and software developers). Also running the software can produce new knowledge: by storing the results annotated by software or by the user.
- Global networks. Today's Internet contains huge amount of data and information, available for processing. Using data mining and other related technologies it is possible to use this information. Furthermore intelligent processing of this information can produce knowledge about real and virtual object, implicitly or explicitly presented within the Internet.
- Social networks. Containing dramatically growing amount of information social networks presents personally related facts that can be processed for knowledge producing. Social network information is a part of global networking, but as the most important problem of contemporary world is related to or even defined by social factor personal data processing became urgent for solving such problems.
- Human resources. Human resources today are still main sources of knowledge. Available directly for asking questions or problem solving experts are required for most of today's problem. But as we can see today a lot of expert knowledge is available in formal way: stored in bases, tracked by network services. So it can be expected that in the nearest future almost every expert decision can be tracked and processed as knowledge by automatic systems. In this case direct experts' involvement for solving the problem will be required only in the most complicated cases because in typical scenarios decision will be made automatically.
- Bases of ideas and hypothesis. This type of storage may contain knowledge about hypothesis, ideas, models that yet can't be verified or checked. Being semantically described element of these bases can be used for analysis in future. Knowledge processor can use this source of knowledge for searching solution for unknown problem.

Today's information environment is rapidly growing and became more and more integrated. Concerning mentioned sources of knowledge we should take into account that there are possible relationship between almost every of them. Thus one of the key issues of building a knowledge processor is access to integrated and interrelated network of knowledge sources of different types.

3. System-Based Decision Workflow

According to basic conceptual hierarchy defined earlier the core concept of computer simulation (an urgent computing as a consuming task of the simulation process) is a system defined as a set of objects which are interacting with each other. Then there are set of models defining the objects' structure, behavior and interaction. Taking these theses as a starting point we are trying to develop a conceptual user interface (UI) which allows defining the user's view of the system by a structure containing blocks, representing these concepts (S-DWF). Fig. 2 represents sketch of the sample UI which shows such a description. The task of Saint-Petersburg flood protection with a storm surge barriers [10] is taken as an example of urgent computing task here and further.

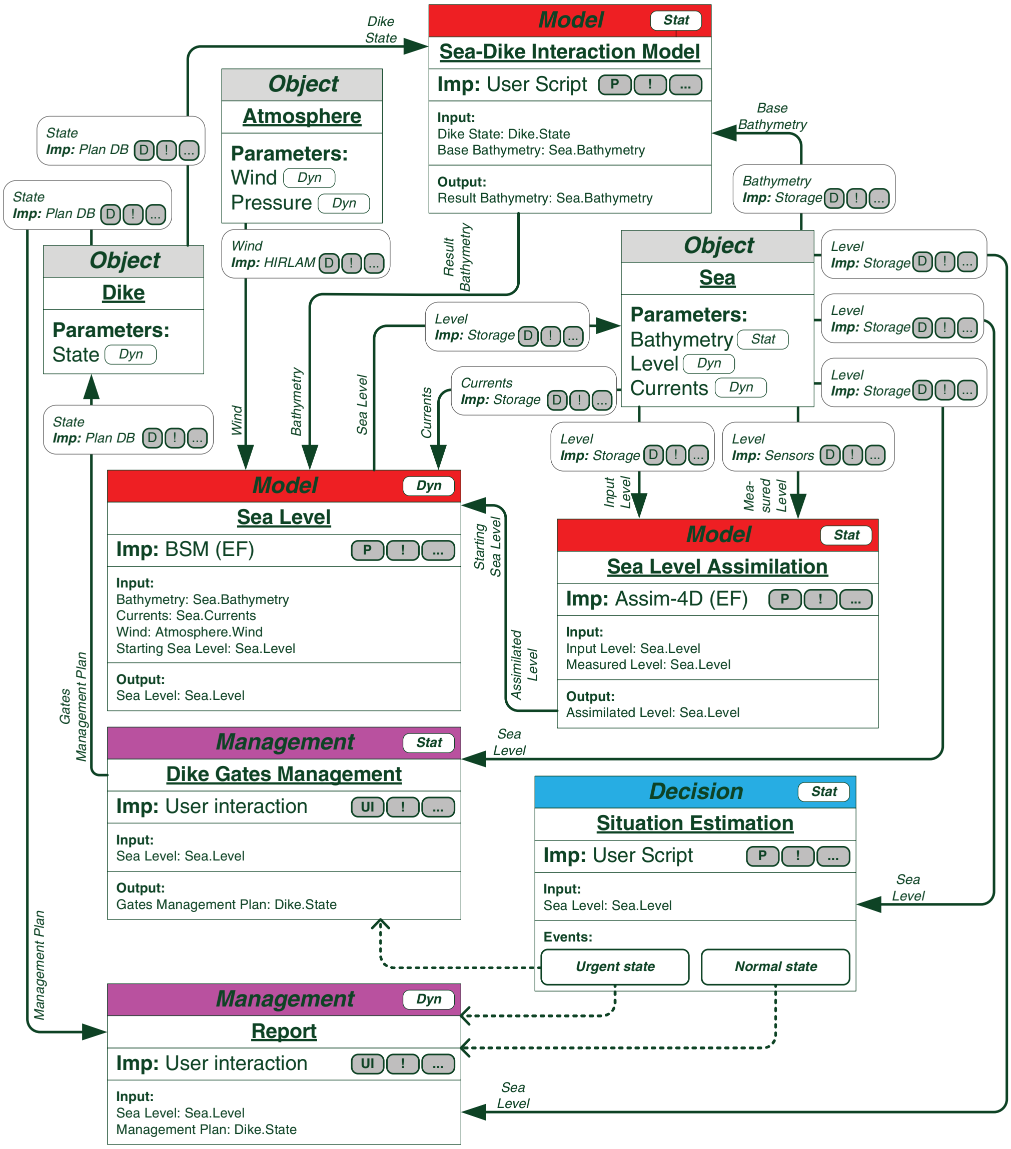

Fig. 2. System-based decision workflow (sketch of sample UI)

The core blocks on this diagram represent objects, which are considered during simulation and decision processes. These blocks are Dike, Sea and Atmosphere (as an object, which affects on the Sea object). All objects have set of values, representing main characteristics of the object. These values are marked as static or dynamic – these terms are referred to static and dynamic models (see Section 2.1).

The object blocks are connected using data flows to activity blocks. Activity blocks are marked as static or dynamic, depending on type of model, they are representing. Activity blocks can be then interpreted as calling of resources (first of all – software and human resources) within an abstract workflow or user script. Abstract workflow (marked EF – EasyFlow on the diagram) defines software calling and can be executed using general functionality of CLAVIRE. User script is simplification of procedural activity this script is defined by the user on any common language (like Ruby or Python) and also can be executed using CLAVIRE. Interpretation section (market with key-word "Imp" on the diagram) contains three buttons: first button define type of interpretation ("P" for procedure (workflow or user script), "UI" – for calling an HR, "D" – for calling data source), second button "!" calls intelligent support of the user for this block interpretation, last button "..." allow user to tune parameters of interpretation in manual mode. The most important button here is "!": intelligent support system interacts with user for additional active fact collection, then search the knowledge base for possible solutions, estimate the solutions by set of quality metrics also defined by knowledge base and at last offer to user result of comparative analysis of possible solutions. Such intelligent support can be performed by knowledge based user support within CLAVIRE environment (see Section 5). There are different types of activity blocks:

- Model blocks. Represent second concept of basic hierarchy. Input and output of these blocks represent values, which can be referred to characteristics of some objects in the S-DWF. Implementation of each block at the end of DWF interpretation process defines the remaining concepts of the basic hierarchy within regular workflow.
- Decision block is defined as activity which controls execution flow by rising events of different kind by analyzing input data. Blocks of these types incorporate models and methods to estimate incoming values and interprets outgoing values as event rising. Within presented structure rising the event causes execution of other blocks within the result workflow (shown by dotted lines with arrows).
- Management blocks represent management procedures, which define user action (real or virtual), related to considering object. E.g. changing state of storm surge barriers or proposing plan for doing this.

While data-flows between activity blocks define only passing the values from one block to another, data-flows between activity block and object block define acquisition process for the data (see blocks on data-flow arrows), which also can be interpreted. Acquisition blocks have the same set of buttons as activity blocks. This block also can be interpret as data sources, procedure (for simulation of the data) or HR interaction (for manual data input).

The process of the diagram composition starts with definition of core objects. Then the core objects are processed by intelligent support system, which (by analyzing knowledge base) defines available models and corresponding activity blocks. Then user select the activity blocks of own interest according to recommendation of the system and connect them to object blocks and between each other. After every user activity intelligent support system repeats analyzing procedure and update set of available activity blocks. After finishing construction the structure can be interpreted with support of intelligent system, which can advice the best possible solution (according to knowledge base) for every activity or acquisition block. When the interpretation process is finished the defined structure can be executed (being interpreted as composite solution allowing different resource calling). Or the structure can be mapped onto P-DWF structure, presented in the next section.

4. Procedure-Based Decision Workflow

Procedure-based decision workflow (P-DWF) can be concerned as next-step interpretation of the S-DWF presented earlier. On the other hand P-DWF can be constructed completely separately using its own block palette and user interface features. Fig. 3 shows sketch of the sample UI defining decision workflow for flood protection task. For the sake of clear explanation this example was changed against the system structure presented on the previous figure, so it can be concerned as separate representation of the same task.

DWF contains blocks, which are connected using data or control flows. Additional type of connection is metadata flow, which means that execution of the block affects another block in run-time (e.g. changing parameters or implementation of the block). Every block can be tuned by changing individual parameters presented on sliding pane, which can be opened using triangle button in bottom-right corner of the block. These parameters then will be translated into abstract workflow. Some of blocks can be interpreted differently (e.g. by selection of particular data source or working with meta-workflow) in this case block has interpretation button in top-right corner, which allows user to define interpretation or to ask intelligent support system to make the selection. Also there are special system blocks: terminators (first of all for definition of forced execution finishing) and switch blocks that allows user to change between different possible structures of P-DWF. The selection blocks can be defined manually or (in case of construction using defined system structure) automatically as different possible interpretations of the same block. The most important role switch block plays in case of different resources are available for using within the P-DWF (e.g. data can be acquired from data sources or can be simulated by software).

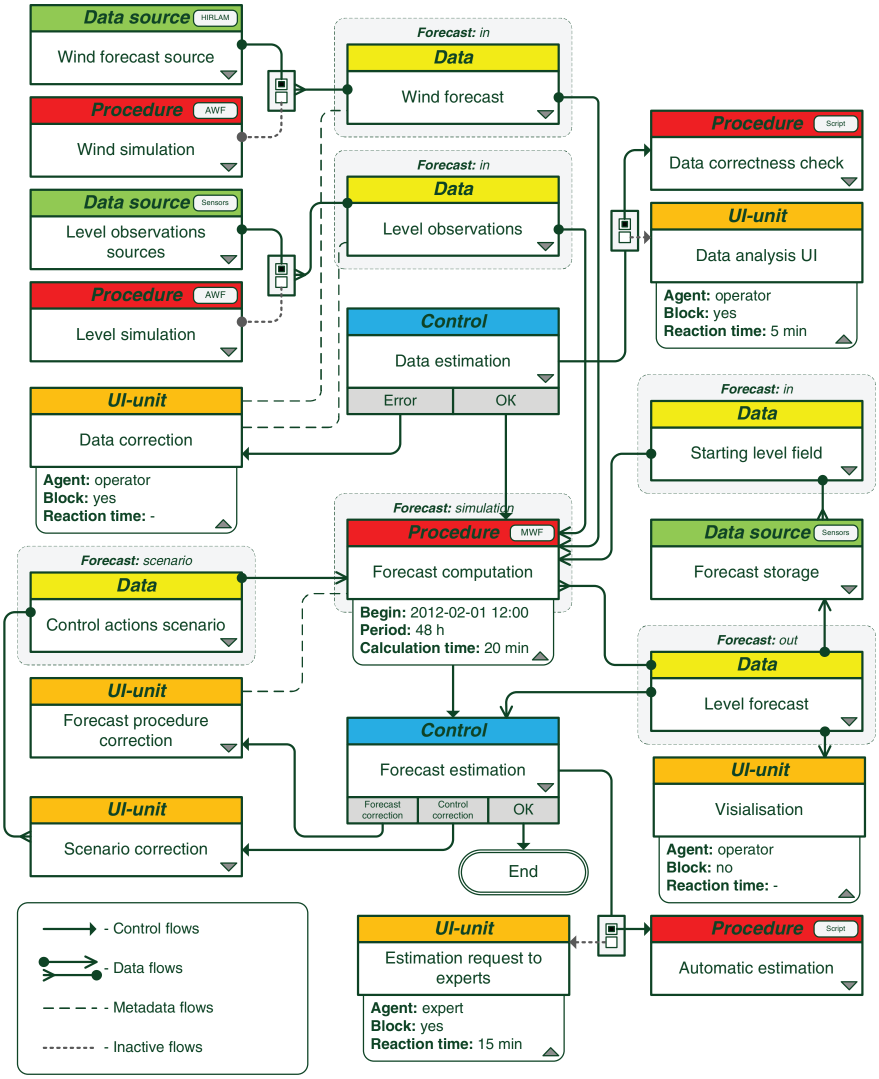

Fig. 3. Procedure-based decision workflow (sketch of sample UI)

There are following classes of blocks within P-DWF:

- Procedure blocks (red) which corresponds to models and methods of basic conceptual hierarchy. These blocks need to be interpreted as workflows (abstract or meta-workflows in terms of CLAVIRE) or user scripts (using button in top-right corner). Tuning parameters are defined by the interpretation of the block (e.g. they can be defined from abstract workflow, selected during interpretation).
- UI-units (orange) represents calling of any users (including operators themselves). Tuning parameters defines user's class and requirements along with request parameter (e.g. required reaction time).
- Control blocks (blue) which represents decision. These blocks are interpreted as procedure block or UI-unit returning values that cause control flow selection. Tuning this block include definition of possible result cases.

- Data sources (green) represent available resources of corresponding type available for data acquisition. Tuning this block include parameter setting for data acquisition procedure.
- Data blocks (yellow) define datasets, which are considered as input and output data during P-DWF interpretation. Tuning the data allows changing structure and formats which are used for data storing and presentation.

One of the most important features of the P-DWF structure is meta-procedure role definition. By meta-procedure we mean high-level method independent from problem domain. E.g. forecasting, control actions selection or system optimization can be considered as a meta-procedure. Tagging some blocks of P-DWF in terms of selected meta-procedure allows implementing active user support during execution process by presenting to the end user exactly the results he/she is looking for. E.g. on the Fig. 3 some of blocks are tagged within meta-procedure "forecasting". This procedure include following roles: simulation – making the forecast using available data; input and output data sets – are used during the simulation; scenario – arranged set of control actions applied to the system during simulation.

Another important feature is comparative analysis of possible interpretation of DWF (available both for S-DWF and P-DWF), considering basic DWF as invariant structure. Using intelligent support system the user can in manual or automatic way compare different variants of DWF interpretation and select the best of them. Usually comparison is done using various interpretation of the block, but also different variants of switch blocks can be concerned as variants of implementation. There are two types of comparative analysis available using DWF structure:

1. Static comparative analysis is performed using knowledge base only. Quality estimation is assigned to every interpretation variant and can be compared during design-time.
2. Dynamic comparative analysis is performed by running different interpretation of the DWF on the same data and comparing quality of gained results (e.g. within P-DWF they can be defined using meta-procedure roles). Quality of result data can be compared by using knowledge base or calling experts (as HRs) with request of quality estimation for simulation results.

At last P-DWF as well as S-DWF can be transformed into abstract workflow (after interpretation of its blocks) and executed using CLAVIRE environment as regular workflow.

5. Knowledge-Based Support of Workflow Composition

Knowledge-based intelligent support for quality comparison is already implemented within CLAVIRE environment [4]. This feature allows defining possible solution as abstract workflow by interacting with user via special tree-based user interface implemented as web-application using .NET Silverlight technology. This interface guides user through domain-specific level of conceptual hierarchy presented in Section 2.1. Interaction with user is performed using terms from selected problem domain and requires from user no technological background or experience with software execution. At the end of this interaction script available for running using CLAVIRE is presented to the user. Going through developed tree-based interface user can see different quality estimations, which were computed using criteria presented in knowledge base. This approach allows CLAVIRE to lead the user to abstract workflow without requiring any knowledge of solution's implementation. Within DWF concept this extension of CLAVIRE capabilities can be used for automatic or semiautomatic inference of optimal interpretation of DWF of defined system structure. In first case intelligent support system working with knowledge base select optimal interpretation using predefined criteria by analyzing semantic structure. In second case the system interacts with the user by mentioned tree interface allowing entering extra parameters for finer tune of DWF interpretation. This approach allows providing user with intelligent support while interpreting blocks of defined semantic structures (DWF and system description) and estimation quality of the whole structures.

6. Discussion and Conclusion

Usually simulation process is described by workflow which represents sequence of service (abstract or concrete) calling. There are a lot of frameworks [11, 12] which use textual or graphical notations of workflows describing simulation by computations within high-performance distributed environment. These frameworks present reach technical tools for calling existing simulation methods available via remote services. But usually they deal with lack of domain-specificity and require a lot of knowledge about calling services as they are built around these services. On the other hand there is a set of standards describing high-level simulation process [13, 14] and works devoted to development of more or less detailed conceptual and technological frameworks for high-level simulation process support [15, 16]. But these works are mostly conceptual and has a weak relation to technological question of distributed resources integration. In contrast presented work is focused on interconnection of high-level simulation concepts and resource integration within a workflow as a concept for in-silico experiments. Generally this work has two main goals: a) to develop a conceptual and technological framework (providing the user with graphical toolset for simulation process description) to fill the gap between high-level simulation theory and technological platforms; b) extend this framework to make it more usable in case of urgent computing and decision support by adding special concepts and interactions.

This paper presents high-level semantic structures allowing describing simulation process in respect to decision support and urgent computing processes. The structures are assumed as basis for knowledge-based intelligent support for the users which allow them to define simulation and decision processes using domain-specific terms. The structures are developed as prototypes of user interfaces which allows user to edit the structures in drag-n-drop way. Developing solution is concerned as an extension to existing iPSE concept [4] and is planned to be implemented as a separate UI module within the CLAVIRE platform [6].

Acknowledgements

This work was supported by project “Multi-Disciplinary Technological Platform for Distributed Cloud Computing Environment Building and Management CLAVIRE” performed under Decree 218 of Government of the Russian Federation and project “Urgent Distributed Computing for Time-Critical Emergency Decision Support” performed under Decree 220 of Government of the Russian Federation.

References

1. Urgent Computing: Exploring Supercomputing’s New Role, CTWatch Quarterly, Vol. 4, №1, (2008). 60.

2. Y. Gil [et al.], Examining the Challenges of Scientific Workflows, IEEE Computer, Vol. 40, Issue 12. (2007) 24-32.

3. A.V. Boukhanovsky, S.V. Kovalchuk, S.V. Maryin, Intelligent Software Platform for Complex System Computer Simulation: Conception, Architecture and Implementation. Izvestiya VUZov. Priborostroenie. No. 10. (2009) 5-24 (in Russian).

4. S. Kovalchuk, A. Larchenko, A. Boukhanovsky, Knowledge-Based Resource Management for Distributed Problem Solving, Proceedings of the Sixth International Conference on Intelligent Systems and Knowledge Engineering, Shanghai, China. (2011) 121-128.

5. A.V. Boukhanovsky [et al.], CLAVIRE: Perspective Technology for Second Generation Cloud Computing, Izvestiya VUZov. Priborostroenie. No. 10. (2011) 7-14 (in Russian).

6. B. Lublinsky, Defining SOA as an architectural style. (2007) [http://www.ibm.com/developerworks/architecture/library/ar–soastyle/]

7. B. Chandrasekaran, J.R. Josephson, V.R. Benjamins, What Are Ontologies, and Why Do We Need Them?, IEEE Intelligent Systems. Vol. 14, No. 1. (1999) 20-26.

8. T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American Magazine. (2008).

9. N.A.L. Khac, L.M. Aouad, M-T. Kechadi, Toward Distributed Knowledge Discovery on Grid Systems, Emergent Web Intelligence: Advanced Semantic Technologies. st. Edition., XVI. (2010) 213-244.

10. A.V. Boukhanovsky [et al.], High-Performance Technologies of Urgent Computing for Flood Hazard Prevention, storm surge barriers, Izvestiya VUZov. Priborostroenie. No. 10. (2011) 14-20 (in Russian).

11. E. Deelman [et al.], Pegasus: Mapping scientific workflows onto the grid, European Across Grids Conference. (2004). 11-20.

12. T. Oinn [et al.], Taverna: a tool for the composition and enactment of bioinformatics workflows, Bioinformatics, vol. 20, iss. 17. (2004). 3045-3054.

13. Standard for Modeling and Simulation High Level Architecture – Framework and Rules, IEEE 1516–2010. (2010).

14. Base Object Model (BOM) Template Specification, Simulation Interoperability Standards Organization, SISO-STD-003-2006. (2006).

15. A. Verbraeck, E.C. Valentin,. Design Guidelines for Simulation Building Blocks, Proceedings of the 2008 Winter Simulation Conference. (2008) 923-932.

16. Q. He, M.-X. Zhang, J.-X. Gong, An Introduction of BOM Modeling Framework, International Journal of Machine Learning and Computing, Vol. 1, No. 4. (2011) 354-358.