Decision-Making Methods and Models for Intelligent Business Analytics Systems in Online Tourism

Victoria Vysotska[1], Yevhen Burov[1], Volodymyr Hnatushenko[2], Lyubomyr Chyrun[3], Sofia Chyrun[1], Liliia Diakoniuk[3], Liubov Kolyasa[1], Oksana Huhul[4] and Oleh Naum[5]

[1] Lviv Polytechnic National University, Stepan Bandera Street, 12, Lviv, 79013, Ukraine
[2] Dnipro University of Technology, av. Dmytra Yavornytskoho, 19, Dnipro, 49005, Ukraine
[3] Ivan Franko National University of Lviv, University Street, 1, Lviv, 79000, Ukraine
[4] West Ukrainian National University, Lvivska Street, 11, Ternopil, 46004, Ukraine
[5] Ivan Franko Drohobych State Pedagogical University, I. Franko Street, 24, Drohobych, 82100, Ukraine

Abstract

The article offers a formal specification of a situational model for intelligent systems of business analytics in the online tourism field. The presentation of the situational model in the XML language is also described. Business analytics models in the intelligent decision support system have been developed to support online tourism. A general scheme for using the resource model has been developed. The architecture for the simulation environment and functional specifications of its services are described.

Keywords

Business analytics, model, online tourism, intelligent decision support system, system architecture

1. Introductions

Intelligent systems for online tourism are a new class of business systems that serve a large number of loosely connected organizations that provide services to tourists, such as airlines, hotels, transport organizations, restaurants, etc. Clients of tourist systems are characterized by high mobility both in the sense of movement in space and in the sense of frequent changes in types of activity. The works [1-4] noted such properties of the tourism industry as globality, dynamism and interdependence of services, heterogeneity of data formats, incompleteness of information, and proposed a method of using executable conceptual models that receive information from the context for making decisions on tourist service in conditions of incomplete information. An important task of tourist IS today is also the constant support of the client during his journey, the provision of services in the context of the specific situation in which the tourist is. Solving this task requires the creation of a mobile intelligent information environment that monitors and identifies the current situation and makes decisions about the relevance, content and scope of the offered services. At the same time, all the information necessary for IS decision-making is obtained from the contexts of the identified situation, the tourist, and his journey [5].

The construction of such an environment requires solving several problems and tasks, in particular, the organization of semantically interpreted access to information, the development of methods and technologies for processing the context, methods for identifying the current situation, etc. These problems are studied by several related scientific fields, such as the semantic

COLINS-2024: 8th International Conference on Computational Linguistics and Intelligent Systems, April 12–13, 2024, Lviv, Ukraine

victoria.a.vysotska@lpnu.ua (V. Vysotska); yevhen.v.burov@lpnu.ua (Y. Burov); hnatushenko.V.V@nmu.one (V. Hnatushenko); Lyubomyr.Chyrun@lnu.edu.ua (L. Chyrun); sofiia.chyrun.sa.2022@lpnu.ua (S. Chyrun); liliia.diakoniuk@lnu.edu.ua (L. Diakoniuk); kolyasa.lubov@gmail.com (L. Kolyasa); OksanaHuhul@gmail.com (O. Huhul); oleh.naum@gmail.com (O. Naum)

0000-0001-6417-3689 (V. Vysotska); 0000-0001-8653-1520 (Y. Burov); 0000-0003-3140-3788 (V. Hnatushenko); 0000-0002-9448-1751 (L. Chyrun); 0000-0002-2829-0164 (S. Chyrun); 0000-0001-5663-0501 (L. Diakoniuk); 0000-0002-9690-8042 (L. Kolyasa); 0000-0003-3052-5759 (O. Huhul); 0000-0001-8700-6998 (O. Naum)

© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

CEUR-WS.org/Vol-3688/paper24.pdf

web, contextual technologies, technologies of distributed intelligent environments, and decision support technologies. In particular, a popular direction of research in the e-tourism field, aimed at solving the problems of dynamism and incompleteness of data, is the application of methods and technologies of the Semantic Web [6, 7]. At the same time, it is often believed that existing sites have enough information to provide semantically oriented services, but only semantic markup is lacking. Works [5, 9, 8] show that in reality, such basic information is often lacking, which is a significant obstacle to the Semantic Web services introduction in the tourism field. In work [10], the application of semantic Web methods for building dynamic packages of tourist services is considered. Such packages combine different types of services from different companies. An ontology of the e-tourism field has been developed, which allows the user to obtain additional information about the place of his trip. To solve the problem of obtaining data from disparate sources, it is proposed to use semantic mediators (mediators) that form a hierarchical structure that corresponds to the structure of concepts in the ontology. A dynamic service package is generated by a web process. At the same time, IS [10] is not flexible enough, does not support dynamic rescheduling of the service package, and works only in manual mode. One of the ways to solve the problem of incomplete data about the client is to obtain additional information from the context, the development of contextual services for the provision of services in the field of tourism. The issue of context-dependent services is currently being actively researched. For example, in [11], an overview of the basic technologies and the architecture of context services developed within the framework of the CONTEXT research project funded by the European Union is proposed. This work demonstrates the possibility of building context services for creating mobile work environments, and fault-tolerant communication services. At the same time, the approach proposed in [11] does not use intellectual, semantically oriented systems, the content of the concept of context is determined in advance for each application. The scientific direction of artificial intelligence, related to the construction of distributed intelligent systems and environments (pervasive, ubiquitous computing, ambient intelligence) solves similar tasks [12]. Today, within this direction, the construction of mobile computing infrastructure is taking place, which is expressed in the development and distribution of mobile computers and services. Some popular services, including Google Places [13], provide context-sensitive services in the context of the user's current location, reporting important and interesting objects in the environment. At the same time, existing services mainly focus on using only the context of the accommodation or the context of the person (personalization services), and do not take into account what the tourist does - and, more generally, the context of the situation in which he is. The task of assessing a problematic situation is one of the central tasks of the theory of decision support systems (situation awareness – SA). A correct assessment of the current situation is necessary for making a correct decision. Solving this problem by an expert (decision-maker), as a rule, is carried out in the presence of incomplete, poorly structured, often contradictory information about the subject area, constant changes in the state of this area, large volumes of data irrelevant to decision-making, as well as strict restrictions on the time of decision-making [5, 14]. The use of decision-making support information systems aims to help the decision-maker in the correct assessment of the problem situation by quickly selecting the information necessary for decision-making and presenting it in a form convenient for the expert to perceive. One of the most difficult components of this task is determining which data in the problem area are relevant to the set goal or identified problem. Traditionally, this problem is solved by the expert himself, guided by his experience in solving similar problems. At the same time, human determination of data relevance is time-consuming and expensive. Today, in the theory of decision support systems (DSS), the direction of cognitive CDSS (Cognition decision support systems) [15] has been determined, which aims to formalize the expert's experience in the form of significant configurations of domain parameters that reflect previously identified and solved problems. Experience formalized in this way in the form of relevant models is used by the decision support system to search for relevant information in similar situations in the future. In IS of active conceptual modelling, formal conceptual models are constantly updated with SA data, providing the expert with relevant and relevant data on a defined range of problems in real-time [1-16]. At the same time, existing decision support systems are usually aimed at DS by an expert, and not at automatic identification of the existing situation.

This article purpose is to develop methods and models for identifying the situation and working out its context for the SA provision of tourism services.

2. Related works

2.1. The architecture of processing situational models in an intelligent service

The tracking changes task in the context of a tourist trip requires taking into account the relevance of changes occurring and other information in the context of this trip [5]. Determining the relevance of this or that information is a difficult expert task, the solution of which should be entrusted to an expert. Thus, the expert predicts and foresees the possible situations in which the tourist may find himself, determines the rules for their identification and the actions that must be performed when this or that situation occurs. The situations defined in this way are formalized in the situational conceptual models form. The intelligent system periodically checks the facts of the information base for the presence of this or that situation and automatically initiates the actions specified by the corresponding model if the situation is detected. Thus, the situational conceptual model consists of 2 parts. The first specifies the conditions that the facts of SA must meet to identify the situation. These conditions are specified as requirements for the presence or absence of facts of specified types with specified properties. The second part of the situational model is the action specification that must be performed when the specified situation is detected. Actions are defined as command sets with parameters determined based on the properties of the facts of the information base and the context of the situation. An important feature of situational models, the solution of which is reflected in the metadata of the model, is the determination of the conditions when and how often it is necessary to analyze the state of the software, that is, to activate the model. At the same time, it is necessary to take into account both the features of the situation detected by the model, in particular the expected probability of its occurrence, and the loading of the modelling system. So, if you activate the model too often, the modelling IC will be overloaded and may not have enough resources to execute other models. On the other hand, if you rarely activate the model, you can miss an important situation and react to it too late. If the IS envisages the simultaneous processing of many situational models, then it becomes appropriate to implement a separate component - the model launch manager. This component reads from the metadata of the models their launch conditions, summarizes them and initiates the launch of all models according to the defined conditions. The model launch manager monitors the degree of use of IS simulation resources and optimizes this use by unifying the launch of models that they have the same launch conditions. The launch manager determines with which data the models work and, if necessary, formulates tasks for monitoring services to replenish the DB with such data. The functions of the model interpreter can also be delegated to the model launch manager. In the previous sections, the architecture and principles of operation of an intelligent system that uses executable conceptual models are given. Support for situational models adds new components to the proposed IS modelling (Fig. 1). The central component of IS modelling is the subject area ontology, which provides a semantic interpretation of all the facts of the information base. Models are also built based on this ontology [5]. The information base is constantly updated with new facts that reflect the state of the subject area. The sources of such facts are employees, IS of travel companies and companies that are providers of travel services, as well as specialized context providers - for example, those that provide the location of a tourist in real-time, or compare this location with a certain organization or institution, for example, a restaurant, or a museum. A separate place among the sources of replenishment of the information base is occupied by monitoring services, which periodically read the specified data in the subject area and enter them into the database. The work [4] formulated the principles of processing contextual information for a tourist portal and described a system that works within the context of a tourist trip. In the case of a designed tourist service, situational models process information that comes from the general context of a person-tourist, or more precisely, from three parts of this context: the tourist's data, his location, and the context of the task that the tourist is currently performing [5].

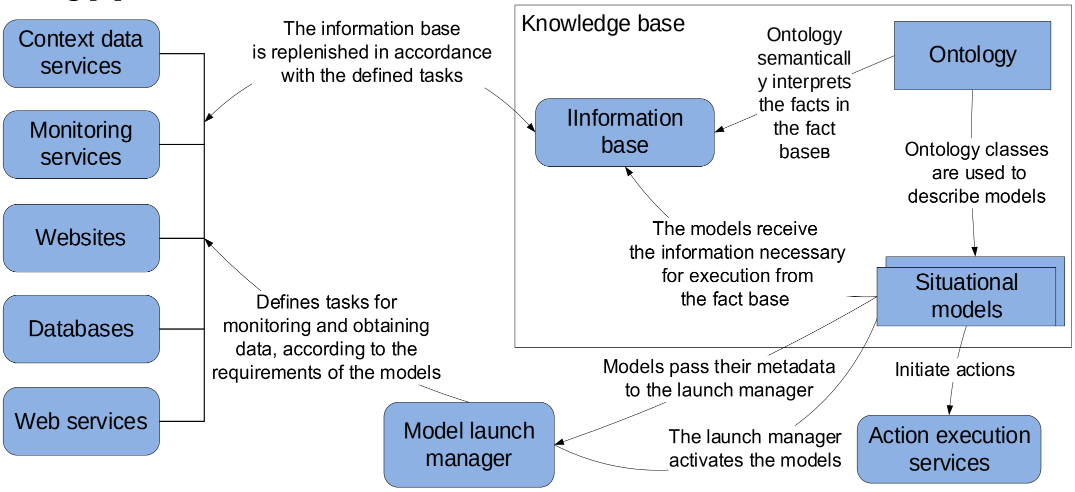

Figure 1: Components of the modelling system [5]

2.2. Formal specification of the situational model

The scheme of the situational model consists of the *BodSit* model body and *MtdSit* metadata [5]: $ScMdSit = (BodSit, MtdSit)$. The body of the model defines the logic of the model and its actions: $BodSit = (SgSit, AcSit)$ and consists of the signature specifications of the *SgSit* model and the *AcSit* actions. The signature is a predicate given on Fc_i facts from the *InBd* information base.

$$SgSit = P(M(Fc_i))|Fc_i \in InBd. \quad (1)$$

The information base is a time base and contains facts that took place in the past. We present the signature as a disjunctive normal form of individual Asr_i^j predicate statements given on the facts of the base [5].

$$SgSit = (Asr_1^1 \wedge Asr_2^1 \wedge \ldots \wedge Asr_{m1}^1) \vee \ldots \vee (Asr_1^n \wedge Asr_2^n \wedge \ldots \wedge Asr_{mn}^n). \quad (2)$$

An *AcSit* action specification is a set of actions defined in a type ontology.

$$AcSit = M(Ac_i)|Type(Ac_i) \in On. \quad (3)$$

Such actions can be, for example, loading a Web page or accessing a Web service. The most general type of action is the execution of the algorithmic model *MdAlg*, specified on the facts of the information base [5]:

$$AcSit = MdAlg(M(Fc_i))|Fc_i \in InBd. \quad (4)$$

MtdSit model metadata includes several sections, but the most important to the operation of the situational model is the *ActCd* activation condition section, which is passed to the model launch manager for launch scheduling. An activation condition is a disjunction of conjunctions of atomic predicates that are true if a specified period has passed or a specified event has occurred.

$$ActCd = (Cd_1^1 \wedge Cd_2^1 \wedge \ldots \wedge Cd_{k1}^1) \vee (Cd_1^2 \wedge \ldots \wedge Cd_{k2}^2) \vee \ldots \vee (Cd_1^n \wedge Cd_2^n \wedge \ldots \wedge Cd_{kn}^n), \quad (5)$$

where atomic predicates Cd_i^j can be [5]:

- predicate *TimeIntCd*(*TimeInt*) – is true if a period of *TimeInt* has passed since the last activation;
- $P(M(Fc_i))|Fc_i \in InBd$ – is the arbitrary predicate on the facts of the base;
- a reference to the model that is activated and returns a Boolean value. Thus, for example, you can set time-fixed moments of activation, or activation with a defined repetition mode;
- predicates tied to specific types of fact-based events, including the addition of new facts or modification of fact types.

Let's consider in more detail the processing of contextual data by the tourist service, taking into account the situation. As an example, a tourist service has been developed that provides the necessary auxiliary information for the tourist depending on the context of the situation and the tourist's geographical location. The ontology developed in the DERI project (http://e-tourism.deri.at/ont/e-tourism.owl) was adopted as the basis of the ontology for the service and modified by adding new entities and connections. A fragment of this ontology, which displays contextual data used to solve problems, is shown in Fig. 2. The same figure shows other structural components of the intelligent service.

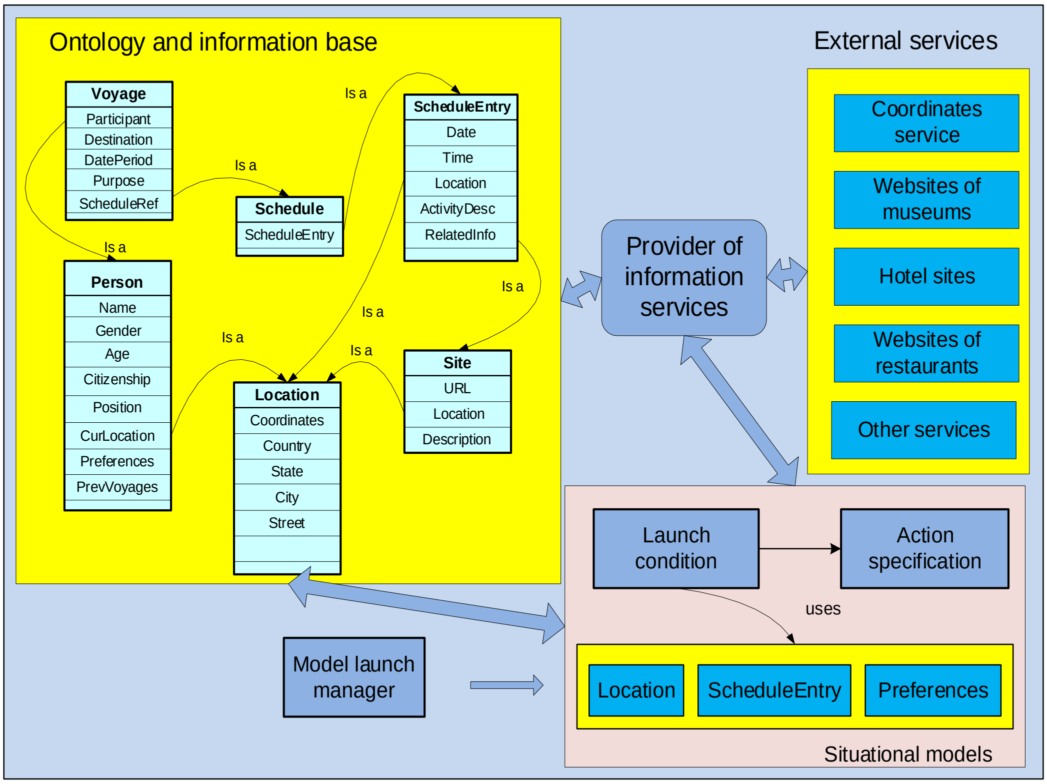

Figure 2: Scheme of obtaining contextual data by models [5]

Voyage and Person are the main components of the ontology whose context is used in the system. Information about tour participants, tour schedule, places of interest planned to be visited during the tour, and addresses of providers of information about them are obtained from the context of the tour. From the context of the person, the preferences of the person are derived, which are used in situational models to select information. In addition, the current location of the person is also obtained from this context, which is also used in the models [5].

The model launch manager periodically activates the models according to the specified activation modes. Activated models check for the conditions specified in the models. Such conditions, for example, include verification of the correspondence of the current time and the time of the planned event in the event, verification of the coincidence of the current location of the tourist with the given area of the location of the object visited on the tour, etc. If necessary, situational models receive additional information from the context of the tourist or tour person and use it to make a decision [5]. If the situational model has detected the presence of the specified trigger conditions, then the algorithmic model is activated, which specifies the sequence of actions to be performed. In the case of the tourist service, the class of actions was considered, which consisted of providing the tourist with additional information, and the task of information filling for the tourist's current information page. In the process of its work, the intelligent service uses the services of external services that provide information about the location of the tourist, information from the sites of the objects visited by the tourist, etc. [5].

2.3. Presentation of the situational model in the XML language

Situational models are created in the "Model Editor" tool and saved in XML format. Let's consider an example of presenting a situational model for a planned tourist service. Let's consider a simple model, which detects the situation based on the coincidence of two conditions - the presence of a tourist near the object planned to be visited and the presence of a visit to this object in the tour plan at present [5].

```
<Model>
 <ModelMetadata>
  <GeneralInfo>
   <ModelId> id </ModelId>
   <ModelType> SituationalModel </ModelType>
   <OntologyURI> www.acme.org/TourismOntology</OntologyURI>
<InfoBaseURI> www.acme.org/InformationBase </InfoBaseURI>
 <ModelRepositoryURI>www.acme.org/ModelRepository</ModelRepositoryURI>
  </GeneralInfo>
  <ActivationInfo>
   <Condition>  <ConditionId> cd1</ConditionId>
   <ConditionBd> CurrentDate()in InBase(Voyage.DatePeriod) <ConditionBd>
   </Condition>
   <Condition><ConditionId> cd2</ConditionId>
    <ConditionBd> Every(5 min) <ConditionBd>
   </Condition>
   <Activate> (cd1) and (cd2)</Activate>
  </ActivationInfo>
 </ModelMetadata>
 <Signature>
  <Condition>
   <ConditionId> sigcd1</ConditionId>
   <ConditionBd> InBase(Some(Site).Location) nearDistance InBase(Person.CurLocation) <ConditionBd>
   <Result> InBase(Site)</Result>
  </Condition>
  <Condition>
   <ConditionId> sigcd2</ConditionId>
   <ConditionBd> CurrentTime() nearTime InBase(Some(Voyage.Schedule.ScheduleEntry.Time))<ConditionBd>
   <Result> InBase(Voyage.Schedule.ScheduleEntry)</Result>
  </Condition>
  <Condition>
   <ConditionId> sigcd3</ConditionId>
   <ConditionBd> Result(sigcd1, InBase(Location)) nearDistance Result(sicd2, InBase(Location))
   <Result> </Result>
  </Condition>
  <Execute> (sigcd1) and (sigcd2) and (sigcd3)</Execute>
 <Signature>
 <ActionSpecification>
  <ActionType>LoadContentfromURL</ActionType>
  <URL>Result(sigcd1,URL)</URL>
 </ActionSpecification>
</Model>
```

The model description contains metadata sections and the model body. In the metadata section, for simplicity, only two subsections are given - general data and activation mode. The general data section lists the model identifier, its type, and links to the ontology, model repository, and fact base used. In the subsection of the activation mode, two activation conditions are indicated - the model is activated only during the duration of the tour and every five minutes. The body of the model has subsections of the firing signature and the action specification. Three conditions are specified in the signature subsection. The first condition determines the fact that the tourist is near one of the objects that are planned to be visited. The condition uses the *NearDistance* predicate, which evaluates to true if the two placements match within a specified margin of error. If the condition is fulfilled for some object, then its placement is remembered as a result of the condition check. The second condition checks whether a visit to a certain object is planned for the current time. In this case, the *NearTime* predicate is used, which takes the value true if the two times coincide within the specified error. If such a visit is scheduled, its ID is also stored as a result of the condition check. Finally, the third condition combines the results obtained in the two previous conditions and checks whether a visit to the object near which the tourist is located is planned [5]. In the action specification section, the type of action is indicated - receiving information content from a certain URL, and a link to this URL. The proposed approach to modelling and identification of the current situation is illustrated in the example of the field of electronic tourism, it is advisable to expand and apply it in other fields as well. The implementation of such an approach will allow the creation of IS and services that quickly respond to changes in the environment [5].

3. Materials and methods

3.1. Semantic structure development for business analytics of the subject area

The task of building an ontology for solving BA problems is one of the mandatory research tasks, because only with the use of such an ontology can you build models for solving problems and performing BA operations. The form of presentation, essence and relation of this or that ontology depends on the SA and the tasks of a specific study. Thus, in [17], an ontology for building business models was developed, which covers product characteristics, interactions with customers, infrastructure, and financial aspects of the business model. This ontology mainly takes into account the economic aspects of business activity. In this work, the main attention is focused on the part of the general business model, which is performed to a large extent with the use of computer technology and is promising for the introduction of automation - the level of process implementation (process, implementation layer [1, 17]). The task is to build the upper level of the BA ontology, which would allow the addition of new objects for different SAs, into which the work results are implemented. The ontology is based on the existing BPMN business process modelling language supplemented and modified by several new objects. The main entities and relations of the developed BA ontology are presented in Table 1-Table 2.

Table 1
The essence of the ontology of business analytics

N	The name of the entity	Characteristic
		Individuals and organizations
1	Person	Individual entity
2	Organization	Legal entity
3	Actor	A role in the model that is associated with individuals and legal entities
4	Position	Defines the rights and duties associated with the position
		Resources
5	Resource	An item required to execute a business process
6	Resource with controlled access	A resource to which access is restricted
7	Material resource	
8	Financial resource	
9	Human resource	
		Process and operations
10	Business process	The process resulting in the creation of an artefact important to the client
11	Business operation	A time-limited part of the business process, with a planned result
12	Competence	A complex of rights, unlimited in time
13	Assessment of the situation	Part of the decision-making process
14	Information search	Search for information necessary for decision-making
		Events
15	Business event	A generic, unspecified business event
16	Meeting	
		Artifacts

N	The name of the entity	Characteristic
17	Artifact (Generic)	A generic, unspecified artefact
18	Document	General, unspecified document
19	Business Solutions	Make a decision
20	Agreement	Agreement as a type of document
21	E-mail Message	
22	Software product	
		Teams and services
23	Fulfillment service	A service that accepts commands generated by the simulation system
24	Team	A separate team
25	Program script	A script as a sequence of commands
26	Model	
		Rules and Restrictions
27	Business rule	A business rule set at the firm level
28	Corporate limitation	The limit is set at the organization level
29	Legal restriction	Mandatory restrictions formulated in-laws
30	Useful practice	No Mandatory restriction or recommendation
Auxiliary		
31	Abstract	Additional, meaningful details

Table 2
Ontology relations of business analytics

N	Relationship name	Characteristic
1	To be part	An entity is part of another entity
2	To be a subclass	An entity is a variety, a subclass of another entity
3	Follow up	Determines the sequence of execution of operations and processes
4	Associated	General association
5	Use	Defines resources used in business processes
6	Access	Used in access control models
7	Have	Ownership relationship between the Person (Organization) and the resource
8	Have the right	Defines the rights to perform a defined range of operations
9	Initiate execution	Defines the launch for the execution of specified commands, or access to the service
10	Decide	Defines a decision-making operation

3.2. Development of business analytics models

3.2.1. Decision support models for business processes

One of the directions of building new-generation business analytics systems is the integration of the model-oriented approach of MDA and methods of knowledge engineering and IS [1-2]. The purpose of this section is to validate the proposed approaches in practice by designing data structures and layout of a real ISBA and decision support. The contractual process for concluding a software development agreement was chosen as the SA. Taking into account that the negotiation process typically requires an intensive exchange of e-mail messages, the final result of the work is a mock-up of a software complex that processes e-mail messages and tracks the negotiation process and related artefacts (agreement text, and other documents). The process of agreeing occurs as a result of the contractual process, which takes place through the exchange of e-mail messages between the Customer and the Contractor. During this process, documents related to the agreement (text of the agreement, technical proposal) are changed and supplemented. The process is considered successful and completed when the text of the agreement and technical proposal is acceptable to both process participants and is ready for signing. The process is considered unsuccessful if it is interrupted at the initiative of one (or both) participants before its completion (Fig. 3) [2].

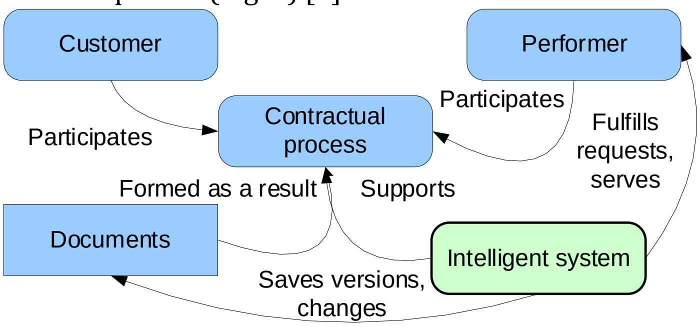

Figure 3: Participants and components of the contract process support system

Presentation of knowledge in the system. Knowledge in IS is stored in the form of ontology, models and rules [1-2]. Models are used for analysis and forecasting, storage of information about process events, ontologies - for providing model elements with content and conducting logical inference, rules formulate decision-making restrictions and provide corporate and regulatory rules and restrictions. Data and knowledge in the system are organized hierarchically, on three levels (Fig. 4). At the top level are common models, ontologies and rules. They do not belong to the selected SA and can be used by different systems. An example of such knowledge is time models and ontologies, based on which an understanding of the time dependence of terms is formed, as well as general knowledge about documents, physical and legal entities, etc. At the same level, the model of the process of exchanging mail messages and the model of the message itself are stored. At the middle level, there is knowledge related to the selected SA - the software development project and the contractual process in particular. The ontologies and models available at this level describe the stages of the project and dependencies between them, requirements for the content of the agreement, types of agreements and restrictions related to them, and other documents that are formed during the contractual process. Corporate rules and restrictions are kept at the same level.

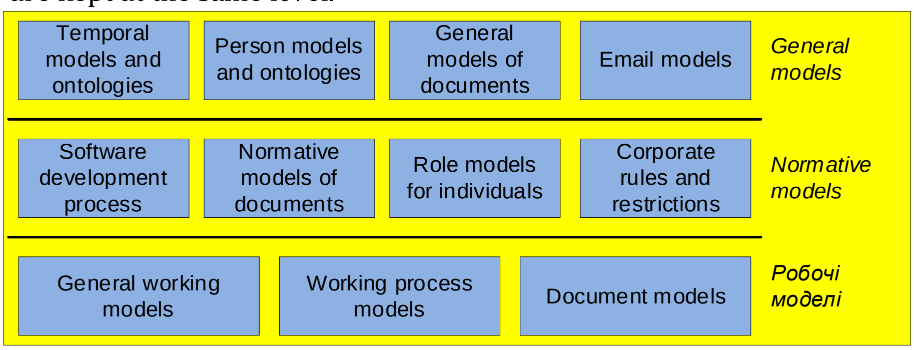

Figure 4: Models in the intelligent decision support system

At the lower level, some models are formed during the contractual process. They structurally reflect real events, and changes in documents concerning both time and defined semantic elements of the agreement, process participants and other constituent parts of the model.

Top-level models (generic models). Temporal models and ontologies are used to correctly interpret the following terms: "before", "after", "past", "future", "now", "hour", "day", "week", "month", "year", "period", "moment", "today", "tomorrow", "yesterday". It also stores models of the working day and working week with restrictions arising from labour legislation, the model of the work schedule. The block of time models also contains calendar information with the indication of holidays and weekends for all participants in the contractual process. Temporal models are used to interpret the elements of user requests, in the formulation of constraints and rules. Person models and ontologies store knowledge and define semantic dependencies between such concepts as "person", "employee", "role (position)", "qualification", "natural person", "legal entity" (organization). They are used to interpret in SA models such concepts as "Customer" and "Executor", definition of persons - participants in the contractual process, human resources necessary for the execution of the contract [1-2].

General models of documents define the generalized concept of a document, forms of its submission, software associated with certain forms of document submission, document properties, restrictions and rules. The type of document is determined in the system according to its content, for example, "agreement", "technical proposal". Document submission form – word processor file, video or audio recording, mail message. The restriction that the agreement and the technical proposal must be submitted as a text document is accepted. Supporting and working documents may exist in other forms. E-mail models contain knowledge about the structure of an e-mail message (sender, recipient, subject, message text, attached documents, time of sending or receiving), and about processes related to correspondence (reply to a letter, sequence of letters about united by a common theme). E-mail models contain associations with time models and ontologies, models of persons, documents. In addition, there are procedures that allow you to find both the message itself and to select its individual parts [1-2].

Medium-level models (normative models). Medium-level models describe knowledge specific to a chosen SA and specific to a specific developer firm and its production processes [1-2]. The models of this level are a detailing and concretization of the general models of a higher level and reflect the regulatory requirements of the company - the developer and the customer regarding the organization of the process, the content and form of documents, etc.

A model of the software development process and the contractual process. The work uses a classic waterfall model with such stages as pre-project work (concept), analysis (formulation), development (implementation), support (sustaining) [18]. The model specifies that the contractual process is part of the pre-project works. As a result of this process, the text of the agreement agreed and signed by the Customer and the Executor is formed. An appendix to the agreement is a document - a technical proposal in which it is determined how the executor plans to perform the agreement, the platforms and means for execution are determined, the risks are analysed, the resources and the work schedule are determined. Permissible types of agreements are also defined here (for example, with a fixed price, with a price based on actual costs, with payment of employees at specified rates). A list of business events and related documents and metadata is defined. For example, such standard business events are receiving (sending) a letter, a meeting, a teleconference [1-2].

Document models. They contain models (templates) of such documents as an agreement and a technical proposal. The text of these documents is divided into meaningful sections and structured in the XML language with a semantic interpretation of the content and purpose of the sections [1-2]. Individual fields of the agreement, such as the customer, the executor, the price, the period of validity, the phased plan is formalized and semantically interpreted, which allows the system to follow the change of values in these fields, apply the relevant rules and make decisions. In addition to the main documents, the form of additional, working documents is defined here.

Role models for individuals and legal entities. The role models of participants in the contractual process are defined - the customer, the executor (legal entities), the project manager of the executor, the project manager of the customer, the main managers, the head of the quality control group, etc. Authority to perform operations and change and access to certain sections (fields) of the agreement for each role is defined [1-2].

Corporate rules and restrictions are set on elements of models of middle and higher levels and cover complexes of interconnected models. For example, the rules stipulate the mandatory presence of a certain list of documents, the permission to correspond with the customer only to the project manager and the main manager. It is quite simple to formulate and follow the basic rules of correspondence, for example, such that every letter from the customer must be answered within the same working day. Violation of the rules causes certain defined actions - a reminder, a warning, informing the chief manager [1-2].

Lower-level models (working models). Models of this level contain information about real events and objects. They are formed during the negotiation process based on ready-made templates (prototypes) defined at the upper levels. Individual elements of these models contain links and specify dependencies with models and ontologies of higher levels, which allows for analysis, interpretation of changes, decisions and monitoring of the execution of corporate rules during the negotiation process. Lower-level models can be divided into [1-2]:

- general, containing permanent information about the project;
- process models containing information about events in the negotiation process are ordered by time;
- document models with version support and tracking of editorial changes and comments.

General working models. In these models, information about the customer and the executor is stored, identifying who exactly performs the roles in the project with the definition of additional parameters for these persons (surname, postal address, etc.). Process models are formed as a sequence of business events ordered in time. Receiving (sending) a letter, meeting, meeting, or teleconference is considered a business event. Participants in the process create their events, marking certain milestones of the project. For each event, set dependencies with documents and common models. So, for example, for a received letter, a dependency is determined by the document that is attached to the letter and the editorial changes that were made in this document compared to the previous version. For the teleconference event, its date and time, participants, reference to the decision protocol and changes in the documents that were made based on the results of the teleconference are determined. Document models contain information about the content of real documents that are developed in the contractual process. Such models support versioning and track all changes in documents, their authors, and comments on changes. In addition, changes in all formalized fields are monitored using corporate rules and restrictions related to the content of these fields [1-2].

Procedure of IS for supporting decision-making. The contractual process takes place in the course of communication between representatives of the customer and the executor through IS. This system is intended to support the work of representatives of the executor, in particular the project manager. In the process of communication, a working model is built, which acts as a repository of data and knowledge about the progress of negotiations, documents and events related to negotiations. Some components of working models are determined manually, others - automatically. For example, business events are automatically generated upon receipt or sending of e-mail messages, connections of this business event with documents attached to the message, changes in the agreement and formalized fields initiated by the message are established. The project manager manually defines meaningful connections and dependencies with other documents, business events, and formalized fields. The manager manually enters such events as meetings, teleconferences and determines their meaningful connections. This activity contributes to a deeper understanding of the progress of the process by the manager, the formed model is used in the future to analyse the progress of negotiations. For each business event, its context is defined as a set of associated objects. At the same time, both direct associations and inheritance and obtained by logical deduction are taken into account. For example, a specific mail message is associated with the date of its receipt, the stage of the process, the author and recipients, changes in the text of the agreement and other documents, the content fields of the agreement that were changed as a result of receiving the letter, the content of the message, the sequence of messages in which this message is included, etc. [1-2].

Training of the system, i.e. the formation of new models, rules and ontology, is done both manually by the system administrator and auditor, who monitor the integrity of the data and taking into account corporate rules, and by the user - the project manager. If a certain concept used by the user in the query does not have a definition in the system, then the user is informed about this, who formulates a definition that is stored for the future. For example, let the user formulate the query "How long have the negotiations been going on?". To find the answer, the system decides to apply the time duration model for the "negotiation" object. The duration model attempts to define duration as the time interval between two events. The final event of the interval is "today", and the initial event is the beginning of negotiations. If the event - the start of negotiations is not defined, the system asks the user to define the corresponding rule (or constant). Let the user define the start of negotiations as the business event date of receiving the first letter. This rule (at the request of the user) will be saved and the system will further determine the start of negotiations upon the beginning of the correspondence process (including for subsequent projects). The user interacts with the system using a set of text and graphic interface tools. For text interaction, a language similar to the "action language" (action language) [19] or the semi-formal language given in [20] is used. Such a language uses and relies on the SA terms defined in the top-level and middle-level ontologies. A significant advantage of the proposed system is the ability to work on the project regardless of the user's location. All logic, knowledge and documents of the system are stored on the server, and only copies (snapshots) of documents, letters, and other project information necessary for decision-making are transferred to client computers [1-2].

4. Experiments, results and discussions

4.1. Resource access control models

Currently available resource access control mechanisms include such approaches as selective (DAC - Discretionary Access Control), mandated (MAC - Mandatory Access Control), and one that uses roles (RBAC - Role-based access control). In DAC, the access policy is defined by the user-owner of the object. The owner grants rights to other users. Therefore, with this approach, each managed resource has an owner. In practice, in many enterprises, users do not "own" information resources, but the sole owner is the enterprise itself. In addition, it is not always advisable to give ordinary users the right to provide access to other users [1-2]. DAC does not always adequately reflect existing dependencies between subjects and access objects. Using DAC, it is difficult to administer some resources, and it is impossible to define and maintain complex access policies. In the MAC method, the access policy is determined by the system, not by the owner. MAC is used in multi-level access control systems where each computer can handle many levels security labels associated with resources. To obtain permission to access a resource, the access subject must have an access level no lower than the label associated with the specified security object (managed resource). The main problem solved by MAC is the control of access to confidential information. MAC is widely used in military systems where there is a need to control access to classified documents and the number of levels of secrecy is relatively small. For use in industrial systems, the abstraction with access labels is not flexible enough and does not reflect the realities of industrial BAs [1-2, 21]. In role-based access control (RBAC), access rights are also determined by the system rather than by the user owning the resource as in DAC. A role in RBAC corresponds to a set of resource access rights. The RBAC mechanism allows you to define access to complex business operations, such as transactions. Users are granted the appropriate rights by association with certain roles. Roles are grouped into hierarchies in which the permissions of higher-level roles are inherited by lower levels. Access control using roles has the greatest effect when there are many users in an enterprise with the same set of access rights, which boil down to defined roles. At the same time, changing access rights for a role immediately applies to all users associated with this role. [21]. Role-based access control is currently the most advanced approach to access control. Draft standards have been developed for RBAC [22]. At the same time, in RBAC, access rights are assigned manually by the administrator and are static. In work [23], the main shortcomings of the RBAC access control mechanism are identified [1-2]:

- large organizations have many users whose access rights sets change and depend on the current tasks being performed. Access control using RBAC in these circumstances results in the creation of a large number of roles that are difficult to administer collectively;

- when new subsystems are added to the existing system, the number of roles increases in arithmetic progression, which at a certain stage makes effective administration impossible;
- changing the content of roles requires correcting their definition. These functions should naturally be performed by business employees who make decisions and issue tasks for execution. However, correcting resource access permissions requires significant technical knowledge that business employees do not possess. Therefore, support for RBAC requires a staff of technical workers - system administrators, the number of which increases with the increase in the complexity of the system;
- over time, access rights for employees expand, because new rights are added when performing new tasks. The reverse process (withdrawal of rights) is not carried out in practice or is carried out with a significant delay. As a result, employees receive unreasonably large access rights, which reduces the overall degree of system security and contradicts the principle of minimum access rights.

In [23], to solve the last problem, it is suggested to periodically audit the system and remove irrelevant rights. At the same time, such an operation requires considerable time and is quite complicated [1-2]. The principle of least rights [21] is generally accepted in the administration practice. It consists of the fact that the user is granted access rights no more than the tasks that this user performs require. Compliance with this principle makes it impossible for the user to perform unnecessary and potentially dangerous actions. In practice, compliance with the principle of least rights requires a detailed specification of the necessary access rights in the conditions of constantly changing tasks performed by the user, which is a difficult task [1-2]. The evolution of methods of managing access to information system resources largely took place in two directions [1-2]:

- solving the problem of single-user registration and authentication within the existing IS. At the same time, the user is authenticated not to work on a specific server, but in the entire information system, or part of it - the domain. Directory services with appropriate authentication infrastructure and system-wide policies were developed to solve this problem;
- consideration of business abstractions to create a correspondence between business processes and user access rights. Such business abstractions are users, groups, positions, organizations, organizational units, and roles. Abstractions and related mechanisms for assigning access rights are described in detail in [24].

Existing access control approaches have the following significant drawbacks.

- lack of documented justification for the decision to grant or revoke access rights;
- access management decisions are not related to the business processes and rules existing in the enterprise. At the same time, actual business processes and rules are the source and basis for assigning access rights;
- the request for the assignment of access rights is accepted by the system administrator, who is a technical employee, not a project manager or another person who is authorized to make management decisions and directly manages the execution of business processes;
- the manual nature of the assignment of access rights, often reactive, and the non-proactive assignment process, leads to delays in the execution of processes;
- lack of templates, typical configurations of access rights that can be reused for different users leads to unproductive use of human resources;
- the availability of various managed IT resources, such as file systems of different computers, DBMS tables, access to premises (electronic locks), and various information services, is not taken into account. The lack of planning access to various IRs in the context of the performance of a certain production task or role leads to a lack of coordination of access, errors and delays in the performance of tasks;
- the presence of different types of managed resources, which store separate versions of accounts of the same person, leads to the need for the user to remember a large number of passwords. Password complexity is limited by a person's ability to remember it. If you refuse to remember the password, you can create longer and more complex passwords and thus better protect the system.

As a result, assigned access rights are often inadequate to the tasks performed by the employee - there are too many or too few access rights [1-2]. The lack of access rights leads to the impossibility or significant delays in the performance of the task, which negatively affects the business process as a whole. An excess of access rights, although it creates additional opportunities for the employee to choose different ways of solving the task, generally reduces the security of the information system and does not comply with the principle of the least access rights. The use of models makes it possible to develop a method of managing access to resources that takes into account the structure of business processes and largely solves the above problems. Management of access to resources using models occurs in IS based on modelling and supporting the execution of business processes [1]. Constituent parts of this modelling system (Fig. 5) are the fact base, ontology, and model repository. Ontology occupies a central place in the system because all other components are formulated on its basis. So, facts from the fact base are objects defined in the ontology. Each model from the model repository is formulated using concepts also defined in the ontology. In the ontology, general restrictions and dependencies between types of objects are formulated, which are taken into account when creating and using models [2].

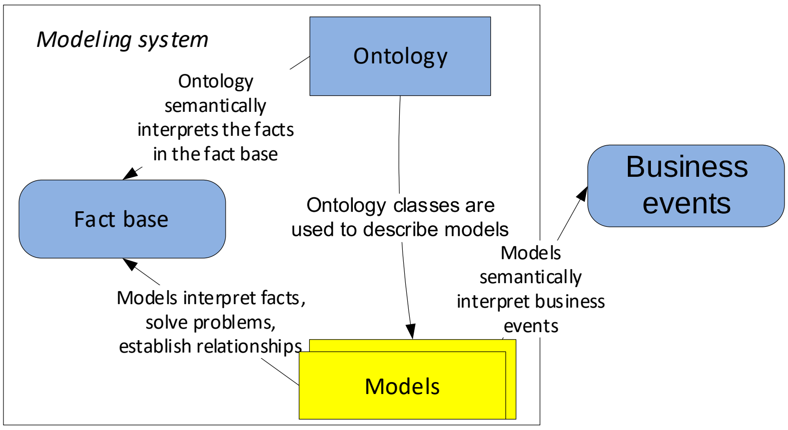

Figure 5: Components of the modelling system

Models are used to perform operations in the system. Each model is designed to solve one specific problem and is relatively simple. At the same time, a model in the process of execution can activate other models. In this way, complex conglomerates of models are formed, which collectively solve complex tasks. Models are created by a person-expert and reflect his knowledge about how to solve a specific problem. Work [1] defines two types of business models used in the system - regulatory models and working models. Working models reflect real events, documents, and business operations as ontology objects (facts). Working models are used and created during the execution of business processes. To ensure correspondence between the realities and the facts of the fact base, the information system mustn't allow other ways of conducting business operations than through the mediation of the modelling system. Normative models reflect business rules, standard procedures, and document templates and impose additional restrictions on working models to achieve compliance with corporate and state standards [2]. Resource access management models (hereinafter - resource models) are essentially regulatory models that limit access to specified resources and are associated with individual operations of the work model of the business process, managed resources, and with a certain set of workers [2]. Fig. 6 shows the general scheme of the resource model. A resource model generally specifies the functional roles of employees and the resources needed to solve a specific (often typical) business task [1-2]. A set of roles and a set of resources are defined in the model. Business abstractions that are understandable to a business employee (for example, a document, an e-mail service, or the Internet) are used as resources. Relationships are defined between roles and resources, weighted (parameterized) by such characteristics as access restrictions (for example, read-only) and quantitative restrictions (for example, maximum traffic). Before starting a task, role models are associated with specific employees who perform these roles. This operation is implemented by a manager, not a system administrator. After carrying out such an association and under the condition that the corresponding business model is active, the associated employees acquire the access rights defined by the resource model. After the completion of the task, the granted access rights are automatically withdrawn. Granting access rights occurs by creating XML files with the model that contain all the necessary information and sending these files to the appropriate resource management services. Resource models work with the semantic abstraction of a person - a company employee, not a computer user, as is done, for example, in operating systems. This allows you to track access rights to various resources in the context of the employee's performance of production tasks. Accordingly, the fact base for each employee stores information about budgets, usernames and passwords with which this employee works with various services and computers of the information system. This information is used to provide access to specific resources. The employee may not have information about his accounts and passwords. At the same time, it is necessary to ensure its authentication at the working beginning with the system using, for example, biometric methods or smart cards [2].

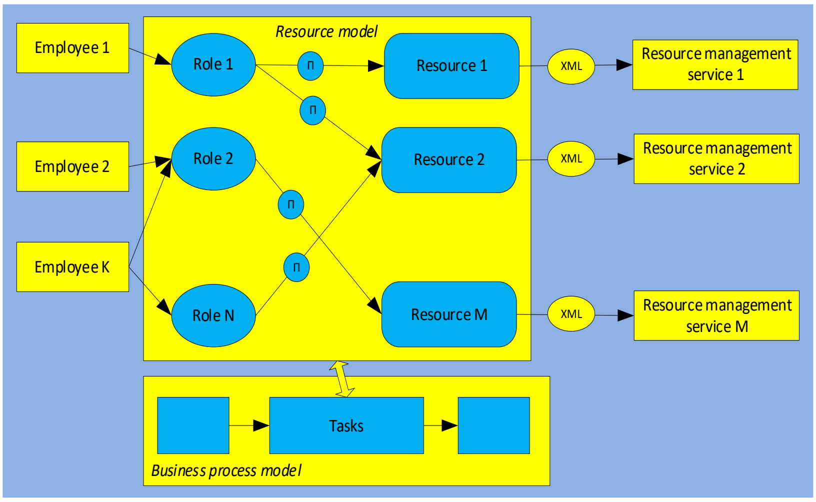

Figure 6: The general scheme of using the resource model [2]

Models are associated not only with individual operations of the business process model but also with certain facts of the fact base. This ensures the granting of rights to the employee depending on his position, age, and gender, and supports the implementation of corporate-wide access policies, regardless of the tasks performed by the employee. For example, such rules as "Every employee of the company has access to the e-mail service", "Every employee of the company has access to the corporate intranet portal", and "The project manager is allowed to initiate international calls" are implemented in this way. Let's consider the resource model used in the business process example of creating a Technical Proposal for a software development project (Fig. 7). The process of developing a technical proposal begins after receiving a request for a technical proposal (RFP) from a potential customer. In the request, the customer's requirements regarding the final product, technologies and development process, and business expectations regarding development terms and product quality are formulated. In the process of developing a technical proposal, a document is created - a "Technical proposal" in which the components, technologies and method of development are determined, the terms, stages and intermediate stages of development are determined, the functional capabilities of the final product are specified, additional limitations and the cost of development are determined. The technical proposal is developed by a narrow circle of experienced specialists under the leadership of the project manager. As a rule, in the development of a technical proposal, in addition to the project manager, who is responsible for planning and technical implementation, the head of the quality control (QA) department, who is responsible for the development of the product quality control process at all stages of its development, participates. If necessary, if the product is technically complex enough or requires testing on many hardware and software platforms, an expert in Configuration Manager is also involved in the development of the proposal [1-2]. This expert plans the processes of purchasing additional equipment, installing it, and setting it up.

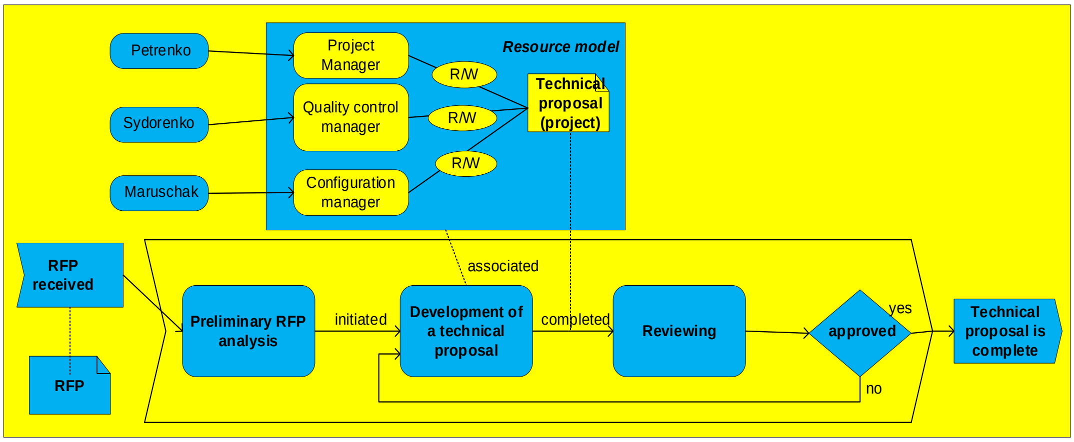

Figure 7: Use of the resource model in the development of a technical proposal

The procedure for developing a technical proposal is determined by corporate standards. Let the development of the proposal begin after the approval of the RFP by the senior manager and the formation of the project team. The manager also associates workers with model roles. An initial technical proposal document is generated based on a template and stored in a document repository with controlled access, such as VSS [1-2]. During the creation of the document, the members of the project team have access to the document both for reading and writing. They create separate sections of the document and read and comment on information added to the document by others. The final responsibility for the content and quality of the document generally rests with the project manager. When the document is created, it goes to the reviewer. At the same time, the proposal creation model ceases to be active and all members of the project group are deprived of access rights. The reviewer is a senior manager. He reads but does not change the sentence, or express his comments. If the draft technical proposal is approved by the reviewer, it is sent to the customer. If changes need to be made to the document, it is sent to the project team for revision. At the same time, the proposal creation stage model becomes active again and all members of the team again get full access to the document. After that, the process is repeated until the document is approved or finally rejected by the reviewer. Compared to the RBAC resource access control method, the proposed method has the following advantages [1-2]:

- the granted access rights correspond to the tasks performed by the employee at this moment in time and, in general, to the set of business processes performed in the organization;
- dynamic nature of assigning and withdrawing access rights;
- lack of expansion of access rights over time;
- more detailed access control using semantic abstractions. Ability to build complex access control rules using business object properties, time parameters, and all relevant knowledge base facts;
- granting access rights is performed by a business employee authorized to make management decisions, not a system administrator. This simplifies the process of assigning rights and frees system administrators from routine work;
- granting of access rights is documented and retained even after the model becomes inactive. The document becomes the resource model itself. This simplifies system auditing;
- the possibility of reusing resource models reduces the labour intensity of the assignment of rights;
- an opportunity is created to standardize business processes and individual business operations, as well as appropriate resource models and ensure compliance by employees;
- the task of allocating access rights is solved simultaneously and in combination with the tasks of planning the task execution process, estimating the required number and parameters of resources;
- automatic assignment and withdrawal of rights according to the formed resource model.

The disadvantage is the greater complexity of the system, and the need for preliminary implementation of the business process modelling system using the knowledge base. Formally, the resource model is presented as a tuple [2]:

$$MdRes=(M(Rol),M(Res),M(LnRolRes)), \quad (6)$$

where $M(Rol)$ is a set, all elements of which are objects of the Role class defined in the *On* ontology: *Role*: $M(Rol) = (Rol|Type(Rol) = Role)$ similarly,

$$M(Res) = (Res|Type(Res) = Resource), \quad (7)$$

$$M(LnRolRes) = (LnRolRes|\ Type(LnRolRes) = LinkResourceRole).$$

Ontology classes corresponding to roles generally form a class hierarchy. For each class, hierarchies define rules and restrictions that are relevant to the given class and must be followed by all derived classes. So, for example, you can define a general class <u>*Role*</u>, and a more detailed class *PMRole* to describe a role, for example, the role of a project manager. In the *PMRole* class, restrictions valid for project managers and independent of a specific resource model will be displayed [1-2]. In turn, the *Rol* object is itself a class, but defined and valid only within the *MdRes* resource model. In this sense, different instances of these objects can exist in different instances of *MdRes*. As a class, *Rol* contains additional constraints formulated at the *MdRes* model level [2]: $Rol=(M(SlRol),M(CsRol))$, where $M(SlRol)$ are the properties (slots) of the role, $M(CsRol)$ are the restrictions defined for the role. The most important category of *CsRol* restrictions is the restriction on the characteristics of the people who are allowed to assign a role to a *Rol*. Such restrictions may, for example, be requirements regarding experience, length of service, position, etc. Among the properties of *SlRol* roles, both properties inherited from parent classes $M(SlRol)_{inh}$ and properties defined at the model level $M(SlRol)_{md}$ [2] are defined: $M(SlRol)=M(SlRol)_{inh} \cup M(SlRol)_{md}$.

Similarly, role restrictions are divided into inherited restrictions, and restrictions are additionally defined within the model [2]:

$$M(CsRol) = M(CsRol)_{inh} \bigcup M(CsRol)_{md}. \quad (8)$$

Ontology classes corresponding to resources also form a hierarchy and inherit their properties from the general Business Resource class. The ontology of managed business resources includes, for example, such important types of resources as Documents and Services. The resource specification contains a set of properties $M(SlRes)$, a set of restrictions $M(CsRes)$ and a set of references to system services that implement and maintain the type of resource defined in the model: $M(SvRes)$: $Res=(M(SlRes), M(CsRes),M(SvRes))$ [2].

Similarly, to roles, properties, constraints, and a set of service references are inherited from the parent classes of the *On* ontology [2]. The relationship between the role and the resource *LnRolRes* is defined on the pair (Rol,Res): $LnRolRes=((Rol,Res), M(SlLnRolRes), M(CsLnRolRes))$, $M(SlLnRolRes)$ is communication parameters, $M(CsLnRolRes)$ is a limitation.

Creation and use of resource models. The development of resource models is carried out at several stages - creation, initialization, and use. The prerequisite for using resource models in the system is the development and implementation of a business process modelling system and the development of an ontology of business objects. Resource models are created by a business employee or manager who has a deep understanding of both the enterprise's business processes and the resources that are needed to perform these processes, by configuration. A reusable resource model is created in a graphical editor – a modelling environment. The author of the model defines roles, and resources, and describes their relationships. In the defined model, it includes restrictions that are dictated by business rules or technical reasons. Ranges of permissible types of values are defined for the main components of the model. A special component of the modelling environment – the validator – checks the constraints existing in the ontology and requires their compliance. The finished model is validated for the absence of ambiguities and uncertainties. The finished and validated model is stored in the model repository as an XML file [2]. In the process of resource model initialization, the template model is configured from the model repository associated with a specific business process operation. The initialization operation is performed by the business worker. In the process of this operation, based on the ready-made model template, an instance of the model is created, which is associated with a defined business operation. Constraints and parameters are added to the model instance that are important to the implementation of that model in the context of a business operation. An important task is to specify the values specified in the template as a range (set) of possible values. The initialized model is validated and in the case of a positive result, it is ready for execution [1-2]. The initialized resource model is executed automatically. The signal to activate the resource model is the activation of the associated business operation model. After activating the model, the component of the modelling system - the model interpreter - interprets the XML file describing the initialized model and generates service management commands of the managed resources, which add the corresponding ACEs to the ACL list of the resource. The signal for deactivating the resource model is the event of the end and deactivation of the associated business operation. In this case, the model interpreter generates commands that remove access rights granted by the model [2].

Resource model presentation language. To describe the resource model, a data presentation format was developed based on the XACML (Extensible access control language) resource access description language standard [3]. An example of part of the XML file of the resource model for the process of developing a technical proposal is given below [1-2].

```
<Model>
 <ModelMetadata>
  <ModelId> id </ModelId>
  <ModelType> ResourceAccessModel </ModelType>
  <OntologyURI> www.acme.org/ResourceOntology</OntologyURI>
 <ModelRepositoryURI>www.acme.org/ModelRepository</ModelRepositoryURI>
  <BusinessOperation Datatype="&xml;string" Name="BusinessOp"
OntologyType="business_operation">ProposalCreationURI</BusinessOperation>
 </ModelMetadata>
 <PolicySet PolicySetId= "PPS:projectmanager:role">
  <Role OntologyType="ProjectManagerRole">
   <AttributeValue Datatype="&xml;string" Name="RoleName"> ProjectManager</AttributeValue>
   <AttributeValue Datatype="&xml;string" Name="Instance" OntologyType="person"> Marushak</AttributeValue>
   <AttributeValue Datatype="&xml;string" Name="Constraint"
OntologyType="constraint">Constraint</AttributeValue>
  </Role>
  <Rule RuleId="Permission:to:write:and:read:proposal">
   <Target>
    <Resources>
     <Resource>
      <AttributeValue Datatype="&xml;string" Name="ResourceName"
OntologyType="document">Proposal</AttributeValue>
     <AttributeValue Datatype="&xml;string" Name="Instance">DocumentURI</AttributeValue>
     <AttributeValue Datatype="&xml;string" Name="Constraint"
OntologyType="constraint">Constraint</AttributeValue>
     </Resource>
    </Resources>
    <Actions>
     <Action>
     <AttributeValue Datatype="&xml;string" Name="ActionName" OntologyType="action">Write</AttributeValue>
     <AttributeValue Datatype="&xml;string" Name="ActionName" OntologyType="action">Read</AttributeValue>
   <AttributeValue Datatype="&xml;string" Name="ServiceURI"
OntologyType="AccessControlService">ServiceURI</AttributeValue>
     <AttributeValue Datatype="&xml;string" Name="Constraint"
OntologyType="constraint">Constraint</AttributeValue>
     </Action>
    </Actions>
   </Target>
  </Rule>
 </PolicySet>
 <PolicySet PolicySetId= "PPS:qamanager:role">.......</PolicySet>
</Model>
```

The model description contains metadata sections and access policy description sections. The metadata section contains general information about the model: identifier, ontology reference, model repository, and business operation associated with this model. The access policy sections for each role define resources and allowable operations. An important part of the description is the reference to the appropriate ontology type and the definition of additional constraints acting at the model level. The developed method of access to resources using models allows building an information system in which access rights are assigned dynamically, in the context of performing business tasks. At the same time, the administering access rights task is significantly simplified when the level of system protection is increased [2].

4.2. The architecture for the simulation environment and functional specifications of its services

4.2.1. Scenarios for using the modelling system

The solution of the set tasks involves the analysis of actors, and methods of use, with further definition and detailing of the functioning processes of IS modelling. The main actors in IS modelling are:

- **SA expert** – creates or modifies the conceptual model, validates it, and supervises its use;
- **programmer** – creates, maintains, modifies and tests model interpreters;
- **simulation agent** – executes models in the simulation environment.

The usage scenario "Creating a model" is, in turn, divided into the scenarios "Model tasks", "Evaluation of execution results and modification of the existing model", "Ontology processing", and "Model testing". The scenario "Execution and support of models" is divided into scenarios "Execution of the model", "Support of interaction of models", and "Search for information". Each of these scenarios is executed by a separate simulation environment service. Summary information about actors, scenarios, and relevant software tools or services of the modelling system is displayed in the Table. 3.

Table 3
Scenarios for the use of the modelling complex

Actor	Usage scenario	Software (service)
Subject matter expert	Creating a model. During model creation, it is validated and tested, and rules for automated verification are set Evaluation of the results of operation of models. Modification of models. Model testing Development of ontology	Model editor
Programmer	Creation, modification and testing of the model interpreter	Integrated programming environment
Modelling agent	Execution of models	Interpreter of models
Simulation environment	Organization of model interaction.	Model interaction broker
	Information search	Provider of information services

4.2.2. Presentation of conceptual models in the modelling system

The Md conceptual model consists of the $ScMd$ circuit and the $RlMd$ implementation: $Md = (ScMd, RlMd)$. A schema is part of a model that is created by a human expert in SA. The scheme specifies the objects involved and the logic of their processing, thereby determining the way to solve a certain problem. The scheme is described in terms of SA, regardless of the possible technologies for its implementation. The presence of a formalized scheme allows not only to ensure its machine interpretation and implementation, but also creates an opportunity for other experts to analyze and evaluate this scheme and, if necessary, correct it. The presence of a scheme that can be changed without changing the implementation at the same time gives the system built using models the necessary flexibility - because when the environment changes, it is enough to change the scheme to change the behaviour of the system. At the same time, such time-consuming and lengthy stages of the traditional software development process as coding and testing become unnecessary. In addition, the presence of a formalized scheme serves as a means of documenting the knowledge of the specialist - the author of the model regarding the method of solving the problem. On the other hand, it is advisable to simplify the implementation of the model by creating reusable components or a single implementation for a wide class of models. The versatility and simplicity of the model implementation give the system flexibility, ensuring the immutability of the implementation when the model scheme is changed. The main requirements for the scheme:

1. The scheme should be simple. The scheme is created and pre-validated by a person - a specialist, and it reflects a certain mental model of that person. The recommended number of concepts in the scheme does not exceed 5-8, because according to the research of psychologists, a person can keep in focus at the same time [25]. This requirement directly follows from the requirement for the cooperation of the intellectual system with a person. In addition, simple, universal models that do not have functional redundancy can easily be combined into more complex ones.
2. The scheme is created using ontology concepts common to all models. When creating a scheme, the constraints defined in the ontology are checked. Thus, when creating a scheme, the experience of other experts is taken into account and formalized in the ontology, and compliance with the same restrictions is ensured by different models. Instead, when creating a scheme, new, additional restrictions are added.

Let's take a closer look at the main components of the model scheme. Some of these parts are created by the author, others are automatically filled in by the model design tool.

The model diagram has the following information sections:

- **metadata** is information about the model in general: author, creation time, modification history, ontology reference, and model type(s). Determining the model type allows you to assign the model to a certain classification category depending on the types of problems and methods of solving them and to find the necessary models based on a known class of problems;
- **base model** – defines the reference to the base schema (data and constraints on it) that the activator of this model needs to fill. Also specifies the mapping between the activator and base model slots. Contains the definition of the relevance function of the problem-solving method.
- **information for initialization** – defines mandatory and optional model slots, indicates possible additional sources for obtaining data;
- **prerequisites** is a list of conditions that must be met to activate the model. The conditions are checked against the underlying model data and additional data obtained from the context. If the prerequisites are not fulfilled, the model is not activated, but a message about the violated conditions is returned;
- **model body** – the format is determined depending on the type of model. Specifies the objects involved and the logic of solving the problem;
- **information for verification** is the verification section is used when modifying the model, or when creating another model based on the data. This section defines the model integrity constraints that must be met for all modifications;
- **additional sections** are defined depending on the type of model. For example, for models that perform monitoring tasks, there is an additional section that determines the frequency of model activation.

The model implementation is a software component that processes the model scheme, initializes it with facts from the fact base, and performs the actions specified by the model. Unlike the scheme, the implementation of the model is created by a specialist programmer.

The main requirements for the implementation are related to the need to ensure the possibility of changing the scheme within wide limits without changing the implementation. At the same time, it is allowed to change not only the parameters of the scheme or restrictions but also to a certain extent to change the structure of the scheme. Implementation requirements:

- the implementation should, if possible, be universal and work for a wide class of model types. At the same time, a semantic interpretation of model data is used;
- to reduce the complexity of reconfiguration, the implementation should use reusable components, and be based on simple functional software components that are the same for all implementations.

The implementation of the model corresponds to the software component – the model interpreter. This component receives the model specification in the form of an XML file, as well as the initialized base model from the model interaction broker, activates and processes the activated model. The main functions of the model interpreter are as follows: initialization of the model - obtaining all the necessary facts from the DB; verification of fulfilment of prerequisites; development of the model's body; support of the model interaction protocol and cooperation with the model interaction broker. The work offers some recommendations for creating a model interpreter, which will significantly simplify the process of creating and modifying the interpreter:

- it is advisable to create a model interpreter in a script programming language (such as Python, JavaScript, VBasic, or Perl) - this reduces the time for modifying the program and does not require a high qualification of the programmer;
- because the activator model transmits only the minimum amount of data (in the base model) to the activated model, and the model itself receives the rest of the data from the context - it is impossible to ensure the presence of the required data model in the fact base in advance. Therefore, it is important to take into account the possibility of missing the necessary data. The interpreter must process the situation of the lack of necessary data in the KB and have "default" solutions ready, or report the impossibility of solving the problem and the reason for this;
- to reduce the complexity of creating a model in the structure of the interpreter, it is advisable to use basic functional and logical components, for example, those that implement the operations "Search in the fact base", "Check the prerequisites", "Request the service", "Check the condition", "Create or modification of the fact" etc. Such functional primitives, which are used by all interpreters, will not only simplify the procedures for creating and maintaining the interpreter but also implement a unified approach to the implementation of typical model interpretation operations;
- one interpreter should be created for a whole class of models. It should be independent of the execution platform and the logic provided in the model. At the same time, the "logic of interpretation" is displayed in the interpreter - the logic of processing models, which makes it impossible to use a single interpreter for all classes of models;
- the actions performed by the interpreter are suggested to be formalized as requests to the services of the information system with defined formats of these requests. Thus, the available services determine the entire range of possible actions in the system.

4.2.3. Development of conceptual models by the modelling system

Let's take a closer look at individual operations and aspects of model processing. These operations are performed by different software components and at different processing stages. For example, the model interpreter performs the operations of model initialization, precondition checking, and model execution. The model editor implements model creation, ontology-based verification, testing, and simulation. The interaction broker supports the process of model interaction. We will pay special attention to issues of interaction of models and solving complex problems using models. Important tasks implemented at all levels of model development are verification, validation, relevance assurance, and selection of a model for implementation from several alternatives.

Initialization. The initialization operation is performed by the model interpreter immediately after its activation and filling of the slots of the base model. The purpose of initialization is to obtain all the data necessary for the execution of the model from the local fact base and, possibly, from external sources. The initial data for initialization are the semantically interpreted data of the base model, including the specification of the purpose with which the model is activated. Filling the slots of the basic model, which occurs in the process of interaction with the activator model and is carried out by the model interaction broker, will be considered the first stage of initialization. During the second stage of initialization, the necessary data is extracted from the context of the basic model in the fact base. In the model scheme, in the section that specifies the data for the initialization process, acceptable configurations of slot fillings are defined depending on the goal of the problem posed to the model. In the simplest case, mandatory and optional model slots are defined. In more complex cases, when there is no data in the fact base, a model is indicated that can be used to search for the required data.

Check prerequisites. Prerequisite checking is performed by the model interpreter after filling the model slots with data. The purpose of this check is to determine the relevance of the model based on the available data in the fact base. In the process of checking the prerequisites, the relevance conditions specified by the author of the model are checked. For example, the model that manages access to the company's house, after identifying the person who claims access, defines positive prerequisites for granting access in such cases: a) the applicant is a permanent employee of the company b) is a temporary employee under a contract c) is an invited person, which there is an entry in the relevant register. If none of these prerequisites are met, the access request is denied.

Execution of models. It is implemented by the model interpreter and is discussed in the section devoted to the principles of constructing interpreters.

Interaction of models. General principles of model interaction were considered in [26]. In particular, the concepts of model - activator and activated model, basic model, relevance functions and model selection, model interaction protocol were introduced. Here we will consider the mechanism of interaction of models in more detail, from the point of view of its technical implementation. (Fig. 8) [27-33]. The initiator of the organization of the interaction of models is the currently active model, which needs to solve its main task by solving secondary, auxiliary tasks [34-39]. For example, the model that forms a tourist trip, at a certain stage of its implementation, needs to solve the task of buying bus tickets. At the same time, she will contact the model who will make such a purchase [40-54]. In order to reduce the complexity of implementing the model - activator, we will introduce a separate service of the modelling system - "Model Interaction Broker", which will ensure the implementation of model interaction. The functions of an interaction broker include: goal analysis and search for relevant models; definition of relevant models; choosing one from a set of relevant models and activating it; initialization of the activated model; tracking its execution process and returning the result to the activator; support, if necessary, of the connection between the activator and the activated model.

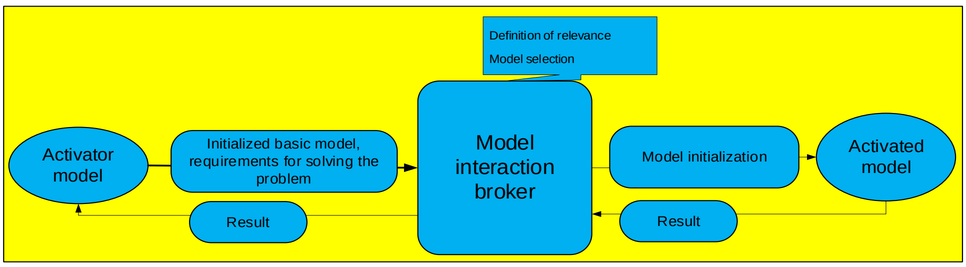

Figure 8: The process of interaction of models

As can be seen from the description of the service functions, it is a rather complex software component, at the same time it is created once and serves the interaction of all components of the modelling system, thus providing a unified approach and rules for supporting the interaction of models. The base model is the data structure used to transfer parameters between the activator model and the activated model. We will consider such a model as a general statement of the problem, which specifies the type of problem, the objects for which this problem is solved, and restrictions on the process of solving the problem. The activator transmits to the interaction broker a data structure in which the mapping of the activator slots into the slots of the base model and the requirements for the process of solving the task and the expected results are entered.

The base model is a schematic, not a complete model, meaning it has no implementation. Instead, the basic model specifies a problem statement for a whole class of similar problems. The hierarchy (taxonomy) of basic models corresponds to the taxonomy of goals (tasks) that are solved using models. Models that solve a similar problem, although perhaps with different methods and for different situations, have a common basic model. Base model slots correspond to roles that are filled with the values of the activator model slots. For example, the general task of conducting a financial transaction has such slots in the base model as Buyer, Seller, Goods, and Money. It serves as the basis for such more specific models of conducting financial transactions as "Purchase of real estate", "Payment for a service", "Purchase of a food product" and others, which differ in the way of conducting the transaction. The choice of one or another basic model in the process of interaction of models is carried out by the activator. At the same time, the following options for defining the basic model are possible:

a) the relevant model (or set of models) is explicitly specified by the author of the model. In this case, the base model is defined through the already defined model, since each model has a reference to the base model that it uses;

b) only the activation goal is given, and the relevant base model needs to be defined. In this case, the basic model and the corresponding set of subordinate models are determined by the ontology of goals (tasks), which is part of the general ontology of the system.

Based on the completed basic model, the interaction broker defines a set of models that implement relevant methods for solving the given task. The relevance function *ReLnMd* is used to determine the set of relevant models. Note that the concept of relevance in the case of models has two aspects. In the first, the relevance of the method of solving the given problem is determined. The other is relevance based on the current state of the fact base. The relevance of the method of solving the problem is determined by the basic model, and the relevance based on the state of the fact base is determined by the prerequisites of the model execution based on the filled slots of this model. Therefore, it is advisable to split the relevance function into two - Re_{mt} and Re_{ft} and assume that

$$ReLnMd = Re_{mt} \& Re_{ft}. \quad (9)$$

Such an approach, which separately considers the relevance of the method and the relevance of the data, allows to reduce the time of model processing, because the initialization of the model and the search for all the necessary data in the context of this model require more time. Therefore, preliminary determination of the relevance of models is carried out only in the context of the base model. The basic model, as a rule, corresponds to a group of problems that have a similar formulation. In the context of an initialized base model, the list of models subject to this base model is reviewed. The relevance function Re_{mt} is associated with each subordinate model, which, based on the data of the base model, determines the relevance of the application of the method of solving the given problem implemented in the model. The relevance function Re_{mt} is defined as a set of statements in the context of the base model.

$$Re_{mt} = M(Asr(Con(Md_b))). \quad (10)$$

For example, if you want to buy a bus ticket, the common basic business transaction model is activated. The Buyer is defined as a *Tourist*, the *Money* slot is initialized with a reference to the tourist's bank account, the *Goods* is a transportation service, and the *Seller* is a carrier. Based on the filled slots of the basic model, and referring to the context of the facts defined in it, the relevance function determines that there is one relevant model in the list of trade transaction models - "Service payment", because the product is a transportation service.

Among the set of relevant models, one must be chosen, which will be used to solve the given problem. To select a model, use the selection function *Fc*, which maximizes/minimizes a certain defined criterion (such as the execution time of the model, or the estimation of the accuracy of the result). The purpose of the model selection operation is to determine one model in a non-empty set of relevant models, which will be activated to solve the given problem. The range of complexity of approaches in its implementation varies from simply random selection of any model from the list of relevant ones to the application of complex methods of evaluating candidate models according to various criteria. In general, to solve the problem of model selection, it is necessary to create a whole selection infrastructure that uses its own models, taxonomies, sections in the schema of the activator model and the activated model. Let's define the components of this infrastructure.

- ontology of selection methods and associated selection models. It includes, for example, random selection, single- and multi-criteria selection methods, etc.;
- ontology of selection criteria. Defines the semantics of possible model selection criteria, for example, time (complexity) of model development, expected accuracy of results;
- part of the enabler metadata that specifies the benefits and requirements for the model execution process. This data defines the constraints and expectations of the activator regarding execution time or accuracy of results. The use of this data will make it possible to decide on the requirements for the selection and execution of the activated model and will be used to form a similar section in the metadata of the activated model. For example, if the activator is limited by execution time, then it will give preference to fast models in the selection of activated models;
- the metadata part of a candidate model contains information about the characteristics of that model, such as complexity, execution time, and accuracy. At the same time, there may be references to other models that will allow you to estimate the execution time or the accuracy of the results in advance;
- choice models that form requirements and preferences for the activated model and implement the choice itself. In addition, these models take into account restrictions on the selection process itself. For example, in some cases, the choice must be made as soon as possible, and spending time on an accurate assessment of the parameters affecting the choice is impractical. Choice models are universal and are used in other situations when it is necessary to choose from several alternatives.

The selection of the model for activation is carried out by the selection model at the initiative of the model interaction broker. The order of interaction between the models and the broker is determined by the interaction protocol. At its simplest, this protocol contains the commands to request a broker, request an activated model, respond to a broker, and activate a model. In a more complex case, if the activated model is defined as a multifunctional service, the protocol additionally defines commands and responses according to the interface published by the model.

5. Conclusions

Models are created and validated by a person who is an expert in a given subject area and reflects this person's knowledge of how to solve a certain problem. To create or modify a model, use a tool – the Model Editor program. The main functions (use options) of this program are as follows:

- creating a model diagram using ontology classes and facts. Definition of additional restrictions. The model scheme is created in a graphical representation, and based on the finished scheme, an XML file of the model description is generated;
- work with ontology. Searching for ontology classes and facts. Modification of classes and relations. Definition of dependencies between certain ontology classes and existing models. Determination of the impact of changes in ontology on existing models;
- work with the fact base. Searching for the necessary facts, adding or removing facts from the database;
- working with the model repository. Search for template models using different criteria;
- defining the XML template of the model and specifying the method of generation of the XML description of the model scheme for the specified type of models;

- the task of the model verification block is a set of rules and dependencies that the model must satisfy. For this, you can use, for example, OCL. At the same time, different sets of validation rules are used when creating a new model based on this model, or during the initialization of the model;
- defining/configuring an existing model interpreter - a component that executes a model;
- testing of models and networks (aggregates) of models. Launching and executing models on the test base of facts, checking the compliance of the behaviour of the models with the expected results.

The functions of the model editor include both the operations of creating a scheme and creating a model implementation. The process of creating a model differs between creating a model "from scratch" and creating it based on an existing template model. Consider the sequence of stages of model creation.

Model type assignment. When creating a model scheme, its type is determined using the ontology of model types. The type is determined by the purpose or class of problems that the model solves. As a rule, models of the same type are interchangeable and can be used to solve similar problems. Thus, all models in the model repository are ordered according to the ontology of the problems they solve. The type of the model also determines the implementation - since the model interpreter should create one for a whole group of models of the same type. The model type imposes restrictions on the allowable slots and relations in the model. If a model is created based on a template, then it is a detail of an existing template model and its type is a subtype of this template model. Accordingly, such a model inherits the definition of slots and constraints from the template model. At the same time, new slots and restrictions may be added to the template. For each type of model, a basic model is defined, which reflects the minimum set of concepts and limitations present in all models of this type. Base models are available to other models that activate models of this type. Having a basic model allows you to use different models and methods to solve the same problem. For example, for classification models, the base model consists of two elements - a classified fact (arbitrary type) and a classification (type - classification), presented as a list of categories. The mechanism of construction of correspondence of a fact to a certain category is determined in specific models. The models at the lower level of the hierarchy define how to solve the problem and the additional information to be obtained from the context.

Determination of the relevance of the method of solving the problem by the model. In the context of a predefined base model, specify the model relevance conditions, which are recorded in the "Base model" section.

Identifying options for using the model. Determining the model's use cases involves identifying the operations that the model performs, if the model is multi-functional. For example, a trade transaction model will involve the operations of modelling and evaluating the results, actually conducting the transaction, and searching for the product for the transaction. As a rule, a multifunctional model can perform several operations with one initialized scheme.

Determination of constituent parts of the model. For each use case, essential roles and ontology types that can replace these roles are defined. Determine the connections and restrictions essential for the model. In general, roles, relationships, and constraints form the basis of a model's schema. At the same stage, it is advisable to establish correspondence between the defined model slots and the base model slots according to the predefined model type.

Specification of mechanisms for implementation of operations. Methods of achieving the goal, inputs and outputs are defined. Specify constraints on inputs and outputs.

Definition of prerequisites. Prerequisites are defined for each use case - as requirements for the types of input data and the permissible ranges of their values.

Assignment of initialization conditions and procedures. Using the results obtained in the previous stages, for each use case, a set of input data is analysed, determining configurations of mandatory and optional data. For example, in the trade transaction model for the use case *Search for a product seller*, the required slots are *Buyer*, *Money*, *Product*, and the *Seller* is determined during the model operation. All slots must be filled in for the *Make a Transaction* use case. To reduce the dependency of initialization success on the state of the fact base, default values are specified for missing values whenever possible. An alternative is to define relevant data retrieval models with the task of mapping the model slots to the slots of the base retrieval model.

Specification of conditions for model verification. As a rule, verification rules are associated with models that determine the way to solve a certain problem when creating a model. Verification conditions are used to check the integrity of the model if it changes. For example, if a classification model is defined using a given scale, then the verification rules check whether all intervals of the scale collectively overlap the definition area of the classification parameter and whether there is no overlapping of the intervals.

Development or modification of the model interpreter. The completed model scheme is passed to the programmer, who creates a new one to execute the model, or defines one of the existing model interpreters.

Model testing. The completed model is tested and validated for different states of the fact base and for different use cases in order to achieve the goal. Identified shortcomings are the basis for changing and retesting the model. The proposed architecture of the simulation system is the basis of the ISBA intelligent controlled model simulation tool set.

References

[1] P. Kravets, V. Lytvyn, V. Vysotska, Y. Burov, I. Andrusyak, Game Task of Ontological Project Coverage, CEUR Workshop Proceedings 2851 (2021) 344-355.

[2] Y. Burov, The Introduction of Attentional Mechanism in the Situational Awareness Process, CEUR Workshop Proceedings 3171 (2022) 1076-1086.

[3] Core and hierarchical role based access. control (RBAC) profile of XACML v2.0, URL: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.

[4] T. Hovorushchenko, Y. Hnatchuk, A. Herts, O. Onyshko, Intelligent Information Technology for Supporting the Medical Decision-Making Considering the Legal Basis, CEUR Workshop Proceedings 2853 (2021) 72-82.

[5] Y. Burov, A. Horodetska, M. Bublyk, M. Nashkerska, V. Vysotska, Intellectual Tourist Service with the Situation Context Processing, in: International Conference on New Trends in Languages, Literature and Social Communications, 2021, May, pp. 233-243. Atlantis Press.

[6] D. Allemang, J. Hendler, Semantic web for the working ontologist: effective modeling in RDFS and OWL. Elsevier, 2011.

[7] F. H. Abanda, J. H. Tah, R. Keivani, Trends in built environment semantic Web applications: Where are we today?, Expert Systems with Applications 40(14) (2013) 5563-5577.

[8] M. Hepp, K. Siorpaes, D.Bachlechner, Towards the semantic web in e-tourism: lack of semantics or lack of content?, 3rd European Semantic Web Conference (ESWC 2006).

[9] M. Hepp, K. Siorpaes, D.Bachlechner, Towards the semantic web in e-tourism: can annotation do the trick?. (2006). URL: www.heppnetz.de.

[10] J. Cardoso, C. Lange, A framework for assessing strategies and technologies for dynamic packaging applications in e-tourism, Information Technology & Tourism 9(1) (2007) 27-44.

[11] D. Raz, A. T. Juhola, J. Serrat-Fernandez, A. Galis, Fast and efficient context-aware services. John Wiley & Sons, 2006.

[12] J. Krumm, (Ed.). Ubiquitous computing fundamentals. CRC Press, 2018.

[13] Introducing google places, URL: http://googleblog.blogspot.com/2010/04/introducing-google-places.html.

[14] T. Hovorushchenko, A. Herts, Y. Hnatchuk, Concept of Intelligent Decision Support System in the Legal Regulation of the Surrogate Motherhood, CEUR Workshop Proceedings 2488 (2019) 57-68.

[15] L. Niu, J. Lu, G. Zhang, Cognition-driven decision support for business intelligence. Models, Techniques, Systems and Applications. Studies in Computational Intelligence, Springer, Berlin, (2009) 4-5.

[16] B. Hutchinson, et al., Towards accountability for machine learning datasets: Practices from software engineering and infrastructure, in: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021, March, pp. 560-575.
[17] A. Osterwalder, The business model ontology a proposition in a design science approach, Doctoral dissertation, Université de Lausanne, 2004.
[18] R. Turner, Gower handbook of project management. Routledge, 2016.
[19] S. J. Mellor, M. J. Balcer, Executable UML: a foundation for model-driven architecture. Addison-Wesley Professional, 2002.
[20] B. Goertzel, C. Pennachin, The Novamente artificial intelligence engine, Artificial general intelligence, pp. 63-129. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.
[21] D. F. Ferraiolo, J. F. Barkley, D. R. Kuhn, A role-based access control model and reference implementation within a corporate intranet, ACM Transactions on Information and System Security (TISSEC) 2(1) (1999) 34-64.
[22] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, R. Chandramouli, Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC) 4(3) (2001) 224-274.
[23] Beyond Roles: A Practical Approach to Enterprise User Provisioning, URL: http://www.idsynch.com/docs/beyond-roles.html.
[24] U. R. Saxena, T. Alam, Provisioning trust-oriented role-based access control for maintaining data integrity in cloud. International Journal of System Assurance Engineering and Management 14(6) (2023) 2559-2578.
[25] M. B. Gunjal, V. R. Sonawane, Multi authority access control mechanism for role based access control for data security in the cloud environment. International Journal of Intelligent Systems and Applications in Engineering 11(2s) (2023) 250-264.
[26] I. Zavuschak, Y. Burov, The Context of Operations as the basis for the Construction of Ontologies of Employment Processes. International Journal of Modern Education and Computer Science 9(11) (2017) 13.
[27] V. Lytvyn, V. Vysotska, Y. Burov, A. Demchuk, Architectural ontology designed for intellectual analysis of e-tourism resources, in: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 2018, pp. 335-338.
[28] N. Antonyuk, A. Vysotsky, V. Vysotska, V. Lytvyn, Y. Burov, A. Demchuk, I. Lyudkevych, L. Chyrun, S. Chyrun, I. Bobyk, Consolidated Information Web Resource for Online Tourism Based on Data Integration and Geolocation, in: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 2019, pp. 15-20.
[29] A. Vysotsky, V. Lytvyn, V. Vysotska, D. Dosyn, I. Lyudkevych, N. Antonyuk, O. Naum, A. Vysotskyi, L. Chyrun, O. Slyusarchuk, Online Tourism System for Proposals Formation to User Based on Data Integration from Various Sources, in: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 2019, pp. 92-97.
[30] A. Berko, V Vysotska., O. Naum, S. Chyrun, V. Panasyuk, N. Borovets, Big data analysis for startup of supporting Ukraine internet tourism, in: Advanced information and communication technologies : proceedings of the 5th IEEE International conference, Lviv, Ukraine, November 21–25, 2023, pp. 164–169.
[31] N. Shpak, I. Novakivskyi, I. Kulyniak, V. Kharchuk, I. Oleksiv, Modeling the Development of the Tourism Industry Using the Koba-Douglas Function for Intelligent Management Systems, CEUR Workshop Proceedings, Vol-3426, 2023, 308-321.
[32] Y. Tverdokhlib, V. Andrunyk, L. Chyrun, L. Chyrun, N. Antonyuk, I. Dyyak, O. Naum, D. Uhryn, V. Basto-Fernandes, Analysis and estimation of popular places in online tourism based on machine learning technology, CEUR Workshop Proceedings, 2020, 2631, pp. 457–470.
[33] N. Antonyuk, et. al., Online Tourism System Development for Searching and Planning Trips with User's Requirements, Advances in Intelligent Systems and Computing IV, Springer Nature Switzerland AG 2020, 1080, (2020) 831-863.
[34] R. Yurynets, Z. Yurynets, M. Denysenko, I. Myshchyshyn, A. Pekhnyk, The Influence of Educational Competencies of the Staff on the Efficiency of Hotel Companies in the Tourism Sector, CEUR Workshop Proceedings, Vol-2870, 2021, pp. 1225-1237.

[35] O. Artemenko, V. Pasichnyk, N. Kunanets, K. Shunevych, Using sentiment text analysis of user reviews in social media for e-tourism mobile recommender systems, CEUR workshop proceedings, Vol-2604, (2020) 259-271.
[36] P. Zhezhnych, O. Markiv, Recognition of tourism documentation fragments from web-page posts. In: 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET, (2018) 948-951.
[37] P. Zhezhnych, O. Markiv, Linguistic comparison quality evaluation of web-site content with tourism documentation objects, Advances in Intelligent Systems and Computing 689, (2018) 656-667.
[38] P. Zhezhnych, O. Markiv, A linguistic method of web-site content comparison with tourism documentation objects. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, (2017) 340-343.
[39] V. Lytvyn, Y. Burov, V. Vysotska, V. Hryhorovych, Knowledge novelty assessment during the automatic development of ontologies, in: Third International Conference on Data Stream Mining & Processing (DSMP), 2020, August, pp. 372-377. IEEE.
[40] Y. Burov, V. Vysotska, V. Lytvyn, L. Chyrun, Software Based on Ontological Tasks Models. Lecture Notes in Data Engineering, Computational Intelligence, and Decision Making, 2022, pp. 608-638. Cham: Springer International Publishing.
[41] J. Yang, et al.,, Social media data analytics for business decision making system to competitive analysis. Information Processing & Management 59(1) (2022) 102751.
[42] M. Fuchs, W. Höpken, M. Lexhagen, Big data analytics for knowledge generation in tourism destinations–A case from Sweden. Journal of destination marketing & management, 3(4), (2014). 198-209.
[43] M. Lee, et al., Artificial intelligence for hospitality big data analytics: developing a prediction model of restaurant review helpfulness for customer decision-making. International Journal of Contemporary Hospitality Management 33(6) (2021) 2117-2136.
[44] C., Marcolin, et al., Business analytics in tourism: Uncovering knowledge from crowds. BAR-Brazilian Administration Review 16 (2019) e180136.
[45] U. Gretzel, Intelligent systems in tourism: A social science perspective. Annals of tourism research 38(3) (2011) 757-779.
[46] S. J. Miah, H. Q. Vu, J. Gammack, M. McGrath, A big data analytics method for tourist behaviour analysis. Information & Management 54(6) (2017) 771-785.
[47] W. Höpken, et al., Business intelligence for cross-process knowledge extraction at tourism destinations. Information Technology & Tourism 15 (2015) 101-130.
[48] S. Babu, S. Subramoniam, Tourism management in internet of things era. Journal of Information Technology & Economic Development 7(1) (2016).
[49] K. Hemachandran, et al. Machine Learning for Business Analytics: Real-Time Data Analysis for Decision-Making. CRC Press, 2022.
[50] S. Akter, et al., Analytics-based decision-making for service systems: A qualitative study and agenda for future research. Inter. Journal of Information Management 48 (2019) 85-95.
[51] Y. Yuan, Decision support system for evaluating English media analytics and the role of visualization in online business development using CRITIC and TOPSIS approaches. Soft Computing (2023) 1-15.
[52] H. Zhang, et al., Big data-assisted social media analytics for business model for business decision making system competitive analysis. Information Processing & Management 59(1) (2022) 102762.
[53] R. A. Hamid, et al., How smart is e-tourism? A systematic review of smart tourism recommendation system applying data management. Computer Science Review 39 (2021) 100337.
[54] O. Boulaalam, et al., Proposal of a big data system based on the recommendation and profiling techniques for an intelligent management of Moroccan tourism. Procedia Computer Science 134 (2018) 346-351.