УДК 519.7:007.52

А.Ю. Шевченко, Е.Л. Лещинская

*Харьковский национальный университет радиоэлектроники, Харьков*

# МОДЕЛЬ РАСПРЕДЕЛЕННОЙ ОНТОЛОГИЧЕСКОЙ БАЗЫ ЗНАНИЙ ДЛЯ ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

*За последнее время значительно увеличилось внимание к онтологическим базам знаний (ОБЗ) при построении интеллектуальных информационных систем. Это обуславливается рядом их преимуществ перед традиционно используемыми реляционными базами данных (БД), главным из которых является возможность динамически менять набор содержащихся в хранилище сущностей без изменения его внутренней структуры. Но ОБЗ также не лишены и недостатков. Одним из них является низкая производительность вычислительных систем, использующих ОБЗ. В данной статье предложена модель распределенной ОБЗ. Такая модель позволяет распараллелить вычислительные процессы, проходящие в онтологической базе знаний, и, как следствие, увеличить быстродействие, защищенность и отказоустойчивость всей системы в целом.*

***Ключевые слова:*** *онтология, распределенная база знаний, OWL, концепт, ABox, TBox, логический вывод, web-сервис.*

## Введение

**Постановка проблемы.** Развитие методов разработки программного обеспечения привело к увеличению внимания к интеллектуальным системам, основанным на онтологических базах знаний (ОБЗ). Как одно из преимуществ применения онтологий в информационных системах можно отметить возможность добавлять не предусмотренные при первоначальном проектировании онтологии знания без изменения единой для всех онтологий мета-структуры ОБЗ. Причем все запросы к любой ОБЗ полностью опираются на её универсальную мета-структуру, и это также обеспечивает удобство в применении онтологий. Также онтология содержит большое количество информации о структуре конкретной базы знаний (объектах и классах, хранящихся в онтологии). При каждом выполнении запроса к конкретной онтологии проводится предварительная обработка её структуры, например вычисление всех возможных отношений типа "подкласс" между всеми парами классов, и только затем происходит вычисление запроса, которое сводится к проверке согласованности ОБЗ путем логического вывода и поиска противоречий. Очевидно, что скорость работы такой системы будет значительно ниже, чем у классических баз данных. Описанные проблемы вызывают необходимость искать способы повышения производительности БЗ интеллектуальных информационных систем без отказа от их онтологической структуры.

**Целью настоящего исследования** является поиск решения обозначенной проблемы путем создания распределенной ОБЗ. Объем информации на один узел такой системы по сравнению с общим объемом информации во всей распределенной ОБЗ должен быть значительно меньше и вычислительные процессы могут проходить в разных узлах параллельно. Следовательно, суммарное время отклика такой системы может быть приближенно к традиционным базам данных. Для реализации такого метода вычислений предложена модель распределенной ОБЗ, для глобальных информационных систем, содержащих элементы, соединённые ненадёжными информационными каналами. Такая ОБЗ позволяет создавать распределенные онтологические системы, более производительные, защищенные и отказоустойчивые, чем их локальные аналоги.

**Анализ последних исследований и публикаций.** Существует около двух десятков развивающихся систем управления ОБЗ. Среди них наиболее популярными являются Protégé, KAON2, Sesame, IBM SHER, Joseki Jena, Oracle Spatial и проч. На данный момент попытки решить проблему увеличения производительности онтологических хранилищ делаются многими из них. В качестве решений исследуются несколько типовых подходов. Рассмотрим каждый из них детальнее.

Онтология содержит два различных типа информации: а) перечень классов онтологии, перечень свойств, информация о взаимоотношении элементов этих классов (знания характеризующий структуру онтологии), в предыдущих стандартах W3C этот тип знаний был выделен в отдельный стандарт RDFS [1], в терминологии OWL эти две составляющие онтологии носят названия TBox и RBox, для классов и свойств соответственно; б) перечень информации, привязанной к реальным объектам, эта информация полностью соответствует общей структуре онтологии, этот тип информации в предыдущих стандартах W3C был выделен в стандарт RDF [2], в терминологии OWL эта составляющая онтологии носит название ABox. Пример элементов TBox, RBox и ABox фрагмента онтологии образовательного процесса показан на рис. 1.

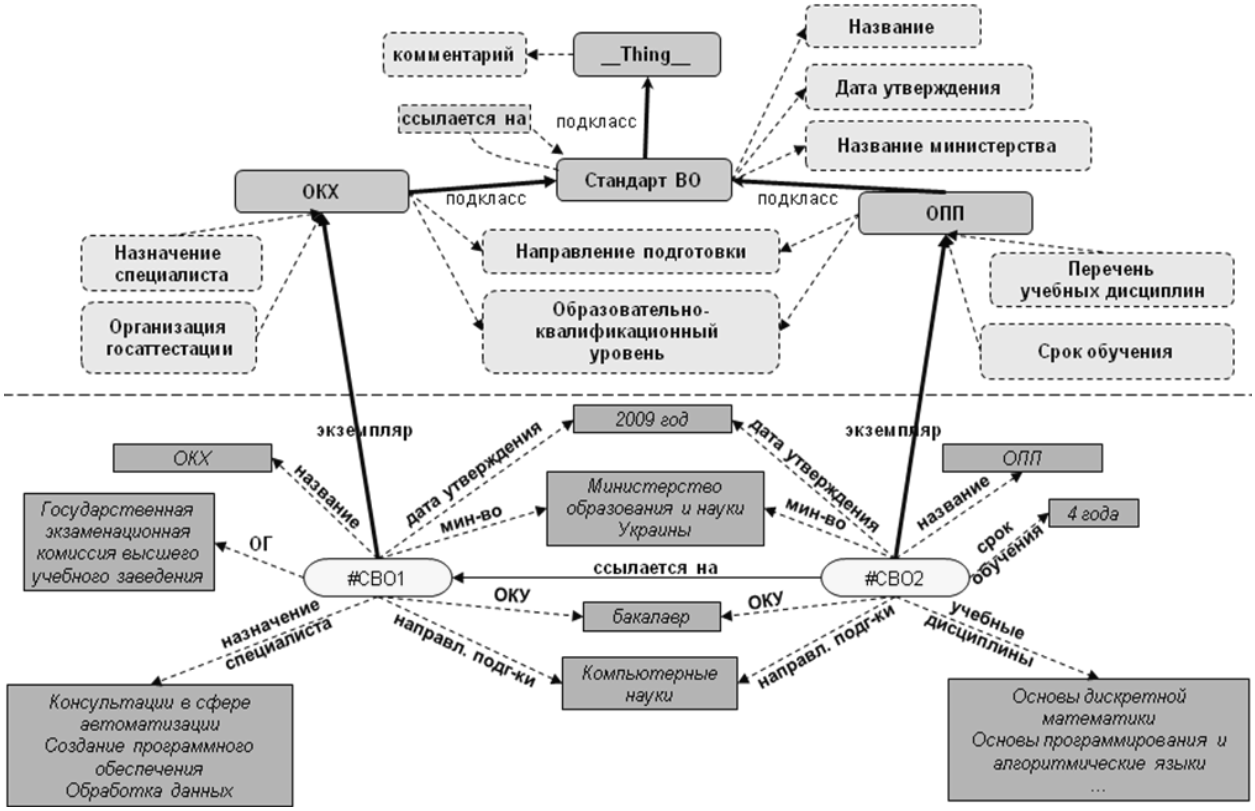


Рис. 1. Фрагмент онтологии образовательного процесса (в верхней части изображена структурная часть онтологии, в нижней приведено несколько примеров описания реальных объектов)

В рамках первого подхода к оптимизации времени выполнения запросов к онтологии развиваются алгоритмы трансляции подмножеств DL-моделей (таких как SHIQ и др.) и запросов к ним в дизъюнктивный Datalog [3]. Это позволяет ускорять логический вывод в онтологических системах за счет применения практически проверенных оптимизационных техник, разработанных для дедуктивных БД, например, индексация, magic sets и join-order optimization. Так, например, поступили разработчики KAON2 [4].

В рамках второго подхода развиваются алгоритмы предварительного уменьшения объема ABox, над которыми будет произведен логический вывод, к общему объему ABox всей онтологии в целом. Объемы ABox, обычно, значительно превосходят объемы TBox и RBox и в реальных онтологиях достигают миллионов фактов. В то время как используемый алгоритм логического вывода tableaux algorithms имеет экспоненциальную оценку времени выполнения [5]. В то же время, разработчиками SHER было замечено, что похожие инстансы связаны с другими инстансами одинаково, например, *отец* и *мать*, связанные друг с другом отношением *иметьСупруга*, связаны со своими *детьми* отношением *иметьСына*. На этом наблюдении разработчики SHER основали алгоритм, который строит сокращенный ABox A′ из оригинального ABox A, путем агрегирования похожих инстансов и утверждений о них [6]. Таким образом вывод над A′ изолирует малую релевантную порцию A, необходимую для получения корректного ответа. Кроме того, поскольку подавляющее большинство систем управления ОБЗ реализуются поверх систем управления реляционными БД, алгоритмы выполнения запросов к ОБЗ могут напрямую использовать интегрированные в них механизмы выполнения запросов к реляционным наборам данных со всеми реализованными в них алгоритмами оптимизации, наподобие, например, индексирования с помощью $B^+$ деревьев [7]. Полезным также является тот факт, что A′ может быть вычислен всего один раз, но повторно использоваться при этом для выполнения последующих запросов без перевычисления.

## Сокращение объемов онтологии путем создания распределенных ОБЗ

Сравнение описанных выше подходов позволяет сделать вывод, что идея уменьшения объемов онтологии, участвующей в обработке конкретного запроса, еще не исчерпала своих возможностей. Например, онтология в SHER, аналогично другим системам управления ОБЗ, построенным поверх реляционных СУБД, расположена в едином хранилище, несмотря на то, что поддержка распределенных БД давно стала классической возможностью полнофункциональных реляционных СУБД, таких как MS SQL Server и Oracle. Наличие распределенной ОБЗ позволяет создать стационарно сокращенные ABox и устраняет необходимость вычислять их перед об-

работкой конкретного запроса. Такое разделение является рациональным, когда запросы к одному сокращенному ABox значительно доминируют среди всех запросов одного клиентского приложения к ОБЗ. При такой организации ОБЗ вычисление ABox путем интеграции знаний из нескольких сокращенных ABox по-прежнему должно оставаться возможным.

Распределенная таким образом ОБЗ идеально подходит для крупномасштабных интеллектуальных информационных систем крупных корпораций, в которых, при наличии общей интегрированной БЗ, отдельным подразделениям чаще всего требуются для корректной работы лишь обособленные её части.

К сожалению, онтологическое ПО, имеющее исследовательскую OpenSource основу в большинстве своем построено поверх таких бесплатных СУБД как MySQL и PostgreSQL, не поддерживающих возможности распределения. Поэтому для осуществления возможности создания распределенных ОБЗ необходимо разрабатывать собственную теоретическую базу.

Глобальные информационные системы, как правило, состоят из компонентов, которые могут находиться на достаточно большом расстоянии друг от друга, и даже если эти компоненты соединены через высокоскоростные каналы, скорость их взаимодействия значительно снижается. Поэтому, выбирая структуру распределенной онтологической базы, мы руководствовались следующим требованием: необходимо обеспечивать возможность работы не только в рамках общей онтологической базы знаний, но и в режиме локальной системы, где фрагмент онтологии, находящийся на удаленном клиенте, представляется как полностью независимая онтологическая структура и может автономно функционировать. Эти два режима могут быть совмещены и разделяться только характером запроса к серверу онтологической базы.

## Выбор общего принципа распределения элементов онтологической базы знаний

При создании распределенной БЗ первичным является определение принципа распределения. Для БЗ, построенных на основе онтологической модели представления знаний, необходимо задаться правилами, по которым между различными узлами системы управления ОБЗ будут распределяться элементы из подмножеств TBox, RBox и ABox. Здесь возможны два варианта: а) схема модели знаний одинакова на всех узлах распределенной БЗ; б) схема модели знаний на различных узлах распределенной БЗ может различаться. В первом случае разделению подлежит только ABox, во втором TBox и RBox также могут быть распределенными. Необходимость разделять элементы подмножеств TBox и RBox требует возможности определения их логически независимых «модулей». Под логической независимостью понимается, что логический вывод факта, в котором фигурируют только классы из одного модуля, эквивалентен его выводу из общей онтологии (т.е. либо следует, либо не следует). Осуществимость и способы эффективного решения этой проблемы, называемой ontology modularity, активно обсуждаются исследователями [8], но авторам пока не известно о существовании её production quality решений.

В связи с описанными особенностями в данном исследовании выбрана первая модель распределения.

Для реализации распределенной ОБЗ необходимо задаться признаком, определяющим, какой из узлов сети содержит необходимое для выполнения запроса подмножество элементов. Одними из возможных вариантов решения данной проблемы может стать определение двух атрибутов у каждого класса: *находитсяВ* и *являетсяЧастью*. Оба атрибута должны иметь множественность 0..1.

Отношение *являетсяЧастью* считается одним из базовых для множества моделей представления знаний и в описанном варианте позволяет задавать иерархические композиции объектов.

Отношение *находитсяВ* с помощью URI задает принадлежность экземпляра одному из узлов распределенной онтологической сети, но необходимости задавать его явно в каждом элементе ABox нет. Явно данное свойство необходимо задавать только для верхнего объекта структурных иерархий. Для остальных элементов значение свойства *находитсяВ* возможно проводить на основе следующего элементарного логического правила:

$$\text{находитсяВ}(x, y) \wedge \text{являетсяЧастью}(z, x) \rightarrow \text{находитсяВ}(z, y).$$

Описанный принцип разделения является естественным, если элементы разрабатываемой распределенной онтологической системы находятся в различных организациях, имеющих общее подчинение.

## Архитектурные решения для системы управления распределенной ОБЗ

Архитектурная модель системы управления распределенной ОБЗ может представлять собой набор равноправных серверов, на каждом из которых располагается доступное для удаленных клиентов программное обеспечение системы управления.

Для географически распределенной системы важной особенностью является возможность одноразового изменения продублированного на всех серверах TBox и RBox, а также его автоматическая синхронизация. Для решения этой задачи один из серверов системы необходимо назначить синхронизирующим. Модификация TBox и RBox на клиентах возможна только в случае, если изменения были сделаны в синхронизаторе онтологической базы знаний. Общая структура взаимодействия узлов распределенной ОБЗ представлена на рис. 2.

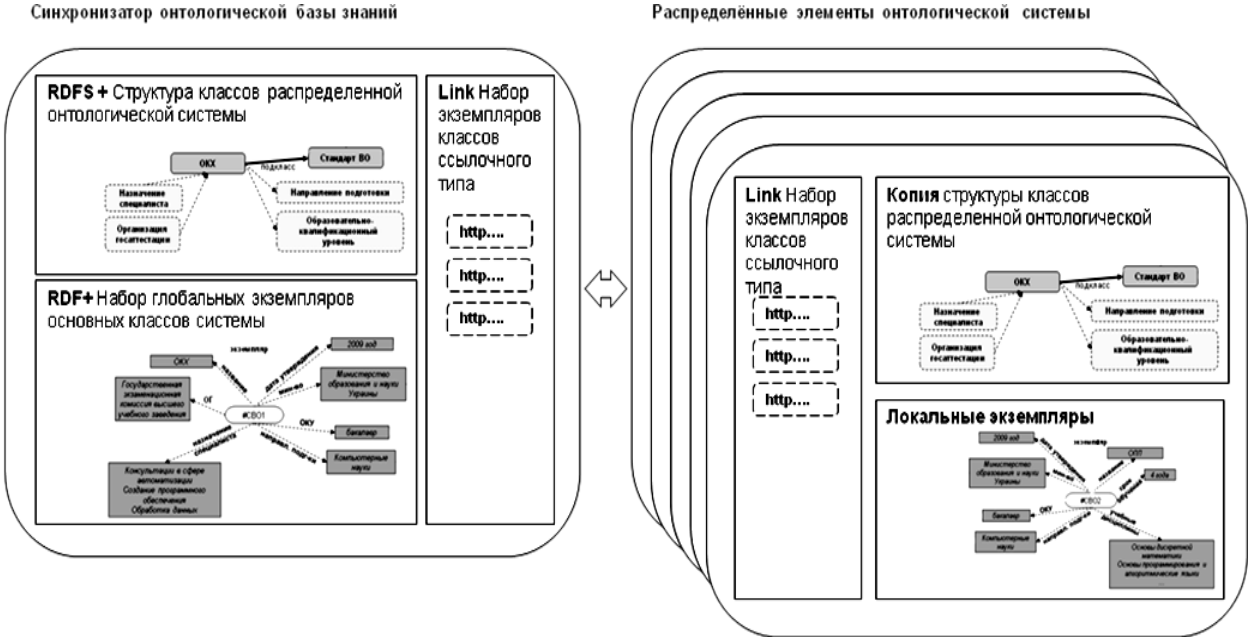


Рис. 2. Принцип построения распределенной онтологии,
основанной на функциональном разделении онтологической базы знаний

Для упрощения процесса модернизации и развития системы управления распределенной ОБЗ с архитектурной т.з. был применен шаблон проектирования Model-View-Controller (**MVC**).

Шаблон проектирования MVC предполагает разделение данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер – таким образом, что модификация каждого компонента может осуществляться независимо. Модель (**Model**) предоставляет данные предметной области представлению и реагирует на команды контроллера, изменяя свое состояние. Представление (**View**) отвечает за взаимодействие с клиентом системы, транслируя его запросы в запросы контроллеру. Контроллер (**Controller**) интерпретирует действия клиента, оповещая модель о необходимости изменений.

В случае с распределенной ОБЗ моделью является сама онтология. Роль контроллера выполняют модуль логического вывода Pellet и Sezam – API доступа к данным онтологии.

Представление предоставляет программный интерфейс клиентским приложениям, который кроме стандартных функций системы управления БЗ реализует также набор системных функции для синхронизации TBox.

## Организация взаимодействия между компонентами распределенной ОБЗ посредством использования Web-сервисов

Предложенная в статье организация распределенной ОБЗ позволит в значительной степени повысить уровень быстродействия и отказоустойчивости онтологической системы.

Важными при проектировании распределенной ОБЗ являются также вопросы обеспечения безопасности передачи информации, а также невысокие, желательно, однотипные трудозатраты адаптации существующих клиентов к новой ОБЗ.

Для решения этих вопросов уровень представления в интерпретации модели MVC, описанной выше, реализован как Web-сервис.

Web-сервис – программная система, идентифицируемая строкой URI, чьи общедоступные интерфейсы определены на языке XML. Описание этой программной системы может быть найдено другими программными системами, которые могут взаимодействовать с ней согласно этому описанию посредством сообщений, основанных на XML и передаваемых с помощью интернет-протоколов. Web-служба является единицей модульности при использовании сервисно-ориентированной архитектуры приложения.

Web-сервис обеспечивает нужный уровень безопасности и целостности информации. Каждая команда Web-сервиса является завершенной транзакцией и в случае обрыва связи гарантирует сохранение целостности БЗ.

В каждый из узлов системы управления распределённой ОБЗ одновременно будут встроены как сам Web-сервис, реализующий синхронизацию TBox, так и небольшой клиент, отвечающий за подтягивание, в случае необходимости, недостающих элементов ABox с соседних узлов.

Если для выполнения действия необходимо активизировать другой элемент распределенной сис-

темы, то используется клиент для отправки вызова и получения ответа от удаленного компонента, если производится ожидание вызова от внешних клиентских систем, то в таком случае используется Web-сервис.

Организация взаимодействия между клиентом и Web-сервисом осуществляется при помощи XML-подобного языка, соответствующего протоколу XML-RPC. При передаче фрагментов онтологии по сети, например, между обычным и синхронизирующим узлом или недостающего фрагмента ABox, используется механизм сериализации объектов.

## Выводы

В статье предложена модель распределенной ОБЗ и системы управления ею, использование которых позволит создавать надежные распределенные интеллектуальные приложения.

Выгодной особенностью предложенной модели является её принцип распределения, позволивший сохранить полную функциональность каждого отдельного компонента онтологической системы, которые могут, в случае необходимости обрабатывать запросы автономно. Поскольку признаки, управляющие методом поиска удаленных объектов онтологии, заложены в структуру самой онтологии, то реализация предложенной модели распределенной ОБЗ возможна с использованием любого из существующих API систем управления ОБЗ.

Предложенная модель распределенной ОБЗ идеально подходит для крупномасштабных интеллектуальных информационных систем, элементы которых предназначены для различных организаций имеющих общее подчинение.

## Список литературы


*1. Broekstra J. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema / J. Broekstra, A. Kampman, F. Harmelen //* LNCS:*The Semantic Web. – ISWC-2002. – Springer Berlin, 2002. – Vol. 2342. – P. 53-68.*

*2. Noy N.F. Creating Semantic Web Contents with Protege-2000 / N.F. Noy, M. Sintek, S. Decker, M. Crubezy, R. Fergerson, M.A Musen // IEEE Intelligent Systems. – IEEE Educational Activities Departmen,2001. – Vol. 16 , Iss. 2. – P. 60-71.*

*3. Hustadt U. Reducing SHIQ description logic to disjunctive datalog programs / U. Hustadt, B. Motik, U. Sattler // Journal of Automated Reasoning. – Springer-Verlag Inc., 2001. – Vol. 39, Iss. 3. – P. 351-384.*

*4. Motik B. Optimizing query answering in description logics using disjunctive deductive databases / B. Motik, R. Volz, A. Maedche // Theory and Practice of Logic Programming. – Cambridge University Press, 2008. – Vol. 8, Iss. 3. – P. 393-409.*

*5. Horrocks I. Practical Reasoning for Very Expressive Description Logics / I. Horrocks, U. Sattler, S. Tobies // Logic Journal of the IGPL. – Elsevier Science Publishers, 2001. – Vol. 8, Iss.3. – P. 239-263.*

*6. Fokoue A. The Summary Abox: Cutting Ontologies Down to Size / A. Fokoue, A. Kershenbaum, L. Ma, E. Schonberg, K. Srinivas // LNCS: The Semantic Web - ISWC 2006. – Springer Berlin, 2006. – Vol. 4273. – P. 343-356.*

*7. Seltzer M. Beyond Relational Databases: There is more to data access than SQL / M. Seltzer // Communications of the ACM. – ACM. – Vol. 51, Iss. 7 – 2008. – P. 52-58.*

*8. Pathak J. Survey of modular ontology techniques and their applications in the biomedical domain / J. Pathak, M.T. Johnson, C.G. Chute // Integrated Computer-Aided Engineering. – ACM, 2009. – Vol. 16 , iss. 3. – P. 225-242.*


*Поступила в редколлегию 30.06.2010*

**Рецензент**: д-р физ.-мат. наук, доц. А.А. Галуза, Национальный технический университет «ХПИ», Харьков.

### МОДЕЛЬ РОЗПОДІЛЕНОЇ ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ ДЛЯ ІНТЕЛЕКТУАЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

О.Ю. Шевченко, О.Л. Лещинська

*За останній час значно збільшилася увага до онтологічних баз знань (ОБЗ) при побудові інтелектуальних інформаційних систем. Це обумовлено їх перевагами перед традиційно використовуваними реляційними базами даних (БД), головною з яких є можливість динамічно міняти набір суті, що міститься в сховищі, без зміни його внутрішньої структури. Але ОБЗ також не позбавлені і недоліків. Одним з них є низька продуктивність обчислювальних систем, використовуючих ОБЗ. У даній статті запропонована модель розподіленої ОБЗ. Така модель дозволяє розпаралелювати обчислювальні процеси, що проходять в онтологічній базі знань, і, як наслідок, збільшити швидкодію, захищеність і відмовостійкість всієї системи в цілому.*

***Ключові слова:*** *онтологія, розподілена база знань, OWL, концепт, ABox, TBox, логічний вивід, web-сервіс.*

### DISTRIBUTED ONTOLOGICAL KNOWLEDGE BASE MODEL FOR INTELLIGENT INFORMATION SYSTEMS

A.Yu. Shevchenko, Ye.L Leshchynskaya

*In latter days attention was considerably increased to the ontological bases of knowledges (OBK) at the construction of the intellectual informative systems. It conditioned by a number of their advantages before traditionally in-use relational databases (RD), main from which is possibility dinamically to change the set of the essences contained in a depository without the change of his underlying structure. But OBK also not deprived failings. One of them is the low productivity of the computer systems, utillizings OBZ. The model of up-diffused OBK is offered in this article. Such model allows parallel calculable processes, passing in the ontological base of knowledges, and, as a result, to increase a fast-acting, protected and faulttolerance of all of the system on the whole.*

***Keywords:*** *ontology, up-diffused base of knowledges, OWL, concept, ABox, TBox, inferencing, web-service.*