

ТЕХНОЛОГИЯ ОПЕРАТИВНОГО УПРАВЛЕНИЯ РАСПРЕДЕЛЕННЫМИ ОРГАНИЗАЦИОННЫМИ СИСТЕМАМИ

The paper describes the use of organization theory, theory of active systems and fuzzy set theory for constructing systems that operative manage distributed organizations. Unification models and algorithms for fuzzy data and their use for choosing management of distant objects are presented.

Адаптация социально-экономических организационных систем к условиям формирующейся информационно-коммуникативной среды требует существенной модернизации традиционных методов управления. Еще Конфуций, Платон и Аристотель утверждали, что в «эпоху перемен» выживают и успешно развиваются только те страны, руководство которых обеспечивает адаптацию и усовершенствование структур, поддерживающих жизненный цикл государства.

Основателем теории организационных систем (ТОС) как науки, позволяющей эффективно управлять организациями и предсказывать их поведение, считается немецкий ученый М. Вебер, идеи которого развили К. Боулдинг, Ч. Барнарду, А. Файоль, Р. Мертон, Г. Саймон и др. Современные исследования ОС в трудах Б.З. Мильнера [1], М.Д. Месаровича [2], Л.А. Растригина [3], В.Н. Буркова, Д.А. Новикова [4], В.Б. Тарасова [10] и др. выполняются на стыке смежных научных направлений – теории организации, общей теории систем, общей теории управления, теории распознавания образов, теории коммуникации и др. Соответственно ОС трактуется с разных точек зрения: как иерархическая структура [1], большая система [3], адаптивная динамическая структура [2], активная система [4] и т. д. в зависимости от направления исследований и конкретной научной школы.

В настоящее время традиционные модели ОС под влиянием глобализации существенно изменились, поэтому методы и средства эффективного управления новыми структурами представляют большой научный интерес [1–4]. В данной работе рассмотрен подход к решению задач оперативного управления наиболее распространенных типов организационных систем – территориально распределенных.

Анализ проблемы

Традиционные локальные структуры ОС с вертикальной системой управления постепенно заменяются на гибкие, территориально распределенные организационные системы (РОС) с горизонтальным типом управления, множеством подразделений, количество и расположение которых постоянно меняется. Типичные РОС – это промышленные компании с распределенным производством товаров, торговые фирмы с сотнями филиалов, крупные научные международные проекты, министерства и ведомства, органы регионального управления [5]. Крупнейшей проблемой поддержки жизненного цикла РОС является решение задач оперативного управления [5, 8, 10]. Они возникают стихийно под

влиянием внешних причин, приобретают все больший масштаб и решать их приходится в крайне сжатые сроки. Исходными данными могут служить не только числовые показатели, но и лингвистические переменные (словесное описание проблемной ситуации, мнения экспертов или заинтересованных лиц). Предметная область в таких задачах не определена и строится каждый раз заново [10].

В целом для управления РОС характерно: 1) переменное число задач управления; 2) переменное количество и семантика числовых и лингвистических диагностических показателей; 3) неопределенная предметная область; 4) преобладание оперативных задач; 5) большое количество объектов управления; 6) гибкая организационная структура.

Наиболее эффективным средством решения оперативных задач в современных условиях являются автоматизированные системы управления (АСУ). Рассмотрим возможность их применения для оперативного управления РОС в условиях 1) – 6).

Существующие АСУ можно условно разбить на две группы: системы для решения плановых задач на основе статических предметных областей и локальных баз данных (технологии MRP – CFA – ERP – CRM); системы для решения оперативных задач в случайные моменты времени в соответствии с динамическими предметными областями, формируемыми каждый раз заново на основе информации из корпоративных хранилищ (технологии CALS – CASE – OLAP) [9].

Технологии обеих групп формировались в 1960–2000 гг. в относительно стабильном индустриальном обществе и обладают свойствами, аналогичными ОС того времени: 1) постоянное число задач решаемых задач; 2) постоянные семантика и количество числовых диагностических показателей; 3) известная, хорошо структурированная предметная область; 4) преобладание плановых задач; 5) небольшое количество объектов управления; 6) жесткая архитектура программных средств.

Очевидные противоречия между свойствами РОС и существующими системами управления говорят о необходимости разработки третьей группы АСУ, изначально ориентированных на решение оперативных задач на основе «внешних» предметных областей, позволяющих анализировать объект с точки зрения среды и использования различных типов показателей. При этом главная проблема заключается не только в разнообразии задач, удаленности и большом количестве объектов, но и в необходимости комплексной оценки ОС, совмещенной с синтезом управления для каждого из объектов в минимальное время на основе как числовых, так и лингвистических типов данных [7, 8]. Применение традиционных механизмов управления в таких условиях неэффективно [10].

Комплексный характер указанных проблем требует разработки новых компьютерных технологий на базе синтеза элементов смежных наук: теории организаций, информатики, общей теории управления, теории активных систем, теории нечетких множеств и теории коммуникации.

Постановка задачи

Пусть имеется ОС W , которая функционирует в среде S и состоит из центра $Center$ и n территориально распределенных объектов управления P :

$$W = (Center, P^1, P^2, \dots, P^n).$$

В момент времени t центр W решает задачу управления Z , состоящую в определении состояния $V = V^1, V^2, \dots, V^k$ объектов P^1, P^2, \dots, P^n и выработке для них управления $U = u^1, u^2, u^k$ на основе конечного множества внешних диагностических переменных $X = x^1, x^2, \dots, x^m$ и их значений $\langle X \rangle = \langle x^1 \rangle, \langle x^2 \rangle, \dots, \langle x^m \rangle$.

Необходимо разработать технологию выработки управления U для объектов P за время $t + \Delta$.

Требования к решению: 1) семантика Z определяется средой S ; 2) $n \rightarrow \infty$; 3) количество, тип и значения X, V, U, m, k зависят от семантики Z ; 4) значения X могут носить как цифровой, так и лингвистический характер; 6) $\Delta \rightarrow 0$; 7) время решения инвариантно количеству объектов: $\Delta \approx \text{const}$ при $n \rightarrow \infty$.

Поставленная задача масштабна и технологически сложна, поэтому ее целесообразно декомпозировать на более простые подзадачи.

Наличие центра, множества территориально распределенных объектов управления и необходимость сбора информации о каждом из них свидетельствуют о возможной аналогии с известной в экологии задачей мониторинга [6]. Оценка состояния каждого из удаленных объектов также имеет аналог – задачу комплексного оценивания в теории активных систем [5]. Методы управления объектами в зависимости от их состояния рассматриваются в общей теории управления [3, 9]. Таким образом, задача оперативного управления РОС сводится к решению трех известных задач: синтезу мониторинга, комплексного оценивания и управления. Их эффективное решение требует уточнения некоторых базовых определений [6, 4, 3] с целью их адаптации к специфике РОС:

мониторинг – механизм моделирования, получения и интеграции значений показателей, характеризующих участников РОС:

$$\langle X \rangle = f1(X, V, U);$$

комплексное оценивание – механизм оценки состояния участников РОС на основе результатов мониторинга:

$$V = f2(\langle X \rangle);$$

управление – механизм выработки управлений для участников РОС, соответствующих результатам комплексного оценивания:

$$U = f3(V).$$

Наличие трех автономных составляющих говорит о необходимости разработки общей концепции их совместного решения.

Концепция комплексного решения

В основе деятельности любой ОС лежат процессы [3, 8], а три фундаментальные составляющие дают основание для сведения решения к выполнению трех соответствующих процессов $f1, f2, f3$.

Необходимость организации общения участников РОС, возможность решения базовых задач в рамках автономных процессов свидетельствуют о возможности применения многоагентного подхода, изначально ориентированного на использование инфраструктуры Интернета [10]. Агентов, реализующих процессы $f1, f2, f3$, обозначим как $aMonit, aEstim, aManag$. Для обмена информацией между агентами введем понятие посредника-медиатора и обозначим его как Med . В результате получим концепцию решения, представленную на рис. 1.

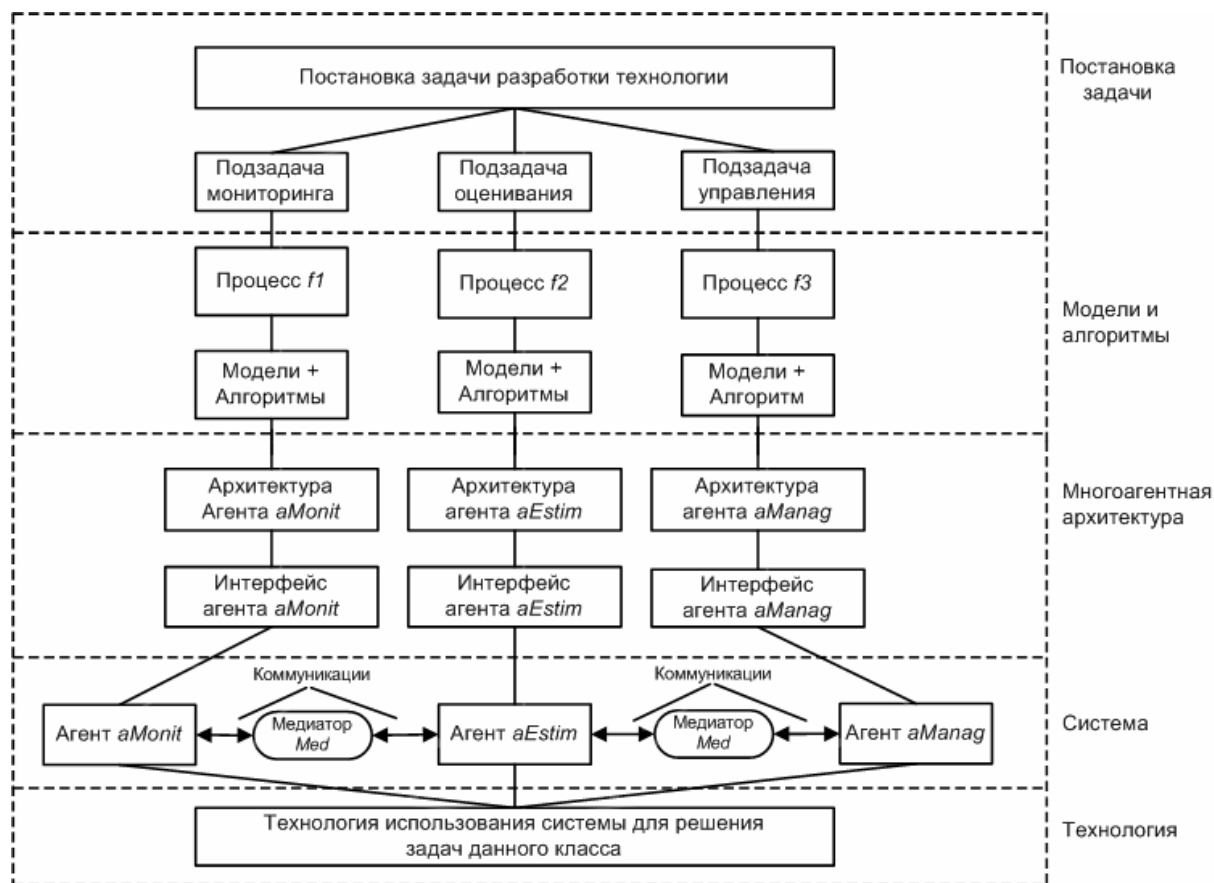


Рис. 1. Процессно-агентная схема решения

Данная схема решения обладает рядом преимуществ: технологии эволюционируют от «естественной основы», представленной определениями, до интерфейсов целевой системы; процессный подход гарантирует соответствие моделей и алгоритмов реалиям управления РОС; сопоставление процессов и агентов решает проблему определения количества и функциональности агентов; прозрачность и полнота схемы позволяет руководителю проекта контролировать любой этап его разработки. Результаты реализации схемы – модели, алгоритмы, интерфейс-программы представлены ниже.

Модели и алгоритмы

Модель предметной области. Решение любой задачи основано на некоторой предметной области (ПрО), содержащей необходимые для решения элементы. В данном случае информационную составляющую модели ПрО можно построить на основе переменных, представленных в постановке задачи:

$$\text{mod}KD = (Z, X, V, U).$$

Синтез элементов модели ПрО и известных элементов идентификации в инфраструктуре Интернета дает возможность построить модели участников реализации процессов f_1, f_2, f_3 . Основное требование к моделям в данной работе – возможность прямого отображения в архитектуру целевой системы и наличие интерфейсов с целью сокращения общего времени разработки.

Модели и алгоритм мониторинга. Согласно схеме (см. рис. 1) в мониторинге участвуют центр, объекты, агент $aMonit$ и посредник mod . Следовательно, их модели необходимо построить в первую очередь.

В качестве модели центра используем результат синтеза типовой модели активной системы [4], элементов модели ПрО, интерактивного средства получения их значений и элементов позиционирования в Интернет:

$$mCenter = (adrCenter, adrP, Z, Q1, Q2, X, V, U, aMonit, aEstim, aManag, Med),$$

где $adrCenter$ – адрес центра в Интернете; $adrP$ – адреса управляемых объектов; $Q1$ – интерактивный механизм формирования центром небогащенной модели ПрО; $Q2$ – интерактивный механизм обогащения модели ПрО на уровне объектов управления.

Модель посредника аналогична, но содержит адрес конкретного объекта, которому будет доставлен экземпляр агента мониторинга с целью получения значений X :

$$mMed = (adrCenter, adrP^i, Z, Q2, X, aMonit).$$

Модель объекта включает элементы позиционирования и фактические значения показателей, требуемых центром:

$$mP = (adrCenter, adrP, Z, \langle X \rangle),$$

где $\langle X \rangle$ – фактические значения показателей X , требуемые центром от объекта P .

Алгоритм реализации процесса мониторинга на базе приведенных моделей состоит из двух логически обособленных частей: построение модели ПрО с помощью механизма $Q1$, входящего в составе $aMonit$, и обогащение модели ПрО с помощью механизма $Q2$, входящего в Med . Обе части формирует агент $aMonit$. В силу сложности и большого объема алгоритмов приведем их универсальную схему, достаточную для понимания:

Шаг 1. Старт агента $aMonit$, активизация $Q1$.

Шаг 2. Формирование идентификатора Z и массива адресов объектов $adrP$.

Шаг 3. Вывод вопросов $\langle Q1 \rangle$ для построения небогащенной модели ПрО.

Шаг 4. Формирование на основе семантики Z значений m, k, n .

Шаг 5. Формирование идентификаторов k возможных состояний V .

Шаг 6. Формирование k, x, m эталонных векторов $[0...1]$ для каждого значения V .

Шаг 7. Формирование k управлений U , соответствующих состояниям V .

Шаг 8. Формирование m идентификаторов X , выделение бифуркационных (влияющих на U) идентификаторов X .

Шаг 9. Формирование k ранжированных вопросов $\langle Q2 \rangle$ с целью получения значений X .

Шаг 10. Формирование функции принадлежности μ^1 для отнесения $\langle X \rangle$ к V .

Шаг 11. Формирование функции μ^2 для коррекции результата работы μ^1 для бифуркационных переменных X .

Шаг 12. Формирование посредника и отправка его экземпляров по адресам $adrP$.

Шаг 13. Получение Med , активизация $Q2$.

Шаг 14. Поочередный вывод вопросов по X с ранжированными ответами.

Шаг 15. Ввод значения $\langle X \rangle$.

Шаг 16. Отправка Med в центр по адресу $adrCenter$.

Шаг 17. Получение Med в центре, интеграция $\langle X \rangle$ в базу данных для решения Z .

Алгоритм формирует ПрО для решения задачи Z и, получив последний экземпляр Med , передает управление агенту оценивания $aEstim$.

Модель и алгоритм оценивания. В рамках процесса комплексного оценивания выполняется задача распознавания текущего состояния объекта V в зависимости от значений диагностических пока-

зателей $\langle X \rangle$ с учетом бифуркационных значений $\langle \underline{X} \rangle$ [4]. Модель агента оценивания $aEstim$ можно описать кортежем:

$$aEstim = \langle Z, P, \langle X \rangle, algRecog, \langle V \rangle \rangle,$$

где $algRecog$ – алгоритм распознавания текущего состояния объекта P на основе $\langle X \rangle$.

Алгоритмов $algRecog$ существует множество, каждый из них выбирается субъективно в зависимости от семантики задачи. Один из простейших, но достаточно эффективных по скорости выполнения вариантов:

Шаг 1. Активизация агента $aEstim$.

Шаг 2. Чтение из базы обогащенной модели ПрО Z, X, V, U по объекту P .

Шаг 3. Распознавание образа состояния объекта P : вычисление расстояния (в зависимости от семантики – Евклидова, Хаусдорфа, Хемминга) вектора $\langle X \rangle$ до каждого из эталонных векторов $\langle V \rangle$.

Шаг 4. Фиксация номера i образа-вектора V , расстояние до которого было минимальным.

Когда состояние всех объектов определено, управление передается агенту выбора управления $aManag$.

Модели и алгоритмы управления. Процесс выбора управления U в зависимости от состояния V объекта P осуществляется на основе заранее установленного соотношения $V \leftrightarrow U$. Соответственно модель агента $aManag$ можно описать кортежем:

$$aManag = \langle Z, P, \langle V \rangle, algChoice, \langle U \rangle \rangle,$$

где $algChoice$ – алгоритм выбора управления для объекта в зависимости от его состояния.

Алгоритмов $algChoice$ существует множество, каждый из них выбирается субъективно в зависимости от семантики задачи. Простейшая схема:

Шаг 1. Активизация агента $aRecog$.

Шаг 2. Чтение из базы данных $Z, P, adrP, \langle V \rangle$.

Шаг 3. Выбор управления U с номером i .

Шаг 4. Коррекция U с помощью μ^2 при наличии бифуркационных значений $\langle \underline{X} \rangle$.

Шаг 5. Отправка управления U объекту по адресу $adrP$.

Универсальная модель агента. Общепринятые универсальные подходы к реализации агентов пока не разработаны, тем не менее некоторую унификацию интерфейса выполнить можно. В данном случае общий интерфейс для всех трех агентов можно моделировать как конструкцию из пяти частей: анализатора среды, организатора диалога, ввода-вывода данных, хранения данных, коммуникации. Один из вариантов модели показан на рис. 2.

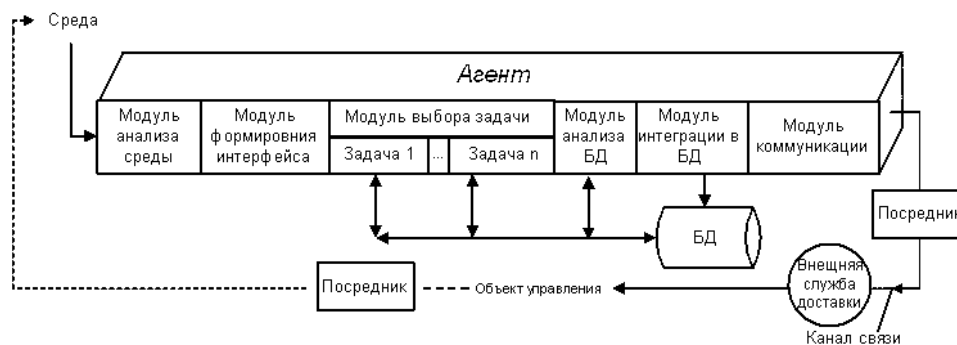


Рис. 2. Универсальная модель интерфейса агента

Разработанные модели и алгоритмы дают основание перейти к построению архитектуры компьютерной системы для их практической реализации.

Архитектура системы

Для комплексной реализации разработанных моделей и алгоритмов предлагается архитектура, представленная на рис. 3, которая включает трех программных агентов, посредника и базу данных, где формируются и аккумулируются элементы ПрО. Архитектура достаточно проста, прозрачна и интуитивно понятна как разработчикам, так и участникам ОС.

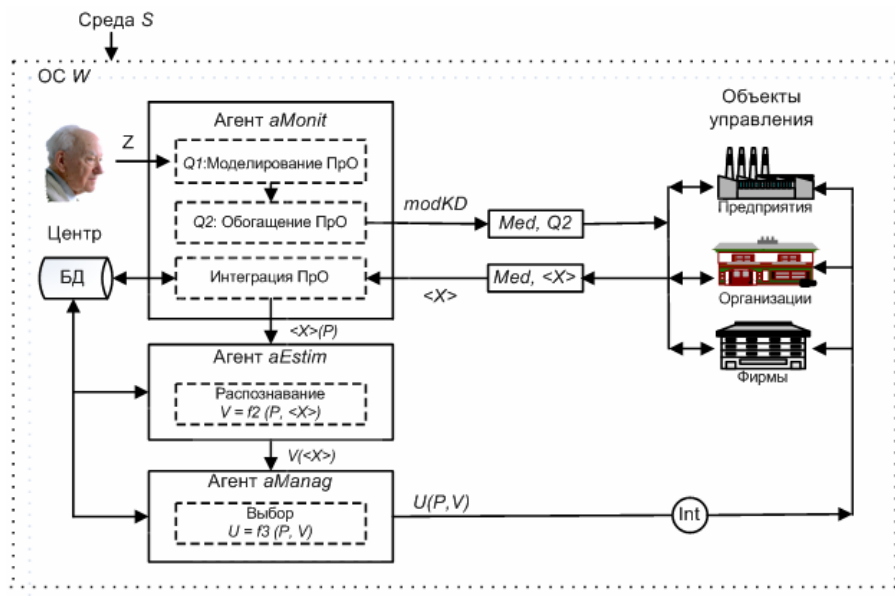


Рис. 3. Архитектура системы

Важнейшей проблемой реализации архитектуры является выбор и обоснование программно-технических средств. Возможны два варианта технологий: Java 2 Platform, Enterprise Edition (J2EE) от SUN и .Net от Microsoft. Они являются конкурентами, хотя позволяют создавать Web-службы (Web services) с практически одинаковой функциональностью.

Платформа фирмы Sun J2EE (J2SE, J2ME) включает инструментальные средства разработчика Forte for Java для операционной среды Solaris, семейство iPlanet – для решения коммерческих задач и др. Java-направление активно поддерживает фирма IBM (WebSphere Application Server). Основным преимуществом J2EE считается защищенность программных продуктов и переносимость на другие платформы, основные недостатки – ориентация на серверную часть, один язык программирования Java, невысокая скорость выполнения, отсутствие пользовательских value-типов.

Главное преимущество .Net заключается в возможности создания web-приложений на разных языках, включая C++, C#, Visual Basic, Fortran, Cobol. Кроме того, глубокая программно-модульная интеграция с Windows обеспечивает высокую скорость выполнения приложений. Главный недостаток – быстрая эволюция, не позволяющая объективно оценить качество предлагаемых технологий.

По мнению авторов, при наличии опыта работы и соответствующих библиотек программ на Java лучше оставаться на этой платформе. Если начинать разработку «с нуля», следует ориентироваться на Microsoft, продукты которой преобладают на рынке IT. В этом случае оптимальным можно считать следующий комплекс технологий: Windows Server 2008; СУБД – Microsoft SQL Server 2008; web-сервер – Microsoft IIS 7.0; язык разработки – C#; среда выполнения – .NET Framework 3.5; среда разработки – Microsoft Visual Studio 2008; доступ к данным – ADO.NET; стандарт структуризации при передаче данных – XML.

Реализация архитектуры, технология, опыт применения

Представленная на рис. 3 архитектура реализована как библиотека программных агентов, используемых для решения прикладных задач. При реализации универсальной модели агента (см. рис. 2) применялся стиль ribbon, т. е. для выбора задач и управления решением используется лента и панель быстрого доступа. Типичный пример интерфейса этого типа представлен на рис. 4.

Технология использования библиотеки агентов достаточно проста:

- 1) появление новой задачи Z ;
- 2) старт в центре агента *aMonit*, активизация механизма $Q1$;
- 3) формирование небогатой модели ПрО по задаче Z ;
- 4) формирование посредника, отправка экземпляра *Med* по сети объектам;
- 5) получение *Med* на объекте, старт механизма $Q2$;
- 6) обогащение ПрО, отправка *Med* в центр;
- 7) интеграция данных всех экземпляров *Med* в базу данных центра;
- 8) автоматический анализ полноты базы данных;

- 9) автоматическая активизация агента оценивания *aEstim*;
- 10) распознавание состояния каждого из объектов;
- 11) автоматическая активизация агента синтеза управления *aManag*;
- 12) выбор управления для каждого из объектов;
- 13) печать отчета центру по результатам решения *Z*;
- 14) отправка управления каждому объекту.

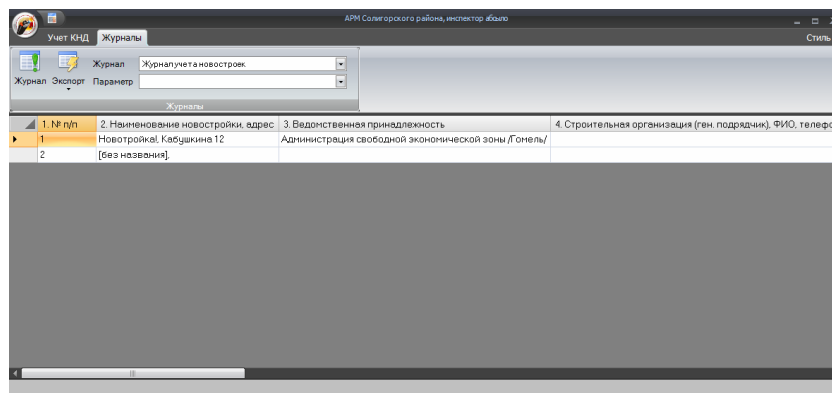


Рис. 4. Пример реализации универсальной модели интерфейса

Благодаря прозрачности технологии ее освоение неподготовленным пользователем не занимает более одного дня.

Рассмотренные в статье теоретические результаты применялись при реализации ряда государственных программ Республики Беларусь, включая «Информационные технологии», «Научные основы информационных технологий и систем», «Защита от чрезвычайных ситуаций». В целом описанные решения, включая принципиальное разделение технологии на этапы синтеза мониторинга, оценивания и управления, а также применение простых жестких моделей с неограниченными возможностями уточнения их элементов, оказались достаточно эффективными. В частности, наличие DLL-библиотеки методов распознавания и открытых структур агентов позволяет подбирать методы, соответствующие специфике решаемой задачи.

* * *

Таким образом, нами рассмотрены проблемы разработки систем для решения оперативных задач оперативного управления распределенными организационными системами. Показано, что для них характерны минимальное время решения, большое количество удаленных объектов управления и нечеткая исходная информация. Обоснована необходимость разработки технологии, инвариантной количеству объектов. В качестве теоретической основы предложено использовать теорию активных систем, дополненную элементами теории нечетких множеств и коммуникации.

В итоге проведенных исследований получены следующие результаты:

- сформулирована задача построения технологии для оперативного управления территориально распределенными организациями на основе нечеткой информации;
- обоснована целесообразность поиска решения на основе синтеза элементов мониторинга, комплексного оценивания и управления;
- построены модели: центра, объектов управления, посредника для обмена информацией;
- разработаны алгоритмы распознавания состояния объектов на основе нечетких данных и выбора соответствующего управления;
- разработаны программные агенты для практической реализации моделей и алгоритмов;
- разработана многоагентная архитектура и компьютерная технология для реализации разработанных моделей и алгоритмов, инвариантная количеству объектов;
- обобщен опыт применения технологии для решения прикладных задач.

1. Мильнер Б.З. Теория организации. М., 2004.
2. Месарович М.Д., Мако Д., Такахага И. Теория иерархических многоуровневых систем. М., 1973.
3. Растрингин Л.А. Адаптация сложных систем. Рига, 1981.
4. Бурков В.Н., Новиков Д.А. Теория активных систем: состояние и перспективы. М., 1999.
5. Иванов Н.П. Мировая экономика и международные отношения. 2000. № 2. С. 18.
6. Мониторинг окружающей среды: Сб. тр. 2001.

7. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. М., 1976.
8. Краснопрошин В.В., Шаках Г., Вальвачев А.Н. // Информатика. 2004. № 3. С. 49.
9. Клир Дж. Системология. Автоматизация решения системных задач. М., 1990.
10. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям. М., 2002.

Поступила в редакцию 27.11.08.

Виктор Владимирович Краснопрошин – доктор технических наук, профессор, заведующий кафедрой математического обеспечения автоматизированных систем управления.

Александр Николаевич Вальвачев – старший научный сотрудник кафедры математического обеспечения автоматизированных систем управления.