

# ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

## INTELLIGENT SYSTEMS AND TECHNOLOGIES

УДК 004.023/004.825

**А. Б. Сорокин**, канд. техн. наук, доц., e-mail: ab\_sorokin@mail.ru,  
**Д. А. Лобанов**, магистрант, e-mail: tty66@mail.ru,  
Московский технологический университет (МИРЭА)

### Концептуальное проектирование интеллектуальных систем

*Рассмотрена методика выделения концептуальных планов из структуры акта деятельности для проектирования систем, основанных на знаниях. На графическом уровне предложены четыре плановых представления: функциональной структуры, процессов, контекста и закономерностей. Данные планы соотносятся с определенными методами искусственного интеллекта. Реализован программный комплекс для выделения этих представлений в виде текстовых файлов.*

**Ключевые слова:** планы концептуальной структуры акта деятельности, план функциональной структуры, план процессов, план контекста, план закономерностей

#### Введение

При возрастании сложности процессов управления человеческий разум не в состоянии познать многомерность проблемы, что приводит к упрощению реальности человеком и способствует принятию неадекватных решений. Наиболее упрощенной картиной реального мира является эмоциональное мышление, которое поддается всякого рода манипуляциям, ориентирует на краткосрочный результат и приводит к "легким" решениям, не требующим доказательств и логического обоснования.

Проектирование систем, основанных на знаниях (систем искусственного интеллекта — далее СИИ) и, соответственно, их комбинаций должно основываться на постоянном концептуальном осмыслении управленческих процессов. При этом центральным компонентом таких систем является база знаний экспертов, которая может быть извлечена из концептуальной структуры акта деятельности. Таким образом, реализуется экспертная система продукционного типа [1].

Элементы экспертных систем не только используются практически в каждом программном продукте, но и тесно связаны с другими подходами искусственного интеллекта. Примером могут служить интегрированные экспертные системы [2]. Поскольку такое сближение приводит к идее выделения из концептуальной структуры акта деятельности других представлений для проектирования СИИ, данную статью можно назвать продолжением работы [1].

#### 1. Постановка задачи для выделения концептуальных планов

Общеизвестно, что в CASE-технологии (*Computer-Aided Software Engineering* — автоматизированная разработка программного обеспечения) используют два принципиально разных подхода к проектированию: структурный, основанный на общелогическом методе познания — анализе контекстного представления (контекстной диаграммы), и объектно-ориентированный, основанный на синтезе различных диаграмм для создания UML-модели (*Unified Modeling Language* — унифицированный язык моделирования).

Рассмотрим подход к проектированию СИИ, основанный на методе дедукции, как логический прием для выделения нового знания (планы акта деятельности) из ранее полученных знаний (концептуальная структура акта деятельности) [см. 3, с. 139].

Предложено выделять динамические и статические представления концептуальной структуры акта деятельности. Динамические представления определены двумя планами, относительно которых имеются следующие утверждения.

*Утверждение 1.* Функциональный план позволяет разрабатывать дискретно-событийные модели.

*Утверждение 2.* План процессов позволяет разрабатывать диаграмму потоков и уровней для имитационно-динамических моделей.

Статические представления также определены двумя планами, обусловленными следующими утверждениями.

**Утверждение 3.** План закономерностей служит своего рода интерпретатором графического представления в аналитическое представление в виде простейших математических уравнений.

**Утверждение 4.** План контекста позволяет разрабатывать различные модели, основанные на идеях когнитивного подхода.

Под концептуальным планом понимается определенная часть концептуальной структуры акта деятельности, которая рассматривается как проект для разработки моделей, основанных на знаниях. Задача реализации концептуальных планов на графическом уровне требует не только понимания методов искусственного интеллекта, но также больших временных затрат. Поэтому предлагается создать программное обеспечение "Интерпретатор" для выделения планов в виде текстовых файлов.

## 2. Функциональный план для проектирования дискретно-событийных моделей

Дискретно-событийная модель рассматривается как глобальная схема обслуживания заявок. Аналитические результаты таких моделей используются в теории массового обслуживания, как переход системы из одного состояния в другое дискретным образом в виде события [4].

Для проектирования сложных дискретно-событийных систем используют SADT-диаграммы (*Structured Analysis and Design Technique* — метод структурного анализа и проектирования), которые дают промежуточное представление, помогающее разработчику избежать ошибок при проектировании [5, с. 91].

Методология SADT представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной структуры сложных иерархических систем в виде блочной модели [6, с. 19]. Основным концептуальным



Рис. 1. Контекстная диаграмма SADT-модели

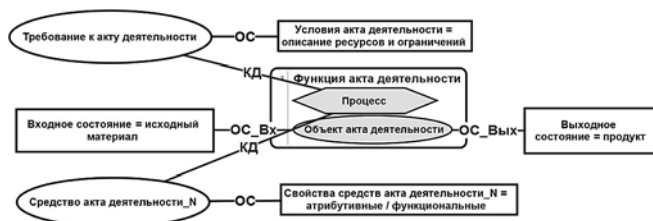


Рис. 2. Функциональный план акта деятельности

принцип методологии SADT — представление любой изучаемой системы в виде набора взаимодействующих и взаимосвязанных функциональных блоков (рис. 1).

В рамках методологии все, что происходит в системе и ее элементах, принято называть функциями, которые соответствуют определенному блоку. Функция  $F$  представлена процессом или действием  $Xa$  в зависимости от рассматриваемого уровня иерархии, и объектом  $Xao$ , на который воздействует данная активность.

$$F = \langle Xa, Xao \rangle. \quad (1)$$

При этом каждая сторона функционального блока (рис. 1) имеет стандартное значение [6, с. 16]: входная и выходная стрелки отображают данные или материальные объекты, управляющие стрелки — условия, влияющие на выполнение функции, а стрелки механизма — средства для исполнения работы. По аналогии функциональный план акта деятельности должен быть представлен: входными характеристиками ( $Vx$ ), которые определены исходным материалом; выходными характеристиками ( $Vyx$ ) — продуктом акта деятельности (далее продукт); управлением — требованием к продукту акта деятельности; механизмами — средствами акта деятельности [7, с. 20] (рис. 2).

Согласно графической нотации акта деятельности, изложенной в работе [1], функция образует структуру из двух возможных типов связующих звеньев:  $Lac$  (КД) — связь между компонентами процесса и процессом;  $Lp$  (ОС) — связь между элементами структуры и их свойствами (характеристиками)  $P$ .

Одним из важных моментов при разбиении является точная согласованность типов связей между функциями. Введем типизацию связей между функциями.

- Пересечение по исходному материалу, который является входной характеристикой нескольких функций. Функции группируются вследствие того, что они используют одни и те же входные данные.
- Пересечение по продукту и исходному материалу. Выход одной функции служит входными данными для следующей функции с меньшим доминированием. При последовательной связности имеет место обратная связь, когда выход функции становится входом другой функции с большим доминированием. Данная связь используется для описания циклов.
- Пересечение по продукту и свойству одного из компонентов. Выход одной функции служит свойством компоненты для другой менее доминирующей функции. При этом функции направлены на достижение одной цели. Поэтому реализуется полная зависимость одной функции от другой.

План функциональной структуры акта деятельности описывает функцию верхнего уровня, которая редуцирована на частные планы функциональных структур единичных решений. Функциональный план можно использовать не только для разработки дискретно-событийных моделей, но и для проектирования архитектуры гибридных интеллектуальных систем.

### 3. План процессов для проектирования имитационно-динамических моделей

Методология построения имитационно-динамического моделирования включает качественную и количественную стадии. Базовая структура количественной стадии содержит четыре существенных элемента: уровень (накопитель), функция решения, материальные и информационные потоки [8, с. 212] (рис. 3). Уровни представлены количественными показателями исследуемого объекта и характеризуют возникающие накопления внутри системы, они представляют собой те значения переменных в данный момент, которые они имеют в результате накопления вследствие разности между входящими и исходящими потоками. Содержимое накопителей может иметь любую природу. Материальный поток, вливаясь в уровень или вытекая из него, определяет изменение уровня. Кроме того, различают информационные потоки, с помощью которых принимается решение по управляющим воздействиям, т. е. определяется значение темпа потока на следующий интервал времени (функция решений) [9, с. 94].

Для реализации базовой структуры количественной стадии предлагается использовать процессный план акта деятельности, который обу-



Рис. 3. Базовая структура количественной стадии

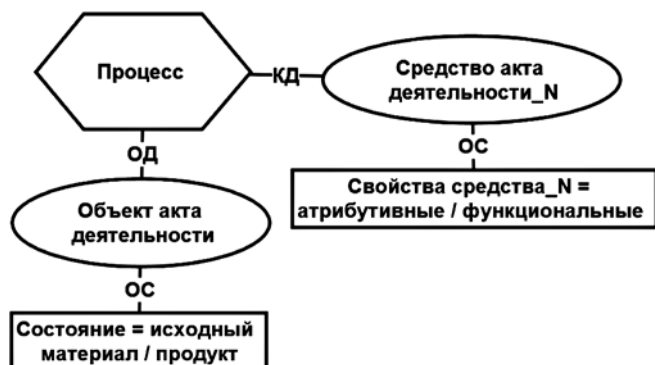


Рис. 4. Процессный план акта деятельности

словлен: процессом как совокупностью действий; объектом акта деятельности; его состояниями — исходными материалами или продуктами; средствами достижения цели и их свойствами (рис. 4).

Согласно графической нотации акта деятельности в процессный план добавлена связь *Lao* (ОД) между объектом процесса *Xao* и процессом *Xa* [1].

Процессный план акта деятельности описывает процессы верхнего уровня, которые могут быть редуцированы на частные планы единичных решений. При этом одним из важных моментов при разбиении должна быть точная согласованность типов связей между действиями. Введем типизацию связей между процессными планами структур единичных решений.

- Тип последовательной связности — исходное действие должно полностью завершиться, прежде чем начнется выполнение другого действия. Данный тип связи моделирует причинно-следственные зависимости трансформации исходного материала.
- Тип последовательной связности с обратной связью — реализует цикл процесса.
- Тип логической связности. Возможна ситуация, когда завершение одного действия может инициировать начало выполнения сразу нескольких других действий. Данный тип связи моделирует логические зависимости.
- Тип обратной логической связности. Определенное действие может требовать завершения нескольких других действий для начала своего выполнения.

Процессный план оперирует множеством действий, определяется, в какое количественное состояние перейдет в будущем объект из заданного текущего состояния. Тогда уровень накопителя представлен как объект действия *Xao*, определенный через связь *Lp* (ОС) одним из количественных состояний — исходный материал или продукт. Поток определен непрерывной активностью *Xa*, которая представлена действием, перемещающим содержимое между накопителями. Используя гидродинамическую метафору, отметим, что функция решений есть вентиль, управляющий потоком, который обусловлен средствами акта деятельности  $\{Xac_1, Xac_2, \dots, Xac_N\}$  и связью *Lp* (ОС) с его функциональными свойствами *P*.

Функции решений есть уравнение темпов, отражающее понимание факторов, определяющих действия, которые будут совершаться непосредственно в следующий момент времени. Закон изменения темпа задается функциональной зависимостью [9, с. 94]:

$$V(t) = F(p_1(t), p_2(t), \dots, p_k(t)), \quad (2)$$

где  $V(t)$  — входящий поток (исток) или выходящий поток (сток);  $F$  — произвольная функция от

числа аргументов  $k$ ;  $p_i(t)$  — параметры модели, значения которых в момент  $t$  известны.

Таким образом, функция решений не является столь очевидной и простой. Поэтому она должна быть определена другой деятельностью и планом зависимости. Также качественная стадия будет рассмотрена в рамках плана контекста как исследованная структура причинно-следственных связей.

На основе плана процессов разрабатывается диаграмма потоков и уровней для имитационных динамических моделей [10].

#### 4. План закономерностей для проектирования аналитических моделей

Статические аспекты описывают концептуальную структуру акта деятельности в определенный момент времени (до или после действия). При этом отношения и соотношения являются основными связующими "строительными блоками" таких концептуальных структур.

Аналитические модели представляют собой уравнения или системы уравнений, записанные в виде алгебраических, интегральных, дифференциальных, конечно-разностных и иных соотношений и логических условий. Такая модель, как правило, имеет статическое представление, которое подходит лишь для очень простых и сильно идеализированных задач и объектов.

План закономерностей определяется множественностью проявляемых свойств объектов концептуальной структуры, которые характеризуются определенными отношениями ( $R_{oc} = 1$ ,  $R_{cc} = 1$ ) и соотношениями ( $R_p$ ), тем самым фиксируются закономерности преобразования исходного материала в продукт.

При построении плана закономерностей единичного решения необходимо учитывать следующие правила.

- Аналитическая закономерность может быть выявлена объектной частью концептуальной структуры единичного решения. При этом компонент "Требование к объекту действия" в плане закономерностей не учитывается (рис. 5).
- Состояние объекта действия (продукт) ассоциируется с решением, а средства действий — с внутренними параметрами уравнения.
- Отношения в плане закономерностей однопавлены и единичны, т. е. из одного элемента может выходить или входить в него одно отношение.
- Отношение между объектом и орудием действия  $R_{oc}$  и отношения между средствами  $R_{cc}$  представлены отношениями зависимостей и обусловлены типами "увеличивает" или "уменьшает", отношение  $R_{oc}$  — типом "определяет" (см. таблицу).
- Соотношения являются ассоциациями отношений зависимостей, которые определены сле-



Рис. 5. План закономерностей единичного решения

#### Ассоциативные правила

Тип отношения	Тип соотношения	Тип арифметического действия
Увеличивает	Больше на Больше в	Сложение Умножение
Уменьшает	Меньше на Меньше в	Вычитание Деление

дующими типами: "больше на", "меньше на", "больше в" и "меньше в". Соотношения ассоциаций, в свою очередь, могут быть отображены как арифметические действия (см. таблицу). Отношение  $R_{oc}$ , обусловленное типом "определяет", отображается как знак равенства.

На основе плана закономерностей могут определяться различные аналитические представления, в том числе функции решений (интенсивность потока) для имитационно-динамической модели [10].

#### 5. План контекста для проектирования когнитивных моделей

Когнитивное моделирование ориентировано на помощь управленцу, оно помогает фиксировать свое представление о проблемной ситуации в виде формальной модели. В качестве такой модели обычно используется так называемая когнитивная карта ситуации, которая представляет известные субъекту основные законы и закономерности наблюдаемой ситуации в виде ориентированного знакового графа, в котором вершины графа — это факторы (признаки, характеристики ситуации), а дуги между факторами — причинно-следственные связи между факторами [11, с. 464] (рис. 6).

В когнитивной карте выделяют два типа причинно-следственных связей: положительные (+) и отрицательные (-). При положительной связи увеличение значения фактора-причины приводит к увеличению значения фактора-следствия, а при отрицательной связи увеличение значения фактора-причины приводит к уменьшению значения фактора-следствия.

При этом под понятием "фактор" понимается движущая сила, причина какого-нибудь процесса, определяющая его характер или отдельные его черты. Таким образом, когнитивная структура предметной области — это выявление

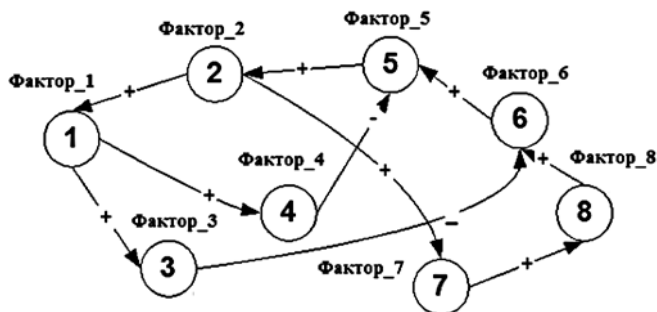


Рис. 6. Пример когнитивной карты

будущих целевых и нежелательных состояний объекта управления и наиболее существенных (базисных) факторов управления и внешней среды, влияющих на переход объекта в эти состояния, а также установление на качественном уровне причинно-следственных связей между ними, с учетом взаимовлияния факторов друг на друга [11, с. 465].

Таким образом, в плане контекста должны быть определены следующие элементы: субъект акта деятельности  $Xas$ , который взаимодействует с объектом действия  $Rso$  и различными компонентами действий  $\{Rsc_1, Rsc_2, \dots, Rsc_N\}$ ; объект акта деятельности  $Xao$ , который существует во взаимодействии с субъектом  $Ros$  и с различными отношениями-компонентами  $\{Roc_1, Roc_2, \dots, Roc_N\}$ ; компоненты действия в виде требований и средств акта деятельности  $\{Xac_1, Xac_2, \dots, Xac_N\}$ , которые взаимодействуют с субъектом  $\{Rcs_1, Rcs_2, \dots, Rcs_N\}$  и существуют в отношениях между собой  $Rcc_{jN}$ . При этом необходимо учитывать двойную направленность взаимодействий ( $Rcs$  или  $Rsc$ ) и отношений ( $Roc$  или  $Rco$ ) (рис. 7).

В данном случае под термином "контекст" понимается структура внешней среды, в пределах которой выявляются факторы управления. Статистическое представление плана контекста определено взаимодействиями и отношениями, равными 1, и типами "увеличивает" или "уменьшает".

На уровне концептуальных структур единичных решений возможны следующие виды пересечений.

- По субъекту акта деятельности. Когда субъект взаимодействует с различными объектами действия как в одном акте деятельности,



Рис. 7. Контекстный план акта деятельности

так и в разных актах. При этом связь  $Las$  (СД) между субъектом действия  $Xas$  и действием  $Xa$  обрывается.

- По объекту акта деятельности. Когда один и тот же объект в одном акте деятельности представлен как "Продукт\_1", а в другом — как "Продукт\_2".
- По одному из компонентов действия (требованию или средствам акта деятельности). Когда один и тот же компонент используется в различных актах деятельности.

Соответственно, возможны различные сочетания данных пересечений. Например, по субъекту и средствам, по субъекту и объектам акта деятельности и т. д.

План контекста предназначен не только для проектирования когнитивных моделей, но для лучшего осмысления определенной деятельности, а также как дополнение к остальным концептуальным планам, например, для определения качественной стадии в имитационном динамическом моделировании.

## 6. Программная реализация концептуальных планов

Концептуальные планы по своей сути являются базой знаний для проектирования интеллектуальных систем [12]. Соответственно, задача выделения таких знаний из концептуальной структуры акта деятельности может быть решена на программном уровне.

Для разработки программного обеспечения (ПО) "Интерпретатор" использовался фреймворк Qt 5.3, в котором предоставлен класс `QXmlStreamReader`. Данный класс представляет собой быстрый синтаксически корректный XML (*eXtensible Markup Language* — расширяемый язык разметки) — анализатор с простым потоковым интерфейсом прикладного программирования. Базовая концепция потокового чтения состоит в представлении XML-документа в виде потока маркеров и позволяет создавать рекурсивные спускаемые анализаторы [13, с. 660]. Это означает, что можно разделить код анализа XML на различные классы.

Таким образом, происходит считывание структуры дерева и классификация его объектов по связям (СД, ОД, КД и ОС). После анализа XML-документа и формирования начального набора данных осуществляется вызов функции поиска объектов. Выполнение этих действий приводит к структуризации данных, необходимых для поиска пересечений, и выделению необходимых плановых представлений. На основе найденных пересечений выводятся необходимые отчеты о концептуальных планах в HTML-файле (*HyperText Markup Language* — язык гипертекстовой разметки).

На основе данного алгоритма разрабатываются четыре модуля программы "Интерпретатор", которые разделяются по двум блокам: динамических и статических знаний (рис. 8).

В отчете динамической базы знаний функционального плана определяются следующие элементы: функция — [Действие\_N Объект\_N]; входное свойство функции — [Свойство объекта\_N до действия\_N]; выходное свойство функции — [Свойство объекта\_N после действия\_N]; механизмы функции — [Средства действия\_N]; инструкции по управлению функцией — [Требования к акту деятельности\_N]. В HTML-файле функционального плана указываются выявленные пересечения и предлагаются рекомендации по его построению в виде различных конструкций. Например: Вход <Свойство объекта\_N до действия\_N> → Функция <Действие\_N Объект\_N> → Выход <Свойство объекта\_N после действия\_N>.

В отчете основными элементами базы знаний процессного плана [14] являются: уровень — [Объект\_N Количество: = <Свойство объекта до действия>] и [Объект\_N Количество: = <Свойство объекта после действия>]; поток — [Действие\_N]. В HTML-файле процессного плана указываются выявленные пересечения и предлагаются рекомендации по его построению в виде следующего конструкта: Уровень\_1 <Объект\_N Количество: = > = <Свойство объекта\_N до действия\_N> → поток <Действие\_N> → Уровень\_1 <Объект\_N Количество: = > = <Свойство объекта\_N после действия\_N>.

В отчете основными элементами базы знаний плана закономерностей являются: уравнение — [Объект\_N]; параметр — [Объект\_N Свойство средства\_N после действия\_N]; математический знак — [Соотношение]; знак равенства — [Отношение <Определяет>]. В HTML-файле плана закономерностей предлагаются рекомендации по его построению в виде следующего конструкта: Уравнение <Объект\_N> Знак равенства <Отношение 'Определяет'> Параметр <Объект\_1 Свойство средства\_1 после действия\_1> Математический знак <Соотношение> Параметр <Объект\_2 Свойство средства\_2 после действия\_2> и т. д.

В отчете основными элементами базы знаний контекстного плана являются: факт взаимодействия — [Субъект Взаимодействие Объект\_N], [Субъект Взаимодействие Средства\_N] и [Субъект Взаимодействие Требования\_N]; факт отношения — [Средство\_N-1 Отношение Средства\_N], [Требование\_N-1 Отношение Требования\_N] и [Средство\_N Отношение Требования\_N]; факт появления продукта — [Объект\_N Свойство объекта\_N после действия\_N]. В HTML-файле контекстного плана указываются выявленные пересечения, и установленные факты являются рекомендациями к разработке когнитивных моделей.

Таким образом, ПО "Интерпретатор" расширяет возможности программного комплекса (ПК) "Оформитель + Решатель" [1]. Появляется новая составляющая данного ПК, которая обуславливает иную организацию совокупностей программ.

Создается ПК "Оформитель + Решатель + Интерпретатор", который позиционируется как интеллектуальная надстройка для проектирования автономных интеллектуальных моделей (рис. 9).

Использование программного комплекса определено следующим алгоритмом:

- после определения актов деятельности [1] в программе "Оформитель" реализуются их концептуальные структуры единичных решений, которые определены как целостная концептуальная модель принятия решений;

- в программном комплексе "Решатель" целостная концептуальная модель принятия решений проверяется на полноту и адекватность (при необходимости генерируется отчет о базе знаний в виде продукционных правил);

- после подтверждения полноты и адекватности модели в ПО "Интерпретатор" генерируются отчеты о базах знаний концептуальных планов; синтез баз знаний соответствует определенным моделям и системам искусственного интеллекта.



Рис. 8. Структурная схема программного обеспечения "Интерпретатор"



Рис. 9. Программный комплекс "Оформитель + Решатель + Интерпретатор"

## Заключение

Практика использования интеллектуальных систем и их синергетических комбинаций показывает, что в настоящее время в ее теоретическом обосновании не существует:

— методики и концептуальной структуры, объединяющей ситуационные, имитационные, экспертные, эволюционные и другие подходы к моделированию СИИ;

— целостной модели знаний и единого языка графического описания для моделирования в области разработки СИИ;

— программного обеспечения представления баз знаний для проектирования СИИ.

Эти обстоятельства указывают на существование в области поддержки принятия решений проблем, состоящих в отсутствии единой концептуальной структуры обоснованных решений относительно управления сложными системами и в программном извлечении различных знаний из концептуальной модели для проектирования СИИ. Предложенный в работе метод и программный инструментарий позволяют эффективно решить данные проблемы.

### Список литературы

1. Сорокин А. Б., Смольянинова В. А. Концептуальное проектирование экспертных систем // Информационные технологии. 2017. Том 23, № 9 (23). С. 634—641.
2. Рыбина Г. В. Интегрированные экспертные системы: некоторые итоги и перспективы // Искусственный интеллект и принятие решений. 2008. № 1. С. 22 — 46.

3. **Философский** энциклопедический словарь / Гл. ред.: Л. Ф. Ильичев, П. Н. Федосеев, С. М. Ковалев, В. Г. Панов. М.: Сов. Энциклопедия, 1983. 840 с.

4. **Боев В. Д., Сыпченко Р. П.** Компьютерное моделирование. М.: ИНТУИТ.РУ, 2010. 349 с.

5. **Емельянов В. В., Ясиновский С. И.** Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Язык РДО. М.: АНВИК, 1998. 427 с.

6. **Марка Д., МакГоуэн К.** Методология структурного анализа и проектирования SADT. М.: МетаТехнология, 1993. 240 с.

7. **Сорокин А. Б.** Полиаспектный анализ при проектировании систем поддержки принятия решений // Научно-техническая информация. Серия 2. Информационные процессы и системы. 2014. № 8. С. 10—23.

8. **Теория** систем и системный анализ в управлении организациями: Справочник: Учеб. пособие / Под ред. В. Н. Волковой и А. А. Емельянова. М.: Финансы и статистика, 2006. 848 с.

9. **Форрестер Дж.** Основы кибернетики предприятия. М.: Прогресс, 1971. 340 с.

10. **Sorokin A. B., Bolotova L. S.** The evolutionary model as the projection methodology situationally — activity analysis and its realization on the example of the model against the development of infectious diseases // Collection of scientific papers "Interactive systems: Problems of Human-Computer Interaction". Ulyanovsk, 2015. P. 120—130.

11. **Болотова Л. С.** Системы искусственного интеллекта: модели и технологии, основанные на знаниях. М.: Финансы и статистика, 2012. 663 с.

12. **Сорокин А. Б.** Свидетельство о государственной регистрации базы данных № 2015621317 "Интеллектуальная база данных для проектирования систем основанных на знаниях". Дата государственной регистрации в Реестре баз данных 26 августа 2015 г.

13. **Шлее М.** Qt 4.8. Профессиональное программирование на C++. СПб.: БХВ-Петербург, 2012. 912 с.

14. **Сорокин А. Б., Лобанов Д. А.** Свидетельство о государственной регистрации программы для ЭВМ № 2015662075 "Интерпретатор — процессный план". Дата государственной регистрации в Реестре программ для ЭВМ 17 ноября 2015 г.

**A. B. Sorokin**, Ph. D., Associate Professor, e-mail: ab\_sorokin@mail.ru,

**D. A. Lobanov**, Master Student, e-mail: tty66@mail.ru, Moscow Technological University (MIREA)

## Conceptual Design of Intelligent Systems

*A methodology for distinguishing conceptual plans from the structure of an activity act for the design of intelligent systems is considered. In conceptual plan is understood as a certain part by conceptual structure of the act of activities which is considered as the project for development of the models based on knowledge. At the graphical level, four planned views are proposed: the functional structure, processes, context and regularities. The functional plan is designed for the design of discrete-event models and architectures of hybrid intelligent systems. The process plan is designed to develop a flow diagram and levels for system dynamics simulation modeling. On the basis of the plan of regularities different analytical representations, including functions of decisions (intensity of a flow) for an imitative dynamic model can be defined. The plan of a context is intended not only for design of cognitive models, but also to better comprehend certain activities and as an addition to the rest of the conceptual plans. However the graphic construction of the conceptual model of a dynamically complex environment and its particular representations is a complex and nontrivial task. Therefore, the program complex "Designer + Solver + Interpreter" is developed, which verifies the conceptual model of a dynamically complex environment on completeness and adequacy, and also generates the necessary knowledge base for the design of intelligent models in the form of text files.*

**Keywords:** plans of conceptual structure of the act of activities, functional structure plan, process plan, plan of a context, plan of regularities

### References

1. **Sorokin A. B., Smol'janinova V. A.** Konceptual'noe proektirovanie jekspertnyh sistem, *Informacionnye tehnologii*, 2017, vol. 23, no. 9, pp. 634—641 (in Russian).

2. **Rybina G. V.** Integrirovannye jekspertnye sistemy: nekotorye itogi i perspektivy, *Iskusstvennyj intellekt i prinjatije reshenij*, 2008, no. 1, pp. 22—46 (in Russian).

3. **Filosofskij** jenciklopedicheskij slovar' / L. F. Il'ichev, P. N. Fedoseev, S. M. Kovalev, V. G. Panov. Moscow, Sov. Jenciklopedija, 1983, 840 p. (in Russian).

4. **Boev V. D., Sytchenko R. P.** Komp'yuternoe modelirovanie. Moscow, INTUIT.RU, 2010. 349 p. (in Russian).
5. **Emel'janov V. V., Jasinovskij S. I.** *Vvedenie v intellektual'noe imitacionnoe modelirovanie slozhnykh diskretnykh sistem i processov.* Jazyk RDO, Moscow, ANVIK, 1998, 427 p. (in Russian).
6. **Marka D., MakGoujen K.** *Metodologija strukturnogo analiza i proektirovanija SADT,* Moscow, MetaTehnologija, 1993, 240 p. (in Russian).
7. **Sorokin A. B.** Poliaspektnyj analiz pri proektirovanii sistem podderzhki prinjatija reshenij, *Nauchno-tehnicheskaja informacija. Serija 2. Informacionnye processy i sistemy,* 2014, no. 8, pp. 10–23 (in Russian).
8. **Teorija** sistem i sistemnyj analiz v upravlenii organizacijami: Spravochnik: Ucheb. posobie / Eds. V. N. Volkovoj, A. A. Emel'janova. Moscow, Finansy i statistika, 2006, 848 p. (in Russian).
9. **Forrester Dzh.** *Osnovy kibernetiki predpriyatija,* Moscow, Progress, 1971, 340 p. (in Russian).
10. **Sorokin A. B., Bolotova L. S.** The evolutionary model as the projection methodology situationally — activity analysis and its realization on the example of the model against the development of infectious diseases, *Collection of scientific papers "Interactive systems: Problems of Human-Computer Interaction",* Ulyanovsk, 2015. pp. 120–130.
11. **Bolotova L. S.** *Sistemy iskusstvennogo intellekta: modeli i tehnologii, osnovannye na znanijah,* Moscow, Finansy i statistika, 2012, 663 p. (in Russian).
12. **Sorokin A. B.** Svidetel'stvo o gosudarstvennoj registracii bazy dannyh № 2015621317 "Intellektual'naja baza dannyh dlja proektirovanija sistem osnovannyh na znanijah". Data gosudarstvennoj registracii v Reestre baz dannyh 26 avgusta 2015 goda.
13. **Shlee M.** *Qt 4.8. Professional'noe programmirovanie na C++,* Saint-Petersburg, BHV-Peterburg, 2012, 912 p. (in Russian).
14. **Sorokin A. B., Lobanov D. A.** Svidetel'stvo o gosudarstvennoj registracii programmy dlja JeVM № 2015662075 "Interpretator — processnyj plan". Data gosudarstvennoj registracii v Reestre programm dlja JeVM 17 nojabrja 2015 goda.

УДК 004.434

**А. О. Сухов**, канд. физ.-мат. наук, доц. каф., e-mail: ASuhov@hse.ru,

**Е. Ю. Медведева**, студент, e-mail: medvedevaeyu@mail.ru,

Национальный исследовательский университет "Высшая школа экономики", г. Пермь

## **Подход к разработке языкового инструментария для создания текстовых предметно-ориентированных языков**

*Описан подход к созданию языкового инструментария, новизна которого заключается в том, что в процессе функционирования языковой инструментальной системы выполняется интерпретация моделей различных уровней иерархии, а не генерация на их основе исходного кода. Это позволяет проводить настройку разработанных языков без регенерации кода редактора, выполнять многоуровневое моделирование, а также определять правила преобразования написанных программ в код на целевом языке.*

**Ключевые слова:** предметно-ориентированные языки, текстовые языки, языковой инструментальной, мета-язык, многоуровневое моделирование, метамодель, трансформация моделей, модельно-ориентированный подход

### **Введение**

Одной из движущих сил развития языков программирования и инструментальных средств разработки программного обеспечения является уменьшение трудовых затрат программистов за счет сокращения семантического разрыва между конструкциями языков программирования и объектами предметной области, которые требуется описать с помощью этих языков. Переход от машинных языков к языкам высокого уровня позволил существенно сократить трудозатраты и время разработки, а также повысить качество создаваемых систем. С начала 1970-х годов прошлого века стали активно развиваться проблемно- и предметно-ориентированные языки программирования, которые предназначены для решения определенного класса задач в конкретной предметной области. Эти языки позволили существенно сократить трудозатраты при раз-

работке приложений, поскольку одной команде такого языка соответствуют десятки строк кода, написанных на языке высокого уровня [1–4]. Примерами предметно-ориентированных языков являются Prolog, VHDL, HTML, GPSS и др. С течением времени такие языки стали все чаще применяться при разработке программного обеспечения. На сегодняшний день разработка любой информационной системы немислима без использования языка запросов SQL, языка проектирования интерфейса пользователя, интегрированного в IDE (Integrated Development Environment).

*Предметно-ориентированные языки (Domain-Specific Languages, DSL) — языки программирования (моделирования), предназначенные для решения определенного круга задач в конкретной предметной области.*

Поскольку такие языки оперируют терминами предметной области, при разработке и сопро-