



УДК 004.89

### СЕМАНТИЧЕСКИЕ МОДЕЛИ ПРЕДСТАВЛЕНИЯ РЕШАТЕЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ

Шалфеева Е.А.\*

\* *Институт автоматизации и процессов управления,  
г. Владивосток, Россия*

shalf@iacp.dvo.ru

В работе предложен набор семантических представлений разрабатываемых интеллектуальных программных систем, их содержание и структура. Предложен метод конструирования моделей решателя системы этапов анализа и проектирования на основе моделей этапа системного анализа. Эти модели необходимы для построения управляемых интеллектуальных систем, а методы являются базой для создания инструментария для поддержки разработчиков.

**Ключевые слова:** интеллектуальная программная система, знания, решатель, постановка задачи, онтология, модель подзадач, архитектурная модель, семантическая сеть.

#### ВВЕДЕНИЕ

В настоящее время продолжают работы по методам автоматизации интеллектуальной деятельности и созданию инструментария, позволяющего комплексно автоматизировать сферы экономики, процессы в которых связаны с использованием постоянно обновляемых *знаний*. Вместе с тем отмечается, что из-за «отсутствия стандартов проектирования и разработки интеллектуальных систем» проблемы, связанные с использованием прикладных моделей представления знаний и моделей обработки знаний, нуждаются в комплексной теоретической и практической доработке [Голенков и др., 2012].

Переход к декларативному представлению знаний в *интеллектуальных программных системах* (ИПС) уже внес свой вклад в сопровождаемость баз знаний. Отвечающая современным требованиям *база знаний* не только отделена в самостоятельный переносимый компонент, но и является *концептуальной*, понятной для сопровождающего ее эксперта [Грибова и др., 2011].

На успех комплексной автоматизации интеллектуальной профессиональной деятельности (ИПД) существенно влияют результаты ее системного анализа. Идентифицированы создаваемые на этом этапе модели предметной области, в числе которых и онтологии знаний и данных, и модель деятельности, и концептуальная модель системы [Клещев и др., 2012].

Представление баз знаний и данных в форме семантических сетей достаточно распространено

[Гулякина и др., 2012, Грибова и др., 2011]. Развиваются языки программирования, ориентированные на обработку таких сетей, представление данных и программ в единообразном виде [Гулякина и др., 2012].

Известны и преимущества декларативного представления программ, по сравнению с императивным: более простое написание, более легкое их понимание программистами и модифицирование, возможность замены сопровождения программы управлением ею. Предложена концепция проектирования интеллектуальных систем, все компоненты (данные, знания, решатель задач с пользовательским интерфейсом) которого имеют единые принципы для их формирования, доступа и модифицирования [Клещев и др., 2010; Грибова и др. 2012]. Разрабатываемая по этой концепции технология создания является развитием технологии построения мультиагентных систем «в сторону» улучшения их сопровождаемости. Самостоятельным компонентом ИПС с концептуальной базой знаний является решатель задач, использующий базу знаний при оказании поддержки в принятии решения.

Целью исследования является разработка метода построения ключевых моделей процесса проектирования решателей интеллектуальной системы с использованием семантических моделей системного анализа автоматизируемой профессиональной деятельности.

# 1. Семантические модели системного анализа

Проектирование эффективных информационных систем невозможно без использования результатов информационного моделирования и стратегического планирования [Штрик, 1993]. При автоматизации ИПД, когда проектируется наращиваемый «долгоживущий» комплекс средств поддержки принятия решения и средств доступа к знаниям и другой информации, этот первый этап является более содержательным по составу работ [Клещев и др., 2012].

Системный анализ ИПД подразумевает выработку следующих моделей (представлений предметной области и разрабатываемой в ней системы).

1. *Схема описания информационного объекта* - объекта автоматизируемой деятельности (примером такого артефакта для задачи медицинской диагностики является схема истории болезни (ИБ) пациента [Клещев и др., 2012]).

2. *Профессиональная терминология* – понятия, которые используют специалисты при описании наблюдаемой действительности и знаний о происходящих в ней процессах или существующих связях (пример терминов для медицины: наблюдения, признаки, события, особенности, измерения, возможные значения, диагнозы, планы лечения и т.д. [Клещев и др., 2005, Черняховская и др., 2007]).

3. *Перечень формируемых в процессе ИПД документов* (например: ИБ, назначения на обследования, протокол лечения, рецепты на лекарственные средства, назначение даты посещения врача).

4. *Схема решаемых задач* – (графовая) модель, упорядочивающая задачи специалистов в рассматриваемой сфере ИПД (на рис. 1 графовая модель упорядочивает интеллектуальные задачи врача – задачи опроса, обследования, постановки диагноза, назначения лечения, наблюдения за состоянием – и прочие задачи медицинского персонала.

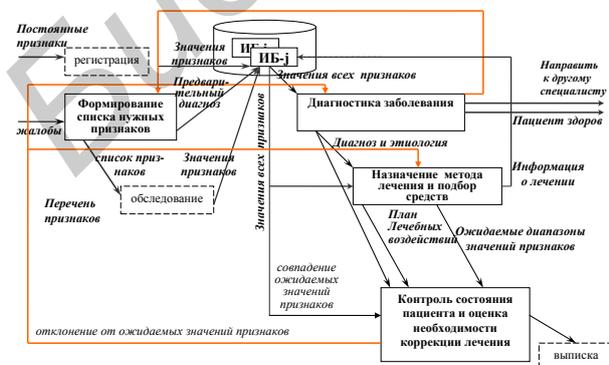


Рисунок 1 – Схема медицинских задач

5. Множество онтологий задачи - своя для

каждой интеллектуальной задачи<sup>1</sup> из схемы решаемых задач.

Каждая онтология задачи имеет следующие составляющие.

O1. *Онтология наблюдений* – терминология, лежащая в основе описания наблюдаемых\управляемых объектов задачи и описания знаний о процессах, в которые эти объекты вовлечены (например, терминология для истории наблюдений и знаний о диагностике, содержащая такие термины, как наблюдение, множество значений наблюдения и т.д.) [Клещев и др., 2005].

O2. *Онтология действительности* – описание структуры наблюдаемых элементов действительности, их области возможных значений; структуры принимаемого специалистом решения и возможные значения решения (например, для медицины - все, что принято включать в карту ИБ пациента).

Например, часть структуры ИБ может быть такой:

паспортная часть,  
особенности пациента,  
(жалобы,  
опрос,  
осмотр)<sup>2</sup>

название признака  
значение  
момент наблюдения)  
предварительный диагноз...

O3. *Онтология знаний* задает структуру хранимых знаний о протекающих процессах, об изменении состояний объекта и влияющих на это факторах и прочих внутренних и внешних связях объекта и закономерностях предметной области.

Например, часть структуры знаний может быть такой:

...Клиническая картина  
(главные жалобы,  
дополнительные жалобы,  
наружный осмотр,  
лабораторные исследования)

(простой признак,  
составной признак  
характеристика признака)  
название признака  
необходимость  
варианты динамики признака

<sup>1</sup> Автоматизация неинтеллектуальных задач (в рамках данного исследования) считается успешно решаемой, и подсистемы, реализующие их, будут интегрироваться в общую систему на соответствующем этапе разработки.

<sup>2</sup> Здесь скобки группируют понятия одного уровня с одинаковой структурой, а «(\*)» указывает на множественное вхождение понятия на этом уровне.

(\*) вариант

(\*) период динамики

номер периода

значение признака

длительность периода...

04. *Соглашения о связях действительности и знаний* (иногда часть онтологии знаний) - формализация знания о зависимости результата решения от различных элементов знаний; каждый элемент онтологии действительности (получаемых результатов) должен быть связан с элементами онтологии входных данных или некоторых «промежуточных», отражающих их причинно-следственные связи [Клещев и др., 2006].

05. *Онтология документов* - архивных, отчетных и печатных, которая задает структуру этих документов (например, структура медицинского архива, т.е. всех ИБ, формат описаний схем лечения и структура рецептов).

06. *Онтология объяснения* результата решения, которая задает структуру получаемой консультации, «прописывая» требования к содержимому объяснения. Например, часть структуры объяснения - сам диагноз плюс все возможные схемы развития во времени процессов в организме пациента, соответствующие диагнозу (заболеванию):

название заболевания

(\*) признак наличия заболевания

название признака

ссылка на вариант динамики

объяснение развития этого варианта

(\*) период развития

номер периода динамики этого варианта

значение признака, произошедшее в этот период

момент его наблюдения [Москаленко, 2006; Клещев и др., 2006].

6. Множество *моделей знаний* (баз знаний или наборов формализованных знаний).

*Знания* для каждой интеллектуальной задачи – формально представленные знания (под управлением онтологии).

Пример. При описании знаний о заболеваниях дается описание нормы: каждому наблюдению из базы наблюдений сопоставляются нормальные значения – собственное подмножество множества возможных значений этого наблюдения в простой базе наблюдений. Описание каждого заболевания состоит из его клинической картины (см. рис. 2), содержащей описания клинических проявлений, названиями которых являются названия некоторых наблюдений в простой базе наблюдений. Описание каждого клинического проявления состоит из конечного множества вариантов, не имеющих названий, а описание каждого варианта – из конечного упорядоченного множества периодов

динамики, не имеющих названий. Описание каждого периода динамики состоит из области значений и границ длительности [Черняховская и др., 2007].

Пневмококковый (стафилококковый) конъюнктивит [заболевание]
Эпидемический конъюнктивит Коха-Уикса [заболевание]
Конъюнктивит острый неспецифический катаральный [заболевание]
Диплобациллярный (ангулярный) конъюнктивит [заболевание]
Дифтерийный конъюнктивит [заболевание]
Клиническая картина [Клиническая картина]
Главные жалобы [Главные жалобы]
Светобоязнь -- Присутствие [клиническое проявление]
Светобоязнь (ссылка в ИР База наблюдений в области офтальмологии)
Присутствие (ссылка в ИР База наблюдений в области офтальмологии)
модальность [модальность]
выбор типа значений [выбор типа значений]
качественные значения [качественные значения]
1 [вариант КП]
1 [период динамики]
область значений следствия [область значений следствия]
постоянно (тип: STRING) [комментарий о границах дл
0.0 (тип: REAL) [верхняя граница длительности /в сутк
0.0 (тип: REAL) [нижняя граница длительности /в сутк
[ жесткая связь значений {И}; (тип: STRING) ]
[ условие на воздействующие факторы ]
[ вариант КП ]
[ клиническое проявление, измененное воздействием события
[ Необходимое условие ]
Слезотечение -- Характер начала [клиническое проявление]
модальность [модальность]
Слезотечение (ссылка в ИР База наблюдений в области офтальмологии)
Характер начала (ссылка в ИР База наблюдений в области офтальмолог
выбор типа значений [выбор типа значений]
[ Необходимое условие ]
Слезотечение -- Периодичность [клиническое проявление]
Резь в глазах -- Присутствие [клиническое проявление]
Резь в глазах -- Глаз [клиническое проявление]
Резь в глазах -- Характер начала [клиническое проявление]
Слезотечение -- Глаз [клиническое проявление]

Рисунок 2 – Фрагмент редактируемого описания заболевания

7. Множество *постановок задач* - своя для каждой интеллектуальной задачи (в терминах ее онтологии).

*Постановка задачи* включает 1) название задачи, 2) текстовое описание задачи, 3) названия и форматы (ссылки на фрагменты онтологии действительности) входных данных задачи, 4) знания о решении интеллектуальной задачи; 5) что требуется найти (в том числе объяснение решения) со ссылками на область значений самого решения (в модели знаний) и на структуру (онтологию) объяснения и сопутствующих документов; 6) описание или указание метода решения.

Пример постановки задачи диагностики заболевания» [Москаленко, 2011]:

1) Задача медицинской диагностики

2) Задача состоит в определении всех возможных альтернативных диагнозов больного на основе знаний предметной области и данных его обследования, к которым относят значения признаков (в моменты их наблюдения), значения анатомио-физиологических особенностей (постоянные во времени) и значения произошедших событий (в моменты, когда они происходили).

3) Наблюдавшиеся признаки<sup>3</sup>, моменты их наблюдения и их значения в эти моменты;

<sup>3</sup> а также произошедшие события и наблюдавшиеся особенности

формат: фрагмент Онтологии действительности (см. О2 выше).

4) Знания о наблюдениях (признаках, событиях, анатомо-физиологических особенностях пациента) и их возможных значениях + знания о заболеваниях и длительностях их периодов развития<sup>4</sup> + знания о значениях признаков при отсутствии заболеваний - знания о нормальных реакциях и знания о реакциях на воздействие событий.

5-1) все альтернативные диагнозы пациента;

формат: множество названий известных заболеваний.

5-2) объяснение каждого диагноза через объяснение всех наблюдаемых значений признаков;

формат: онтология объяснения (см. Об).

6) «Естественным методом решения частной обратной задачи медицинской диагностики является перебор всех возможных значений диагнозов, при котором для каждого заболевания выполняется решение прямой задачи - построение всех возможных схем развития ...» [Москаленко, 2011].

Наличие в постановке «Описания метода решения» увеличивает вероятность реализуемости соответствующей подсистемы для рассматриваемой задачи.

8. Множество алгоритмов решения задач - свой для каждой интеллектуальной задачи.

Алгоритм решения задачи на этапе системного анализа не обязателен для «обычных» задач. Технология систем рекомендует построение алгоритмов для решения задач на более ранних этапах для получения гарантии реализуемости соответствующей подсистемы для рассматриваемой задачи [Martin, 1989; Данилов, 2006]. Алгоритмы могут быть представлены математически, структурно, на псевдо-языке, графически. Но, как минимум, эти алгоритмы содержат информацию об отдельных решаемых подзадачах (разного уровня вплоть до элементарных) информацию об их связях, нужную для создания архитектурных проектов подсистем).

9. Концептуальная архитектура системы - схема взаимодействия подсистем – решателей задач с пользователями, с общими информационными компонентами и друг с другом. Для каждой интеллектуальной задачи требуется подсистема поддержки (решатель и пользовательский интерфейс), которая вырабатывает соответствующее объяснение своего заключения.

Пример. Врач подтверждает результат задачи либо формирует свое решение. Решения задач обследования, диагностики, лечения и коррекции являются элементами ИБ, поэтому подсистема ее

редактирования доступна врачу (рис. 3). Каждый случай расхождения решения врача с автоматическим решением сохраняется в архиве с тем, чтобы стать одним из прецедентов для будущих научных исследований по формированию новых знаний. Знания, общие для задач прогноза и лечения - фармакологические знания; знания о реакциях на воздействие событий используется в задачах мед диагностики, назначения лечебных мероприятий, мониторинга.

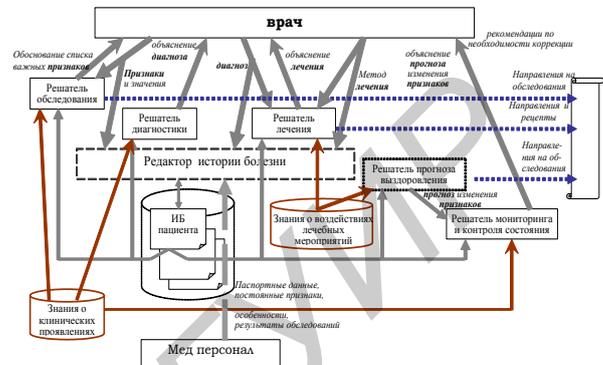


Рисунок 3 – Вариант концептуальной архитектуры медицинской системы

Структурное описание такой архитектурной модели требует несколько видов узлов и соединений между ними:

- (\*) узел-подсистема;
- (\*) узел-пользователь;
- (\*) узел-хранилище (или адресуемый элемент хранилища);
- узел-устройство печати;
- (\*) связь по информации пользователь-подсистема;
- (\*) связь по информации подсистема - подсистема;
- (\*) связь по информации подсистема - хранилище.

Структурированное содержание каждой модели, разрабатываемой при системном анализе позволяет иметь представление такой модели в виде семантической сети.

9. Множество документов «Пользовательские требования к решателю» или единый документ (если разные задачи ИПД решаются одним и тем же специалистом).

Ввиду того, что для формирования предыдущих моделей привлекались эксперты предметной области, которые в общем случае могут не быть конечными пользователями, задача формирования пользовательских требований к подсистемам-консультантам (решателям задач) остается почти такой же, как в традиционных разработках, за исключением того, что главные функции подсистем уже выявлены.

Документ включает «описания назначения решателя» (возможно совпадающее с названием задачи или ее текстовым описанием из Постановки

<sup>4</sup> а также знания об этиологиях и знания об осложнениях, знания о причинно-следственных связях между заболеваниями и наблюдениями - знания о клинических проявлениях и т.д.

задачи) и списка сформулированных требований к интерфейсу, дополнительным функциям, эксплуатационным характеристикам.

## 2. Формирование функциональных моделей решателя по ранним семантическим представлениям системы

Этап детального анализа функциональности каждой разрабатываемой подсистемы (приложения) позволяет выработать одно или несколько представлений программных подсистем, на основе которых может быть создан проект качественной реализации каждой подсистемы и проект интеграции подсистем в единую развивающуюся систему.

Наиболее распространенное представление функциональности приложения (функциональный взгляд на требования к решателю) - *модель подзадач*. (Представление процесса решения в виде системы взаимодействующих объектов не является естественным, т.к. причинно-следственные связи – не объекты и не атрибуты объектов). На основе описания назначения решателя, постановки задачи решателя и пользовательских требований к решателю аналитик строит модель разбиения функциональности решателя на более мелкие подфункции (подзадачи), возможно учитывая при этом структуру (онтологию) знаний и данных.

Структура обрабатываемых данных действительности и структура знаний о связи решения (искомого результата) с элементами наблюдаемой действительности (т.е. онтология предметной области) играют здесь определяющую роль (оставляя, впрочем, «место» для творчества аналитика при выборе им необходимого числа подзадач). Тем не менее, чем больше узлов в сети-онтологии, тем больше узлов в модели подзадач.

Например, в соответствии с фрагментом онтологии знаний «Если в ситуации присутствует некоторая нормальная реакция, то в множестве знания о нормальных реакциях присутствует такой элемент, у которого следствие совпадает со следствием нормальной реакции, в множестве вариантов нормы этого элемента содержится вариант нормы из нормальной реакции и для него выполнено условие на воздействующие факторы.» (из [Клещев и др., 2005]), алгоритм должен включать шаг (*подзадачу*) «Проверить гипотезу о том, что пациент здоров», организовав Цикл по наблюдавшимся признакам и для наблюдавшегося признака проверить гипотезу о том, что все его наблюдавшиеся значения могут иметь место у здорового пациента.

Универсальным представлением для алгоритма и функционального разбиения задачи может считаться иерархия подзадач с возможностью элементу иерархии иметь полустепень захода больше единицы. Элемент иерархии – подзадача некоторого уровня (последовательная или с-ветвлением).

Элемент иерархии верхнего уровня имеет в подчинении (чтобы обращаться к ним) элементы иерархии нижнего уровня, некоторые из которых состоят в «простом» подчинении, а некоторые – в «цикличном» подчинении (как тело цикла):

```
задача {
  название;
  (*) подзадача {~alt
    подзадача в «простом» подчинении
  OR
    подзадача в «цикличном» подчинении
    (название;
    [(*) подзадача]
  });
};
```

Если на более ранних этапах математик или аналитик или инженер знаний строит алгоритм на основе «метода решения задачи» и связей знаний с действительностью (онтологии знаний), то специальное разбиение на подзадачи не нужно, достаточно рассмотреть альтернативное представление (в виде семантической сети) алгоритмов подзадач.

Например, в соответствии с методом решения задачи «перебор всех возможных значений выходных данных (диагнозов – отдельных заболеваний). Для каждого заболевания выполняется решение прямой задачи – построение всех возможных вариантов развития причинно-следственных связей (ПСС)... и поиск среди них всех тех, которым соответствуют все наблюдаемые значения признаков пациента» (из [Москаленко, 2006]) модель подзадач (сформулированных упрощенно) в виде сети может выглядеть так<sup>5</sup>:

*постановка и объяснение диагноза*

*Получить данные наблюдений пациента (В)*

*Проверить гипотезу о том, что пациент здоров (П)*

*проработать каждый признак (Ц)*

*проверить гипотезу о соответствии значений признака норме (П)*

*запомнить значения признака, опровергающие гипотезу (П)*

*подытожить гипотезу о том, что пациент здоров. (В)*

*проработать каждое заболевание (Ц)*

*проверить выполнение необходимого условия для текущего заболевания (В)*

*проверить гипотезу о заболевании(П)*

*попытаться отвергнуть гипотезу о заболевании по не-КК-признаку (П)*

*проработать каждый КК-признак (Ц)*

*проверить возможность наблюдавшихся значений (В)*

*проработать каждый вариант проявления КК-признака (Ц)*

*составить динамику знаний варианту ПСС (П)*

*добавить анализ этого варианта в объяснение (П)*

<sup>5</sup> каждый сдвиг означает здесь подчиненность подзадачи, а буквами размечены подзадачи последовательные (П), с ветвлением (В) и в цикличном подчинении (Ц);

*подытожить результаты анализа всех вариантов в объяснении КК-признака (П)*

*подытожить результаты анализа всех признаков в объяснении заболевания (П)*

*выдать результат – множество диагнозов и их объяснение (П).*

Для построения этой семантической сети использованы сети этапа системного анализа. Для каждой подсистемы из *концептуальной архитектуры системы* строится своя «модель задач».

Традиционно вместе с разбиением функциональности подсистемы на подфункции (подзадачи) выполняется специфицирование входов и выходов каждой подфункции [Pressman, 2001]. *Расширенная модель подзадач* [Шалфеева, 2011] позволяет охватить информацию о каждой подзадаче и ее связях с обрабатываемыми данными (в виде сетей).

Например, подзадаче «ввод истории болезни» требуется записывать информацию от пользователя - паспортные данные, жалобы, текущее наблюдение и др. – в информационную структуру «история болезни». Подзадача формирования объяснения должна записать в соответствующий информационный ресурс (ИР) объяснение подтверждаемых и опровергаемых заболеваний. Подзадаче «Оценка гипотезы здоров» – требуется обращаться к ИБ за множеством пар <название признака, значение признака>. Подзадаче «проверить гипотезу о соответствии значений признака норме» надо сопоставлять один элемент этого множества – хранимым знаниям о нормальных значениях и т.д.

*Расширенная модель подзадач* добавляет в *модель подзадач* дуги-ссылки от подзадачи к компоненту обрабатываемой информации. Указаны связи подзадач «*проработать каждое...*» с компонентом-множеством, по которому организуется цикл (связь этой подзадачи с «циклично» подчиняемыми) [Шалфеева, 2011].

Пример. Узел «проработать каждый признак» имеет связь с узлом-множеством «наблюдения» структуры ИБ, узел «*проверить гипотезу о соответствии значений признака норме*» имеет связь получения информации из подструктуры «нормальные значения», узел «*подытожить гипотезу о том, что пациент здоров*» имеет связь записи информации в подструктуру «объяснение».

При разработке решателей (подсистем) для каждой интеллектуальной задачи эти модели являются ключевыми, т.к. по ним определяется внутренняя архитектура решателей и требования (спецификации) к разработке отдельных компонентов решателей. Сложность модели может быть «перенесена» на архитектуру, по мере необходимости и возможности либо архитектура либо модель подзадач может быть оптимизирована (например, для задачи диагностики может быть предложен метод сокращения списка гипотез для

проверки за счет предварительной проверки наблюдавшихся признаков в клинических картинах заболеваний-гипотез).

Единое декларативное представление всех моделей создаваемой системы [Грибова и др. 2012] позволяет модели функциональности решателя формировать в виде семантических сетей.

### **3. Формирование архитектурных представлений решателя по моделям анализа функциональности**

Ключевыми моделями для обеспечения сопровождаемости и управляемости ИПС являются архитектурные модели. К ним, в частности, относятся: архитектура каждого решателя, архитектура хранимой информации, проекты программных единиц. Достаточной для производства проектных представлений программных подсистем является *расширенная модель подзадач*.

Построение первой версии архитектуры решателя целесообразно по более компактной модели - *модели подзадач*, оно осуществляется по схеме: 1) сопоставить подзадачам – программные единицы (ПЕ) отвечающие за выполнение соответствующей подфункции; 2) сопоставить связям подчинения среди подзадач – передачи управления между ПЕ. В рамках технологии [Грибова и др. 2012] ПЕ - агенты (или блоки агентов), среди которых можно различать обрабатывающие, сопоставляемые обычно подзадачам в «листовых» узлах семантической сети, и управляющие ПЕ, сопоставляемые часто подзадачам в узлах с ненулевой полустепенью исхода.

Параллельно с архитектурой решателя строятся «внешние» спецификации каждой ПЕ этой архитектуры. *Спецификация ПЕ* показывает входные параметры (например, требуется передать как параметр элемент множества, для которого должна быть решена «циклично-подчиненная» подзадача) и необходимость формирования ответной информации о результате решения подзадачи. Кроме того, важной составной частью *спецификации* является указание входных и выходных нелокальных данных. Эта зависимость ПЕ от хранимых данных уже прописана в *расширенной модели подзадач*.

Проект (архитектура) хранимого ИР – описание его структуры и набор спецификаций операций, достаточных для манипулирования ими. Такой вариант абстрактного типа данных позволяет иметь гибкость при дальнейшей реализации системы (удобен, как для объектно-ориентированного, так и для процедурного и агентного подходов). Проект каждого типа обрабатываемого ресурса представляется как семантическая сеть.

Обрабатывающие агенты взаимодействуют с ИР через операции. Например, подзадаче «*Получить*

данные наблюдений пациента» – соответствует обрабатывающий агент, который реагирует на сообщение-задание «ввести» с параметром «ссылка на ИБ». Агент должен записывать информацию от пользователя в ИР, для этого необходим набор операций, таких как *записать паспортные данные*, *записать жалобы*, *записать текущее наблюдение* и др.

Подзадаче формирование объяснения – соответствует агент или блок «Начать формирование объяснения», который вносит в ИР-результат ссылку на ИБ (или пациента), чей диагноз начинает устанавливаться в виде объяснения (возможных невозможных заболеваний). Требуется операция над ИР «объяснение» «добавить ссылку на ИБ» с параметрами «ссылка на ИБ» (или номер ИБ или уник идентификатор пациента), «ссылка на выходной ИР – объяснение».

Достаточность представляемых (проектируемых) наборов операций над ресурсами проверяется их присутствием в спецификациях множества проектируемых ПЕ.

Для операций (вновь разрабатываемых абстрактных типов данных) формируются спецификации на разработку.

Семантические сети продолжают оставаться средством представления все этих архитектурных моделей.

Пример архитектуры решателя. В примере упрощенной агентной версии архитектуры решателя задачи диагностики (на рис. 4) подзадаче «Проверить гипотезу о том, что пациент здоров» – соответствует блок реагирования на сообщение-задание «оценить норму» агента «Оценка гипотезы здоров» и блок реагирования на сообщение-обратная связь (того же агента).

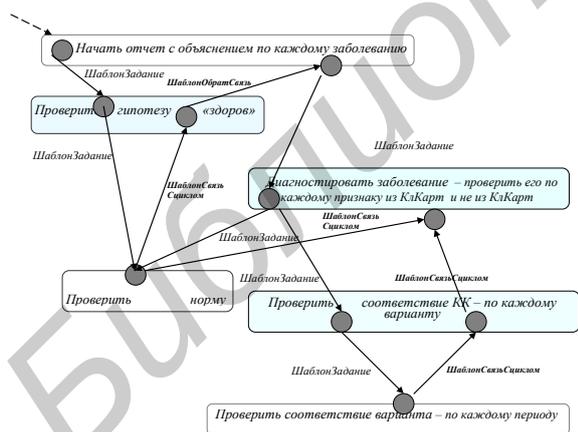


Рисунок 4 – Вариант архитектуры решателя задачи диагностики

Пример. Спецификация агента «проверить норму признака»:

«проверить норму признака» [название]

«проверяет, все ли значения указанного признака соответствуют норме» [описание]

Множество блоков продукции

Шаблон-задание [шаблон входного сообщения]

форматы используемых ИР

Онтология ИБ [формат первого используемого ИР]

Онтология описания нормы [формат второго используемого ИР]

форматы изменяемых ИР

шаблон-обратная связь [Шаблон выходного сообщения]

Пример архитектуры хранимого ИР «история болезни». Модель содержит структуру (см. выше) и список операций доступа:

«записать паспортные данные»,

«записать жалобы»,

«записать текущее наблюдение»,

«взять названия всех признаков»,

«взять уникальный идентификатор пациента»,

«взять все наблюдения указанного признака,

...

«записать предварительный диагноз».

Пример спецификации операции чтения над ресурсом ИБ:

«Взять названия всех признаков»

онтология ИБ [Формат входного ИР];

множество названий [Формат результата].

#### 4. Заключение

Разработка описанных моделей и методов является продолжением работы по формированию методологии системного анализа и моделирования произвольных сфер деятельности с интеллектуальными процессами и концепции комплексной автоматизации для систематической поддержки принятия ответственных решений специалистами.

Предложенный метод построения ключевых моделей процесса проектирования решателей интеллектуальных программных систем состоит в разработке трех групп семантических сетей с учетом их взаимовлияния друг на друга. Предложен необходимый набор моделей каждой группы и структуры семантических сетей для их представления. Эти модели обеспечивают необходимый минимум для построения управляемых интеллектуальных систем. Предложены методы перехода от предшествующей модели к последующей. Они являются базой для создания инструментария для автоматизации интеллектуальной профессиональной деятельности.

Работа выполнена при поддержке грантов РФФИ № «12-07-00179-а «Интернет-сервисы для преобразования семантических сетей» и ДВО РАН № 12-III-A-01И-019 «Мониторинг концептуальных информационных ресурсов для управления интеллектуальными системами».

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Голенков и др, 2012] Голенков В. В., Гулякина Н. А., Колб Д. Г. унификация подходов к разработке интеллектуальных систем в internet // // Материалы 22-ой международной - Крымской конференции «СВЧ-техника и телекоммуникационные технологии» (КрыМиКо'2012). Севастополь, 10—14 сентября 2012 г. Севастополь: Вебер, 2012. Т.2 С. 28-31.

[Грибова и др, 2012] Грибова В.В., Клещев А.С., Крылов Д.А., Москаленко Ф.М., Тимченко В.А., Шалфеева Е.А. Агентный подход к разработке интеллектуальных интернет-сервисов // Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'12». М.: Физматлит, 2012 –Т.1. с. 218-223.

[Грибова и др. 2011] Грибова В.В., Клещев А.С. Проблемы создания жизнеспособных интеллектуальных систем и методы их решения // International Journal "Information Technologies & Knowledge". Bulgaria. Sofia: ITHEA. 2011. Vol. 5, № 3. P. 250-258.

[Гулякина и др, 2012] Гулякина Н.А., Пивоварчик О.В., Лазуркин Д.А. Языки и технологии программирования, ориентированные на обработку семантических сетей // OSTIS-2012, стр. 221-228, Минск, БГУИР

[Данилов, 2006] Данилов Н. Н. Курс математической экономики: учеб. пособие / Н. Н. Данилов. - М. : Высшая школа, 2006. - 407 с.

[Клещев и др., 2012] Клещев А.С., Шалфеева Е.А. Системный анализ при автоматизации интеллектуальной профессиональной деятельности // XIII Национальная конференция по искусственному интеллекту с международным участием «КИИ-2012», 16–20 октября 2012 г. ISBN: 978-5-361-00182-8 Труды конференции, т.2. Белгород: Изд-во БГТУ, 2012. С.128–135.

[Клещев и др., 2010] Клещев А.С., Грибова В.В., Управление интеллектуальными системами. Известия РАН. Теории и системы управления. 2010. № 6. С. 122-137.

[Клещев и др., 2005] Клещев А.С., Черняховская М.Ю., Москаленко Ф.М. Модель онтологии предметной области «Медицинская диагностика». Часть 1. Неформальное описание и определение базовых терминов / Журнал НТИ - Серия 2. #12, 2005.

[Клещев и др., 2006] Клещев А.С., Черняховская М.Ю., Москаленко Ф.М. Модель онтологии предметной области "медицинская диагностика". Часть 2. Формальное описание причинно-следственных связей, причин значений признаков и причин заболеваний / Журнал НТИ, Серия 2, №2, 2006.

[Москаленко, 2011] Москаленко Ф.М. Задача медицинской диагностики, методы и алгоритм её решения // Владивосток: ИАПУ ДВО РАН, 2011. 40с.

[Москаленко, 2006] Москаленко Ф.М. Алгоритм диагностики, основанный на реальной онтологии медицины, для многопроцессорной ЭВМ / Доклад РАСО2006, 2006. Издатель: Институт проблем управления им. В.А. Трапезникова РАН

[Черняховская и др., 2007] Черняховская М. Ю., В. Я. Мельников, Л. П. Догадова, В. И. Негода, Москаленко Ф.М. Формальное представление знаний о конъюнктивитах // ИАПУ ДВО РАН, 2007. 51с.

[Шалфеева, 2011] Шалфеева Е.А. Использование онтологий при разработке управляемых интеллектуальных систем. Владивосток: ИАПУ ДВО РАН, 2011.

[Штрик, 1993] Штрик А.А. Эволюция методологий разработки программного обеспечения, поддерживаемых зарубежными CASE-системами. 1993 г.

[Martin 1989] Martin, J. Information Engineering: Book I - Introduction. Englewood, NJ: Prentice Hall, 1989.

[Pressman, 2001] Pressman R.S. Software Engineering: Practitioner's Approach. Fifth edition. McGraw-Hill Inc., 2001. 860 p.

\* *The Institute of Automation and Control Processes, Vladivostok, Russia*

shalf@iacp.dvo.ru

The set of semantic representations of intelligent software, their contents and structure is offered. The construction method for the problem solver analysis and design models on the basis of domain models is offered. These models are necessary for development of managed intelligent software. These methods form the base for development of tools supporting of intelligent software developers.

## INTRODUCTION

Nowdays the methods of automation of intellectual activity and of the help in acceptance of crucial decisions are being developed. In such spheres of economy decision-making is carried out with use of constantly updated knowledge.

The success of complex automation of professional activity is influenced essentially by the next results of its system and domain analysis: knowledge' and data' ontologies, activity model, and conceptual model of system.

The research objective is development of a method of creation of main designing models of intelligent software' problem solver with use of semantic models of the system analysis of professional activity being automated.

## MAIN PART (REPORT THESES)

First all main semantic models of the domain add system analysis are presented. The formation of functional models of a problem solver on the base of early semantic representations of intelligent system is shown in the second part. The formation of a problem solver' architectural representations from the models of the analysis of its functionality is shown In the third part.

## CONCLUSION

The offered method of creation of key models of designing of intelligent software' problem solver consists in development of three groups of semantic networks. The necessary set of models of each group and structure of semantic networks for their representation are offered. These models provide a necessary minimum for creation of managed intelligent software. The methods for transformation of previous model to the subsequent are offered.

## THE SEMANTIC MODELS FOR REPRESENTATION OF INTELLIGENT SOFTWARE' PROBLEM SOLVER

Shalfeeva E. \*