

**СЛОЖНЫЕ ТЕХНИЧЕСКИЕ  
СИСТЕМЫ УПРАВЛЕНИЯ**

УДК 621.865.8–5.001.5

**ВИРТУАЛЬНАЯ СРЕДА ДЛЯ ПРОЕКТИРОВАНИЯ СИСТЕМЫ  
УПРАВЛЕНИЯ РОБОТОМ СЕРВИСНОГО ОБСЛУЖИВАНИЯ\***

© 1996 г. В. Н. Захаров, Л. В. Литвинцева, С. В. Ульянов

Москва, ВЦ РАН; Переславль-Залесский, Исследовательский центр искусственного интеллекта  
Ин-та программных систем РАН; Москва, АОЗТ “Робоцентр”

Поступила в редакцию 28.11.95 г.

Отмечается сложность задачи проектирования системы управления роботом сервисного обслуживания, способного к автономному перемещению в реальном мире, ограниченном общим планом передвижения. Подчеркивается необходимость имитационного моделирования действий робота с использованием средств виртуальной реальности. Вводится понятие виртуальной поддержки процессов проектирования и описывается общая структура взаимодействия оператора и робота с виртуальной средой проектирования. Рассматриваются основные модули интерфейса и системы виртуальной поддержки, язык описания динамических состояний виртуальной реальности и блок моделирования поведения робота сервисного обслуживания в виртуальной среде.

**Введение.** К числу мобильных роботов, способных к автономному перемещению в динамическом внешнем мире, принадлежит важный их класс – роботы сервисного обслуживания (РСО). Эти роботы с успехом применяются на производстве и в строительстве при организации погрузочно-разгрузочных работ, для уборки (очистки) производственных помещений, для снабжения производственных участков материалами и полуфабрикатами, в медицинском обслуживании (в госпиталях и т.п.) и во многих других сферах. Роботы сервисного обслуживания выделяются в отдельный класс в силу своей специфической особенности функционирования, состоящей в том, что в процессе выполнения своей главной задачи (реализации функций обслуживания) они вынуждены перемещаться хоть и в ограниченном (обычно планами помещений) пространстве, но в условиях возможности появления непредвиденных препятствий как статических (например, штабель временно выгруженных для дальнейшей перевозки изделий), так и динамических (например, таких же роботов). Это приводит к необходимости решения задачи навигации средствами системы управления робота в динамически меняющемся внешнем мире.

Разработка систем управления автономных РСО является сложной научно-технической задачей. Управляющие системы такого типа относятся к интеллектуальным системам, или системам, построенным с использованием новейших методов и средств современной информационной технологии, в частности, методам обработки знаний. При проектировании модулей управления слож-

ными системами, функционирующими в реальном времени в динамически меняющемся внешнем мире, требуется промоделировать большое количество ситуаций и алгоритмов принятия решений в них. Натурные испытания подобных систем достаточно дороги, связаны с возможностью поломок и проведением соответствующих восстановительных работ, а иногда и просто невозможны (например, в труднодоступных средах). Поэтому в последнее время разрабатываются новые подходы к проектированию интеллектуальных управляющих систем, основанных на использовании идей когнитивной графики и виртуальной реальности.

В данной работе приведены некоторые результаты исследований в области организации систем виртуальной реальности для целей проектирования интеллектуальных управляющих систем РСО.

**1. Особенности организации автономного мобильного РСО.** В качестве примера рассмотрим организацию РСО из [1], ориентированную на выполнение роботом функции обслуживания путем автономного перемещения по коридорам и комнатам в многоэтажном здании, оборудованном лифтом, по имеющимся в памяти робота планам поэтажного расположения помещений. РСО с такими возможностями можно использовать для перевозки небольших партий грузов или передачи посылок (например, документов в офисе или лекарств в больнице). Кроме этих основных функций РСО способен выполнять функции сопровождающего, охранника, а по окончании рабочего дня ему можно поручить уборку помещений (например, мытье пола).

\*Работа выполнена при поддержке РФФИ по грантам №95-01-00729а и 95-01-00730а.

С конструктивной точки зрения РСО из [1] представляет собой автономное мобильное транспортное средство в виде специальной тележки с двумя степенями свободы с установленным на ней трехзвенным манипулятором с пятью степенями свободы, оканчивающимся специальным трехпальцевым схватом. Снабженный многоуровневой иерархической интеллектуальной системой управления РСО способен к автономному перемещению в среде в условиях появления неожиданных препятствий. При необходимости переместиться из одного помещения в другое с целью выполнения полученного задания РСО решает навигационную задачу в реальном масштабе времени. В процессе ее реализации роботу часто приходится выполнять операции открывания и закрывания дверей, которые, как указано в [1], могут рассматриваться в качестве тестовой задачи эволюционного моделирования для оценки эффективности различных генетических алгоритмов, используемых при обучении. Задача открывания двери разбивается на две подзадачи: планирование и реализация перемещения робота в правильную позицию, которую необходимо занять перед дверью (задача, напоминающая парковку автомобиля) и планирование и реализация действий манипулятора, обеспечивающих открывание двери.

Система управления РСО выполнена в соответствии с пятью принципами организации интеллектуальных управляющих систем. Структура управления является многоуровневой, построенной по иерархическому принципу. Главными решающими подсистемами верхних уровней иерархии являются: система распознавания окружающей обстановки, включая анализ текущего положения РСО и наблюдаемое препятствие; система планирования перемещений и действий РСО, включая обход препятствий, и система обучения, позволяющая корректировать поведение РСО на базе накопленного опыта выполнения сервисных функций. Информация с главных решающих подсистем поступает в систему управления перемещением платформы и манипулятора робота для формирования соответствующих команд на исполнительные механизмы.

**2. Виртуальная среда проектирования.** Чтобы лучше представить себе это понятие, рассмотрим пример, аналогичный приведенному в [2], применительно к проблеме проектирования управляющих систем автономных РСО.

Представим, что разработчик робота находится в просторной комнате с тремя огромными расположенными перед ним экранами. Разработчик надевает специальные очки, гасится свет, включается компьютер и в мгновение ока комната превращается в производственное помещение, в котором действует проектируемый робот. В ко-

ридоре открываются двери и в проеме появляется робот. Он останавливается, затем разворачивается и направляется в другой конец коридора. С целью проверки динамики поведения РСО разработчик может “передвинуть” стоящую у стены колонну и посмотреть, как робот станет обходить возникшее на его пути препятствие. Пусть при обходе препятствия робот случайно задевает “локтем” манипулятора стоящую сбоку колонну. Колонна с грохотом обрушивается, а манипулятор робота безжизненно повисает на полуоборванных проводах. Что-то не сработало. Что именно? Разработчик включает систему принятия роботом решений. На экране появляется блок-схема управления, иллюстрирующая процесс принятия решения. Все ясно: ошибка в формировании функций принадлежности меры близости, из-за чего и произошло столкновение в наблюдаемом виртуальном мире. В реальной действительности такой эксперимент обошелся бы довольно дорого. Разработчик исправляет ошибку и приступает к испытанию динамики поведения робота в новой ситуации.

Проблемы разработки систем виртуальной реальности (VR-систем), подобных описанной, рассмотрены в обзоре [2]. Цель данной публикации – определить контуры необходимого программного обеспечения, поддерживающего процессы проектирования сложных робототехнических систем.

Под виртуальной поддержкой процессов проектирования РСО подразумевается комплекс вспомогательных программных средств, который, в совокупности с программно-аппаратным обеспечением, необходимым для реализации виртуальной проблемной среды (в рассматриваемом случае пространства действий РСО, ограниченного планами помещений), позволяет пользователю активно вмешиваться в динамику поведения робота и его внешнего окружения с целью оценки эффективности разрабатываемых алгоритмов обработки знаний и логических выводов интеллектуальных управляющих систем.

Ядром системы взаимодействия с VR-средой является интерфейс (рис. 1), позволяющий оператору и роботу активно взаимодействовать с модельной средой в прямом контакте. Интерфейс состоит из трех основных компонент: блока взаимодействия робота с VR-средой; блока взаимодействия оператора с VR-средой и блока взаимодействия оператора и РСО.

Задача проектирования VR-среды с функционирующим в ней роботом относится к классу задач построения искусственного агента, выполняющего функцию взаимодействия с VR. Таким искусственным агентом может быть модельный робот сервисного обслуживания, находящийся в виртуальной среде и взаимодействующий со сре-

дой и с оператором. PCO должен обладать не только способностями, необходимыми для ориентации и поведения в ВР, но и более сложными специальными проблемно-ориентированными стратегиями поведения. Он должен обладать развитой способностью общаться с реальным пользователем: понимать естественно-языковую речь (или тексты), жесты и даже мимику лица [3].

В [4] Badler разработал вариант искусственно-го агента Джека, живущего и взаимодействующего с ВР-средой. Джек может заменить участие человека в ВР-среде, если по каким-либо причинам для человека это невозможно или нежелательно.

В [5] разработан прототип системы, использующей виртуальную реальность для обучения (выполнения ряда гимнастических процедур на спортивных снарядах) посредством совместной деятельности. Демонстрационная версия системы происходит в виртуальном мире, состоящем из зала, в котором находится различное спортивное оборудование. В этом мире находятся и общаются между собой три реальных агента – Энди, Йоко, Марк, один искусственный агент – Майк и новичок Мерл. Мерл в процессе общения с ними постигает, как Майк работает на спортивных снарядах. Агенты общаются на простом английском языке и взаимодействуют между собой посредством компьютерной сети. Агенты находятся в различных смежных комнатах (однако, это не принципиально, и они могут находиться как угодно далеко друг от друга). Каждый агент работает на графической станции типа Silicon Graphics и имеет свое отображение в виртуальной среде – так называемое “виртуальное тело”. Пользователь смотрит на экран и видит изображение среды вместе с агентами, как если бы он находился непосредственно в зале.

Архитектура такого взаимодействия включает: модуль графического изображения, модуль физического моделирования, интерфейс навигации и положения, модуль генерации звуковых сигналов, трехмерную модель среды.

Модуль графического изображения воспроизводит на экране рабочей станции виртуальный мир. Параллельно и почти синхронно модуль звуковой передачи (поддерживаемый специальным hardware) смешивает звуки от разных источников и воспроизводит их агенту посредством стереонаушников. Модуль физического моделирования вычисляет движение виртуального тела агента, который управляет моделированием посредством интерфейса навигации и положения. Основной результат моделирования – информация о положении – передается другим агентам. Система моделирования управляет также генерацией звуковых эффектов при движении виртуального тела. Эта информация посылается также другим агентам. Трехмерная модель мира содержит

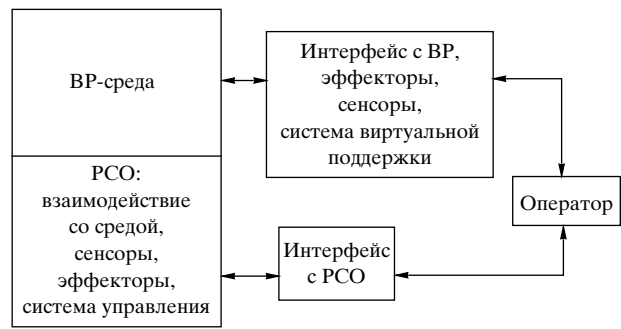


Рис. 1.

иерархические описания каждого объекта, в том числе виртуальные тела агентов. Большая часть информации в модели статична. Описание каждого движущегося или изменяемого объекта разделяется на статическую часть плюс множество параметров, которые изменяются, когда объект движется.

При анимации виртуальных сцен используется физическая модель, включающая такие параметры как масса тела, момент инерции, внешние силы, действующие на тело (например, гравитация). Согласно законам движения моделируется поведение каждого объекта в сцене. Законы движения, однако, могут описывать движения определенного типа (например, так называемую “картонную физику”). Представляет интерес анимация виртуального тела (например, анимация механических структур робота).

В Японии разработками ВР-систем для целей проектирования заняты коллективы, поддерживаемые такими фирмами, как NEC (Nippon Electric Company), ATR (Advanced Telecommunication Research), TEPCO (Tokyo Electric Power Company’s & Communications Research Center) и др. Исследователи от фирмы TEPCO создали ВР-систему для анализа программного обеспечения. В системе строятся трехмерные изображения функционирующих модулей и их взаимосвязей. Таким образом можно воочию увидеть трассировку данных между подзадачами в программе.

Японская фирма Fujita совместно с UPL ведут проект по использованию ВР-технологий для разработки системы дистанционного управления коллективом из роботов. В Германии во Фраунгоферовском институте промышленности и автоматизации (г. Штутгарт) создаются ВР-рабочие станции, позволяющие оператору проводить моделирование поведения робота, погружаясь в его виртуальный мир.

Нетрудно видеть, что создание даже демонстрационного образца системы взаимодействия с ВР требует больших затрат и решения таких проблем, как: разработка программного и аппаратного обеспечения взаимодействия с ВР-средой,

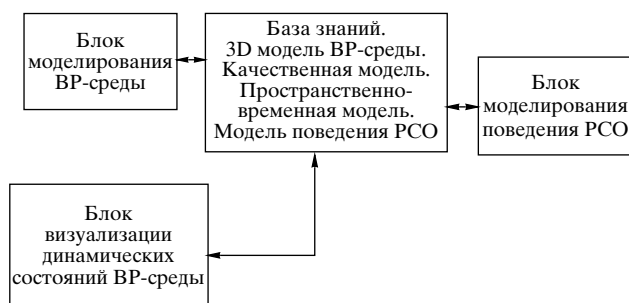


Рис. 2.

моделирование и анимация VR-среды в реальном времени, построение искусственного агента (включая понимание и генерацию речи на естественном языке, распознавание жестов, сенсорные проблемы определения положений в пространстве), разработка языка описания динамических состояний VR-среды, разработка блока имитационного моделирования поведения агента в VR и др.

**3. Архитектура системы виртуальной поддержки.** Рассмотрим задачу исследования алгоритмов поведения PCO с использованием средств системы виртуальной поддержки.

На рис. 2 показана архитектура системы виртуальной поддержки. Основу системы составляют блоки моделирования среды, моделирования поведения PCO в VR-среде и блок визуализации. На каждом этапе такого моделирования оператор оценивает состояние мира и сценарии поведения PCO. Трехмерная модель мира содержит всю информацию, в том числе и числовую, для графического отображения сцены. Однако она слишком детально для блока когнитивного моделирования. Для этого используется блок качественной модели мира. В нем содержатся такие понятия как пространственно-временные отношения (типа “в”, “на”, “близко”). Выявляется также информация о центре координат и ориентации каждого объекта. Кроме того, качественная модель содержит информацию о физических действиях, которые может совершать PCO в среде.

PCO реализует в своем поведении так называемую парадигму целенаправленных действий. Он может выполнять следующие физические и ментальные действия: переместиться из локализации  $A$  в локализацию  $B$ ; изменить скорость, в том числе и остановиться; повернуть (изменить направление) на  $n$  единиц; двигаться в координату  $\langle x, y \rangle$ ; переместить объект из локализации  $A$  в локализацию  $B$ ; произнести некоторую фразу (или выдать некоторое текстовое сообщение).

Тактика поведения робота в среде может быть задана с помощью правил. Например, “если PCO в палате и пациент близко, то необходимо осуществить процедуру 1”. Указанные правила позволяют моделировать определенные типы поведе-

ния и взаимодействия между PCO и другими субъектами среды.

Блоки моделирования среды и визуализации VR-среды являются центральными, поэтому в данной статье им уделяется особое внимание.

Рассмотрим задачу построения языка описания сцен и динамики их изменений на основе реализуемых действий и сценариев для блоков моделирования состояния VR-среды и поведения PCO в ней.

**4. Язык описания сцен и динамики их изменений в виртуальной среде.** Будем рассматривать следующий виртуальный мир для робота сервисного обслуживания. В рамках некоторого ограниченного пространства находится определенная совокупность объектов. В этом пространстве действуют один или несколько PCO, совершающих физические действия и изменяющих пространственные отношения между объектами. Кроме того, могут происходить внешние воздействия на объекты среды, не связанные с действиями PCO (например, природные силы и т.п.). Поставим задачу определить изменения, происходящие в таком мире и разработать язык описания таких изменений. Назовем описанную нами среду динамическим миром и будем рассматривать состояния такого мира в дискретные моменты времени. Язык описания динамических состояний виртуальной среды LVR (Language for Virtual Reality) включает в себя язык описания сцен, их динамических изменений, а также язык описания поведения PCO на основе действий и сценариев.

**4.1. Язык описания сцен. Алфавит.** Будем рассматривать счетное множество имен  $N = \{n1, n2, \dots\}$  и следующие конечные множества сущностей, характеризующихся наличием имени и своей глубинной структурой:

множество объектов  $O = \{o1, o2, \dots, ok\}$ ;

множество замкнутых ограниченных пространств  $LOC = \{loc1, loc2, \dots, lock\}$ ;

множество субъектов (или акторов) действий  $A = \{a1, a2, \dots, ak\}$ ;

множество времен  $T = \{t1, t2, t3, \dots, tk\}$  (отметим, что время  $ti$  может быть как точкой (моментом), так и интервалом на временной шкале в зависимости от того, какая глубинная структура соответствует имени  $ti$ );

множество точек трехмерного пространства  $P(L) = \{p1, p2, \dots, pk\}$ , состоящее из троек вида  $\langle x, y, z \rangle$ , где  $x, y, z$  – координаты точки  $pi$  в системе координат  $L$ ;

множество базовых действий  $Do = \{do1, do2, \dots, dok\}$ ;

множество внешних или внутренних воздействий  $E = \{e1, e2, \dots, ek\}$ ;

множество действий  $D = \{d1, d2, \dots, dk\}$ , представляющих собой определяемые временные последовательности из базовых действий.

С каждым Loc'ом будем связывать множество пространственных сцен  $S(\text{Loc}) = \{s1, s2, \dots, sk\}$ , отражающее изменения, происходящие в этом пространстве.

Определим пространственную сцену как множество объектов и пространственных отношений между ними в момент времени  $t$ . Тогда изменение в заданном Loc'e будем описывать как переход от пары  $\langle s0, t0 \rangle$  к паре  $\langle s1, t1 \rangle$ . Определим сценарий как процедуру, описывающую изменения в пространственных сценах и состоящую из последовательности действий во времени и ряда специальных операторов (условных типа "если, то, иначе" и операторов обработки глубинных структур). Зададим множество сценариев  $SC = \{SC1, SC2, \dots, SCk\}$ .

Рассмотрим следующие множества. Множество временных отношений  $R_{\text{time}} = \{R1, R2, \dots, Rk\}$ , заданных на множествах  $T \times T, S \times S, Do \times Do, D \times D, E \times E, SC \times SC$ . Множество пространственных отношений  $R_{\text{space}} = \{r1, r2, \dots, rk\}$ , заданных на множествах  $(O \cup A) \times (O \cup A)$ , где  $\cup$  – операция объединения,  $P \times P, \text{Loc} \times \text{Loc}$ . Множество каузальных отношений  $R_{\text{cause}} = \{q1, q2, \dots, qk\}$ , заданных на множествах  $D \times D, S \times S, SC \times SC$ . Множество качественных отношений (типа "иметь размер, вес", "обладать свойством хрупкости" и т.п.).

Рассмотрим также следующие операторы:  $\rightarrow$  – переход из состояния  $\langle s, t \rangle$  в состояние  $\langle s1, t1 \rangle$ ; #name – присвоить имя; #realize – исполнить действие, сценарий, а также последовательность специальных операторов, запускаемых данным оператором; #ext-forse – эффект внешнего воздействия; #int-forse – эффект внутреннего воздействия. Результатом выполнения оператора  $\rightarrow$  является переход среды из состояния  $\langle s, t \rangle$  в состояние  $\langle s1, t1 \rangle$ . Результатом выполнения оператора #name является генерация нового имени  $b$  текущей ППФ, такого, что пересечение  $N$  и  $b$  пусто, а затем расширение множества  $N$  до  $N \cup b$ . Результатом выполнения оператора #realize является порождение множества переходов  $\{\langle s, t \rangle \dots, \langle si, ti \rangle\}$ , характеризующих исполнение актором заданного действия или сценария в начальной ситуации  $\langle s, t \rangle$ . Результатом выполнения операторов #extforse, #int-forse является порождение множества переходов  $\{\langle s, t \rangle \dots \langle si, ti \rangle\}$ , характеризующих эффект внешнего или внутреннего воздействия в начальной ситуации  $\langle s, t \rangle$ .

**Синтаксис.** Опишем множество Syn всех правильно построенных формул (ППФ) языка с помощью следующих правил:

1.  $R \in R_{\text{time}}; t1, t2 \in T; s1, s2 \in S; do1, do2 \in Do; d1, d2 \in D; e1, e2 \in E; SC1, SC2 \in SC$ . Тогда  $(t1 R$

$t2), (s1 R s2), (do1 R do2), (d1 R d2), (e1 R e2), (SC1 R SC2) – ППФ; \langle s, t \rangle – ППФ.$

2.  $R \in R_{\text{cause}}; s1, s2 \in S; d1, d2 \in D; SC1, SC2 \in SC$ . Тогда  $(s1 R s2), (d1 R d2), (SC1 R SC2) – ППФ.$

3.  $R \in R_{\text{space}}; p1, p2 \in P; X, Y \in O \cup A; loc1, loc2 \in \text{Loc}$ . Тогда  $(p1 R p2), (X R Y), (loc1 R loc2) – ППФ.$

4.  $d \in D \cup Do; e \in E; SC \in S$ . Тогда  $(\#realized), (\#realize SC), (\#ext-forse e), (\#int-forse e) – ППФ.$

5. Если  $\alpha – ППФ, (\langle s, t \rangle \alpha) – ППФ.$

6.  $(\langle s, t \rangle \rightarrow \langle s1, t1 \rangle) – ППФ.$

7. Если  $\alpha$  ППФ вида  $(\langle s, t \rangle \rightarrow \langle s1, t1 \rangle)$ , то  $(\alpha \rightarrow \langle s2, t2 \rangle) – ППФ.$

8. Если  $\alpha – ППФ, то (\#name \alpha) – ППФ.$

9. Если  $\alpha$  и  $\beta – ППФ, то (\alpha \& \beta) – ППФ, где \& – операция логического умножения.$

10. Если  $\alpha$  ППФ, то  $\exists(s)\alpha – ППФ$  и  $\exists(t) \alpha – ППФ, \forall(s)\alpha – ППФ$  и  $\forall(t) \alpha – ППФ, где \exists – квантор существования, \forall – квантор всеобщности, s принадлежит множеству сцен S, t – множеству времен T.$

**4.2. Язык описания действий и сценариев.** Для описания действий будем использовать следующие отношения: agent – действие реализуется субъектом действия; time – действие реализуется во время  $t$  (интервал или момент);  $d$ -loc – действие локализуется в пространстве loc; obj – действие направлено на объект  $oj$ ; instr – действие имеет инструмент действия  $oj$ ; recip – действие имеет получателя (реципиента); from – действие направлено от исходной пространственной точки или loc; to – действие направлено к конечной пространственной точке или loc; conditions – действие требует необходимых условий реализации  $\phi i$  (где  $\phi i – ППФ$ ); results – действие приводит к  $\phi i$ ;  $p$ -change – действие меняет пространственное отношение между объектами  $oi$  и  $oj$ ; purpose – действие имеет цель  $\phi i$ .

Для описания сценариев будем использовать следующие отношения: agent – сценарий реализуется субъектом действия; time – сценарий реализуется во время  $t$  (интервал); str – сценарий имеет структуру  $\phi i$ , где  $\phi i – формула, выражающая последовательность действий сценария, заданную посредством временных отношений; conditions – сценарий требует необходимых условий реализации  $\phi i$  (где  $\phi i – ППФ$ ); results – сценарий приводит к  $\phi i$ ; add-results – сценарий имеет побочный результат  $\phi j$ ; purpose – сценарий имеет цель  $\phi i$ .$

**Синтаксис.** Расширим множество ППФ языка LVR, добавив следующие правила.

1)  $(x R y) – ППФ, где x принадлежит множеству действий или сценариев, R – отношение "agent", a y принадлежит множеству субъектов действий.$

2)  $(x R y)$  – ППФ, где  $x$  принадлежит множеству действий или сценариев,  $R$  – отношение “time”, а  $y$  принадлежит множеству времен.

3)  $(x R y)$  – ППФ, где  $x$  принадлежит множеству действий,  $R$  – отношение “d-loc” или “obj”, “instr”, “recip”, “from”, “to”, а  $y$  принадлежит множеству локализаций или объектов, субъектов, точек трехмерного пространства соответственно.

4)  $(x R y)$  – ППФ, где  $x$  принадлежит множеству действий или сценариев,  $R$  – отношение (“conditions” или “results”, “add-results”, “purpose”), а  $y$  принадлежит множеству ППФ.

5)  $(x R y; y1)$  – ППФ, где  $x$  принадлежит множеству действий,  $R$  – отношение “p-change”, а  $y, y1$  принадлежат множеству объектов.

**Семантика языка LVR.** Как было сказано выше, каждый элемент языка LVR (за исключением множества имен) характеризуется двумя признаками: именем и глубинной структурой, отражающей семантику данного элемента в модели предметной области. Данные глубинные структуры содержатся в базе знаний (которые используются вышеприведенными операторами выполнения действий и сценариев, а также специальными операторами обработки). Для описания глубинных структур используется фреймовый язык представления знаний [6].

**4.3 Модель интерпретации LVR.** Модель интерпретации представляет собой модель вида  $\langle Z, H, V \rangle$ , где  $Z$  – некое множество интерпретируемых значений;  $H$  – правила отображения  $A \rightarrow Z$  ( $A$  – множество базовых элементов языка, алфавит), приписывающие каждому элементу из  $A$  некоторое интерпретирующее значение;  $V$  – правила интерпретации, приписывающие любой ППФ, состоящей из базовых элементов, некоторое интерпретирующее значение.

Для определения истинностных значений ППФ языка LVR будем рассматривать следующую модель  $M = \{Z, H, V\}$ , где  $Z = W \times T$ ,  $W$  – множество всех возможных сцен,  $T$  – множество времен. Таким образом, каждую формулу  $\Phi$  языка LVR будем оценивать по отношению к каждой паре  $\langle s, t \rangle$ ;  $V$  – функция, определенная на множестве  $\{O \cup A \cup S \cup SC \cup D \cup Do \cup R_{\text{cause}} \cup R_{\text{time}} \cup R_{\text{cause}}\}$  такая, что для каждой ППФ  $\phi$  значение  $V(\phi)$  принадлежит  $(W \times T)$  либо  $(W \times T) \times \dots \times (W \times T)$ .

С выражениями языка LVR связаны два процесса: 1) интерпретации выражений языка, реализуемой блоком моделирования поведения РСО, результатом которой является построение последовательности переходов  $\langle s_0, t_0 \rangle, \dots \langle s_n, t_n \rangle$ ; 2) отображение последовательности переходов в графические представления (последовательность виртуальных сцен), реализуемой блоком визуализации.

**5. Логические модели динамических действий, процессов и состояний в виртуальной среде.** В рамках логических моделей предпринимаются попытки описать на логическом языке различные динамические явления: процессы, действия и т.д. Для представления знаний о динамических явлениях существует несколько подходов [7–10]. Первым по времени была ситуационная схема [7], предназначенная исключительно для логического описания действий. Примитивами в такой модели являлись: ситуация – моментальный слепок мира в целом, и действия – элементы, благодаря которым происходит переход от одной ситуации к другой. Существенным недостатком ситуационного исчисления является невозможность рассуждать о деятельности в физическом мире из-за того, что предлагаемая онтология очень бедна.

В [11] предпринята попытка более полного описания динамики действий в рамках так называемой динамической логики действий. Действия рассматриваются как некоторые сущности, трансформирующие состояния мира и приносящие изменения или препятствующие изменениям. Для описания свойств действий в [11] предлагается использовать так называемую пропозициональную динамическую логику (PDL – propositional dynamic logic) [12]. Действия в этой логике описываются с помощью PDL-программ, отражающих степень изменения состояний. Дальнейшим развитием этой концепции является динамическая логика действий, описываемая в [11].

При описании действий в явном виде рассматриваются статические и динамические аспекты мира. Каждое состояние описывается различными фактами (например, “дверь открыта”, “свет распространяется”) и может быть изменено действиями (например, “закрыть дверь”, “выключить свет”). Факты являются статическими элементами, в то время как действия относятся к определенной степени динамики. Поэтому представляется неадекватным использовать одну и ту же синтаксическую категорию для их описания.

В [11] используются две синтаксические категории: формулы для описания фактов (статических элементов) и программы, которые используются для представления типов действий. Таким образом, статические и динамические аспекты легко разделяются. Эти идеи соответствуют идеям von Wright’a [13] с некоторыми усовершенствованиями.

В рассмотренных моделях действия описываются на уровне пропозициональной модальной логики, которые не отражают такой важный аспект действий, как их динамический характер. Для отражения этого свойства необходимы более сложные модели и представления.

В следующем разделе рассматривается дальнейшее развитие логического подхода к описа-

нию динамических действий и процессов. Логика действий в системе моделирования поведения робота сервисного обслуживания в виртуальной среде представляет собой тройку  $\langle LVR, Ax, Pv \rangle$ , где LVR – рассмотренный выше язык описания динамических состояний, Ax – множество аксиом, Pv – правила вывода.

**Аксиоматика.** Аксиоматика (множество Ax) включает системы аксиом, отражающих свойства пространственно-временных отношений, действий и операторов, рассмотренных нами выше. Полная аксиоматика пространственно-временных отношений описана в [14]. Здесь представлены некоторые примеры аксиом, используемые в блоках моделирования ВР-среды и поведения РСО: 1)  $(o1 \text{ “находиться на” } o2) \models (o2 \text{ “иметь точку опоры” } o3)$ ; 2)  $(o1 \text{ “находиться в” } o2) \& (o1 \text{ “иметь размер” } L) \& (o2 \text{ “иметь размер” } S) \models (s \text{ value False})$ , где  $L$  – “большой” и  $S$  – “маленький”, “value” – специальное отношение, характеризующее корректность сцены  $s$  с точки зрения определенной физической модели мира.

Приведем несколько схем аксиом, описывающих действия во времени и пространстве: 4)  $(d1 \text{ Rcause } d2) \models (d1 \text{ R } d2)$ , где  $R$  отношение “быть раньше”; 5)  $(d1 \text{ R } d2), (d2 \text{ R } d3) \models (d1 \text{ R } d3)$ ,  $R \in \{Rcause\}$ ; 6)  $(d1 \text{ agent } s1) \& (d1 \text{ d-loc } loc0) \& (d1 \text{ obj } o1) \models (s1 \text{ r } o1) \& (o1 \text{ r1 } loc0) \& (s1 \text{ r1 } loc0)$ , где  $r$  – отношение “находиться около”,  $r1$  – “находиться в”; 7)  $(d1 \text{ conditions } d2) \models (d2 \text{ “раньше” } d1)$ ; 8)  $(d1 \text{ p-change } o1; o2) \& (d1 \text{ d-loc } loc0) \& (o1 \text{ r1 } loc0) \& (o2 \text{ r1 } loc0) \& (o3 \text{ r1 } loc0) \& \dots \& (ok \text{ r1 } loc0) \models (d1 \text{ not p-change } oi; oj)$ , где  $i, j = 3, \dots k$ .

Данная аксиома утверждает следующий важный факт, связанный с действиями: если действие изменяет пространственное отношение между объектами  $o1$  и  $o2$  в сцене  $loc$ , то считаем, что остальные пространственные отношения в последующей сцене после выполнения действия не изменяются.

Рассмотрим примеры аксиом, описывающих состояния, действия и операторы:

9)  $(\langle s0, t0 \rangle \#realize d) \models \exists(t1, t2, \dots tk) \text{ и } \exists(s1, s2, \dots sk)$

$(\langle s0, t0 \rangle \longrightarrow \langle s1, t1 \rangle \longrightarrow \langle s2, t2 \rangle \longrightarrow \dots \longrightarrow \langle sk, tk \rangle)$ ;

10)  $(\langle s0, t0 \rangle \#realize SC) \models \exists(t1, t2, \dots tk) \text{ и } \exists(s1, s2, \dots sk)$ ,

$(\langle s0, t0 \rangle \longrightarrow \langle s1, t1 \rangle \longrightarrow \langle s2, t2 \rangle \longrightarrow \dots \longrightarrow \langle sk, tk \rangle)$  (здесь  $\exists$  – квантор существования);

11)  $(\langle s, t \rangle \#realize (d1 \& d2)) \models (\langle s, t \rangle \#realize d1) \& (\langle s, t \rangle \#realize d2)$ ;

12)  $(\langle s, t \rangle \#realize (d1 \text{ R } d2)) \models (\langle s, t \rangle \#realize d1) \text{ R } (\langle s, t \rangle \#realize d2)$ , где  $R$  принадлежит  $\{Rtime \cup Rcause\}$ .

**6. Пример работы блока моделирования РСО в ВР-среде.** Рассмотрим работу блока моделиро-

вания поведения РСО, на вход которого поступило следующее выражение языка:  $(\langle s0, t0 \rangle \#realize Sc1)$ . Здесь  $\langle s0, t0 \rangle$  – начальное состояние ВР-среды, где  $t0$  – начальный момент времени, а  $s0$  – исходная сцена. Пусть в глубинном фреймовом представлении  $s0$  описывается как комната N 317 ( $loc1$ ), в которой находится субъект действий – РСО (текущее положение которого – точка  $p(x, y, z)$ ). РСО выполняет сценарий  $Sc1$  – “переместить объект  $o1$  в локализацию  $loc2$ ”, где  $o1$  – “лекарство  $C$ ”,  $loc2$  – “комната N 105”.

Данное выражение соответствует следующему тексту на естественном языке, переданному оператором на вход РСО: “отнести лекарство  $C$  больному в комнате N 105”.

Блок моделирования поведения РСО анализирует выражение языка LVR и строит последовательность состояний среды, соответствующую изменениям виртуальной среды, заданных данной ППФ. Для этого используются глубинные фреймовые представления используемых сценариев и действий. Рассмотрим их.

Sc1:

(Sc1 : переместить-объект-в-локализацию;  
agent:  $a1$ ;  
object:  $o1$ ;  
loc:  $loc2$ ;  
conditions :  $\#f1$ ;  
actions:  $\#if f1 = \langle false \rangle$  then  $\#realize Sc3$  else  $\#realize Sc2$ ;  
results :).

Здесь  $\#f1$  и  $\#if$  then else – специальные процедуры, запускаемые оператором realize,  $Sc2$  – сценарий “переместиться из точки или локализации в точку или локализацию”;  $Sc3$  – сценарий “найти и взять объект”.

SC2:

(Sc2: переместиться-из-в;  
agent:  $a1$ ;  
from:  $loc1$ ;  
to:  $loc2$ ;  
conditions:  $\#f2$ ;  
actions:  
 $\#if f2 = \langle false \rangle$  then  $\#realize ask$ ;  
 $\#f3a$ ;  
 $\#realize d1$ ;  
 $\#realize d2$ ;  
 $\#realize d3$ ;  
(выйти за пределы локализации  $loc1$ )  
 $\#realize d4$ ;  
 $\#f4$ ;  
 $\#if f5 = \langle true \rangle$  then  $\#realize Sc4$ ;  
 $\#f3b$ ;  
 $\#realize d1$ ;

#realize  $d2$ ;  
 #realize  $d3$ ;  
 (войти в пределы локализации loc2);  
 #realize  $d4$ ).

В описаниях используются следующие действия:  $d1$  – “подойти к объекту (дверь) из текущей точки  $p(x, y, z)$ ”;  $d2$  – “открыть объект дверь”;  $d3$  – “продвинуться немного вперед”;  $d4$  – “закрыть объект дверь”; ask – специальное действие, связанное с коммуникацией, а также следующие функции:  $f1$  – проверка наличия ППФ “ $a1$  держит (обладает) объект  $o1$  (лекарство)” в текущем состоянии;  $f2$  – “проверка существования loc2 или объекта”;  $f3a$  – “определить место выхода (дверь) из локализации loc1”;  $f3b$  – “определить место входа (дверь № 105) в локализацию loc2”;  $f4$  – “определить этаж локализаций loc1, loc2”;  $f5$  – “этаж (loc1) отличен от этаж (loc2)”.

Описание сценария Sc4:

(Sc4: переместить-себя-с-помощью-лифта;  
 agent:  $a1$ ;  
 from: этаж (loc1);  
 to: этаж (loc2);  
 conditions: # $f2$ ; (проверка существования лифта)  
 actions:  
 #if  $f2 = \langle \text{false} \rangle$  then #realize ask;  
 #realize Sc5;  
 #realize  $d4$ ;  
 #realize  $d5$ ;  
 #realize  $d6$ ;

#realize ( $d7$  Ro  $d8$ ) (где Ro – временное отношение “одновременно”);

#if ППФ\* = “true” then #realize  $d9$  else #realize ( $d7$  Ro  $d8$ )).

В данном описании используются следующие действия:  $d4$  – “войти в лифт”;  $d5$  – “найти кнопку этажа”;  $d6$  – “нажать кнопку”;  $d7$  – “стоять”;  $d8$  – “следить за движением лифта и определить остановку лифта – (ППФ\*)”;  $d9$  – “выйти из лифта”, а также сценарий Sc5 – “вызов лифта”, включающий такие действия, как “найти кнопку лифта”, “нажать кнопку”, “ждать, когда лифт остановится перед агентом”.

В результате работы блока моделирования поведения получаем последовательность во времени базовых действий, соответствующих полученному от оператора заданию и осуществляемых РСО. Каждое действие изменяет состояние ВР-среды, которое моделируется в блоке моделирования ВР-среды. Последовательность произведенных РСО действий соответствует последовательности изменений сцен среды. Эта последовательность сцен ВР-среды может быть

визуализирована имеющимися в системе средствами визуального отображения.

**Заключение.** В статье рассмотрены проблемы, возникающие при построении системы виртуальной поддержки процессов проектирования системы управления РСО, в частности проблемы моделирования поведения робота сервисного обслуживания на основе системы виртуальной реальности. В связи с этим обсуждены такие важные вопросы как, особенности организации робота сервисного обслуживания, взаимодействие робота с виртуальной средой и оператором, язык описания динамических состояний ВР-среды, а также логические модели динамических действий, процессов и состояний в виртуальной среде.

## СПИСОК ЛИТЕРАТУРЫ

1. Ulyanov S.V., Yamafuji K. et al. Intelligent Fuzzy Motion Control of Mobile Robot for Service Use // Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. August 5–9. 1995. Pittsburgh, Pennsylvania, 1995.
2. Литвинцева Л.В. Виртуальная реальность – новый шаг в технологии человеко-машинного взаимодействия. // Изв. РАН. Теория и системы управления. 1995. № 5.
3. Cassell J. Animated conversation: Rulebased Generation of Facial Expressions, Gesture and Spoken Intonation for Multiple Conversational Agents // Proc. Siggraph, 1994. № 2.
4. Philips C.B., Badler N.I. Interactive behavior for Bipedal Articulated Figures // Computer Graphics. 1991. V. 25. № 4.
5. Rich C., Waters R.C., Schabes Y. et al. An Animated On-Line Community with Artificial Agents // IEEE Multimedia. 1994. V. 1. № 4.
6. Литвинцева Л.В. Концептуальная модель системы визуализации трехмерных динамических сцен // Программные продукты и системы. 1992. № 2.
7. McCarthy J., Hayes H. Some philosophical problems from the standpoint of artificial intelligence // Machine Intelligence. V. 4. N.Y., 1969.
8. Kuipers B. Qualitative simulation // Artificial Intelligence. 1986. V. 29. № 3.
9. Forbus K. Qualitative reasoning about space and motion // Mental models / Eds D. Gentner, A. Stevens. Hillsdale, 1983.
10. De Kleer J., Brown J.S. The origin, form and logic of qualitative physical laws // Proc. 8th Intern. Joint Conf. on Artif. Intell. V. 2. Karlsruhe, 1983.
11. Pentner B.A. Dynamic Logic of Action // J. of Logic, Language and Information. 1994. V. 3. № 3.
12. Segerberg K. Actions in Dynamic Logics // Symbolic Logic. 1988. V. 53.
13. Von Wright G.H. The logic of Action // The logic of decision and action / Ed. N. Rescher. Univ. of Pittsburg, 1967.
14. Кандрашина Е.Ю., Литвинцева Л.В., Поспелов Д.А. Представление знаний о времени и пространстве в интеллектуальных системах. М.: Наука, 1989.