OSTIS-2016
(Open Semantic Technologies for Intelligent Systems)

УДК 004.92

ОНТОЛОГИЧЕСКИЙ ПОДХОД К ВИЗУАЛИЗАЦИИ РЕЗУЛЬТАТОВ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Коршунов С.А.[*]

[*] *Институт динамики систем и теории управления имени В.М. Матросова СО РАН, г. Иркутск, Россия*

grey.for@gmail.com

В работе рассмотрен подход к визуализации результатов имитационного моделирования на основе онтологий и концепция программного средства, реализующего этот подход. Разрабатываемое средство позволит повысить эффективность процесса визуализации результатов имитационного моделирования, автоматизировав построение визуальных моделей. Также описана архитектура предлагаемого программного средства, аспекты его программной реализации, предлагаемые модели онтологий, приведены примеры визуальных сцен, созданных при помощи предлагаемого средства.
Ключевые слова: онтология, визуализация, имитационное моделирование, машинная графика.

Введение

Проблема визуализации результатов имитационного моделирования является достаточно актуальной на данный момент, что обусловлено широким применением данного подхода в самых различных областях. При этом, визуализация результатов имитационного моделирования традиционно считается второстепенной задачей при исследовании и моделировании систем различной природы. Подобное отношение обуславливает недостаточное теоретическое и практическое развитие данной области.

В частности, современные средства визуализации результатов имитационного моделирования являются либо «встроенными» системами и обеспечивают низкую когнитивность [Зенкин, 1991] (свойство, характеризующее насколько полно визуальные модели отражают основные свойства реальных объектов-прототипов) создаваемых пространственно-временных сцен (на уровне пикселей и абстрактных графовых структур), используя простейшие универсальные алгоритмы и математические выражения, либо «внешними» специализированными программными комплексами, требующими навыков профессиональных дизайнеров и использующих сложные математические и физические абстракции.

При этом отсутствуют системы визуализации, использующие унифицированный подход, например, на основе онтологического моделирования, и обеспечивающие синтез (автоматическое создание) описания пространственно-временных сцен.

Создание графических объектов, соответствующих определенной предметной области, является довольно трудоемкой задачей, для решения которой исследователи в своих работах чаще всего используют различные графические пакеты (3DSMax, Blender, Maya и др.). При этом выбор определенного пакета обусловлен опытом работы исследователя с выбранным программным обеспечением.

Стоит отметить, что графические пакеты являются довольно сложными системами и в том случае, если опыт работы с ними отсутствует, время, затраченное на их освоение, может существенно замедлить процесс исследовательской работы. Из этого можно сделать вывод, что использование стороннего программного обеспечения является достаточно неэффективным способом визуализации.

Наиболее перспективное направление решения данной проблемы – автоматизация исследований при помощи создания пространственно-временных сцен на основе онтологии.

1. Основные принципы разработки программного средства визуализации

Предлагаемое программное средство призвано учесть недостатки существующих систем и обеспечить:

- возможность создания сложных визуальных объектов на основе геометрических примитивов;
- отображение динамики поведения визуальных моделей (анимация, динамическое изменение текстур);
- использование принципов порождающего программирования для синтеза программного кода, описывающего визуальные объекты и пространственно-временные сцены;
- веб-интерфейс создания визуальных объектов и пространственно-временных сцен.
- независимость от имитационной модели;
- возможность адаптации к предметной области;
- отчуждаемость программного кода.
- отсутствие необходимости установки дополнительного программного обеспечения (библиотек и т.д.);

С точки зрения визуализации, степень адаптации к предметной области определяется способностью отобразить все необходимые объекты исследуемой области, что невозможно при использовании ограниченных библиотек визуальных объектов. В этих условиях возникает проблема создания визуальных объектов для исследуемой области.

Наиболее перспективным направлением решения этой проблемы является их создание на основе онтологического моделирования. При таком подходе предлагается инкапсулировать свойства и поведение созданных объектов в объектах онтологии, создав иерархию онтологий (рис. 1).

Риунок 1 -- Иерархия онтологий

При таком подходе хранение онтологий предлагается осуществить с помощью базы данных, что исключает обработку семантически незначимой информацией (служебные теги XML), а также обеспечивает высокое быстродействие и поддержку целостности данных. При этом обеспечивается возможность импорта/экспорта иерархии понятий, их свойств и отношений между ними в форматах OWL и RDF.

Дополнительную сложность решения задачи визуализации придает необходимость моделирования поведения визуализируемых объектов. Данная проблема может быть решена путем императивного описания их поведения в онтологии и его реализации с помощью продукционной модели представления знаний.

Объектно-ориентированная модель некоторой предметной области D, т.е. онтология предметной области Ont^D, включает предметные сущности (классы и экземпляры классов) и отношения между ними:

$$Ont^D = \langle BT\_D, CN, Pr, Obj, R^D \rangle, \quad (1)$$

где Ont^D – онтология предметной области, BT_D – перечень базовых типов данных, $BT\_D = \{\text{литерал}, \text{объект}, \text{коллекция}\}$, CN – имена классов, Pr – имена свойств классов, Obj – понятия (константы, объекты) предметной области, R^D – конечное множество отношений между концептами заданной предметной области,

$$R^D = \{R^D_{is-a}, R^D_{pr}, R^D_c\}, \quad (2)$$

при этом R^D_{isa} – отношение наследования между классами CN, R^D_{pr} – отношение между классами и свойствами, $pr\, R^D_{pr} \langle cn, cn\_bt \rangle$, где $pr \in Pr$, $cn \in CN$, $cn\_bt \in CN \cup BT\_D$, последнее означает, что в предметной области D свойство с именем pr_i характеризует класс cn_j, а значениями свойства могут быть элементы типа другого класса из CN или основного множества типов BT_D, R^D_c – причинно-следственные отношения между концептами.

Онтология визуальных объектов включает классы и экземпляры понятий, описывающие все доступные визуальные объекты и визуальные эффекты:

$$Ont^{VO} = \langle BT\_VP, Pr^{VP}, Obj^{VP}, R^{VP}_{pr}, T\_VE, Pr^{VE}, Obj^{VE}, R^{VE}_{pr} \rangle, \quad (3)$$

где Ont^{VO} – онтология визуальных объектов, BT_VP – базовые типы визуальных примитивов, Pr^{VP} – свойства визуальных примитивов, Obj^{VP} – собственно визуальные примитивы, R^{VP}_{pr} – отношения между базовыми типами и свойствами визуальных примитивов, Pr^{VE} – свойства визуальных эффектов, T_VE – типы визуальных эффектов, Obj^{VE} – визуальные эффекты, R^{VE}_{pr} – отношения между типами и свойствами визуальных эффектов.

$$Ont^{AppV} = \langle Obj^{AppV}, Rule_{VS} \rangle, \quad (4)$$

где Ont^{AppV} – онтология визуализации приложения, Obj^{AppV} – визуальные объекты,

$$Obj^{AppV} = \left\langle Obj^{D}, Obj^{VP}, Obj^{VE} \right\rangle, \quad (5)$$

где $Rule_{VS}$ – правила, описывающие пространственное расположение объектов на сцене приложения и их поведение, представлены в виде цепочки логических правил «если, то», где в качестве условий будут выступать переменные входных данных имитационной модели, а в качестве действий на условие – изменение определенных визуальных объектов.

При отображении онтологии визуальных объектов и онтологии приложения на модель хранения данных получаем ее структуру (рис 2).

Согласно данной модели, каждому объекту предметной области будет соответствовать визуальный объект, наиболее полно отображающий все его основные свойства. В свою очередь, каждый визуальный объект может поддерживать несколько различных визуальных эффектов (анимаций), которые могут изменять определенные свойства объекта. Каждый экземпляр анимации содержит ссылку на ее тип и на атрибуты/свойства объекта, которые будут изменены в процессе анимации.

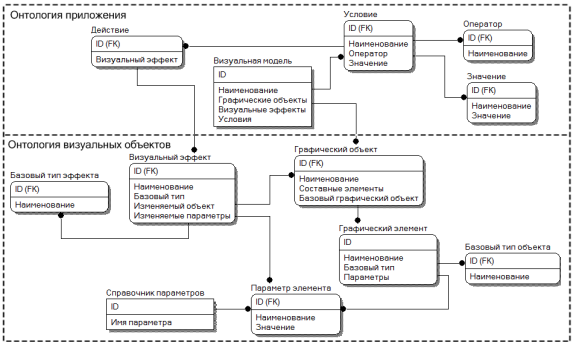

Рисунок 2 -- Логическая модель онтологий

На основе предложенной онтологической модели необходимо разработать отображения, позволяющие синтезировать программный код визуализации, в частности:,

$$R_K^{VP} : Obj^{VP} \to K^{VP}, \quad (6)$$

где K^{VP} – программный код примитива, $R_K^{VE} : Obj^{VE} \to K^{VE}$, K^{VE} – программный код визуального эффекта,

$$R_K^{Rule} : Rule_{VS} \to K^{Rule}, \quad (7)$$

где K^{Rule}– программный код построения визуальной модели, а также заключительное отображение

$$R^{K} : K^{VP} \otimes K^{VE} \otimes K^{Rule} \to K^{AppV}, \quad (8)$$

описывающее динамическую визуальную модель предметной области, где K^{AppV} – программный код, $\otimes$ – обобщенная операция композиции программного кода.

Задача состоит в том, чтобы разработать алгоритмы, реализующие перечисленные отображения: R_K^{VP}, R_K^{VE}, R_K^{Rule}, R^{K}.

2. Архитектура программного средства.

В соответствии с представленными основными принципами разработки сформирована общая концепция архитектуры программного средства (рис. 3). Программное средство состоит из нескольких компонентов, которые можно разделить по принадлежности к серверной или клиентской частям архитектуры.

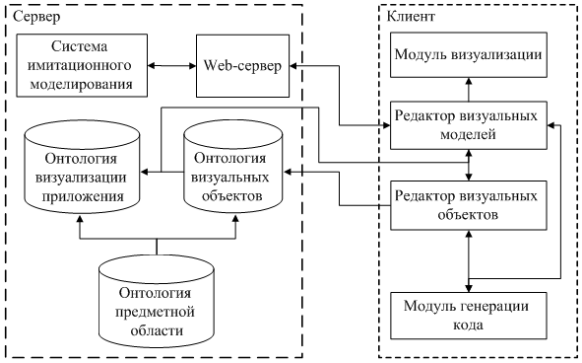

Рис.3. Общая архитектура программного средства.

Клиентские компоненты:

- Редактор визуальных объектов позволяет пользователю создавать свои визуальные объекты разной степени сложности, используя в качестве основы геометрические примитивы (прямоугольник, сфера, цилиндр и т.д.) (рис.3).
- Редактор визуальных моделей для описания структуры и поведения пространственно-временных сцен в виде правил [Коршунов и др., 2015].
- Модуль визуализации позволяет на основе сформированного описания сгенерировать трехмерную пространственно-временную сцену, соответствующую моделируемой области и содержащую визуальные объекты, созданные пользователем. В качестве средства для отображения трехмерной сцены предлагается использовать браузер.
- Модуль генерации JavaScript-кода пространственно-временной сцены и ее объектов отображаемый при помощи графического API WebGL [WebGL].

Серверные компоненты:

- Имитационная модель – источник входных данных для визуализации. В частности, предполагается апробация программного средства при взаимодействии с системой имитационного моделирования на основе агентного подхода [Николайчук и др., 2010].
- Сервер обмена данными между имитационной моделью и модулем визуализации.
- Онтология визуальных объектов, содержащая созданные пользователем визуальные примитивы, которые впоследствии можно

использовать в качестве основы для новых объектов, дополняя или изменяя их структуру.

- Онтология визуализации приложения, содержащая конкретные объекты, предназначенные для моделирования исследуемой области, которые будут находиться в составе визуальной модели.
- Онтология предметной области, содержащая классы и понятия исследуемой области.
- Сгенерированный код будет отчуждаем и может отображаться вне программного средства при помощи любого браузера.

3. Предлагаемые методы реализации компонентов

Программное средство предлагается реализовать в виде веб-приложения, предоставив пользователю доступ к нему через браузер. Главное преимущество такого подхода – отсутствие затрат и дополнительных сложностей, связанных с установкой, обновлением и поддержкой программного средства.

Классическая клиент-серверная архитектура обладает существенным недостатком – обмен данными инициируется по запросу клиента. Данный недостаток можно устранить, используя протокол WebSocket, который осуществляет асинхронный обмен сообщениями между браузером и веб-сервером в режиме реального времени и дает возможность инициировать передачу данных сервером [WebSocket]. Также, клиентская реализация протокола позволит интегрировать программное средство визуализации в систему имитационного моделирования.

В качестве основы для модуля визуализации и графического редактора предлагается использовать графический API WebGL, позволяющую создавать на JavaScript интерактивную графику. Как и WebSocket-протокол, WebGL поддерживается всеми основными браузерами, что позволяет реализовать клиентскую часть в виде веб-страницы.

Заключение

Авторами предложен подход к автоматизации процесса визуализации результатов имитационного моделирования при помощи генерации кода визуальных моделей на основе онтологического описания их структуры и поведения, который позволит пользователям, не имеющим высокой программистской квалификации визуализировать исследуемую область, не прибегая к помощи сторонних графических редакторов, требующих значительных временных затрат на освоение. Предложены возможные технологии реализации компонентов, для нужд программного средства были разработаны модели ряда онтологий (предметной области, графических объектов и визуализации приложения).

Реализация такого подхода в программном средстве позволит создавать подсистемы визуализации, которые в дальнейшем могут быть интегрированы в имитационные системы.

Результаты, представленные в статье, получены при частичной финансовой поддержке РФФИ, проекты 14-07-05641 «Разработка принципов, моделей и методов создания и поддержки интеллектуальных мультиагентных систем для прогнозирования техногенных чрезвычайных ситуаций», 16-37-00122 «Модели, методы и средства синтеза императивного описания пространственно-временных сцен на основе онтологий и порождающего программирования», 15-37-20655 «Разработка моделей, методов и средств сервисно-ориентированной технологии синтеза баз знаний продукционных экспертных систем на основе трансформации концептуальных моделей».

Библиографический список

[Зенкин, 1991] Когнитивная компьютерная графика. Под ред. Д.А. Поспелова. М.: Наука, 1991. – 192 с.

[Коршунов и др., 2015] Коршунов С.А., Николайчук О.А., Павлов А.И. Web-ориентированный компонент продукционной экспертной системы // Программные продукты и системы. – 2015. – №2. – С.20-25.

[WebGL] Документация спецификации Web-based Graphics Library (WebGL). 2015. URL: https://www.khronos.org/registry/webgl/specs/latest/1.0/ (дата обращения: 22.06.2015).

[Николайчук и др., 2010] Николайчук О.А., Павлов А.И., Юрин А.Ю. Система имитационного моделирования динамики состояний сложных технических систем на основе агентного подхода // Автоматизация в промышленности. – 2010, №7. – С.44-48.

[WebSocket] Документация спецификации WebSocket. 2015. URL: https://tools.ietf.org/html/rfc6455 (дата обращения: 22.06.2015).

ONTOLOGY-BASED APPROACH FOR VISUALIZATION OF SIMULATION RESULTS

Korshunov S.A. *

** Matrosov Institute for System Dynamics and Control Theory, Siberian Branch of the Russian Academy of Sciences (IDSTU SB RAS), Irkutsk, Russia*

grey.for@gmail.com

The paper describes the approach of graphical objects creation based on domain ontology description, as well as the hierarchy of the ontology, which includes domain ontology, graphical objects ontology and application's visualization ontology. Concept of software is described, that uses this approach for simulation visualization. The basic requirements for software are formulated, the implementation of which eliminates the most disadvantages of existing imaging systems.

The main components of proposed software architecture considered to be: the simulation model as an input source; graphics editor that lets to create graphical objects based on geometric primitives; editor of visual models for visual scene`s code generation; visualization module to create visual scenes based on generated code; transfer data web-server; domain ontology, graphical objects ontology and application's visualization ontology.