

власнення імен, зарезервовані слова, допустимі відхилення від стандарту (відсутність кінцевого тегу) відповідно до XML-стандарту [2]. Оптимальним є дотримання вимог стандарту XML. Кодування даних задається за бажанням користувача.

Доцільно використовувати Unicode/ISO 10646 для запису даних.

На синтаксичному рівні визначаємо точні назви тегів і синтаксичні правила їх подання. На цьому рівні відповідність синтаксичному стандарту може бути перевірено за допомогою аналізу тексту, тобто формальним способом. На семантичному рівні необхідно гарантувати однозначність розмітки та її інтерпретаційної частини.

Для зняття неоднозначностей при перенесенні текстової інформації між різними системами необхідно вказати для користувача належність використовуваних тегів частинам документа, в нашому випадку – структурним елементам бібліографічного опису.

Такі правила в основному визначаються в супроводжуючих довідниках користувача.

**Висновки**

Вирішення задач розробки засобів автоматизації створення бібліографічних описів дозволяє максимально ефективно інтегрувати систему в існуючі бібліотечні системи та спеціалізовані Web ресурси.

Досліджено структуру даних бібліографічних описів в форматах попередніх ГОСТів і розроблено засоби перетворення бібліографічних описів згідно ГОСТу ДСТУ 7.1:2006 у спеціальну електронну форму. Побудовано концептуальну математичну модель бібліографічного опису.

Розроблено вимоги до мови розмітки бібліографічного опису, розвинуто можливості мови XML з метою застосування її для розмітки структурованих текстів у вигляді бібліографічного опису.

Література

1. ДСТУ ГОСТ 7.1:2006. Бібліографічний запис, бібліографічний опис. Загальні вимоги та правила складання : метод. рекомендації з впровадження [Текст] / уклали: Галевич О. К., Штогрин І. М. – Львів, 2008. – 20 с.
2. Extensible Markup Language (XML) 1.1 (Second Edition) [Електронний ресурс] / Режим доступу: <http://www.w3.org/TR/xml11>.
3. Касилов О. В. Моделирование лексикографических систем [Текст] / О. В. Касилов // Вісник Міжнародного Слов'янського університету. Сер. «Технічні науки». – Харків:– 2004. – Т. 7. – № 1. – С. 13–15.

**Підхід до створення інтелектуальної анімації використовує семантичне анотування поведінки об'єктів та процесів, що моделюються. Це дозволить автоматично генерувати динамічні сцени, керуючись знаннями, отриманими в результаті виконання запитів до онтологічної бази**

**Ключові слова: інтелектуальна анімація, онтологія, OpenGL**

---

**Подход к созданию интеллектуальной анимации на основе семантического аннотирования поведения моделируемых объектов и процессов позволит автоматически генерировать динамические сцены, используя знания, полученные в результате выполнения запросов к онтологической базе**

**Ключевые слова: интеллектуальная анимация, онтология, OpenGL**

---

**Approach for intelligent animation based on semantic annotations of objects and processes allows automated generation of dynamic scenes using knowledge obtained as a result of ontological knowledge base requesting**

**Keywords: intelligent animation, ontology, OpenGL**

УДК 004.822:514

# СОЗДАНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ АНИМАЦИИ НА ОСНОВЕ ОНТОЛОГИЙ

**И. В. Климов**  
 Кафедра искусственного интеллекта  
 Харьковский национальный университет  
 радиозлектроники  
 пр. Ленина, 14, г. Харьков, Украина, 61166  
 Контактный тел.: 099-441-97-95  
 E-mail: igor.klymov@gmail.com

**1. Введение**

Сегодня компьютерная анимация входит в число лидеров программной индустрии, используется

для создания современных фильмов, мультфильмов, игровых приложений, кроме того, применяется в компьютерных симуляторах сложных и дорогостоящих систем: авиа-тренажерах, тренажерах управления

космическими кораблями и станциями, для моделирования промышленных процессов и построения так называемых профессиональных или корпоративных игр.

Однако анимация оказывается трудоемким и временно-затратным процессом, особенно в задачах, требующих высокой детализации и реалистичности отображения среды и объектов.

В данной работе предлагается подход к созданию интеллектуальной программной анимации, способной автоматически строить и воспроизводить трехмерные изображения по их описанию, хранящемуся в онтологической базе знаний [1]. Анимационный фильм представляется в виде набора семантически аннотированных компонентов (герои, среда и другие объекты), свойства которых находятся в онтологической базе знаний, откуда извлекаются для автоматического построения анимации. При этом онтологии используются в качестве хранилища информации как о моделируемой области, так и о способе ее отображения [2].

## 2. Выбор технологии построения автоматической анимации

В условиях актуальности и популярности трехмерной компьютерной анимации растет спрос на инструментальные системы ее создания и развитие их возможностей.

На данный момент существует множество программных продуктов, предоставляющих пользователю возможность создавать объекты и анимировать их поведение.

Для создания интеллектуальной анимации в данной работе предлагается OpenGL- оптимизированная, высокопроизводительная графическая библиотека функций и типов данных для отображения двух-и трехмерной графики. Преимуществами OpenGL является единый API, предоставляемый разработчику, позволяющий скрыть различия в возможностях аппаратных платформ, низкоуровневость и процедурность императивного подхода, который вынуждает программиста диктовать точную последовательность шагов, чтобы построить результирующую растровую графику, что, с одной стороны, требует глубокого знания законов трёхмерной графики и математических моделей, а с другой стороны, даёт свободу внедрения различных инноваций, а также независимость от языка программирования. Приложение, использующее OpenGL, не нуждается в дополнительных модулях и программах. К достоинствам также можно отнести переносимость на большинство операционных систем, открытость системы и постоянство интерфейса.

Основным принципом работы OpenGL является получение наборов векторных графических примитивов в виде точек, линий и многоугольников с последующей математической обработкой полученных данных и построением растровой картинке на экране и/или в памяти.

Для представления знаний о моделируемой предметной области и способах ее визуализации в работе предлагается использовать возможности онтологий

[3]. Онтологии описывают формальные, распределенные концептуализации структурно гетерогенных предметных областей, как правило, в машино-понятном формате OWL, поддерживая синтаксический и семантический обмен информации.

## 3. Онтологическое хранилище анимационной информации

Анимация представляет собой процесс создания и воспроизведения множества изображений, демонстрирующих изменение состояния объектов сцены во времени. Хранение информации о содержании моделируемой сцены и о способах ее трехмерной визуализации с помощью программной прорисовки средствами OpenGL выполняется онтологией анимации.

Ресурсы онтологии, указывающие на объекты или процессы предметной области, характеризуются свойством `shape`, которое ставит в соответствие ресурсу его моделируемое изображение (рис. 1) – некоторый объект анимации сцены типа `Graphic_object`. В онтологии `Graphic_object` – абстрактный концепт для создания графических объектов разных типов, конструируемых из геометрических примитивов: точек, линий и полигонов, которые, в свою очередь, задаются своими вершинами.

Кроме явного низкоуровневого перечисления примитивов, существует способ построения графических объектов по заранее описанным в OpenGL шаблонам для часто используемых фигур, таких как сфера, куб, цилиндр и т.д. Примитивы и фигуры принадлежат классам онтологии `Graphic_primitive` и `Shape`, которые рекурсивно наследуют класс `Graphic_object`.

Вершины в онтологии являются экземплярами класса `Vertex` и обладают цветом и координатами местоположения – `x`, `y`, `z`. Цвет – экземпляр класса `Color`, устанавливается четырьмя параметрами: красная, синяя, зеленая составляющие и прозрачность. Эти параметры варьируют в диапазоне от нуля до единицы.

Вершины передаются в качестве аргументов в конструкторы примитивов. Количество аргументов, передаваемых в конструктор, зависит от типа создаваемого примитива. Для создания связи между вершиной и примитивом в онтологии определено свойство примитива `argument`.

Если объект строится по шаблону, то вместо вершин в конструктор объекта передаются такие свойства объекта, необходимые для его отрисовки, как ширина, высота, глубина, радиус и подобные. Для каждого типа шаблона определен свой набор свойств. Текстура фигуры представлена свойством `texture`, значение которого является ссылкой на готовое растровое изображение, которое накладывается на многоугольники каркаса модели и вносит в нее эффект поверхности реального материала.

Также в онтологии анимации описаны способы освещения моделируемой сцены. Можно использовать три типа источников света: направленный (`Oriented_light`), точечный (`Pointed_light`) и прожектор (`Projector`).

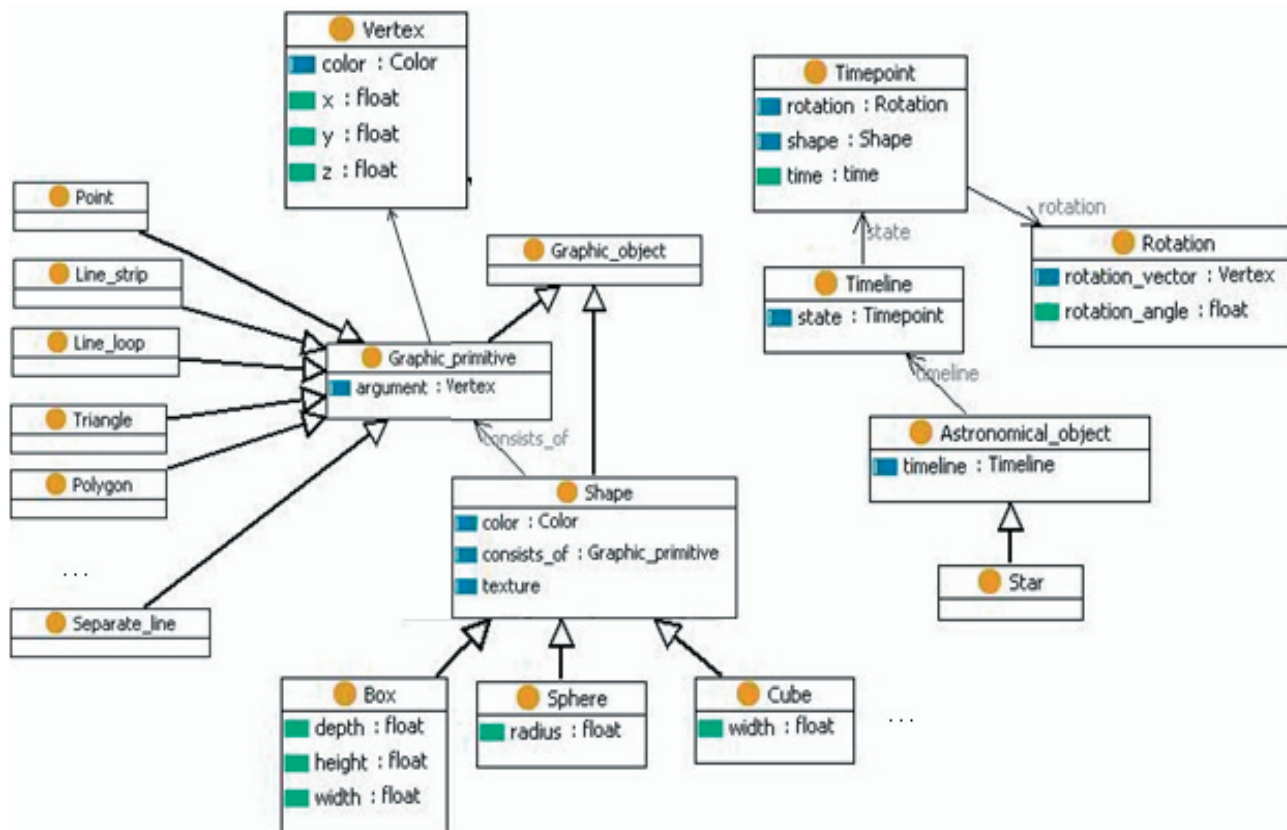


Рис. 1. Определение основных графических объектов в онтологии

#### 4. Создание анимации по онтологии

Анимирование предполагает прорисовку сцены и объектов в различные моменты времени. Для компьютерной имитации движения с помощью изменения и перерисовки формы и положения объектов во времени в онтологии анимации вводится понятие временной шкалы, которому соответствует концепт Timeline. Экземпляры Timeline указывают на критические временные срезы Timepoint, в которые анимируемый ресурс онтологии меняет свое состояние.

Рассмотрим создание объектов анимации на примере автоматической визуализации движения объектов Солнечной системы (рис. 2). Создадим объект Sun как экземпляр класса Star и поставим ему в соответствие временную шкалу Timeline\_1, которая имеет два ключевых состояния Timepoint\_1 и Timepoint\_2, характеризующих изменения в отображении объекта Sun. Первое характерное состояние объекта – в нулевой момент времени. Для статических объектов это состояние является единственным. В онтологии описывается форма отображаемого объекта – сфера, местоположение на

сцене и его поворот, задающийся углом поворота и вектором, относительно которого выполняется вращение. Вектор вращения задается двумя точками: вершиной в пространстве сцены (x, y, z) и началом координат (0, 0, 0).

Машина вывода анализирует значение свойства shape, указывающего на форму объекта – Sphere и подбирает соответствующую функцию OpenGL, которая рисует сферу в начале координат: auxSolidSphere(), передавая ей в качестве аргумента значение радиуса сферы. Отдельной функцией glColor3d(0,0,0) предварительно задается текущий цвет будущей сферы – значение свойства color соответствующей фигуры. Аналогично машиной вывода анализируются все ключевые состояния визуализируемого объекта для его дальнейшего преобразования во времени.

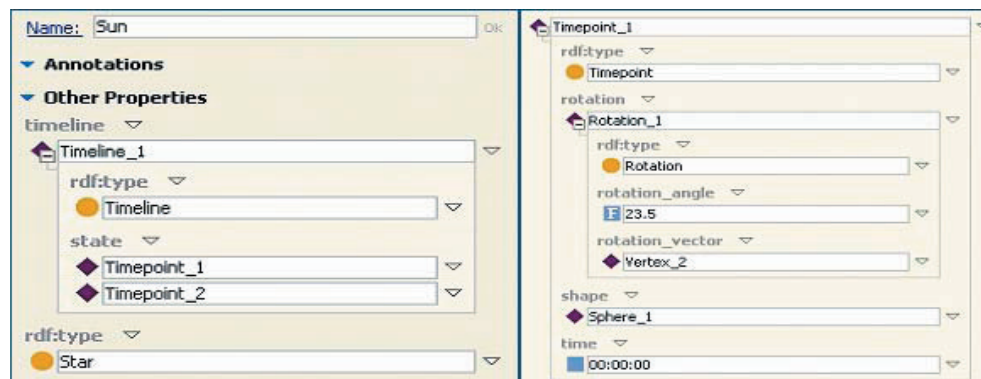


Рис. 2. Создание временной шкалы анимации объектов Солнечной системы

Результатом выполнения запросов к базе знаний является отображение всех объектов анимации в программе в начальный момент времени, дальнейший вызов функций анимации для соответствующих объектов и отрисовка последующих кадров анимации (рис. 3).

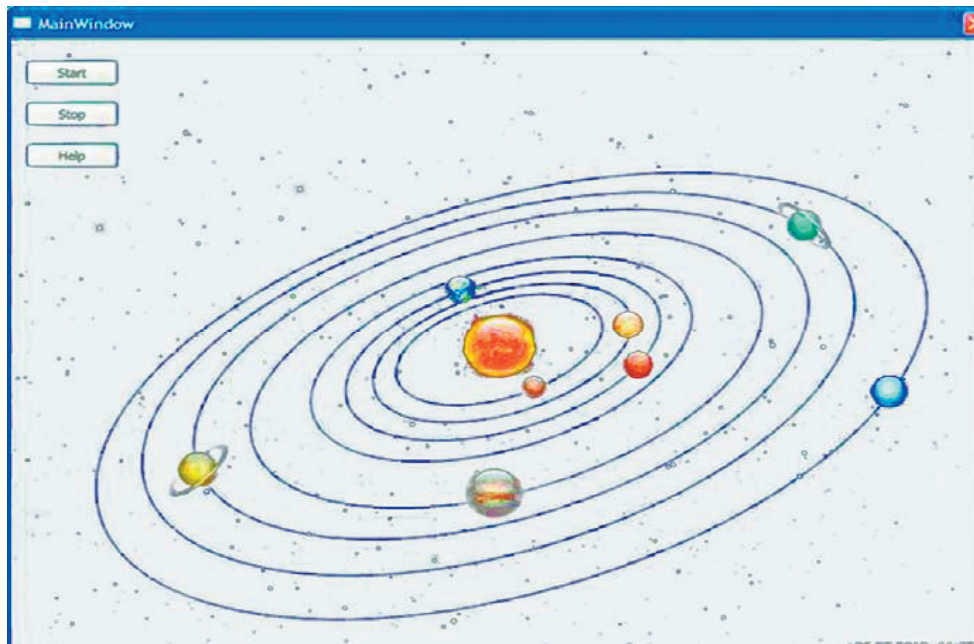


Рис. 3. Прорисовка онтологии «Солнечная система» в программе-просмотрщика

## 5. Выводы

Предложенный подход к созданию интеллектуальной анимации объектов позволяет существенно упростить процесс создания анимации благодаря использованию семантически проаннотированных

моделей поведения объектов и процессов предметной области.

Разработанный алгоритм способен автоматически сгенерировать динамическую сцену некоторой предметной области на основе знаний, полученных в результате выполнения запросов к онтологической базе знаний. Онтологический подход обеспечивает гибкость полученной анимационной системы, возможность расширения структуры заданных объектов и компонентов анимации, повторного их использования в различных предметных областях и для разнотипных задач.

## Литература

1. Шевченко, О.Ю. Метод побудови інтелектуальних систем обробки інформації та документообігу за допомогою онтологічної бази знань [Текст] / О.Ю. Шевченко, М.В. Климова // Штучний інтелект. – 2009. – Вип. 2. – Донецьк. – С. 91–97.
2. Gruber, T. R. A translation approach to portable ontology specifications. [Text] / T.R. Gruber // KnowledgeAcquisition, 6(2):199–221, 1993.
3. Using Ontology to create 3D Animations for Training [Electronic resource] / SERSC: Science&EngineeringResearchSupport society.– Mode of access: \www/ URL: [http://www.sersc.org/journals/IJSEIA/vol1\\_no1\\_2007/IJSEIA-2007-01-01-05.pdf/](http://www.sersc.org/journals/IJSEIA/vol1_no1_2007/IJSEIA-2007-01-01-05.pdf/) – 10.05.2012 – Titlefromscreen.