



ЛИТЕРАТУРА

1. Еремина В.В., Костюков Н.С., Тюрина С.Ю. Моделирование оптического спектра воды в области упругой ионной поляризации // Информатика и системы управления. 2004. – № 2(8). – С. 32-39.
2. Сакнави Г.И. Физика диэлектриков. – М.-Л.: Госэнергоиздат, 1952.
3. Юхневич Г.В. Инфракрасная спектроскопия воды. – М.: Наука, 1973.
4. Эйзенберг Д., Кауцман В. Структура и свойства воды. – Л. Гидрометеоздат, 1975.
5. Яворский Б.М., Детлаф А.А. Справочник по физике для инженеров и студентов вузов. – М.: Наука, 1977.
6. Мальдельштамм Л.Е. Лекции по теории колебаний. – М.: Наука, 1972.
7. Еремина В.В., Костюков Н.С., Тюрина С.Ю. Математические модели процессов упругой ионной поляризации воды // Вестник АмГУ. – 2004. – № 25. – С. 20-22.
8. Золотарев В.М., Морозов В.Н., Смирнова Е.В. Оптические постоянные природных и технических сред: справочник. – Л.: Химия, 1984.

Статья представлена к публикации членом редколлегии А.Д. Плутенко.

УДК 519.86:793.3

© 2006 г. С.Л. Перцовский

(Институт автоматизации и процессов управления ДВО РАН, Владивосток)

ОНТОЛОГООРИЕНТИРОВАННЫЙ ПОДХОД К РАЗРАБОТКЕ САПР СОВРЕМЕННОГО СОЛЬНОГО ТАНЦА

Рассматривается онтологоориентированный подход к созданию САПР современного сольного танца и методы разработки САПР на основе этого подхода. Приводятся примеры компьютерных моделей основных терминов онтологий танца, танцора и процесса проектирования танца, из которых формируется ядро САПР современного сольного танца.

Введение

В настоящее время при создании музыкальных композиций [1, 2], разнообразной дизайнерской деятельности [3], в киноиндустрии и мультипликации [4] и даже в хореографии [5] находят применение системы автоматизации проектирования (САПР).

При проектировании танцев использование хореографами универсальных программных средств трехмерного моделирования движущегося человека [6, 7] практически невозможно, так как требует профессиональ-

ной подготовки в области трехмерного компьютерного моделирования. Однако существуют программные средства, предназначенные для проектирования хореографами балетных танцев. Они позволяют задавать последовательность поз классического балета на графической модели танцора [8-10]. Но такие программные средства не могут быть использованы при проектировании современных сольных танцев. Причин этого несколько. С одной стороны, эти программные средства основаны на использовании записи Лабана [11], которая не позволяет фиксировать все варианты поз и движений современного танца. С другой стороны, появляются новые стили современных сольных танцев, все чаще хореографы используют такие позы и движения танцев, которые не являются позами и движениями классического балета.

Популярность современных танцев приводит к тому, что появляется все больше танцевальных школ, в которых создаются новые танцы современных стилей. Для распространения новых танцев используются только видеозаписи. Создание САПР современного сольного танца (САПР ССТ) дает возможность хореографам автоматизировать процесс создания танца, а ученикам школ – изучать новые танцы дистанционно. При этом ученики могут многократно и в деталях рассматривать каждое движение, каждую позу и позицию танца, что невозможно сделать, имея лишь видеозапись.

Этапы создания САПР современного сольного танца

К созданию САПР в “нетехнических” областях не существует единого подхода. САПР является программой, и для создания ее, вообще говоря, могут быть применены разнообразные технологии [12]. При создании программных средств для использования в таких неформализованных предметных областях как создание танца (в особенности современного танца), приходится в первую очередь исследовать данную предметную область и строить ее математическую модель. Именно это открывает онтологоориентированный подход к проектированию программных средств [13-14].

При онтологоориентированном подходе разработка программы предполагает несколько этапов, причем первые два из них – построение онтологии предметной области и ее математической модели. Создание САПР ССТ проходит еще через два этапа (рис. 1): разработка ядра САПР и построение самой САПР, адаптированной к проектированию современного сольного танца определенного стиля.

На первом этапе происходит извлечение знаний о предметной области "Проектирование ССТ". Результатом работы является онтология этой предметной области. Второй этап – построение математических моделей этой онтологии. На обоих этапах применяются известные методы формализации предметных областей [14]. Вся дальнейшая разработка основывается на построенной математической модели онтологии предметной области.

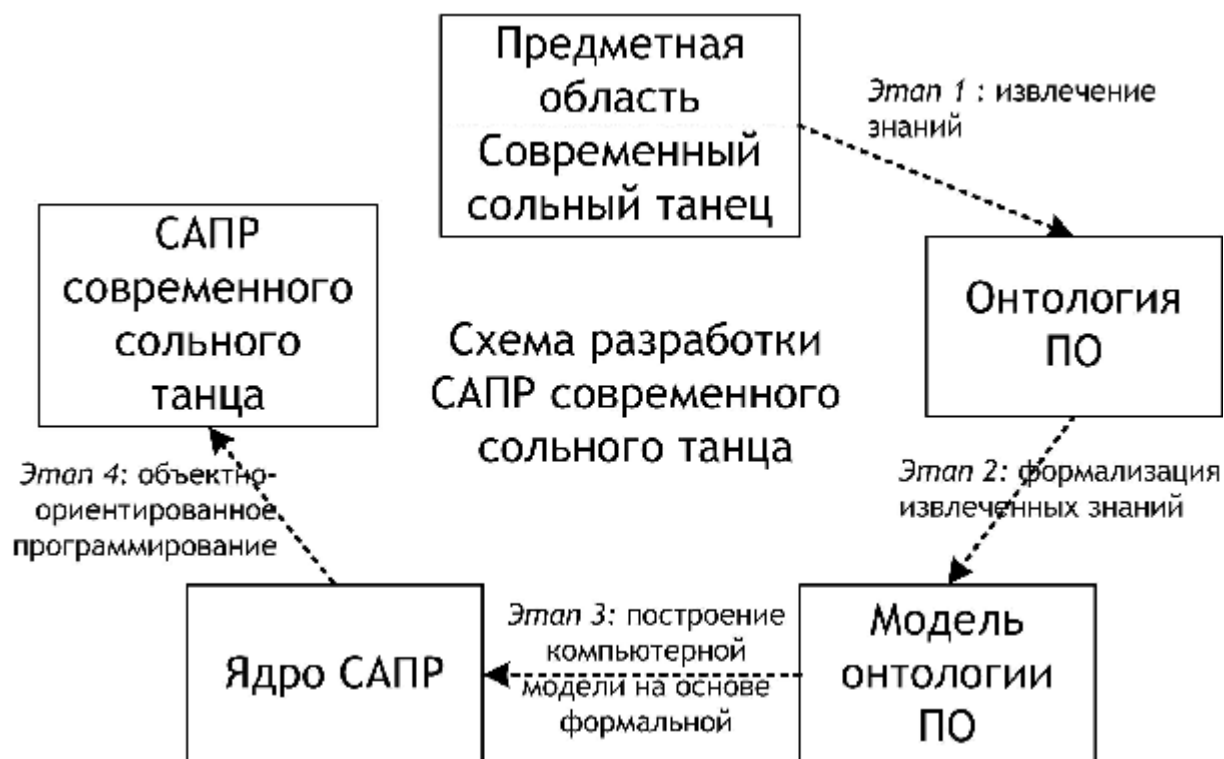


Рис. 1.

САПР – это программное средство, и для его создания невозможно использовать математические модели терминов онтологии непосредственно. Поэтому на третьем этапе создания САПР ССТ строятся компьютерные модели терминов онтологии, которые образуют ядро САПР. На последнем, четвертом этапе на базе ядра САПР методами объектно-ориентированного программирования строится сама САПР ССТ: добавляются процедуры обработки команд пользователя, создаются интерфейс и другие программные элементы.

В данной статье этапы разработки САПР ССТ описываются на примере разработки САПР танца стиля “Фанк”. Танец стиля “Фанк” сходен с такими танцами как “Хип-хоп”, “R’n’B” и “Модерн-джаз” [15, 16] и может служить адекватным представителем современных сольных танцев.

На первом этапе была исследована предметная область (ПО) “Проектирование современного сольного танца стиля “Фанк”. Разработаны три онтологии этой ПО: онтология танца (описываются все движения и позы танца стиля “Фанк”), онтология танцора (выделяются части тела танцора, описываются ограничения на движения, накладываемые физиологией человека) и онтология процесса проектирования танца стиля “Фанк” хореографом (описываются шаги, составляющие этот процесс) [17 – 19]. Выделено около 50 терминов онтологий этой ПО. Анализ видеоматериалов, посвященных современному сольному танцу (см., например, [20, 21]), позволил также выделить 37 вариантов танцевальных позиций стиля “Фанк” и 15 вариантов общепринятых движений различных частей тела танцора – испол-

нителя танца стиля “Фанк”.

На втором этапе были построены математические модели онтологий, которые состоят из моделей терминов этих онтологий и моделей отношений между ними [14, 17 – 19]. Пример одной из них – математической модели танца – представлен на рис. 2. На этом рисунке *fDance* – модель записи танца, *fEight* – модель восьмерки, *fMoveD* – модель движения доли, *fMoveHalfD* – модель движения полудоли, *fPMove* – модель движения части тела, *fBMove* – модель движения всего тела.

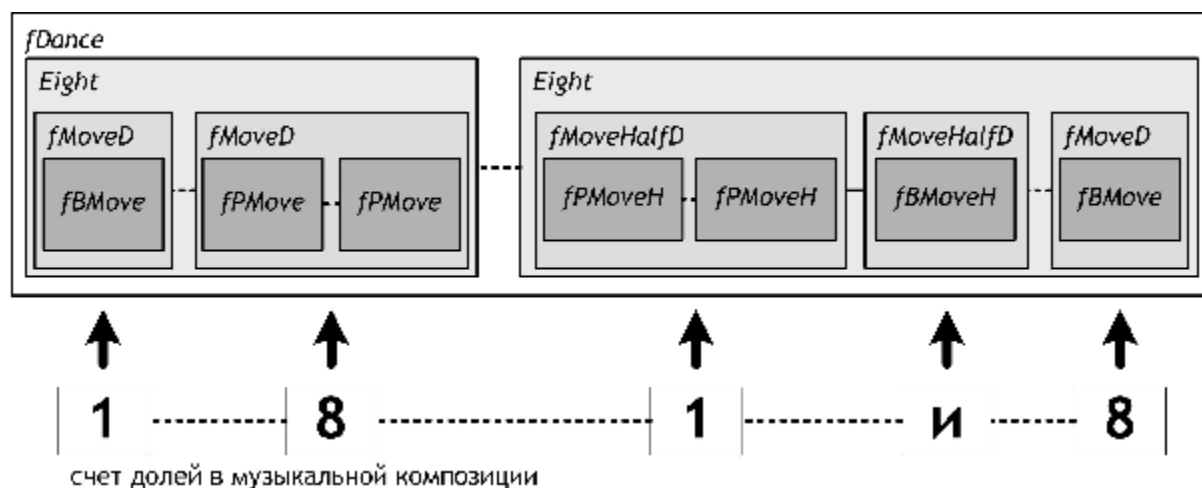


Рис. 2.

Третий этап был посвящен созданию ядра САПР ССТ “Фанк”. Метод создания ядра САПР ССТ (см. ниже) не зависит от стиля современного танца и может быть использован при создании САПР ССТ любого стиля. В созданном ядре САПР ССТ стиля “Фанк” использовано около 50 компьютерных моделей терминов онтологий ПО.

На четвертом этапе на основе этого ядра была разработана САПР ССТ стиля “Фанк”: добавлен графический интерфейс, процедуры обработки команд хореографа и т.п.

Метод создания ядра САПР ССТ подробно описывается в следующем разделе.

Метод создания ядра САПР современного сольного танца

При онтологоориентированном подходе методом создания ядра САПР ССТ является сопоставление каждой модели термина в модели онтологии ПО его компьютерной модели (КМ). Для описания КМ используется объектно-ориентированный язык программирования. В данной статье для этой цели используются конструкции языка С++ [22]. Правила сопоставления определяются особенностями модели онтологии ПО: в ней используются только четыре вида математических объектов (множество имен, кортеж элементов, множество объектов и конечная последовательность, моделирующая процесс, состоящий из конечного числа шагов).

Правила сопоставления таковы:

множеству имен в математической модели онтологии N сопоставляется перечислимое множество $_N$:

$$N \stackrel{\text{def}}{\rightarrow} \{b_1, \dots, b_N\} \rightarrow \text{enum } _Ns \{b1, \dots, bN\},$$

здесь b_1, \dots, b_N – имена в математической модели, знак “ $\stackrel{\text{def}}$ ” обозначает “есть по определению”, а знаком “ \rightarrow ” обозначено сопоставление;

кортежу элементов в математической модели онтологии S сопоставляется класс $_S$:

$$S \stackrel{\text{def}}{\rightarrow} \langle q_1, \dots, q_K \rangle \rightarrow \text{class } _S \{T1_q1; \dots; TN_qN; m1; \dots; mH; \},$$

здесь q_1, \dots, q_K – элементы кортежа, $_q1, \dots, _qN$ – переменные класса $_S$ типа $T1, \dots, TN$ (N может быть равно K), соответственно $m1, \dots, mH$ – методы класса $_S$, которые используются в том случае, когда значения переменных задаются непосредственно (пользователем) или с помощью других классов;

множество объектов математической модели онтологии E моделируется с помощью массива объектов $_E$ класса $_S$:

$$E \stackrel{\text{def}}{\rightarrow} \{f_1, \dots, f_M\} \rightarrow _S _E[M],$$

здесь f_1, \dots, f_M – элементы массива;

процессу проектирования в математической модели онтологии P сопоставляется функция $_P$, телом которой является цикл обработки команд хореографа:

$$P \stackrel{\text{def}}{\rightarrow} STEP_1, \dots, STEP_G \rightarrow _P() \{ \text{for (int } i = 1; i < G; i++) \\ \{ \text{if (...) ...; if (...) ...; } \}; \},$$

здесь $STEP_1, \dots, STEP_G$ – шаги процесса проектирования, if (...) – условный оператор выбора команд хореографа.

Анализ структуры модели онтологии ПО “Проектирование сольного танца стиля “Фанк” показал, что для создания ядра САПР ССТ необходимо выполнить следующие шаги – построить КМ:

записи танца;

реалистичного изображения танцора;

анимации танцора;

процесса проектирования танца.

Совокупность этих четырех КМ образует КМ танца, которая и служит ядром САПР ССТ.

Ниже описано построение всех этих КМ для танца стиля “Фанк”.

Компьютерная модель записи танца

При построении КМ записи танца решаются две задачи: построения модели и отображения этой модели на экране монитора. Процесс построения КМ записи танца состоит из некоторого количества шагов. После ее построения она должна быть отображена на экране монитора.

На первом шаге, вообще говоря, нужно было бы построить КМ описаний поз и позиций. Однако учитывая то обстоятельство, что любая поза

представляет собой совокупность всех позиций частей тела, строить компьютерные модели описаний поз не нужно – достаточно *построить компьютерную модель множества описаний позиций*. Следующий шаг построения КМ записи танца – *построение КМ поз и позиций* (поз и позиций, однозначно задающих положение частей тела) На основе этих моделей строятся *КМ танцевальных движений части тела и всего тела*, а затем – *КМ множества общепринятых танцевальных движений*.

Танец всегда исполняется под музыку, поэтому танцевальные движения характеризуются еще и музыкальным ритмом, в котором они исполняются. Как следует из анализа модели онтологии танца, при построении восьмерки неважно, исполняется движение частью тела или всем телом, поэтому уже на следующем шаге строится *КМ танцевального движения, исполняемого в течение доли (полудоли)*, на ее основе далее строится *КМ восьмерки*, а на последнем шаге – *КМ записи танца*.

Для отображения полученной КМ записи танца предлагается графический элемент *Лента* (рис. 3).

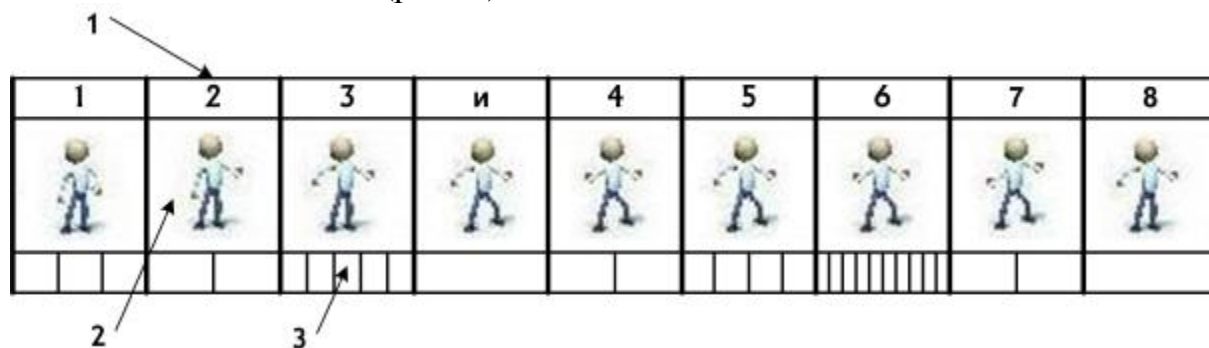


Рис. 3.

Лента состоит из трех частей:

1. *Лента музыкального ритма* (рис. 3.1) – отображение последовательности долей и полудолей, причем каждая восьмая доля отмечает конец очередного музыкального такта и, соответственно, восьмерки, исполняемой в течение этого такта.

2. *Лента танцора* (рис. 3.2) – отображение последовательности изображений танцора в начальной позе танцевального движения доли или полудоли;

3. *Лента отображения “ключей”* (рис. 3.3): танцевальное движение состоит из последовательности поз или позиций, каждая вертикальная линия в линейке отмечает одну из таких поз – “ключей” движения.

Таким образом, создание КМ записи танца сводится к разработке множества классов, сопоставляемых терминам онтологии ПО, использованным при определении термина “запись танца”. При этом разрабатываемые классы должны включать в себя следующую информацию:

“описание позиции”: имя части тела; имя позиции; параметры позиции;

“позиция”: идентификатор объекта класса описания позиции (указатель на объект); значения параметров позиции; функцию (метод класса)

для получения от пользователя-хореографа значений параметров позиции;

- “поза”: массив объектов класса “позиция” для всех частей тела;
- “движение”: количество позиций (или поз), из которых состоит движение; массив объектов класса “позиция” (или объектов класса “поза”);
- “общепринятое движение”: имя части тела (в случае движения части тела); имя движения; значения характеристик движения; объект класса “движение”;
- “движение, исполняемое в течение доли (полудоли)”: значение, показывающее, как исполняется движение (в течение доли или в течение полудоли); номера начальной и конечной “музыкальных точек” на музыкальной дорожке; объект класса “движение”;
- “восьмерка”: порядковый номер восьмерки в последовательности восьмерок; массив объектов класса “движение, исполняемых в течение доли (полудоли)”;
- “танец”: количество восьмерок в танце; время начала танца; протяженность доли; массив объектов класса “восьмерка”.

Например, класс “описание позиции” был определен следующим образом:

```
class PART_POSE_DESC {PART_NAME name; STRING pose;  
    STRING opt; bool ang; },
```

где *name* – имя части тела; *PART_NAME* – множество имен частей тела; *pose* – имя позиции; *STRING* – класс для описания конечной последовательности символов; *opt* – характеристики позиции, причем в строке *opt* ее *i*-я подстрока *opt_i*, начинающаяся символом “#”, задает значения *i*-й характеристики; значения разделены символом “*”; *ang* – значение “1” переменной означает, что для позиции задавать значение угла поворота части тела необходимо, а “0” означает обратное.

Компьютерная модель изображения и анимации танцора

Сначала рассмотрим шаги создания компьютерной модели изображения танцора. В первую очередь строится *КМ ограничений на движения частей тела*: если не задать этих ограничений, модель сможет “совершать” нереальные движения. Для управления моделью человека (в частности танцора) используются методы скелетной анимации и “натяжение кожи” [7]: иерархический скелет помещается внутрь модели, и, управляя им, пользователь изменяет позу модели. При этом к скелету “прикрепляется” поверхность – “кожа” модели. Следующий шаг – построение *КМ иерархического скелета модели танцора*. Используя эту модель, уже можно строить *КМ тела танцора*.

Таким образом, метод создания КМ изображения танцора сводится к разработке некоторого множества классов, сопоставляемых моделям терминов онтологии, используемым в математической модели танцора. При этом эти классы должны включать в себя следующую информацию:

класс “часть скелета”: имя части тела (скелета); идентификатор “родительской” части скелета (указатель на нее); значения углов поворота части скелета относительно “родительской” части тела;

класс “тело танцора”: массив объектов класса “часть скелета”, образующий весь скелет; массив значений координат точек поверхности; массив объектов, описывающий механизм “прикрепления” вершин поверхности к частям скелета; идентификатор файла, где хранится текстура поверхности.

Рассмотрим класс “часть скелета”:

```
class FRAME {PART_NAME_EX name; FRAME* parent; int x, y, z; },
```

где *name* – имя части тела, соответствующей части скелета; *PART_NAME_EX* – множество имен частей тела танцора (это множество имен отличается от множества *BODY_PART_NAME*, используемого в методах построения модели записи танца: в модели танцора части тела танцора выделяются более детально); *parent* – указатель на “родительскую” часть скелета, – например, для *pelvis* (таз) значение этой переменной “NULL”, для *torso* (туловище) – указатель на объект *X* типа *FRAME*, где *X.name = pelvis* и т.д.; *x, y, z* – значения углов поворота части скелета относительно “родительской” части тела.

Теперь охарактеризуем метод создания КМ анимации танцора: он сводится к разработке некоторого множества функций, среди которых основными являются:

функции для вычислений положений частей тела в любой момент времени – функции интерполяции позиций [23];

функции для вычисления положения всего тела в любой момент времени, т.е. реализации известного метода движения модели по опорным следам [6].

Компьютерная модель процесса проектирования

После построения компьютерных моделей записи танца, тела танцора и анимации танцора можно описать, с помощью каких объектов хореограф, используя эти модели, управляет процессом создания танца, т.е. описать методы построения модели процесса проектирования.

Модель процесса формирования танца является основной и единственной КМ процесса проектирования: в любой момент процесса формирования танца хореограф может перейти к формированию восьмерки, к формированию позиции некоторой части тела, к удалению некоторого движения и т.д. Поэтому выделение более “мелких” моделей не является необходимым.

Рассмотрим верхний уровень детализации этой КМ – ее основные функции.

Функция *FormDance()* запускает цикл, на каждом шаге которого хореограф дает команду на добавление восьмерки, просмотр созданных ра-

нее восьмерок или на удаление восьмерки. В начале работы с САПР ССТ хореограф может дать команду только на добавление восьмерки. Функция заканчивает свою работу, когда хореограф заканчивает процесс формирования танца и дает команду на выход из программы.

```
BOOL FormDance() {  
    while (1) { // цикл формирования танца  
        STRING A = GetChCom(); // получить команду от хореографа  
        if (A == "Exit") return; // если команда "Exit", выйти  
        FormDanceStep(A); } // запустить шаг формирования танца
```

КМ отдельного шага процесса формирования танца:

```
BOOL FormDanceStep(STRING Com) {  
    // команда на добавление восьмерки  
    if (Com == "AddEighth") return AddEight();  
    // команда на просмотр всех восьмерок; если танец "пуст", выход  
    if (Com == "ViewEights") { if (d.n == 0) return 0; return ViewEights(); }  
    // команда на удаление восьмерки; если танец "пуст", выход  
    if (Com == "DelEight") { if (d.n == 0) return 0; return DelEights(); }  
    return 0; } // ни одна команда не может быть выполнена
```

В этом разделе описан третий этап разработки САПР ССТ, на котором создается ядро (совокупность четырех КМ). Следующий этап – создание САПР ССТ на основе этого ядра.

Заключение

В статье описан онтологоориентированный подход к разработке САПР ССТ. Этот подход предполагает глубокое изучение ПО. При онтологоориентированном подходе разработка программного средства состоит из четырех этапов: построение онтологии ПО, создание математической, а затем компьютерной модели ПО и построение САПР, адаптированной к проектированию современного сольного танца определенного стиля.

Онтологоориентированный подход дает возможность создать профессиональное программное средство для его использования в сложных и неформализованных областях творческой деятельности человека. Использование этого подхода при разработке САПР для хореографа позволяет получить на выходе такую программу, использование которой сохраняет привычный для хореографа процесс проектирования танца; при этом общение хореографа с программой происходит в его профессиональных терминах.

ЛИТЕРАТУРА

1. Forsberg A., Dieterich M., Zeleznik R. The Music Notepad // Proceedings of the 11th annual ACM symposium on User interface software and technology. – Sun Francisco, CA, 1998. – P. 203-210.

2. *Simons T.* Cakewalk Home Studio XL 2004 // Presentations. – 2004. – V. 18. № 7.
3. *Бэйн С., Уилкинсон Н.* Эффективная работа: CorelDRAW 12: пер. с англ. – СПб.: Питер, 2005.
4. *Porter T., Susman G.* Creating Lifelike Characters in Pixar Movies // Communications of the ACM. – 2000. – V. 43. № 1. – P. 25-29.
5. *Calvert T., Wilke L., Ryman R., Fox I.* Applications of computers to dance // Proceedings of the IEEE – Computer Graphics and Applications. – 2005. – V. 25. № 2. – P. 6-12.
6. *Мэрдок К.Л.* 3ds max 6. Библия пользователя: пер. с англ. – Киев: Вильямс, 2005.
7. *Бриллиант К.* Цифровая модель человека: пер. с англ. – М.: КУДИЦ-Образ, 2004.
8. *Lopez D., Rodriguez D., Meziat M., Carbajo A., Bosque J.L.* A user interface for the design of human figures multimedia animations // Proceedings of HIM, 1997.
9. *Meziat M., Lopez D., Rodriguez D., Carbajo A., Casillas A., Bosque J.L.* HAL: A Multimedia Language for the Creation of 3D Animations of Human Figures // Proceedings of the IEEE Symposium on Visual Languages, 1997. – P. 181-182.
10. *Wilke L., Calvert T., Ryman R., Fox I.* From dance notation to human animation: The LabanDancer project: Motion Capture and Retrieval // Computer Animation and Virtual Worlds, 2005. – V. 16. – P. 201-211.
11. *Тораз М.* Elementary Labanotation: A Study Guide. Second Edition. Hightstown. – NJ.: Princeton Book, 1996.
12. *Уилсон С.Ф., Мэйплс Б., Лэндгрейв Т.* Принципы проектирования и разработки программного обеспечения: пер. с англ. – М.: Русская редакция, 2002.
13. *Gruber T.R.* Towards Principles for the Design of Ontologies Used for Knowledge Sharing // The International Journal of Human-Computer Studies. – 1994. – V. 43. – P. 907-928.
14. *Клещев А.С., Артемьева И.Л.* Математические модели онтологий предметных областей. Ч. 2. Компоненты модели // Научно-техническая информация. Сер. 2. "Информационные процессы и системы". – 2001. – № 3. – С. 19–28.
15. *Никитин В.Ю.* Модерн-джаз танец: История. Методика. Практика. – М.: ГИТИС, 2000.
16. *Johnson V.* Dance training: the funk series // American Fitness Magazine. – 1993. – V. 11. №3. – P. 52–55.
17. *Перцовский С.Л.* Моделирование предметной области “Хореография” // Информатика и системы управления. – 2003. – №2(6). – С. 22–31.
18. *Перцовский С.Л.* Разработка интеллектуальной САПР современного танца на основе онтологий // Вестник ДВО РАН (в печати).
19. *Pertsovskiy S.L.* Computer Simulation of Dance Designing // Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics. – Florida, USA, 2005. – P. 324–329.
20. Базовые движения House, Hip-Hop [Электронный ресурс]: Учимся танцевать клубные танцы (школа танцев SVETIX, г. Москва) Электрон. дан. – М.: ООО “Видеогурман”, 2004. 1 электрон. опт. диск (DVD-ROM). Загл. с контейнера.
21. *Henson D.* Darrin’s Dance Grooves [Электронный ресурс] Электрон. дан. NY: Razor & Tie Direct, L.L.C., 2005. 1 электрон. опт. диск (DVD-ROM). Загл. с контейнера.
22. *Страуструн Б.* Язык программирования C++. Специальное издание: пер. с англ. – М.: Бином, 2005.
23. *Адамс Д.* DirectX: продвинутая анимация: пер. с англ. – М.: КУДИЦ-Образ, 2004.

Статья представлена к публикации членом редколлегии А.С. Клещевым.