

Средства представления структур в системах визуализации программного обеспечения

А.Ю. Байдалин, Д.Р. Исмагилов
ИММ УрО РАН, УрГУ, Екатеринбург, Россия
bajdalin @ imm.uran.ru, idr_uz@mail.ru

Аннотация

В статье обсуждаются вопросы представления структур и проблемы использования трехмерной графики для визуализации параллельных вычислений. Предложен критерий соответствия

Ключевые слова: визуализация, метафора, параллельные вычисления, когнитивные структуры

1. ВВЕДЕНИЕ

В начале 90-х развитие средств компьютерной графики резко усилило выразительные средства визуализации программного обеспечения. Трехмерность, хорошо зарекомендовавшая себя в научной и информационной визуализации, начала активно использоваться в системах визуализации программного обеспечения. Прежде всего был создан целый ряд трехмерных языков программирования. Также была начата разработка трехмерных отладчиков и средств отображения выполнения программ. Необходимым казалось использование трехмерности в таких системах визуализации параллельного программного отображения, как отладчики правильности и системы настройки производительности. В этих системах начали использоваться технологии виртуальной реальности.

Можно предположить, что добавление размерности в любом случае расширяет словарь языка визуализации. Однако анализ работ и наша практика создания систем визуализации параллельного программного отображения показала, что трехмерность не всегда оказывается необходимой и полезной при визуализации. Почему так происходит? В чем причина избыточности трехмерного представления в визуализации параллельного программного отображения? Эти проблемы изучаются в данной работе.

Что касается анимации, то, как и в случае трехмерности, анимация должна повышать выразительность визуализации. Исходя из общих соображений, можно ожидать, что описание процессов, присущих параллельному исполнению, при помощи анимационных образов эффективно. Анимация достаточно успешно использовалась в системах отладки параллельных программ, реализованных еще в 80-ых годах. Однако, в дальнейшем примеры использования анимации в системах визуализации параллельного программного отображения стали редкими.

Наши попытки использования анимации в примерах визуализации параллельного программного отображения не дали эффекта. Есть мнение, что при современных методиках отладки правильности и настройки эффективности параллельных программ анимация вообще не нужна. Так ли это? Вот еще один вопрос для исследования.

В более общем случае рассматриваемые вопросы формулируются следующим образом:

По каким критериям можно оценивать системы визуализации параллельного программного обеспечения?

Как проектировать языки визуализации, чтобы все его элементы улучшали его выразительные свойства?

2. СТРУКТУРЫ ОБЪЕКТОВ ВИЗУАЛИЗАЦИИ

Определим метафору визуализации как отображение, ставящее в соответствие понятиям и объектам моделируемой прикладной области систему сближений и аналогий и порождающее некоторый изобразительный ряд (набор видов отображения) и набор методов взаимодействия с визуальными объектами [1].

Анализ результатов переноса метафор видов отображения в разных областях визуализации показал наличие закономерностей, определяющих его успех или неудачу, независимых от природы визуализируемой информации. Для успеха должна существовать связь между внутренней природой и структурой модельных объектов (и соответствующих им данных) и внутренними ментальными структурами пользователя, в которых представлена сложившаяся у него картина явления (так называемые «репрезентативные когнитивные структуры») [2]. При этом связи между теми и другими структурами не являются произвольными.

Человек выделяет в картинке какие-то общие закономерности, разбивая единую картинку на совокупность (быть может вложенных) фрагментов, абстрагируясь от второстепенных деталей [2].

Таким образом, можно говорить о когнитивных структурах, структурах анализируемых сущностей и структурах визуальных объектов и образов.

При анализе визуального образа интерпретация происходит в двух фазах. Выделяются элементы, интерпретируемые в соответствии со знанием, полученным из предметной области. Из общего визуального образа выделяется (восстанавливается) информация о взаимосвязях между этими элементами и, тем самым, обеспечивается связное адекватное представление о визуализируемом объекте. То есть интерпретация частично происходит на основе самого визуального образа, а частично на основе анализа его частей и их взаиморасположения.

Информация о взаимосвязи элементов указывает на структуру визуализируемого объекта (точнее, нашего представления о нем).

Процесс визуализации можно рассматривать как построение визуального (геометрического) образа на основании абстрактных представлений об объекте. Эти абстрактные представления есть модель (исследуемого объекта, явления, или процесса), как-то связываемая с имеющимся у пользователя когнитивными структурами, описывающими данную сущность. Визуальные образы, представляющие моделируемую сущность, служат для того, чтобы создать или восстановить по нему когнитивные структуры. Цель исследований в области визуализации состоит в создании таких методик и принципов, которые обеспечат восстановление по визуальным образам адекватных когнитивных структур. Порождение когнитивных структур по визуальным образам и есть процесс интерпретации. Этот процесс является обратным или, точнее, двойственным

визуализации. Аналогично принципам визуализации, существуют принципы, на которых должна базироваться интерпретация.

Рассмотрение структуры модельного (исходного) объекта может снизить сложность разработки представляющих его видов отображения. В этом случае сложный вид отображения может разрабатываться как комплексный вид отображения. Для представления компонент исходного объекта можно использовать некоторый набор визуальных объектов. Также предполагается использование набора приемов визуализации отношений между элементами.

Виды отображения структуры предполагают не зависящие от конкретики способы изобразить элементы, находящиеся в данном отношении. Вид отображения структуры есть способ задания методик ее интерпретации. Конкретизация вида отображения с учетом специфики исходной сущности определяют сами объекты интерпретации.

Задача визуализации – получить виз.образ, по которому будет правильно восстановлен ментальный образ (представление) об исходном объекте.

Таким образом, разработка конкретного вида отображения сложной исходной сущности может состоять из выбора способа отображения соответствующей структуры и способов представления ее элементов. Расположение визуальных образов элементов структуры необходимо проводить в соответствии с правилами, предписывающими способы визуализации отношений между этими элементами.

Если выбрать неправильные способы визуального представления исходных сущностей, то интерпретация будет невозможна, так как образы непонятны. Также может случиться, что выбран неправильный способ передачи отношения между элементами. В этом случае будет воссоздана ошибочная когнитивная структура, тем не менее по составу похожая на исходную. Поэтому необходим какой-то критерий правильности видов отображения. В данном случае построение видов отображения могут основываться на правиле, что структура построенного визуального образа не должна противоречить структуре исходного объекта. При интерпретации визуального представления объекта не должно возникать отношений, отсутствовавших в прообразе.

Показ как трехмерного объекта сущности, имеющей двумерную структуру, может привести к неправильной интерпретации. (Хотя часто повышение размерности изображения оказывается полезным для удобства взаиморасположения объектов в сцене). В тоже время редукция размерности, также как и редукция всей структуры не есть ошибка при условии, что пользователь (интерпретатор) проинформирован о ее осуществлении. Естественно, когда элемент не отображается, то отношения, в которых он находился с другими элементами, также не отображаются. Хотя не отображать какие-либо отношения нежелательно, но в ряде случаев это неизбежно.

Можно рассмотреть с позиции критерия сохранения структур трехмерные способы представления для визуализации параллельных вычислений. В ряде случаев трехмерность избыточна, так как визуализируемые сущности не являются трехмерными. Например, в системах настройки эффективности используется графы вызовов. Для адекватного изображения этих графов достаточно плоскости. Трехмерность же необходима скорее для повышения наглядности, увеличения информационной плотности, а также чтобы выделять более важные элементы. Так, конические деревья (располагающие граф-дерево в трехмерном пространстве) обеспечивают фокусировку

внимания пользователя на конкретном пути от корня к листу за счет выноса соответствующих ребер на передний план. Это связано с особенностями человеческого восприятия визуальной информации. Однако, если рассматривать более сложные сущности, например, граф вызовов, включающий некоторый цикл, распараллеленный по двумерному массиву процессоров, то трехмерность оказывается полезной, чтобы сущность "выполнение операций на матрице процессоров" было отображено с сохранением отношений.

3. МЕТАФОРА КОМНАТЫ И ЕЕ АНАЛИЗ

В связи с изучением структурного соответствия при визуализации важно отметить, что метафора позволяет понимать довольно абстрактные или по своей природе неструктурированные сущности в терминах более конкретных или по крайней мере более структурированных сущностей [4].

В качестве объекта изучения была выбрана метафора комнаты. Реализации этой метафоры часто характеризуются сочетанием трехмерного пространства самой комнаты с двумерностью объектов, помещенных ее на «стенах», потолке или полу. Такое сочетание с одной стороны не нарушает принципов структурного соответствия модельных сущностей и визуальных объектов, а с другой обеспечивает удачное пространственное размещение образов.

Анализ метафоры комнаты проводился с двух позиций – что включает в себе метафора комнаты и как с ее помощью можно представлять данные.

Метафора комнаты обладает следующими свойствами:

1. Способность содержать какие-либо объекты внутри себя. Комната не только представляет собой отдельный объект, но и является контейнером для других ones.
2. Ограничение контекста восприятия. Объекты внутри комнаты можно рассматривать в отрыве от «внешнего мира».
3. Замкнутость. Для работы с метафорой комнаты не требуется дополнительных элементов, кроме, возможно, объектов, расположенных внутри комнаты.
4. Включение в структуру. Из комнат можно «строить здания», рассматривая совокупность комнат. Поэтому комната может являться элементом построения некоторой сложной конструкции.
5. Естественность метафоры. Комната является естественной метафорой, с наличием соответствующего объекта в реальном мире. Это свойство делает интуитивно понятными все вышеописанные свойства. При использовании метафоры не возникает дополнительных аналогий и неестественных образов. Функциональность и характеристики реального объекта просто переносятся (хотя и в несколько расширенном понимании) в виртуальный мир.

Рассмотрим, какими способами можно передать информацию при использовании метафоры комнаты.

Так как комната есть контейнер, то естественно первичный способ передачи информации – помещение в нее объектов, и осуществление передачи с помощью них. Причем можно рассматривать различные способы размещения объектов внутри комнаты. С одной стороны информация может быть передана типом (видом) объекта, без учета его размещения, с другой стороны можно рассматривать однотипные объекты, а основную информацию будет нести их местоположение в комнате. Так же можно рассматривать одновременное применение обоих способов. Располагать объекты внутри комнаты естественней всего на ее «стенах». Однако

расположение элементов во внутреннем пространстве, между стенками также может быть оправдано. Заметим, что расположение объектов на «стенах» комнаты является естественным. В повседневной жизни нас окружают комнаты, в которых на стенках расположены объекты, третье измерение которых часто не играет значительной роли в восприятии (в качестве примера – картина на стене).

Информацию может нести и интерьер комнаты. Под интерьером можно понимать такие характеристики, как цвет и геометрическая форма. То есть, если рассматривать совокупность комнат, то по форме и/или цвету можно выделить различные их классы. Например, возможен обзор динамики выполнения программы «изнутри» посредством использования формы для передачи типа программной конструкции. Можно придумать соответствующую форму для каждой значимой языковой конструкции. Цвет комнаты вычисляется на основе содержания «комнаты»-функции. Цвет, например, может определяться количеством наследников или объемом обрабатываемых данных.

Можно рассматривать совокупность разнотипных комнат. В этом случае связи между комнатами могут передавать структурные отношения в сложном объекте. При таком рассмотрении можно применить еще две характеристики – местоположение комнаты в пространстве и ее положение относительно других комнат. Также можно представить предопределенное, строго заданное расположение комнат в пространстве (стена, небоскреб и т.п.). Но такое расположение может нести меньший объем информации о комнатах, составляющих структуру.

Динамическое изменение характеристик комнат во времени может являться источником информации. Возможна анимация в рамках всей совокупности комнат. Кроме того, анимация может затрагивать не только изменение пространственного положения, но и других характеристик комнат – цвета, размера, формы и т.д.

Имели место несколько попыток использования идеи контейнера (причем не только в рамках метафоры комнаты [5]). Но на наш взгляд они не достигли результатов в виде повышения уровня интерпретации, даже с использованием эффекта прозрачности. Возможно, что частичные неудачи связаны с ограничениями в восприятии структур как объектов, вложенных друг в друга.

Проводя анализ свойства ограничения контекста восприятия, оказывается, что это не всегда хорошо. Например, при работе с совокупностью взаимосвязанных комнат, может быть полезно посмотреть, что находится в комнатах, связанных с данной, причем «не переключая» текущую комнату, то есть, не сменяя всего контекста. (Такая возможность реализована в системе Avatar [6].)

4 ПРИМЕНЕНИЕ МЕТАФОРЫ КОМНАТЫ

Рассмотрим применение метафоры комнаты как с точки зрения интерактивных систем в целом, так и Software Visualization.

4.1 Применение метафоры комнаты в интерактивных системах

В последнее время, впервые появившаяся в компьютерных играх и используемая далее в компьютерной визуализации, метафора комнаты начинает «захватывать» и такую, казалось бы, устоявшуюся в своей двухмерности область как пользовательский интерфейс.

Возможно, основной характеристикой, которая побуждает использовать метафору комнаты в интерактивных системах, является ее естественность, наличие хорошо известного аналога в реальном мире. Примером здесь может служить трехмерное представление такой повсеместно используемой метафоры как «рабочий стол». Точнее расположение нескольких «рабочих столов» на стенках комнаты, внутрь которой «смотрит» пользователь. (Подобное заявлено в будущей системе от Microsoft – Windows Vista.) Аналогично трехмерный браузер (точнее надстройка Cubic Eye от компании 2Ce над браузером Microsoft Internet Explorer) использует эту метафору. There страницы, загруженные из Интернета, на стенках комнаты отображаются.

Хотя существуют примеры использования метафоры комнаты и в двумерном виде, но в таких системах в смысле визуального представления нет сильных отличий от метафоры «рабочего стола», а следовательно, there is no выигрыша в выразительности.

4.2 Применение метафоры комнаты в системах визуализации параллельных вычислений

В качестве объекта предметной области, представляемого метафорой комнаты при параллельном программировании естественным образом напрашивается процесс (как это фактически и произошло в системе с метафорой гостиницы [7]). То есть рассматривается совокупность комнат как параллельных процессов. В данном случае являются передаваемые сообщения или данные описываются связями между комнатами. Возможно и применение метафоры для анимационного описания динамики программы.

В плане описания данных параллельной программ это есть удобно описать с помощью метафоры комнаты составные структуры данных программы (такие как массивы, структуры, объекты). Пользуясь свойством комнаты как контейнера, размещаем внутри нее элементы составных типов данных. Возможно, большой эффект от визуального представления данных будет в системах с общей памятью. Здесь также напрашивается рассмотрение данных в динамике. Таким образом, будет возможность изучить как (и что также немаловажно – где) меняются данные. Это упрощает, например, поиск мест программы, где обрабатывается и/или пересылается большое количество данных, на основании такой информации могут быть сделаны выводы об эффективности ее работы.

Метафора комнаты может быть применена для интерпретации структуры программы. Например, можно отобразить граф вызовов функций, используя метафору комнаты. По такому комбинированное отображению можно судить о «слабых» в смысле эффективности работы местах программы. Можно обнаружить такие важные особенности, как часто вызываемые функции; функции, в которых программа «проводит» значительное время; функции, ответственные за передачу данных между процессами, etc.

Другим примером применения метафоры является визуализация иерархии классов программы. Иерархия классов очень важна на этапе проектирования программы. От удачного ее построения во многом зависит удобство написания и внесения изменения в программу, что в итоге скажется и на эффективности. В качестве объекта предметной области, который представляет комната, рассмотрим класс. Содержимое класса (данные и методы) будет содержимым комнаты, а отношения наследования/агрегации – связями

между комнатами. В итоге можем получить трехмерный аналог диаграммы классов языка UML. Подобные идеи высказывались в [3].

Как уже говорилось, метафора комнаты уже применяется в интерфейсе различных систем. Если существует какое-то разделение системы на отдельные модули со своими параметрами, механизмами работы, то работу с системой можно построить с помощью разделения «рабочего пространства» на комнаты. В качестве небольшого примера рассмотрим систему визуализации неких данных с возможной их фильтрацией. Можно выделить как минимум две комнаты с различными функциями. Первая функция – получение, сохранение, предварительная обработка данных. Вторая функция – создание и применение к ним фильтров. В случае если работа с программой строится в виде последовательных шагов, то это может быть представлено как «путешествие» по комнатам. Из комнаты получения данных движемся в комнату применения фильтров, где на стенах комнаты можем изучить и сравнить результаты.

Из анализа метафоры комнаты и рассмотрения возможностей ее применения можно сделать вывод о ее большом потенциале для систем визуализации параллельного программного обеспечения.

5 ЗАКЛЮЧЕНИЕ

Малое количество систем, использующих 3D графику и анимацию, при визуализации параллельных программ представляется неслучайным. Опыт реализации трехмерных метафор для систем настройки эффективности параллельных программ указывает также на наличие определенных проблем с как с эффективной реализацией, так и с восприятием данных и их интерпретацией. Анимация успешно использовалась нами в простом варианте использования самого процесса вывода диаграмм Ганта для сравнительного анализа параллельных процессов. По нашему мнению сама природа сущностей, связанных с процессом разработки параллельных программ часто ограничивает использование этих мощных средств визуализации. Рассмотрение проблем структурного соответствия между модельными сущностями, визуальными объектами и ментальными образами показывает, что использование трехмерности и анимации для представления заведомо одномерных или двумерных сущностей часто оказывается избыточным. Анализ метафоры комнаты и рассмотрение ее возможных применений прежде всего задает методику изучения метафор, потенциально применимых в системах визуализации для параллельных вычислений. Кроме того, необходимы дальнейшие исследования природы программных объектов и сущностей, используемых при разработке параллельных программ, их отладке и настройке эффективности. Также полезно активное использование таких технологий как виртуальная реальность. Ситуация может измениться при изменении самого процесса разработки программ, например, при создании средств автоматизированного распараллеливания.

6 БИБЛИОГРАФИЯ

- [1] Авербух В.Л. Метафоры визуализации // Программирование, 2001. N 5, с. 3-17.
[2] Чуприкова Н.И. Умственное развитие и обучение (к обоснованию системно-структурного подхода). Москва-Воронеж. Издательский дом РАО. 2003.

[3] K. Alfert, A. Fronk, F. Engelen. Experiences in 3-Dimensional Visualization of Java Classes Relations. // SDPS Journal of Design Process Science, Vol. 5, № 3, pp. 91-106, September 2001

[4] Lakoff G. The contemporary theory of metaphor // Ed. by A. Ortony Metaphor and Thought (2nd ed.). Cambridge: Cambridge University Press. 1993. Pp. 202--251

[5] Lieberman H. Three-dimensional representation for program execution // Visual Programming Environments; Application and Issues. Los Alamitos: IEEE Comput. Soc. Press. 1990. pp. 555-560.

[6] Reed D., Scullin W., Tavera L., Shields K., Elford Ch.} Virtual Reality and Parallel Systems Performance Analysis // IEEE Computer, V.28, N 11, (November 1995) pp. 57-67.

[7] Tscheligi M., Musil S. An Information Manipulation Environment for Monitoring Parallel Programs / AVI 1994: Bari, Italy. AVI'94, Proceedings of the Workshop on Advanced Visual Interfaces, June 1-4, 1994, Bari, Italy. ACM, 1994. P. 246-248

Об авторах

Александр Юрьевич Байдалин – научный сотрудник Института Математики и Механики Уральского Отделения Российской Академии Наук, отдел системного обеспечения, сектор компьютерной визуализации.

Адрес: Екатеринбург, 620219, ул. С.Ковалевской, 16, ИММ УрО РАН.

Телефон: (343)3753512

Email: bajdalin@imm.uran.ru

Дамир Ришадович Исмагилов – аспирант Уральского Государственного Университета

Адрес: Екатеринбург, 620083, ул. Ленина, 51

Email: idr_uz@mail.ru

Structure representation in software visualization systems

Abstract

The problems of 3D graphics and questions of structure representation used in Parallel Software Visualization are considered. A criterion of the structural correspondence between model entities, visual objects and mental images is offered. Room metaphor is analyzed and its real and potential applications are described.

Keywords: visualization, metaphore, parallel programming, cognitive structures and respresentations

About the authors

Alexandr Bajdalin, researcher at Institute of Mathematics and Mechanics. His contact email is: bajdalin@imm.uran.ru

Damir Ismagilov, Ph.D. Student student at Urals State University, Department of Mathematics His contact email is: idr_uz@mail.ru