

## Метафора комнаты. Примеры разработки систем компьютерной визуализации

Исмагилов Д.Р. Шарпан С.В.  
e-mail: idr\_uz@mail.ru

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

Конечно же одно из первых и самое распространенное использование метафоры комнаты было в игровых программах. Наверное каждый из нас, от мала до велика, если не играл, то слышал про такие программы как Wolfenstein, Doom, Quake и т.п. В виду специфики этих программ нельзя, конечно, говорить о решении проблем нехватки экранного пространства и масштабируемости, но проблема повышения реалистичности и естественности представления информации была здесь решена в полной мере, особенно при использовании средств виртуальной реальности.

Дальнейшие известные примеры использования метафоры комнаты были в системах Avatar [4], TeamWave WorkPlace [5] и описанная в [3] метафора гостиницы (как некоторое продолжение развития метафоры комнаты). Основное качественное отличие первых двух систем в "размерности" представления комнаты. В TeamWave WorkPlace мы видим только одну "стену" комнаты. Раз мы добиваемся повышения реалистичности представления, то мы должны сохранять и размерность модельных объектов в системах визуализации.

Основные плюсы при использовании метафоры комнаты вытекают из списка проблем, которые необходимо решить в процессе написания систем. Среди них такие, как:

*Проблемы масштабируемости и нехватки экранного места.* [2]  
Решение здесь заключается в том, что мы рассматриваем именно трехмерную комнату. Визуальное отображение некоторых объектов занимает достаточно много экранного места. "Переход" в трехмерное пространство изображения дает еще одну степень свободы. Мы можем размещать объекты на "стенах" "потолке" и "полу" комнаты.

Это перекликается с вопросом о естественности изображения. Кроме того, количественное улучшение может дополняться качественным, то есть в трехмерном пространстве мы получаем больше возможностей для изображения каких-то связей между объектами, причем делая это достаточно компактно не увеличивая всего визуального представления. Еще можно заметить, что плюс трехмерности в гибкости расположения объектов. В двумерных системах визуального программирования почти невозможно избежать ситуации перекрытия одного объекта другим, либо невозможно без перекрытия отобразить взаимосвязи некоторых сущностей. Теперь мы, получив еще одну степень свободы, и дав возможность изменения положения, поворота объектов, можем всегда изменить положение так, чтобы различить необходимую деталь изображения.

*Проблема повышения реалистичности и естественности изображения.* Использование метафоры комнаты и отображение объектов на "стенках" соответствует бытовым реалиям. Человек, находясь всю жизнь в стенах комнат, естественным образом переносится в рамки этой метафоры на экране компьютера. Кроме того, мы живем в трехмерном пространстве, значит естественней было бы и изображать в трех измерениях основные сущности.

Разработанная нами система Icon3DL является модельным примером трехмерного визуального иконического языка программирования. Здесь мы визуализируем и предоставляем инструменты для изменения структуры данных программы. Основные визуальные идеи нашего проекта это применение понятий - комната-куб, пиктограмма, анимация. Опишем применение этих понятий в нашей работе. Всю программу мы разбиваем на несколько уровней. Количество уровней на единицу больше количества функций в программе. Каждый уровень мы представляем в трехмерном пространстве кубом-комнатой. На стенках комнаты располагаются пиктограммы, которые интерпретируются как функции, переменные или заголовочные файлы, использующиеся в программе. В каждый момент времени на экране отображается только одна комната. Пиктограммы могут быть связаны с помощью визуальных связей.

Все уровни располагаются в некотором подобии информационной стены или "небоскреба". Стена становится видимой на экране только в процессе перехода с уровня на уровень, это сделано для решения проблем экономии экранного места и уменьшения рассеивания вни-

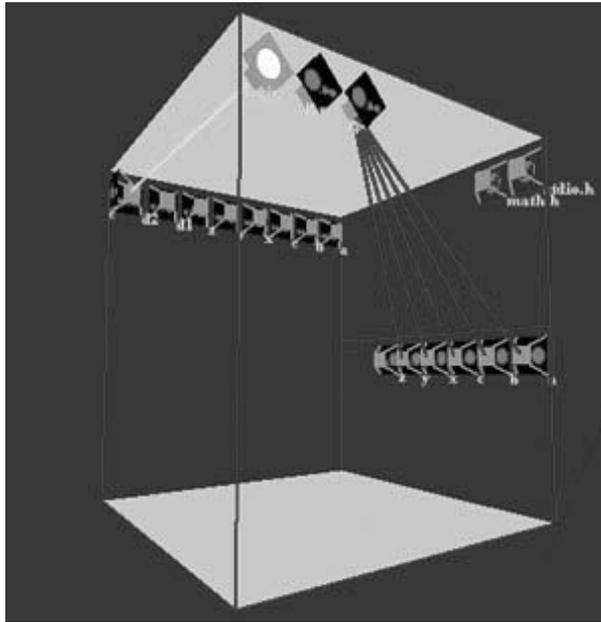


Рис. 1: Комната в Ico3DL

мания пользователя. Сам процесс перехода реализован анимацией.

Для манипулирования структурой программы реализованы возможности добавления переменных, функций, подключения заголовочных файлов. Существует также возможность изменения пространственного положения и возможность поворота изображения, что облегчает восприятие всех деталей визуального представления данной программы. Мы отказались от использования абсолютной трехмерности в отображении комнаты — на стенках выводятся двумерные объекты. Наилучшим способом является на наш взгляд сочетание двух- и трехмерных объектов на стенках комнаты (причем преимущество здесь за двумерностью), что согласуется с визуальным восприятием реальных комнат.

Если в системе Ico3DL на первом месте была сама комната и объекты внутри нее, то во второй нашей разработке - CG Visualization

- на первый план выходит пространственное расположение и взаимосвязь комнат. CG Visualization - система визуализации графа вызовов функций программы. Расположение комнат в этой системе трехуровневое. Все функции программы мы разбиваем на три части: функции пользователя, имеющие в качестве наследников пользовательские функции; функции пользователя, не имеющие таковых и системные функции. Каждой такой части соответствует свой уровень в результирующем изображении.

Аналогично предыдущей системе каждая комната - визуальное представление функции. Кроме того, за функцию "отвечает" и пиктограмма на стене комнаты функции-родителя в графе вызовов. Ребра графа естественным образом переносятся на наше изображение: они следуют от пиктограммы функции на стенке родителя к комнате-кубу данной функции. Изначально весь наш граф затемнен, но при установлении некоторой функции текущей, соответствующая ветвь подсвечивается, привлекая внимание пользователя.

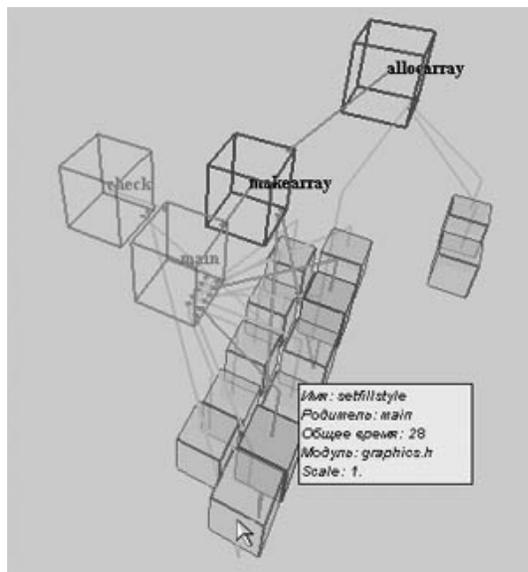


Рис. 2: Комнаты в CG Visualization

Так как вместе с самим графом вызовов включается еще и некоторая совокупность временных отрезков, проведенных потоком управления программы внутри каждой функции, то у нас присутствуют два вида отображения распределения временных отрезков, проведенных в наследниках текущей функции. Эти виды отображения - куб и пирамида. В соответствии с временными показателями куб делится плоскостями, параллельными нижней грани, на несколько частей, каждая из которых подписывается именем функции. Аналогично кубу построено и отображение "пирамида". Однако вместо нескольких плоскостей за временные отрезки отвечают параллелепипеды, поставленные друг на друга.

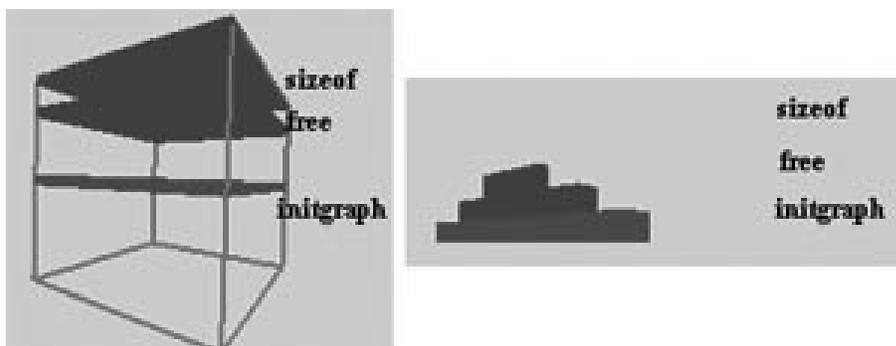


Рис. 3: Куб и пирамида

Наибольшую реалистичность изображения дает вид отображения "изнутри" комнаты в сочетании с возможностью "путешествия" внутри нашего "здания" между комнатами. Но в этом случае мы лишены структурного, объемного видения графа. На наш взгляд сочетание этих двух видов отображения графа дает оптимальный результат.

Дальнейшее развитие метафоры, по нашему мнению, приводит к идее о полном реализме, то есть о полной трехмерности. Метафора комнаты не является универсальной в визуализации, ее применение в каждом конкретном случае должно быть рассмотрено и взвешено. Мы уверены, что метафора комнаты внесет (и вносит) значительный вклад в успехи компьютерной визуализации.

## Список литературы

- [1]. *Авербух В.Л.* Метафоры визуализации // Программирование, 2001. N 5, с. 3-17.
- [2]. *Margaret M. Burnett et al.* Scaling Up Visual Programming Languages // IEEE Computer. Vol.28 No.3, pp.45-54, March 1995.
- [3]. *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
- [4]. *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.
- [5]. *Greenberg S. and Roseman M.* Using a Room Metaphor to Ease Transitions in Groupware. /Research report 98/611/02, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, January.