

Проблемы визуализации социального знания

Г.В. Каныгин, М.С. Полтинникова

Социологический институт РАН
g.kanygin@gmail.com, maria.poltinnikova@gmail.com

Аннотация

В статье мы предлагаем строить структуру знаний на основе элементарных визуальных блоков, для которых определены правила их компиляции в графы. Наглядность визуальных структур для пользователя и естественность правил их компиляции в агрегированные представления открывают новые возможности в развитии информационно-аналитического обеспечения социального управления.

1. Введение

Будем считать, что каждый из нас может иметь свои представления в некоторой области знания. И перед каждым из нас рано или поздно возникает задача предъявить эти представления как самому себе, так и другим людям. Одним из современных направлений решения этой задачи является графическое изображение структуры этих представлений с помощью компьютера. Цель статьи – сформулировать предложения по совершенствованию подобных инструментов с расчетом на неподготовленного пользователя.

2. Методы визуализации представлений пользователя

Пользователи современного компьютера могут по-разному графически выражать свои представления. Мы выделяем четыре возможных способа и различаем их, основываясь на собственном опыте работы с компьютером, и иллюстрируем на примерах известных программных средств.

Во-первых, пользователь может выразить свои мысли в виде картинки с помощью графического редактора. Программа, например, Paint позволит ему рисовать с помощью различных инструментов – перо, заливка, геометрические фигуры и др.

Во-вторых, если человеку удастся выразить свои суждения в виде функциональных зависимостей, то у него возникает возможность их визуализации с помощью средств деловой графики — диаграмм, графиков, таблиц. Таким образом работает, например, Excel.

В-третьих, пользователь может следовать в своих размышлениях некоторой заранее определенной структурной схеме, которую программа будет тут же отрисовывать в виде соответствующего графа. Подобный способ «иллюстрирующей» работы хорошо знаком пользователям, например, Explorer Windows. В этой программе применяется древовидный граф, с помощью которого наглядно представляются связи между файлами, хранящимися на компьютере. Назовем этот случай визуализацией на основе предопределенной структуры или ПС-визуализацией.

В-четвертых, продвинутый пользователь может формализовать свои мысли, воспользовавшись языком спецификации в составе соответствующей инструментальной среды, примерами которой служат известные оболочки визуального программирования — Pascal, C и другие. Выражая свои представления таким образом, пользователь неминуемо создаст программные объекты и отношения между ними. И на каждом этапе этой программистской работы по ее текущим результатам среда даст возможность пользователю наглядно показать создаваемые им собственные представления, например, в виде графа соподчиненности сконструированных им программных объектов. Назовем этот случай визуализацией на основе спецификации.

В последнем случае пользователю придется приобрести некоторую квалификацию, позволяющую работать с современной программной оболочкой. Однако затраченные усилия позволят такому человеку получить в свое распоряжение весь арсенал компьютерных способов визуализации.

Рассмотрим более подробно, как осуществляется работа пользователя по визуализации своих представлений в двух последних случаях. Применение «предопределенной структуры» означает, что в состав компьютерной программы входит структурный компонент, который может редактироваться пользователем. Например, в Explorer таким компонентом оказывается граф типа дерево (tree graph). Наряду с древовидным графом пользователь может визуально представить свои мысли с помощью других предопределенных структур, скажем, концептуальной карты (concept mapping) или сети (network). Так, разработчики одной из программ концептуализации нечисловой информации, осуществляемой с помощью аналитических единиц, называемых кодами, предлагают

их изображать в форме визуальной сети (visual network view of code links and relationships) [1].

В любом случае предопределенная структура составлена из элементов, и пользователю предоставляются средства их редактирования. Человек по своему усмотрению может, например, назначать новые узлы, давать им названия, устанавливать соподчиненность узлов в виде отношения «родитель-дети», устанавливать связи между вершинами одного уровня и др. На каждом этапе таких действий программа позволяет человеку видеть общую картину связей, которая выражает его представления в некоторой области знания. Основываясь на этой общей картине, пользователь имеет возможность осуществлять дальнейшее редактирование.

Как показывает опыт работы с файловыми менеджерами, визуализация на основе предопределенной структуры оказывается естественной и наглядной для пользователя. И эти качества привлекают разработчиков программ разных классов. Например, создателей пакетов анализа нечисловой информации (qualitative data analysis software), нашедших широкое применение в качественной социологии. С помощью этих пакетов исследователи в условиях интервьюирования небольших групп респондентов проводят концептуализацию поставляемой ими речевой, текстовой или другой нечисловой информации. Анализ инструментов пакетов [2] показывает, что существенную роль в таком исследовании играет визуализация с помощью предопределенной структуры.

Такой способ визуализации логических структур оказывается в известной мере универсальным. Это подтверждается его широким применением в случаях, когда приходится выражать не представления отдельных лиц, а знания сообщества специалистов. Для решения подобных задач в настоящее время активно разрабатываются методы управления знанием, в том числе большое распространение получили компьютерные реализации этих методов, известные под названием онторедкторов. Например, Protege [3], HOZO [4], InTez [5].

Не вызывает сомнения, что визуализация «предопределенной структуры» позволяет наглядно представлять знания независимо от предметной области, для которой это знание разрабатывается. Однако следует подчеркнуть, что во-первых, такая наглядность достигается ценой кропотливого участия человека, который вынужден вручную редактировать структуру, предписанные программой. Во-вторых, в этих программах действия человека, осуществляющего визуализацию, не контролируются с помощью алгоритмических средств. Скажем, работая в InTez, пользователь может структурировать и одновременно визуально представить свои онтологии в виде графа. Однако процедура его работы принципиально не изменится по сравнению с Explorer. Она будет по-прежнему состоять в

пошаговом редактировании структуры графа, и не предполагать никаких проверок, например, того, как согласованы между собой разные области создаваемого человеком графа. Пользователь, если захочет и сумеет, должен сам вручную согласовать интересующие его подструктуры.

Выразим тезисно особенности визуализации на основе предопределенной структуры.

Представления отдельного пользователя могут быть наглядно выражены с помощью предопределенной структуры. «Наглядно» означает, что пользователь получает возможность видеть общую картину связей между аналитическими единицами своих представлений, чего он лишен в случае, например, изложения своих мыслей на естественном языке.

Предопределенная структура никак не контролирует «параметры» представлений, выражаемых с ее помощью. Человек непосредственно создает элементы ПС, программа лишь отрисовывает ПС по совокупности действий человека.

Представления отдельного человека по некоторому сюжету не обязательно согласованы с представлениями другого человека по тому же сюжету. ПС никак не позволяет согласовать представления разных лиц.

ПС визуализация представлений пользователя не использует идею их разбиения на модули с последующей «компиляцией» этих модулей.

Рассмотрим теперь второй способ визуализации, который назван выше «основанным на спецификации». Этот случай легко представить, если вспомнить работу программиста, например, в среде объектно-ориентированного программирования. Человек описывает свои представления о некоем сюжете на языке спецификации в виде формально разрозненных программных объектов (типов, форм, экземпляров и т. п.) и действий с ними. В такой форме его представления лишены наглядности, которая была обеспечена при ПС визуализации.

Сложатся или нет эти формально описанные объекты и действия в единое исполняемое тело программы, определяется тем, насколько программистское изложение согласовано с правилами компиляции. В том случае, если компиляция прошла, становится возможным увидеть наглядную структуру отношений между представлениями программиста, выраженными на языке спецификации. Тем самым визуализация осуществляется без непосредственного участия пользователя, но по результатам его работы по созданию текста спецификации.

При таком способе визуализации соглашения и правила, которые применяются при компиляции, оказываются инструментом контроля действий пользователя по выражению его представлений. Вместе с тем одни и те же правила компиляции приложимы к действиям каждого из пользователей, работающих с одной и той же языковой спецификацией. Наличие контроля за действиями пользователя, которые он выражает на формальном

языке, и одинаковость средств такого контроля для каждого пользователя позволяет нам различать *представление* и *знание* пользователя.

Различие между индивидуальным мнением и индивидуальным знанием состоит в наличии или отсутствии правил, которые введены априори и согласно которым человек структурирует свои мысли. Фактически такие правила являются добровольными соглашениями, которые берет на себя каждый из числа лиц, действующих по программистским канонам.

Таким образом, применительно к проблемам визуализации пользователем одного и того же сюжета или объекта, ПС визуализация — это всегда выражение индивидуального мнения. Визуализация на языке спецификации всегда выражает знание пользователя.

Конечно, знание, формируемое в виде спецификаций, существует только для целей программирования и решает относительно ограниченные задачи самоорганизации совместных действий программистов. Однако принципы подобной организации знания, пусть существующие для ограниченной сферы специфических коллективных действий, представляются достойными перенесения в более широкий социальный контекст. Тем более, что современное информационное общество возникло благодаря именно этим принципам.

Модульная организация программ и основанные на ней правила компиляции являются привычными для программистского сообщества. Используя модули, программисты избегают практики многократного создания одной и той же структуры программы, или *структуры знания*. В этом процессе трудно переоценить компилируемость спецификаций на основе предварительно введенных правил. Компиляция реализует проверку структуры знания, которая имеет ряд достоинств. Во-первых, она делает конструктивными действия программиста по определению своих аналитических единиц и отношений между ними. Во-вторых, повышает ответственность за совершаемые действия и задает «стандарт строгости». В-третьих, проверяет формируемую структуру знания как в индивидуальной фазе её создания в виде отдельного программного модуля, так и коллективной фазе, когда объединяются части программы в единое целое.

Резюмируем особенности визуализации на основе компиляции.

Компиляция является способом проверки визуально представленного знания, создаваемого отдельным пользователем, и методом согласования знаний, порождаемых коллективно.

Выражение представлений пользователя осуществляется с помощью утверждений на языке спецификации и не отображается непосредственно в виде воспринимаемой глазами картинке. Тем самым формирование знания на языке спецификации не является наглядным, т.к. разнесены процессы создания знания и его алгоритмической визуальной репрезентации.

Наглядная картина возникает только в том случае, если пользователь сумеет выразить свои мысли согласно правилам компиляции.

Работа со знанием на языке спецификации требует от пользователя квалификации, которую он может приобрести только в результате специального обучения.

Социальное знание и его визуализация

Представим теперь ситуацию, при которой знание создается не командой программистов, а сообществом рядовых граждан. Такое знание назовем социальным знанием. Будем считать, что создателями подобного знания являются разнообразные акторы, в число которых входят органы управления, государственные структуры, общественные организации, отдельные граждане.

Социальное знание характеризуют гигантский объём информации, конструируемость знания участниками социальной коммуникации, разнообразие представлений об одном и том же событии у разных граждан, неподготовленность пользователя к применению каких-либо специальных средств управления знанием. Эти особенности создают проблемы, решение которых невозможно без расширения инструментальных возможностей существующих средств управления знанием.

Отличительной чертой социального знания в сравнении с его аналогом, создаваемым с помощью формальных спецификаций, является неподготовленность пользователя к применению каких-либо специальных средств управления знанием.

Поэтому мы стремимся разработать инструментальные средства, позволяющие расширить включенность рядовых граждан в процессы социального управления. Сочетание наглядного визуального представления социального знания и возможности проверки его логической непротиворечивости в процессе коллективного создания кажутся нам необходимыми условиями широкого прикладного применения информационно-аналитических средств в области социологии управления.

Имеющиеся в современных приложениях методы визуализации знаний, подразделенных нами выше на два случая или группы, не удовлетворяют этим условиям и, следовательно, не смогут нам помочь при формировании знания, создаваемого коллективом рядовых граждан. Действительно, если применять ПС визуализацию знания, то, возможно, результат будет нагляден для самого человека, но не обзорим, а значит, не проверяем при сколь угодно значительном объеме графически выраженных представлений. Ситуация только усугубится, если актором знания окажется сообщество лиц.

Если обратиться к методам визуализации на основе спецификации, то знание будет проверяемо, но практически не реализуемо. Из-за того, что: во-первых, сообщество специалистов в принципе не может адекватно представлять сообщество рядовых

граждан. Во-вторых, мы не можем рассчитывать на то, что рядовые граждане вдруг станут профессиональными программистами.

Мы полагаем, что специфицировать знание, т.е. формально определять типы объектов, экземпляры, фреймы и другие аналитические единицы, равно как и отношения между ними, следует непосредственно в виде структур. Эти структуры должны быть компилируемы. Другими словами, так же как в языковой спецификации, построенной на идеях модульности, в ее структурном аналоге должны существовать базовые «структуры-блоки» или элементарные структуры. Назовем их визуальные модули. Кроме того, должны существовать правила компиляции (соглашения по сборке или агрегирующие правила), которые позволяли бы непосредственно из визуальных модулей получать агрегированные представления, например, в виде графов аналогично тому, как в структурном программировании компилятор из модулей собирает исполняемую программу.

Такой подход в принципе приводит к следующей картине. Знание создается коллективно сообществом социальных акторов (законодателей, специалистов, граждан, органов управления и т.п.). Каждый актор формирует свой фрагмент знания непосредственно в виде интуитивно понятных элементарных структур, основанных на естественных обозначениях. Подход предусматривает правила компиляции этих структур, позволяющие объединять их в конфигурации сообразно задаче, решаемой соответствующим сообществом. Тем самым, давая возможность сообществу акторов проверять связность знания как на этапе его индивидуального создания, так на фазе согласования фрагментов, созданных отдельными акторами.

Литература

- [1] The Intelligent Qualitative Analysis Program – Qualrus. URL: <http://www.ideaworks.com/qualrus/features.html> (дата обращения: 13.09.2014).
- [2] Lewins A., Silver C. Using Qualitative Software: A Step-by-Step Guide. Sage publications, London. 2007.
- [3] A free, open-source ontology editor and framework for building intelligent systems. URL: <http://protege.stanford.edu/> (дата обращения: 13.09.2014).
- [4] Hozo-Ontology Editor. URL: <http://www.hozo.jp/> (дата обращения: 13.09.2014).
- [5] InTez Онтология. URL: <http://www.inttez.ru/> (дата обращения: 13.09.2014).
- [6] Рубашкин В. Ш., Пивоварова Л. М. Методология наполнения онтологий – практика без теории? // Труды Второго Симпозиума «Онтологическое моделирование», г. Казань, 11–12 октября 2010 г. Ред. Калининченко Л. А. - М: ИПИ РАН, 2011. – с. 195 - 214

Практическое применение этого подхода в задачах социального управления, во-первых, позволяет повысить включенность в процесс создания разнообразных социальных акторов (законодателей, специалистов, граждан, органов управления и т.п.), во-вторых, повышает ответственность каждого из акторов за свои социальные действия.

В добавление отметим, что высказанные соображения позволяют решить проблему «бутылочного горлышка», возникающую как раз на фазе применения существующих онтологических методов управления знанием. Суть проблемы в том, что не ясно, кто и как сможет «наполнить» формально специфицированную онтологию [6]. В случае социального знания, создаваемого самими акторами с помощью визуально-структурных средств, эти акторы оказываются, во-первых, «дееспособными» без специального обучения; во-вторых, стимулируются к такому созданию самой социальной коммуникацией.

Заключение

Реализацию описанных идей мы разрабатываем в виде компьютерной модели социального знания. Обязательным условием нашей разработки является интуитивная понятность и наглядность для рядового пользователя базовых структурных объектов, лежащих в основе всей программной реализации.

Мы надеемся, что проводимая разработка позволит нам понять, как работают методы управления знанием, в которых соглашения по компиляции задаются не в виде языковых спецификаций, а с помощью правил, терминами которых являются структуры, например, графы.

Problems of Visualization of Social Knowledge

G. Kanygin, M. Poltinnikova

In this paper we propose to build the structure of knowledge by means of elementary visual units, for which we define rules of compiling them into graphs. This approach provides clear representation of knowledge for ordinary users and opens new opportunities to develop software for social management.