

ЯЗЫК РАЗМЕТКИ КОМПЬЮТЕРНЫХ ПИКТОГРАММ IcoML КАК ИНСТРУМЕНТ ОПИСАНИЯ ИНФОРМАЦИИ В СЕМАНТИЧЕСКОЙ СЕТИ

М. Н. ГРАНИН, И. В. БЫЧКОВ

*Институт динамики систем и теории управления СО РАН,
Иркутск, Россия*

e-mail: graninico@granin.com, bychkov@icc.ru

A new Semantic Web direction is intensively developed to increase the efficiency of internet resources and to move towards a new level of human-computer interaction. According to this concept, a semantic description language of computer icons (Icon Markup Language, or IcoML) is proposed, which allows to automatically process computer icons using their semantics (for example, to organize icons search). Usage of IcoML language will allow developers to significantly reduce the software development timeframe and to increase quality of a graphic interface. The IcoML language is developed in accordance with XML standard and it contains not only description of visual icon presentation, but also the content level which describes the semantics.

Введение

Всемирная компьютерная сеть (World Wide Web) в настоящее время сталкивается со следующими проблемами:

- поисковые системы как основное средство навигации в Интернете недостаточно эффективны;
- полезность сети Интернет для конечного пользователя повышается медленнее, чем растет объем накопленной в ней информации;
- автоматическая обработка информации сильно затруднена.

Для преобразования World Wide Web (WWW) из слабо структурированного набора документов, запрашиваемых пользователями, в хранилище информации, допускающее ее автоматическую обработку, в настоящее время разрабатывается концепция семантической сети (Semantic Web) [1, 2]. В рамках данной концепции предполагается ассоциировать с web-страницей некоторые метаданные. По определению Т. Бернерса-Ли, семантическая сеть — это не какая-то отдельная сеть, а расширение уже существующей WWW; цель ее построения заключается в повышении эффективности совместной работы человека и компьютера, для чего необходимо решить задачи описания “контента”, выделения и формализации смыслового содержимого информации [1].

Переход к семантической сети возможен с помощью создания дополнительного инструментария обработки информации. При этом эффективность будет выше, если будут учтены особенности представления семантического уровня информации. В частности, такой инструментарий требуется создать для формализации представления интерфейсной графики и компьютерных пиктограмм. Как известно, компьютерная пиктограмма (от латинского *Pictus* — нарисованный) — это элемент графического интерфейса пользователя, небольшое изображение на экране, служащее для идентификации некоторого объекта: файла, программы и т. п. В современных системах компьютерная пиктограмма (далее — пиктограмма) представлена набором растровых и векторных изображений, каждое из которых оптимизировано определенными цветом, разрешением и прозрачностью. Это связано с ограничениями, присущими различным дисплеям, операционным системам и приложениям. В качестве примеров форматов файлов пиктограмм можно привести растровые форматы ICO (Microsoft WindowsR), ICNS (Apple Mac OS XR), векторный формат SVG (Linux KDER).

Компьютерная пиктограмма не ограничивается только графическим файлом, поскольку последний не может нести всю информацию, содержащуюся в ней. Как и любой элемент знаковой системы, пиктограмма включает в себя еще *денотат* (отражение предмета из внешнего мира) и нередко *коннотат* (комплекс чувственно-оценочных оттенков, связанных с предметом и понятием о нем). На уровне денотата пиктограммы часто состоят из нескольких элементов, которые могут повторяться в различных пиктограммах.

Назовем модификатором элемент, часто встречающийся в коллекции пиктограмм и занимающий фиксированное положение; он имеет одну и ту же семантику в разных пиктограммах. Проблема заключается в том, что на сегодняшний день средства поддержки метainформации в форматах файлов пиктограмм чрезвычайно бедны. Фактически они ограничены именем файла и порядком изображений внутри файла, поэтому невозможен даже эффективный поиск пиктограмм с учетом их структуры.

Целью настоящей работы является разработка средств и методов обработки формализованного описания информации, содержащейся в пиктограммах, основанных на общепринятых открытых стандартах. Эта формализация должна позволять полностью описать информацию, содержащуюся в пиктограмме, указывая ее четкий смысл в зависимости от контекста. То есть можно считать данную постановку задачи как уточнение формулировки основной задачи Semantic Web в приложении к компьютерным пиктограммам.

В работе предлагается вариант решения этой задачи для компьютерных пиктограмм (icons) с использованием разработанного языка описания компьютерных пиктограмм (IcoML — Icon Markup Language) подобно тому, как языки CML (Chemical Markup Language) и MathML (Mathematical Markup Language) используются для описания химической и математической информации.

1. Задачи

Существующие системы представления и обработки пиктограмм достаточно эффективно реализуют основные функции, однако имеют ряд существенных недостатков. Так, приложение Skype поддерживает множество платформ — Pocket PC, Linux, Mac OS X, Windows, но и оно не позволяет единообразно управлять информацией, содержащейся в пиктограммах (например, информацией о размере и цветности иконки). Большинство программных продуктов содержит избыточный набор пиктограмм, дублирующих либо полностью, либо

частично друг друга. Это не позволяет организовать эффективный поиск необходимых пиктограмм. Так, в продукте LiveLink оказалось возможным оптимизировать существующий набор пиктограмм и сократить его с 1600 до 800 элементов за счет исключения пиктограмм, имеющих одинаковое содержание. При реализации приложений нет инструментария, обеспечивающего автоматическое отслеживание контекста использования визуально схожих пиктограмм. Например, в приложениях звукозаписи пиктограмма “начать запись” имеет вид красного круга, соответственно, пиктограмма такого вида не может быть использована с другим значением в данном контексте (например, о недостатке места на диске красным сигналом светофора в виде красного круга). На данный момент также нет средств для автоматической генерации пиктограмм даже в тех случаях, когда существует необходимый алгоритм (например, все модификаторы со всеми объектами), так как не описано, каким образом пиктограмма связана со своими коннотатом и денотатом.

Таким образом, для устранения отмеченных недостатков разрабатываемый язык должен обеспечить решение следующих задач:

- отображение пиктограммы на различных устройствах и платформах;
- описание содержания пиктограммы, которое позволит эффективно организовывать поиск пиктограмм, визуально схожих с исходной;
- описание обработки пиктограммы в различных контекстах;
- сочетание различных аспектов информации, содержащейся в пиктограмме;
- встраивание пиктограмм в различные контексты.

В решаемой задаче обработки сложных графических данных в первую очередь необходимо определение структуры этих данных, тогда как построение алгоритмов их обработки может быть достигнуто различными средствами (в частности, XSLT), поэтому не вызывает сложностей. Например, можно было бы средствами XSLT генерировать HTML-страницу с полнотекстовым описанием и предварительный просмотр с учетом возможностей терминала, если бы информация, содержащаяся в коллекции пиктограмм, была бы описана XML-языком.

Следующая задача, которую IsoML должен позволить решать в более далекой перспективе, — это автоматическое построение языка компьютерных пиктограмм для какой-либо предметной области, уже описанной в соответствии со стандартами Semantic Web. Так, для предметной области, описанной на OWL (Ontology Web Language), можно автоматически построить язык компьютерных пиктограмм. Это позволит не только упростить разработку графического интерфейса, но и повысить его качество. Возможен будет также “перевод” с одного языка пиктограмм на другой, например, при переносе приложения с платформы Windows XPR на платформу Windows VistaR, так как эти системы различаются визуальным языком пиктограмм.

Разрабатываемый язык должен позволить включить информацию, содержащуюся в пиктограммах, в семантическую сеть. Например, автоматические агенты обработки пиктограмм в семантической сети могут быть использованы для автоматического сбора в сети Интернет разных метафор для одной пиктограммы, при отсутствии в собственной коллекции соответствующих запросу пиктограмм, автоматический агент магазина пиктограмм запрашивает их у бизнес-партнеров.

В результате анализа текущего состояния средств поддержки метаинформации о пиктограммах и для решения перечисленных выше задач были сформулированы следующие требования к языку IsoML. Язык IsoML должен обеспечивать:

- переносимость информации между различными системами;
- разработку более эффективных и гибких средств навигации;

— автоматизацию обработки информации, в частности осуществления поиска по информации, содержащейся в пиктограммах.

2. Описание языка IsoML

Разработанный язык IsoML — это язык семантического описания [4], основанный на XML (Extensible Markup Language). Его синтаксис (детали использования тэгов, атрибутов, сущностей) соответствует правилам синтаксиса XML, а грамматика (детали, касающиеся элементов и атрибутов IsoML, вложенности элементов и т. д.) определяется IsoML DTD (Document Type Definition).

Работа по созданию языка IsoML включала следующие этапы:

- сбор материала (более 5000 компьютерных пиктограмм, примерно 100 контекстов);
- построение онтологии собранного материала;
- разработка спецификации языка IsoML;
- верификация языка IsoML, проверка его полноты (достаточности языка для описания всей значимой информации, содержащейся в пиктограмме) для описания собранных пиктограмм.

Опишем таксономию элементов IsoML. В соответствии с концепциями XML [5] язык IsoML предусматривает описание компьютерных пиктограмм на двух уровнях: на уровне представления (Presentation Markup) и уровне содержания (Content Markup).

Специфичным для IsoML является разделение уровня описания представления на уровень разметки графических данных и уровень разметки метаданных, относящихся к графике (далее — графические метаданные). Соответственно, все элементы IsoML делятся на четыре группы:

- элементы уровня графических данных;
- элементы уровня графических метаданных;
- элементы уровня содержания;
- интерфейсные элементы.

3. Решение поставленных задач средствами языка IsoML

Уровень разметки графических данных (Задача 1. Отображение на различных устройствах). Элементы уровня графических данных описывают визуально ориентированную двумерную структуру компьютерной пиктограммы. Основанный на XML-стандарте язык масштабируемой векторной графики (Scalable Vector Graphic, SVG) удовлетворяет большей части требований, выдвигаемых к разметке графических данных IsoML. Выбор стандарта SVG обусловлен следующими преимуществами:

- SVG позволяет управлять элементами через DOM (Document Object Model);
- отображением элементов в SVG можно управлять с помощью стандартных стилевых механизмов (CSS2, XSLT) либо непосредственно с помощью атрибутов. Последнее позволяет, в частности, менять эффекты отображения элемента управления, представленного в виде SVG-графики, в зависимости от контекста;
- SVG поддерживает анимацию с помощью языка интеграции синхронизированного мультимедиа SMIL (Synchronized Multimedia Integration Language);
- SVG не является собственностью какой-либо компании;
- SVG поддерживает включение растровых элементов (PNG, JPEG);

- SVG (в отличие от своего основного конкурента — формата SWF) позволяет осуществлять XSLT-трансформации;
- SVG может индексироваться поисковыми системами;
- SVG может включать внешние пространства имен XML (foreign XML-namespaces).

Уровень разметки графических метаданных (Задача 2. Описание содержания пиктограммы). Элементы этого уровня описывают элементарные объекты, составляющие пиктограмму: их взаимное расположение и состояние. Описание состояния включает указание варианта объекта и описание модификации его изображения.

На данный момент разметка графических метаданных включает около сорока элементов, принимающих около десяти атрибутов.

Уровень содержания (Задача 3. Значение пиктограммы в разных контекстах). Для того чтобы IcoML мог описывать семантическое содержание пиктограммы, необходимо включение в него соответствующего уровня. Уровень описания семантического содержания позволяет, в частности, описывать не только отдельные пиктограммы, но и наборы компьютерных пиктограмм, связанные между собой сложными отношениями. Кроме того, описание семантического содержания дает возможность учитывать значение компьютерной пиктограммы в зависимости от используемого контекста, а также описывать сдвиг семантического значения, содержащегося в пиктограмме (например, метафору). Таким образом, IcoML позволяет записывать как нотацию, представляющую объект, так и саму структуру объекта, а также метафорический смысл, вложенный в объект.

Смешанная разметка (Задача 4. Описание различных аспектов информации). Можно смешивать все три типа записи для определения графических данных, графических метаданных, содержания пиктограммы. Смешанная разметка, включающая несколько уровней IcoML, исчерпывающим образом описывает всю информацию, связанную с компьютерной пиктограммой или набором пиктограмм. В настоящей редакции IcoML в качестве основного механизма смешанной разметки использованы семантические примечания (элемент `semantic`, аналогичный одноименному элементу в MathML).

Уровень интерфейса (Задача 5. Встраивание в различные контексты). Интерфейсные элементы предназначены для встраивания IcoML-разметки в различное окружение. Например, декларация `!DOCTYPE` указывает приложению на тип обрабатываемого документа. С помощью механизма `namespace` можно в одном документе достаточно гибко использовать несколько языков, определяемых разными DTD.

Приведем пример совместного использования XHTML, SVG и IcoML в одном документе.

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC
"-//W3C//DTD XHTML 1.1 plus IcoML 0.2 plus SVG 1.1//EN"
"http://www.icoml.org/2006/04/xhtml-ico-svg/xhtml-ico-svg.dtd"[
<!ENTITY % SVG.prefix "IGNORE">
<!ENTITY % XHTML.prefix "INCLUDE">
<!ENTITY % XHTML.prefix "xhtml">
<!ENTITY % ICOML.prefix "INCLUDE">
<!ENTITY % ICOML.prefix "ico">
]>
<svg version="1.1" xml:lang="en"
  xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink">
```

```

xmlns:xhtml="http://www.w3.org/1999/xhtml"
xmlns:ico="http://www.icoml.org/2006/icoml">
<desc>SVG as the host language</desc>
... SVG содержание ...
<switch>
  <foreignObject ...>
    <xhtml:p>
      ... XHTML содержание ...
    <ico:ico>
      ... IcoML содержание ...
    </ico:ico>
    ... XHTML содержание ...
  </xhtml:p>
</foreignObject>
</switch>
... SVG содержание ...
</svg>

```

Вслед за стандартным заголовком XML-документа идет заголовок !DOCTYPE, указывающий на профиль xhtml-ico-svg.dtd. Данный профиль позволяет соединить в одном документе расширяемый язык разметки гипертекста (XHTML) 1.1, язык описания компьютерных пиктограмм (IcoML) 0.7 и масштабируемую векторную графику (SVG) 1.1, используя механизм пространств имен, и в то же время дает возможность сделать проверку правильности такого документа.

Данное DTD может использоваться главным образом для XHTML-документа, содержащего фрагменты IcoML и SVG, а также для SVG-документа, содержащего фрагменты XHTML и IcoML. Фрагмент SVG внутри XHTML-документа может содержать другие фрагменты XHTML или IcoML, а фрагмент XHTML может снова содержать другие фрагменты IcoML или SVG. Ограничений на степень вложенности нет.

В пределах XHTML фрагменты IcoML и SVG разрешено использовать практически повсеместно в теле документа. Внутри SVG с помощью элемента foreignObject могут быть внедрены XHTML и IcoML [3]. Тэги <foreignObject> </foreignObject> указывают на то, что содержащийся между ними код не является SVG и не должен обрабатываться как SVG.

4. Пример разметки IcoML

Рассмотрим пример пиктограммы (см. рисунок), которая на уровне графических метаданных представляет собой стрелку, направленную из левого нижнего угла к центру листа,



Пример пиктограммы.

на котором изображена звезда. На уровне содержания пиктограмма означает импортировать в файл “Избранное”. Контекстом для такой пиктограммы может быть приложение Менеджер закладок “Избранное” в различных форматах, использующихся в различных браузерах. IcoML-разметка пиктограммы, описывающая уровень денотата и представления, имеет следующий вид:

```
<?xml version="1.0"?>
<!DOCTYPE IcoML SYSTEM "icoml.dtd»
<mrow id="a0»
<semantics>
  <apply>
    <ПЕРЕД/>
    <mrow id="b1">
      <apply>
        <НА/>
        <ci id="a1"> ЛИСТ </ci>
        <ci id="a2"> ЗВЕЗДА </ci>
      </apply>
    </mrow>
    <cn id="a3">СТРЕЛКА</cn>
  </apply>
  <annotation-xml encoding="IcoML-GraphicalData">
  ... SVG content ...
</semantics>
</mrow>
```

Первая строка — стандартный заголовок XML-документа, во второй строке расположен заголовок IcoML DOCTYPE. Элемент semantics (семантическое примечание) использован для представления выражения разметки графических данных как семантического примечания к выражению разметки графических метаданных. Элемент mrow использован для выделения подструктур пиктограммы на уровне графических метаданных. Данные, содержащиеся между тэгами <mrow> и </mrow>, рассматриваются как аргумент элемента mrow. Таким образом, в элементе mrow содержится целое выражение, которое может рассматриваться как отдельная пиктограмма. Элемент apply обозначает применение операции к выражению. Порядок аргументов этого оператора имеет ключевое значение, так как первый дочерний элемент является оператором, принимающим остальные в качестве аргументов.

В нашем примере применяются пустой элемент ПЕРЕД (для описания расположения объектов СТРЕЛКА и ЛИСТ относительно друг друга) и аналогичный элемент НА (для элементов ЛИСТ и ЗВЕЗДА). Эти элементы имеют несколько аргументов, порядок которых также имеет значение. Для выделения на уровне элементарных объектов ЗВЕЗДА и ЛИСТ применяется элемент ci, элемент cn применен для выделения объекта СТРЕЛКА, который является модификатором.

5. Практическое применение IcoML

Язык IcoML уже сейчас может быть использован следующим образом:

— в коллекциях пиктограмм, распространяющихся на Интернет-сайтах. Таких сайтов

в сети в настоящее время сотни, они содержат десятки тысяч пиктограмм и сотни тысяч файлов. Лишь один из таких сайтов — stockicons.com содержит более 30 коллекций, более 2000 пиктограмм, более 30 000 файлов. Механизмы поиска в этих коллекциях не развиты. Глобальные поисковые системы изображений также не решают проблемы эффективного поиска по информации, содержащейся в пиктограммах. Например, при поиске пиктограмм футбольной тематики на запрос “football icons” в системе поиска изображений Google ImagesR выдается несколько тысяч ссылок на различные изображения, подавляющая часть которых нерелевантна запросу, и существующие способы повышения релевантности недостаточно эффективны;

— как стандартное, не зависящее от платформы средство распространения пиктограмм, а также как открытый формат, основанный на общепризнанных стандартах и поддерживающий отображение пиктограмм на различных устройствах и в различных операционных системах;

— как инструмент минимизации количества противоречий внутри знакового языка, представляемого компьютерными пиктограммами, при разработке интерфейсов приложений.

Предложенный язык описания компьютерных пиктограмм нашел практическое применение при разработке Интернет-сервиса каталогизации и поиска пиктограмм IconExpert [6]. В целях совместимости со стандартом и для большей гибкости в последующих редакциях языка планируется DTD заменить на XML Schema. XML Schema превосходит DTD по многим параметрам, включая поддержку пространства имен (namespaces). Например, XML Schema позволяет определять элемент как integer, float, boolean, URL и др. В представленной редакции языка выбран DTD, поскольку он по сравнению с XML Schema позволяет относительно быстро описать грамматику языка.

В настоящее время ведется доработка уровня описания графических данных с тем, чтобы оно могло включать информацию о различных вариантах детализации векторных пиктограмм. В дальнейшем планируется использовать полученные результаты в смежных областях, таких как каталогизация клипартов, фотобанков, видеоархивов. Не вызовет проблем расширение IsoML до языка описания не только компьютерных пиктограмм, но и всех изображений, используемых в графическом интерфейсе пользователя.

Заключение

Предлагаемый язык IsoML, основанный на открытых и общепринятых стандартах XML, SVG, XSLT, спроектирован для хранения и отображения значительного объема разнородной информации в области компьютерных пиктограмм, а также для поддержки семантики и связей в этой информации. Язык IsoML позволяет разрабатывать автоматизированные средства для повышения эффективности доступа к необходимой информации, что особенно актуально в условиях существования огромного количества систем, использующих пиктограммы.

Список литературы

- [1] BERNERS-LEE T., HENDLER J., LASSILA O. The semantic web // Scientific American, 2001. www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2

- [2] W3C Semantic Web. www.w3.org/2001/sw/
- [3] SCALABLE Vector Graphics (SVG) XML Graphics for the Web. www.w3.org/Graphics/SVG/
- [4] ПРАТТ Т., ЗЕЛКОВИЦ М. Языки программирования: разработка и реализация. СПб.: Питер, 2002. 686 с.
- [5] SEPARATION of semantic and presentational markup, to the extent possible, is architecturally sound. www.w3.org/2001/tag/doc/contentPresentation-26.html
- [6] ICONEXPERT icons search service. www.iconexpert.com/

Поступила в редакцию 29 июня 2006 г.