

# МЕТАСВОЙСТВА ВИКИ-ФОРМАТИРОВАНИЯ И ПРОГРАММ ВЕБ-СЕРВЕРОВ

Колодин М.Ю.

---

УДК 006.72

*Колодин М.Ю. Метасвойства викиформатирования и программ веб-серверов.*

**Аннотация.** Наряду с «большими» форматами в Интернете для представления данных и знаний все чаще применяются «малые» форматы, позволяющие сохранить и даже усилить метасвойства «больших». Это, в частности, упрощенные версии формата XML и разнообразные варианты вики-форматирования. Использование их позволяет получить ощутимый выигрыш в скорости действия систем, в том числе веб-служб, повысить производительность труда человека.

**Ключевые слова:** форматы, вики, веб-сервера, представление и преобразование данных.

*Kolodin M.Y. Metaproperties of wiki-formatting and Web-engines.*

**Abstract.** Along with «big» formats some «small» formats are now widely used for data and knowledge representation in the Internet. These formats allow to preserve and enhance meta-facilities of «big» formats. The «small» formats are simplified versions of XML, various variants of wiki markup. Their application allows to get significant gain in systems data processing speed, including web services, increase human labor productivity.

**Keywords:** formats, wiki, Web engines, data representations and transformations.

---

**1. Введение.** В современной информатике наряду с форматами данных, имеющими полные объемные спецификации (далее называемыми «большими», например, ODF (OpenOffice.org) [1], OOXML (Microsoft Office) [1], HTML, SGML, XML [2, 3], XSLT [3], XSL-FO, TeX [4] и т. п.) все активнее применяются и более простые форматы (будем называть их «малыми»; это, например, разнообразные варианты вики-разметки [5], а также упрощенные форматы, близкие к XML, например, YAML). Многие из «малых» форматов сохраняют основные свойства «больших», в том числе имеют некоторые полезные метасвойства [6]. Реализация таких языков выполняется обычно в виде программных преобразователей, как самостоятельных [5, 7], так и встроенных в веб-движки [8] (например, MediaWiki).

«Малые» форматы имеют свои преимущества, в частности, при сохранении основных функциональных возможностей «больших» форматов по представлению данных они значительно проще для синтаксического разбора (соответственно разбор выполняется быстрее и может быть встроен в динамические системы типа веб-сайтов и служб), проще и быстрее для набора оператором, зачастую нагляднее и компактнее, содержат меньше оформительских элементов при том же или близком наборе параметров, что и в «больших» форматах.

Труды СПИИРАН. 2009. Вып. 11. ISBN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)  
SPIIRAS Proceedings. 2009. Issue 11. ISBN 2078-9181 (print), ISSN 2078-9599 (online)

Задача исследования состоит в выборе способов оптимального применения «малых» форматов с максимальным развитием метафункциональности, получаемой с помощью программ на серверной стороне информационной системы.

**2. Назначение «малых» форматов и сравнение их с «большими».** Все указанные форматы служат для представления данных, выражаемых как гиперструктурированная система произвольно вложенных элементов, каждый из которых может быть пустым или содержать данные или иные вложенные элементы; при этом каждый элемент может иметь свои атрибуты (в том числе именованные, по умолчанию, позиционные, со значениями или без оных). Приставка «гипер» для структурированности означает наличие перекрестных ссылок на другие элементы данной структуры или на внешние объекты.

*Пример «малых» форматов для XML.* Пусть есть структура для описания студента:

```
<?xml version="1.0" encoding="UTF-8"?>
<person id='p001'>
<name first='Ivan' middle='N' last='Petrov' />
<birth date='1968-03-20' place='St.Petersburg, Russia' />
<study-at dept='phys' group='4705' />
<bio>Студент успевающий, активный.</bio>
</person>
```

В формате YAML этот можно записать как

```
person: p001
  name first:Ivan middle:N last:Petrov
  birth date:'1968-03-20' place:'St.Petersburg, Russia'
  study-at dept:phys group:4705
  bio
```

Студент успевающий, активный.

А в формате JSON как

```
{"id": "p001",
  "name": {"first": "Ivan", "middle": "N", "last": "Petrov"},
  "birth": {"date": "1968-03-20", "place": "St.Petersburg,
  Russia"},
  "study-at": {"dept": "phys", "group": "4705"},
  "bio": "Студент успевающий, активный."}
```

**3. Назначение и возможности вики-разметки.** Вики-разметка предназначена в первую очередь для ускоренного набора веб-текста человеком.

Примеры шрифтового форматирования: вместо записи вида

```
<b>полужирный шрифт</b>и <i>курсив</i>
```

пишется просто

```
'''полужирный шрифт''' и '''курсив''',
```

вместо сложного оформления списков типа

```
<ol>
```

```
<li>1 элемент</li>
<li>2 элемент</li>
</ol>
```

пишем

```
- 1 элемент
- 2 элемент
```

и т. п.

В целом это значительно ускоряет набор. Кроме того, перевод из этого «псевдокода» в реальный HTML выполняет соответствующий транслятор, который заодно проверяет текст на наличие ошибок (известно [8], что именно пользователь является самой ненадежной частью системы).

Более того, набор форматирующих элементов, и еще более — гиперссылок в псевдокоде, позволяет существенно повысить защищенность системы, поскольку защищает ее от потенциальных злонамеренных вставок кода; о важности этого говорит и то, что многие веб-системы (например, форумы) не позволяют пользователю напрямую вводить HTML-разметку, заставляя его применять только псевдокоды типа `[b]текст[/b]`.

Кроме того, такой текст весьма просто описывается регулярными выражениями, а для них есть стандартные быстрые преобразователи на многих языках программирования.

**4. Метасвойства вики-разметки.** Помимо простых средств шрифтового форматирования текста вики-разметка предлагает широкие возможности по полуавтоматическому и автоматическому формированию гиперпереходов и автоматическому (программному) построению соответствующих гиперструктур.

Например, вместо ссылки типа

```
<a href="http://www.spiiras.nw.ru">Сайт СПИИРАН</a>.
```

пишем

```
[[www.spiiras.nw.ru|Сайт СПИИРАН]].
```

Здесь важно, что ссылку формирует программа веб-сервера, а он имеет большие возможности, например вариант MediaWiki, используемый во всемирной свободной сетевой энциклопедии «Википедия» (см. <http://ru.wikipedia.org>), записываемый как

```
www.spiiras.nw.ru,
```

достаточен, чтобы сформировалась простая внешняя гиперссылка вида

```
[[СПИИРАН]],
```

которая в свою очередь формирует внутреннюю для веб-системы ссылку на статью «СПИИРАН», причем наличие и действительное полное имя такой статьи определяется программно в момент показа страницы пользователю. Помимо этого между первоначальным вводом

текста страницы и моментом показа текст может быть неоднократно программно обработан (например, расставлены перекрестные ссылки, выполнена категоризация статьи в целом и составляющих объектов, дополнительные проверки на наличие ошибок и т. п.) [7, 8].

Именно здесь есть большие возможности для использования и развития метасвойств. Действительно, объекты, некоторым специальным образом обозначенные (например, набранные в квадратных или фигурных скобках, возможно, с параметрами; специальные виды сетевых локаторов помимо `http:`, `ftp:` и им подобных), могут распознаваться системой и статически или динамически разрешаться в некоторые другие объекты. Нужно только задать правила соответствия элементам форматирования для обрабатывающих эти объекты программ, а контекст в сочетании с явно заданными параметрами определит особенности обработки. В разных ситуациях (определяемых контекстом) соответствующие параметры будут различны, что автоматически определит способ или особенности обработки. Таким образом, сам выбор обработчика будет определяться сочетанием потенциально многоуровневого контекста и набора явных параметров.

Простой пример: вне контекстов ссылка

[Владимир]

не определена; ее можно уточнить как

[город:Владимир], [имя:Владимир]?

или [персона:Владимир (Красно Солнышко)];

в обобщенном контексте, где допустимы все перечисленные значения, ссылка [Владимир] ведет к списку значений в случайном порядке, а при наличии специального контекста типа

[контекст:города] ... [Владимир] ... [/контекст]

ссылка ведет к приоритетной выдаче конкретного значения; контексты могут быть вложены. При этом «город», «имя», «персона» и т. п. — вызовы обработчиков контекстов. Однако во всех перечисленных случаях результатом может быть не готовый ответ, определенный статическим контекстом, а запрос на поиск соответствующей информации, и тогда результат будет сформирован динамически. Если в предыдущем обращении поставить динамический контекст, определяемый контекстной переменной

[контекст:\$динконтекст1] ... [Владимир] ... [/контекст],

то в результате работы интерпретатора получим нужное значение или набор значений. Контексты также могут быть не единственными, фиксированными, а подобно словарям языка Forth задавать стек контекстов, в котором осуществляется поиск подходящих значений; вообще,

язык Forth традиционно очень хорош для реализации систем подобного рода [5, 9].

**5. Метасвойства программ веб-серверов, использующих мета-разметку и аналогичные «малые» форматы.** Из вышесказанного можно выявить некоторые общие метасвойства вики-разметки и близких ей и XML форматов с тем, чтобы получить возможность явно использовать их полезные свойства.

Программа веб-сервера («веб-движок», «web engine») — активная система, действующая на стороне сервера, получающая запрос по сети, выполняющая необходимые преобразования запроса, поиск ответа, формирование результата на основе имеющихся шаблонов и баз данных, и выдающая ответ пользователю. В том случае, если запрос содержит многоуровневую обработку, серверная веб-программа фактически выполняет метаобработку данных; и будет лучше, если мы с самого начала явно наделим его этими обобщающими возможностями.

В отличие от простого использования языков программирования на стороне сервера, при применении серверных программ мы имеем возможность, с одной стороны, упростить работу по написанию страниц веб-сайта, поскольку значительные возможности уже содержатся в движке или прилагаемых библиотеках, а с другой, реализовав некоторые расширения для обработки определенных языковых конструкций, — существенно повысить уровень языка. Так пакет Tk заметно расширяет язык Tcl, а цепочка серверных надстроек над веб-сервером типа «Python—Zope—Plone—Сайт» позволяет получить мощную среду корпоративного уровня, хотя и ценой некоторой потери быстродействия и повышения ресурсоемкости системы, включающей в себя в этом случае несколько уровней интерпретации.

Обобщения могут включать в себя выбор языка данных запроса, языка шаблона, выбор СУБД, определение (каскадное или иное) и формирование стиля выходного документа, выбор способа разрешения заложенных в ответ объектов, в том числе формирование и разрешение гиперссылок, возможно, требующие многократных обращений в сеть для получения полного ответа.

В целом представляется оптимальным использовать упрощенные XML-форматы для представления гиперструктурированных и потоковых данных, а вики-разметку — для форматирования веб-страниц, причем как в традиционном ручном режиме, так и в автоматическом, для последующей обработки в момент выдачи страниц.

**6. Заключение.** Исследование показало, что использование «малых» форматов указанными способами позволяет получить ощутимый выигрыш в быстродействии систем и понизить их сложность, в том числе это справедливо для веб-служб, а также повысить производительность труда разработчика. При сравнении использовались оценки затраченного на разработку и программирование времени, времени выполнения программ, объема и сложности кода. Сложность выполнения оценок вызвана тем, что фактически сравнивались разные программы и разные механизмы их выполнения. Тем не менее, удалось выявить следующие закономерности. Сокращение времени разработки составило около 30 % при выполнении повторяющихся работ (однако на первоначальную разработку требуется до 50 % времени дополнительно), объем кода уменьшился до 50 %. Точно сравнить затраты времени на выполнение данных программ не удалось, поскольку они ничтожны по сравнению с остальными затратами (на передачу запросов и ответов по сети, вызов и компиляцию серверных программ, работу с базами данных и т. п.), однако не замечено ни ускорения, ни замедления работы. В целом положительный эффект составил в среднем 25–30 %.

Важно также отметить, что рассмотренные форматы и процедуры сохраняют, а в ряде случаев и усиливают полезные метасвойства систем; в этом смысле подход остался бы полезен даже при отсутствии выигрыша во времени. При таком подходе повышается профессионализм разработчика, упрощается код, улучшается взаимодействие программ друг с другом. Определение способов оценки такого эффекта входит в дальнейшие исследования по данной теме.

### Литература

1. Перспективы стандарта электронных документов ISO 26300. ODF. Сборник материалов. М.: INFO-FOSS.RU, 2008. 132 с.
2. *Валиков А.Н.* Технология XSLT. СПб: БХВ-Петербург, 2002. 544 с.
3. *Холзнер С.* XSLT. Библиотека программиста. СПб: Питер, 2002. 544 с.
4. *Котельников И.А., Чеботаев П.З.* Издательская система LaTeX2ε. Новосибирск: Сибирский хронограф, 1998. 496 с.
5. *Колодин М.Ю.* Инструментальные средства разработки, реализации и сопровождения гипертекстовых преобразователей // Тр. СПИИРАН. 2008. Вып. 7. С. 64–69.
6. *Колодин М.Ю.* Синтаксические и семантические особенности метасистем // Тр. СПИИРАН. 2009. Вып. 9. С. 168–177.
7. *Колодин М.Ю.* Межформатные преобразования гипертекста // Тр. СПИИРАН. 2008. Вып. 6. С. 168–170.
8. *Колодин М.Ю.* Разработка, реализация и сопровождение веб-сайта научной организации // Тр. СПИИРАН. 2003. Вып. 1, т. 3. С. 217–223.

9. *Баранов С.Н., Колодин М.Ю.* Феномен форта // Методы теоретического и системного программирования. 1995. Вып. 4. С. 193–271.

**Колодин Михаил Юрьевич** — научный сотрудник исследовательской группы информационных технологий в образовании (ИГИТО) Учреждения Российской академии наук Санкт-Петербургский институт информатики и автоматизации РАН (СПИИРАН). Область научных интересов: метасистемы, свободное программное обеспечение, образовательные технологии. Число научных публикаций — 74. myke@iias.spb.su, www.myke.spb.ru; СПИИРАН, 14-я линия В.О., д.39, Санкт-Петербург, 199178, Россия; р. т. +7(812)328-0382. Научный руководитель — канд. физ.-мат. наук, доцент, ведущий научный сотрудник А.Л. Тулупьев.

**Kolodin Mikhail Yurievich** — researcher, Research Group for Information Technologies in Education (RGITE) of Institution of the Russian Academy of Sciences St. Petersburg Institute for Informatics and Automation RAS. Research interests: metasytems, free and open source software, educational technologies. The number of publications — 74. myke@iias.spb.su, www.myke.spb.ru; SPIIRAS, 39, 14th Line V.O., St.Petersburg, 199178, Russia; office phone +7(812)328-0382. Scientific supervisor — Ph.D., associate professor, leading researcher A.L. Tulupiev.

Рекомендовано группой междисциплинарных проблем информатики СПИИРАН. Научный руководитель А.Л. Тулупьев, канд. физ.-мат. наук, доцент, ведущий научный сотрудник.

Статья поступила в редакцию 14.12.2009.

## РЕФЕРАТ

### **Колодин М.Ю. Метасвойства вики-форматирования и программ веб-серверов.**

Наряду с «большими» форматами в интернете для представления данных и знаний все чаще применяются «малые» форматы. Они характеризуются значительной простотой синтаксиса и семантики, что позволяет облегчить и ускорить создание и обработку их как человеком, так и компьютером. Несмотря на свою относительную простоту, эти форматы позволяют не только сохранить, но и усилить метасвойства «больших» форматов.

Прежде всего, это «облегченные» версии XML-формата, в частности, форматы (языки) YAML и JSON. Каждый по-своему, они дают возможность записать ту же информацию, что и многословный и относительно тяжелый XML, не говоря уж об SGML. Они могут также легко встраиваться в языки программирования, например, в Javascript.

Вики-разметка изначально создана как упрощенный язык для форматирования веб-страниц, где без сложного HTML-кодирования можно выразить большинство основных возможностей гипertextовой разметки. Однако на самом деле ее возможности гораздо шире, если они применяются в сочетании с «умным» веб-сервером (типа MediaWiki), позволяющим «на лету» формировать не только шрифтовое выделение элементов текста, но и сложные гиперссылки, управлять разработкой больших веб-сайтов; часто для сложной обработки применяются программы-«роботы» на серверной стороне.

Здесь имеются значительные новые возможности для развития таких систем, поскольку аппараты макрообработки и формирования гиперссылок могут быть усилены, их можно использовать для получения новых объектов по уже имеющимся при помощи параметризации, информации о контекстах и т. п. Поскольку программа веб-сервера в таких случаях выполняет метаобработку данных, наибольший полезный эффект будет достигнут тогда, когда разработчик изначально явно наделит его соответствующими обобщающими способностями.

Обобщения могут включать в себя выбор языка данных запроса, языка шаблона, выбор СУБД, определение (каскадное или иное) и формирование стиля выходного документа, выбор способа разрешения заложенных в ответ объектов, в том числе формирование и разрешение гиперссылок, возможно, требующие многократных обращений в сеть для получения полного ответа.

В целом представляется оптимальным использовать упрощенные XML-форматы для представления гиперструктурированных и потоковых данных, а вики-разметку — для форматирования веб-страниц, причем и в традиционном ручном режиме, и в автоматическом, для последующей обработки в момент выдачи страниц.

## SUMMARY

### *Kolodin M.Y.* **Metaproperties of wiki-formatting and Web-engines.**

Not only «big» data and knowledge representation formats are now used in the Internet, but the «small» ones as well. They are characterized by significant simplicity of syntax and semantics, that allows to simplify and speed up their creation and processing both by human and computer. Despite their relatively simple form they not only preserve but even enhance metafacilities of the «big» formats.

First of all these are «light» versions of XML format, especially formats (languages) YAML and JSON. Each of them in their own ways allow to store the same information as multiword and relatively heavy XML, to say nothing of SGML. They may be also easily be built in programming languages like Javascript.

Wiki markup was initially created as simplified web pages formatting language, where one can express majority of principal features of hypertext markup without complicated HTML coding. But in fact its possibilities are much wider, if they are used together with «smart» web engine (like MediaWiki), that allows to form not only font decorations on-the-fly, but complex hyperlinks as well, to control development of huge web sites; web robots are often used on the server side for complex data processing.

Significant new possibilities appear here for development of these systems, because the apparatus for macroprocessing and hyperlinks formation may be enhanced and used to obtain new objects basing on the existing objects using information about contexts, parameterization etc. Noting that Web engine in these cases performs metaprocessing of the data, better useful effect is achieved when the developer explicitly initially provides it with the corresponding generalizing features.

Generalizations may include selection of data request language, patterns language, DBMS selection, defining (cascade or another) and forming of the output document style, selection of the way of solving of the objects included in the response, forming and solving of hyperlinks, possibly requiring multiple applying to the network in order to receive complete response.

Generally it is supposed to be optimal to use simplified XML formats for hyperstructured and flow data representation, and wiki markup for Web pages formatting, both in traditional manual mode and automatically, for subsequent processing at the moment of pages' output.