

**Т. А. Волянская**

**МЕЖДУНАРОДНЫЕ СТАНДАРТЫ ПРЕДСТАВЛЕНИЯ В СЕТИ  
ИНТЕРНЕТ ИНФОРМАЦИОННЫХ РЕСУРСОВ  
ПО КУЛЬТУРНОМУ НАСЛЕДИЮ:  
СТАНДАРТ ANSI/NISO Z39.50 И ПРОФИЛЬ СИМІ\***

**ВВЕДЕНИЕ**

В настоящее время в сети Интернет представлено большое число разнообразных и обособленных информационных ресурсов по культурному наследию. В их число входят различные электронные музеи, архивы и каталоги. Информационные ресурсы по культурному наследию имеют самый разнообразный характер: это описательные базы данных, базы данных изображений, видео- и аудио- материалов. Эти ресурсы принадлежат различным организациям, которые проводят самостоятельную политику в отношении их описания, использования и публичного доступа к ним.

Предоставление доступа к информации о культурном наследии организовано различными способами и с применением различных технологий и протоколов, которые часто не согласуются с существующими стандартами. Наиболее распространенным способом является использование Web-технологии с доступом по HTTP протоколу, одним важным недостатком которого является отсутствие глобальной стандартизации на уровне организации данных и форматов их представления. Большинство существующих информационно-поисковых систем поддерживают совершенно различные структуры хранения данных, способы доступа и форматы представления информации и, как следствие, имеют свой собственный пользовательский интерфейс.

Одна из важных задач в области разработки распределенных информационно-поисковых систем по культурному наследию — это унификация доступа и интеграция информационных ресурсов [2]. Целью является объединение имеющихся информационных ресурсов по культурному наследию в единую распределенную информационную систему со сквозным поиском. Это позволит открыть для пользователей всю информацию, накопленную за

---

\* Работа выполнена при финансовой поддержке Российского гуманитарного научного фонда (грант № 02-05-12010).

долгое время и в разных местах, и оперативно получать исчерпывающие ответы на сложные запросы. Однако, для разнородных и обособленных информационных систем, не поддерживающих существующие стандарты, возникает проблема с унификацией доступа к данным и интеграцией информационных ресурсов. Все это во многом затрудняет как работу с ними, так и дальнейшее использование предоставляемой ими информации.

Единственная технология в настоящее время, которая позволяет решить вышеперечисленные проблемы, — это технология, основанная на международных стандартах ANSI/NISO Z39.50 (ISO 23950) [1,5,6] и профиле CIMI [4,10]. Стандарт Z39.50 позволяет унифицировать сетевой доступ к любым базам данных, а профиль CIMI регламентирует доступ к информации о культурном наследии по протоколу Z39.50. Информационные системы по культурному наследию, организованные на основе Z39.50 серверов с использованием CIMI профиля, становятся независимыми от конкретных систем хранения данных и, следовательно, могут быть интегрированы с другими подобными системами.

Технологию, основанную на стандарте Z39.50 и профиле CIMI, предполагается использовать при создании виртуального музея истории информатики в Сибири, работа над которым ведется коллективом сотрудников ИСИ СО РАН, ИМ СО РАН и НГУ при финансовой поддержке Российского гуманитарного научного фонда (грант РГНФ N 02-05-12010) [3].

Статья содержит краткий обзор стандарта Z39.50 и профиля CIMI. Разд. 1 посвящен истории разработки стандарта Z39.50 и профиля CIMI. В разд. 2 рассматриваются основы стандарта Z39.50, модель данных и обеспечиваемые сервисные возможности. В разд. 3 кратко изложена модель поиска информации по Z39.50, рассматриваются понятия абстрактной базы данных и пунктов доступа, приводятся наборы и типы поисковых атрибутов, обсуждаются построение RPN запросов и результаты поиска. В разд. 5 приведена модель извлечения информации по Z39.50, рассматриваются понятия схемы базы данных, наборов тэгов, абстрактной структуры записи, на примере иллюстрируются введенные понятия. Пятый раздел статьи посвящен краткому обзору профиля CIMI, рассматриваются следующие основные аспекты профиля: поиск записей и набор атрибутов CIMI-1; выбор записей и связанные с ним набор тэгов CIMI Tag Set, CIMI схема и абстрактная структура записи с различными уровнями семантической интероперабельности, а также передача записей и синтаксис записи GRS-1. В разд. 6 приведен пример распределенной информационной системы на базе Z39.50.

## 1. ИСТОРИЯ РАЗРАБОТКИ Z39.50 И CИМІ

**Стандарт Z39.50** (Information Retrieval (Z39.50): Application Service Definition and Protocol Specification) определяет прикладную службу и спецификацию протокола для поиска и извлечения информации из баз данных. Он предназначен для унификации сетевого доступа к базам данных и определяет процедуры поиска, извлечения и форматы представления информации [1,5,6].

Первая версия протокола, получившего название Z39.50, была подготовлена Комитетом организации по национальным информационным стандартам США — NISO (National Information Standards Organization) и введена в 1988 г. стандартом Z39.50-1988. Действие этой версии распространилось только на работу с библиографической информацией.

В 1989 г. было организовано Агентство поддержки протокола Z39.50 (Maintenance Agency Z39.50) под административным управлением лаборатории Конгресса США, а в 1990 г. сформирована Группа исполнителей Z39.50 (Z39.50 Implementors Group — ZIG). Ее членами стали производители, продавцы, распространители разнородных видов информации (библиографической, текстовой, графической, финансовой, химической и т.д.). Агентство поддержки Z39.50 является постоянно действующим органом, которое занимается сопровождением и развитием этого стандарта. Сетевой адрес агентства — <http://lcweb.loc.gov/z3950/agency>, здесь находятся вся информация по протоколу, новости, документация, реестры объектов и т.п.

В 1992 г. указанными организациями была разработана версия 2 стандарта (Z39.50-1992), заменившая стандарт 1988 г. Версия 3 стандарта (Z39.50-1995) была разработана в 1995 г. [5]. Поскольку стандарт Z39.50-1995 является развитием версии 1992 г. и совместим с ней, он определяет протокол версий 2 и 3. В 1995 г. протокол Z39.50 был принят как американский национальный стандарт ANSI (ANSI/NISO Z39.50-1995), а в ноябре 1998 г. — как международный стандарт ISO-23950.

Стандарт ANSI/NISO Z39.50-2003 [6], действующий в настоящее время, был утвержден в ноябре 2002 г. Он является технической переработкой стандарта ANSI/NISO Z39.50-1995 и также определяет версии 2 и 3, но дополнительно включает различные пояснения, исправления и соглашения, рекомендованные группой исполнителей Z39.50 — ZIG.

В первые годы своего существования протокол использовался преимущественно для организации доступа к библиографическим ресурсам, на сегодняшний день область его применения существенно расширена. Он используется для доступа к научно-технической, финансовой информации,

к геоинформационным ресурсам, к глобальным базам метаданных, тезаурусам и рубрикаторам, цифровым коллекциям [9] и музейной информации [10].

**Профиль CIMI**, предназначенный для работы с информацией о культурном наследии по протоколу Z39.50, был разработан в 1998 г. Консорциумом по компьютерному обмену музейной информацией — CIMI (Consortium for the Computer Interchange of Museum Information) [10]. Это некоммерческая инициатива, занимающаяся развитием коммуникативных стандартов, сохранением и обменом музейной информацией в электронном виде. Сетевой адрес консорциума — <http://www.cimi.org>. При создании профиля рабочая группа CIMI Z39.50 объединила вместе экспертов по Z39.50, экспертов в области музейного дела и музейной информации, разработчиков программного обеспечения и специалистов в области коммерции.

Профиль CIMI служит спецификацией использования Z39.50 для поиска и извлечения информации о культурном наследии. Он определяет подмножество характеристик, опций и параметров Z39.50, необходимых для поддержки функциональности и требований пользователя при поиске и извлечении информации о культурном наследии. Элементы этого профиля имеют глобальные идентификаторы и являются частью международного стандарта ISO-23950 [5,6].

## 2. ОСНОВЫ СТАНДАРТА Z39.50

**Стандарт Z39.50** определяет службу и протокол типа клиент/сервер для информационного поиска. Он специфицирует процедуры и форматы для проведения клиентом поиска в базах данных, предоставляемых сервером, извлечения записей из баз данных и выполнения других функций, связанных с информационным поиском. Протокол Z39.50 определяет взаимодействие только между клиентскими и серверными информационно-поисковыми приложениями, он не определяет взаимодействие между клиентом и конечным пользователем. Точнее говоря, протокол Z39.50 не определяет форматы хранения данных в конкретных базах данных, способы их индексации и процедуры функционирования различных СУБД. Он также не определяет интерфейсы взаимодействия пользователя и клиента.

Не вдаваясь в детали работы протокола, можно сказать, что стандарт Z39.50 определяет такие правила взаимодействия компьютеров, которые позволяют унифицировать доступ к различным базам данных. Таким образом, пользователь, использующий всего лишь одно клиентское приложение,

может производить поиск информации в удаленных распределенных базах данных, имеющих самую различную структуру и форматы представления информации.

Две основные особенности отличают протокол Z39.50 от других протоколов. Во-первых, это *абстрактная модель представления информации*. В идеологии Z39.50 в рамках одной схемы данных все базы данных совершенно одинаковы, несмотря на их физические различия по используемой СУБД, полям и синтаксису запросов. Иными словами, протокол предоставляет абстрактную модель представления информации на каждом этапе взаимодействия клиента и сервера. В Z39.50 клиент работает всегда с одной и той же системой запросов и получает данные в одних и тех же форматах.

Вторая особенность состоит в том, что протокол Z39.50 полностью обеспечивает *сессионное взаимодействие* клиента и сервера. Эта особенность заложена в самом протоколе и реализуется во всех его приложениях, будь то серверная система или программа-клиент.

Z39.50 обеспечивает следующие основные **сервисные возможности**.

- Клиент может производить поиск, указав список серверов и список баз данных для поиска, используя параметры, определяющие записи, идентифицируемые для поиска.
- Сервер отвечает на запрос клиента подсчетом найденных записей (всех или их части).
- Клиент может получить у сервера отобранные записи. При этом клиент принимает условие, что записи, отбираемые при поиске, формируются и определяются установками сервера.
- Клиент может определить установки для получения данных в том случае, когда он не уверен в результатах поиска: в зависимости от количества найденных записей производить их передачу в полном объеме или в сокращенной форме.
- Клиент может установить предпочтительную для него форму (синтаксис) получения данных, например, в USMARC'e.
- Клиент может присвоить имя полученному набору данных для последующего сравнения результатов, а впоследствии удалить названный им результат поиска.
- Сервер может налагать ограничения контроля доступа на клиента, которые определяются результатами аутентификации, предшествующей обслуживанию запроса.
- Сервер может производить контроль ресурсов для отправки незапрашиваемых или запрашиваемых сообщений. При этом сервер может временно приостанавливать обслуживание клиента и пре-

доставлять ему возможность для определения целесообразности продолжения работы.

Цель этого стандарта — облегчить взаимодействие между клиентами и серверами в прикладных системах, в которых клиент ведет поиск и извлекает записи из баз данных сервера. Базы данных могут иметь различную реализацию: разные системы могут иметь разные способы хранения данных и разные способы доступа к ним. Поэтому при описании баз данных в Z39.50 используется общая *абстрактная модель базы данных*, которой каждая отдельная система может поставить в соответствие свою реализацию. Это позволяет разным системам при взаимодействии использовать стандартные и общепонятные термины для задач поиска и извлечения информации из баз данных. Модели поиска и извлечения информации будут рассмотрены позже в третьем и четвертом разделах соответственно.

Термин *база данных*, используемый в данном стандарте, обозначает собрание записей, где каждая запись представляет собой собрание взаимосвязанных данных. Термин *запись базы данных* относится к локальной структуре данных, представляющей информацию отдельной записи. С базой данных связаны одно или более множеств *пунктов доступа*, которые могут быть указаны при поиске записей в базе данных, и одно или более множеств *элементов*, которые могут быть извлечены из записи базы данных. *Пункт доступа* — это уникальный или неуникальный ключ, который может быть указан отдельно или в сочетании с другими пунктами доступа при поиске записей. Пункт доступа может, но не обязан быть связанным с некоторым элементом. Он может быть эквивалентен элементу, может быть производным от некоторого множества элементов или может не быть связанным ни с какими элементами.

Для наглядности рассмотрим следующий простой пример: пусть у нас есть база данных, в которой содержатся данные о музейных предметах. Информация классифицируется по следующим полям: ID, Title, Type, Creator\_name, Date\_of\_origin, Place\_of\_origin, Material, Technique, Dimensions, Period, Keywords, Description.

Таблица 1

**Список полей базы данных музейных предметов**

Название поля	Комментарий
ID	Идентификационный номер предмета
Title	Название предмета
Type	Типология
Creator_name	Создатель
Date_of_origin	Дата создания
Place_of_origin	Место создания
Material	Материал
Technique	Техника
Dimensions	Размеры, вес
Period	Культурный период
Keywords	Ключевые слова
Description	Краткое описание предмета

С этой базой данных можно связать, например, следующее множество пунктов доступа, которое может быть указано при поиске записей (в скобках приведены соответствующие числовые значения поисковых Use атрибутов из набора атрибутов CIMI-1).

Таблица 2

**Множество пунктов доступа,  
связанное с базой данных музейных предметов**

Название поля	Пункты доступа
ID	ObjectID (2024)
Title	Title (4), ObjectTitle (2033), What (2047)
Type	Material type (1031), ObjectName (2032)
Creator	Author (1003), CreatorName (2035), CreatorGeneral (2041), Who (2046)
DateOfOrigin	DateOfOrigin (2022), When (2048)
PlaceOfOrigin	Name geographic (58), PlaceOfOrigin (2023), Where (2049)
Material	MaterialMedium (2008)
Technique	ProcessTechnique (2012)
Dimensions	Dimensions (2074)
Period	StylePeriod (2017)
Keywords	Any (1016), SubjectContent (2040)
Description	Abstract (62)

Также с этой базой данных можно связать множество элементов, представленных в табл. 3, которые могут быть извлечены из записи, представленной в табл. 3.

Таблица 3

### Множество элементов, связанное с базой данных музейных предметов

Название поля	Элементы
ID	Identifier (2,28), ObjectID (5,3)
Title	Title (2,1), ObjectTitle (5,32)
Type	Type (2,22), ObjectName (5,31)
Creator	Creator (2,2), CreatorInfo (5,36), CreatorGeneral (5,49)
DateOfOrigin	Date (2,8), DateOfOrigin (5,45)
PlaceOfOrigin	PlaceOfOrigin (5,11)
Material	MaterialMedium (5,5)
Technique	ProcessTechnique (5,12)
Dimensions	Dimensions (5,13)
Period	StylePeriod (5,14)
Keywords	Subject (2,21), Subject (5,2)
Description	Description (2,17)

В скобках приведены соответствующие числовые значения элементов из наборов тэгов Generic Tag Set и CIMI Tag Set. Эти наборы тэгов будут приведены позже в таблицах 14 и 19 соответственно. Первая цифра в скобках означает номер набора тэгов (2 для Generic, 5 для CIMI), а вторая — номер тэга в наборе.

### 3. МОДЕЛЬ ПОИСКА ИНФОРМАЦИИ

Для проведения поиска записей требуется указать список серверов баз данных и список имен баз данных для поиска, а также сформулировать поисковый запрос, содержащий критерии поиска. В традиционных системах, не использующих Z39.50, для построения запроса необходимо также знать дополнительную информацию: синтаксис языка запросов для каждой СУБД, а также структуру, имена полей и типы данных каждой базы данных. В случае, когда в группе выбранных серверов и баз данных перечисленные характеристики различны, даже очень простой запрос не сможет быть ис-



полнен для всей группы. В Z39.50 эта проблема решается за счет построения специфической модели поиска и стандартизации ее компонент.

В Z39.50 поисковые запросы всегда формулируются не к реальной базе данных, а к некоей абстрактной. Эта *абстрактная база данных* не имеет никакой структуры, она характеризуется только *пунктами доступа* (поисковыми атрибутами). При получении от клиента запроса в виде термов и поисковых атрибутов, сервер конвертирует его в синтаксис реальной базы данных, и эта процедура остается незаметной для клиента. При таком подходе к процедуре поиска все базы данных становятся для клиента одинаковыми, если они поддерживают один и тот же набор пунктов доступа.

**Наборы поисковых атрибутов** составляют класс объектов Z39.50 {Z39.50 3}, подлежащих стандартизации. На настоящий момент стандартизованы наборы атрибутов, представленные в табл. 4.

Таблица 4

**Наборы атрибутов Z39.50**

OID	Набор	Комментарий
{Z39.50 3 1}	bib-1	Библиографическая информация
{Z39.50 3 2}	exp-1	Explain
{Z39.50 3 3}	ext-1	Расширенный сервис
{Z39.50 3 4}	ccl-1	Type-2 (ISO 8777) и Type-100 (Z39.58) запросы
{Z39.50 3 5}	Gils	Для поиска GILS
{Z39.50 3 6}	Stas	Научно-техническая информация
{Z39.50 3 7}	collections-1	Навигация по электронным коллекциям
{Z39.50 3 8}	cimi-1	Информация по музейным коллекциям
{Z39.50 3 9}	geo-1	Пространственные метаданные
{Z39.50 3 10}	ZBIG	Биологическая информация
{Z39.50 3 11}	Util	Утилиты
{Z39.50 3 12}	xd-1	Междоменный набор
{Z39.50 3 13}	Zthes	Навигация по тезаурусам

Набор атрибутов Bib-1 [7], используемый для поиска библиографической информации, является основным, его подмножества включены во многие другие наборы.

**Типы поисковых атрибутов**

Набор Bib-1 включает в себя шесть типов атрибутов с номерами 1-6: Use, Relation, Position, Structure, Truncation и Completeness. При построении

запроса указание поисковых атрибутов в комбинации с поисковым термом определяет критерии поиска. В каждой группе каждый атрибут определяется уникальным числовым значением, поэтому для указания поискового атрибута необходимо задание двух чисел: тип + значение.

*Атрибуты 1: Use (атрибуты использования)*

Атрибуты этого типа указывают, с какой смысловой информацией связывается поисковый терм, т.е. определяют пункты доступа. В наборе атрибутов Vib-1 определено 99 значений. Среди значений Use есть значения, соответствующие автору, заглавию, ключевым словам, году издания и т.п. Среди атрибутов Use есть такие, которые связываются одновременно с несколькими полями («Author-name-and-title»), а значение «Anywhere» связывается сразу со всеми поисковыми полями.

Таблица 5

**Use атрибуты Vib-1**

Значение	Атрибут	Комментарий
4	Title	Название
21	Subject	Ключевое слово
31	Date-publication	Дата публикации
1003	Author	Автор
1000	Author-name-and-title	Автор и название
1035	Anywhere	Везде

*Атрибуты 2: Relation (атрибуты отношения)*

Атрибуты *Relation* описывают отношения между пунктами доступа и поисковыми терминами, т.е. они указывают, как поисковый терм соотносится с выбираемыми данными из полей, определенных атрибутом Use.

Таблица 6

**Relation атрибуты Bib-1**

Значение	Отношение	Комментарий
1	Less than	Меньше
2	Less than or equal	Меньше или равно
3	Equal	Равно
4	Greater or equal	Больше или равно
5	Greater than	Больше
6	Not equal	Не равно
100	Phonetic	Фонетические отношения
101	Stem	Корень
102	Relevance	Релевантность (соответствие)
103	AlwaysMatches	Любое соответствие

*Атрибуты 3: Position (атрибуты позиции)*

Атрибуты *Position* определяют позицию поискового термина внутри поля или подполя, т.е. указывают, в каком месте поля, определенного атрибутом *Use*, должен находиться поисковый терм.

Таблица 7

**Position атрибуты Bib-1**

Значение	Позиция	Комментарий
1	First in field	Первая позиция в поле
2	First in subfield	Первая позиция в подполе
3	Any position in field	Любая позиция в поле

*Атрибуты 4: Structure (атрибуты структуры)*

Атрибуты *Structure* указывают, какую структуру имеет поисковый терм.

Таблица 8

**Structure атрибуты Bib-1**

Значение	Структура	Комментарий
1	Phrase	Фраза
2	Word	Слово
3	Key	Ключ
4	Year	Год
5	Date (normalized)	Дата (нормализована)
6	Word list	Список слов

100	Date (un-normalized)	Дата (не нормализована)
101	Name (normalized)	Имя (нормализовано)
102	Name (un-normalized)	Имя (не нормализовано)
103	Structure	Структура
104	Urx	Идентификатор докумен- та
105	Free-form-text	Текст в свободной форме
106	Document-text	Текст документа
107	Local-number	Локальный номер
108	String	Строка
109	Numeric string	Числовая строка

### *Атрибуты 5: Truncation (атрибуты усечения)*

Атрибуты *Truncation* указывают, может ли один или несколько символов, находящихся в позиции, определяемой атрибутами *Truncation*, не учитываться при сопоставлении с поисковым термом.

Таблица 9

### **Truncation атрибуты Bib-1**

<b>Значение</b>	<b>Усечение</b>	<b>Комментарий</b>
1	Right truncation	Усечение справа
2	Left truncation	Усечение слева
3	Left and right truncation	Слева и справа
100	Do not truncate	Без усечения
101	Process # in search term	Обработать # в каждом термине
102	RegExpr-1	Регулярные выражения 1
103	RegExpr-2	Регулярные выражения 2

### *Атрибуты 6: Completeness (атрибуты полноты)*

Атрибуты *Completeness* указывают, что является обязательной областью совпадения при поиске, т.е. представляет ли поисковый терм полное, неполное подполе или поле, или, другими словами, могут ли дополнительные слова присутствовать вместе с поисковым термом в поле или подполе.

Таблица 10

**Completeness атрибуты Bib-1**

Значение	Полнота	Комментарий
1	Incomplete subfield	Неполное подполе
2	Complete subfield	Полное подполе
3	Complete field	Полное поле

**Поисковые запросы**

В Z39.50 предусмотрено несколько видов запросов: type-0 — ANY, type-1 — IMPLICIT RPNQuery, type-2 — OCTET STRING (CCL ISO 8777), type-100 — OCTET STRING (CCL Z39.58), type-101 — IMPLICIT RPNQuery, type-102 — OCTET STRING, type-104 — IMPLICIT EXTERNAL (SQL).

Запрос *type-0* представляет собой любой запрос в синтаксисе СУБД, с которой связан сервер, он передается без изменения. Запрос *type-104 (SQL)* — это новый тип запросов, которые входят в стандарт Z39.50 с 2000 г. На данный момент практически нет серверов, которые поддерживали бы SQL в Z39.50. Как видно, SQL-запрос — простая текстовая строка. Наибольший интерес представляют запросы *type-1* и *type-101* — запросы RPN, не различающиеся в 3 версии протокола Z39.50. Запросы *type-1* являются обязательными для всех серверов Z39.50, поддержка других типов запросов сервером Z39.50 является факультативной.

*RPN-запрос* — поисковый запрос, представленный в формате обратной польской нотации (RPN — reverse polish notation). *RPN-запрос* состоит из одного или нескольких условий на пункты доступа, связанных логическими операторами. Каждое условие на пункт доступа состоит из поискового термина и атрибутов. Структуру RPN-запроса можно представить в виде дерева, в узлах которого находятся связывающие логические операторы (AND, OR, AND-NOT), а листья — блоки «атрибуты+терм» (APT).

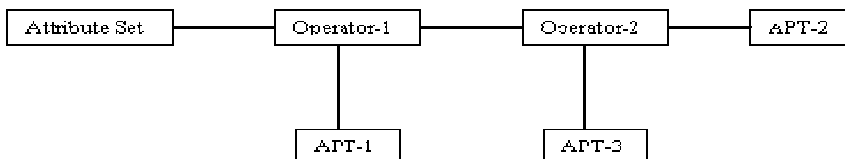


Рис. 1. Структура RPN-запроса

Например, для поиска музейных предметов, создателем которых является Иванов или Иванова, необходимо задать:

(СИМГ-1 1=2035 2=3 3=3 4=2 5=1 6=2 «Иванов»)

Каждый атрибут — это пара, отражающая тип атрибута и его значение. В приведенном примере присутствуют следующие поисковые атрибуты: тип «Use» и значение «CreatorName» (1=2035), тип «Relation» и значение «Equal» (2=3), тип «Position» и значение «Any position in field» (3=3), тип «Structure» и значение «Word» (4=2), тип «Truncation» и значение «Right truncation» (5=1) и тип «Completeness» и значение «Incomplete subfield» (6=2).

### Результаты поиска

В результате выполнения поиска в базах данных клиент может получить от сервера следующую информацию: сообщение об ошибке, количество найденных записей или сами найденные записи. Первый вариант ответа связан с какой-либо ошибкой, второй и третий варианты соответствуют успешному проведению поиска. Какой из них будет получен клиентом, зависит от параметров, которые передаются серверу вместе с поисковым запросом.

Для этого вводятся понятия *малый*, *средний* и *большой наборы*. Здесь под *набором* понимается совокупность найденных записей, пронумерованная сквозным образом. Все записи из малого набора всегда возвращаются, все записи из большого набора никогда не возвращаются, а из среднего набора возвращаются некоторые записи. Далее задаются следующие параметры.

- Верхняя граница малого набора, т.е. максимальный номер записи в малом наборе, который начинается с первой записи.
- Нижняя граница большого набора, т.е. номер, начиная с которого записи попадают в большой набор. Все записи, номера которых больше верхней границы малого набора, но меньше нижней границы большого, считаются записями из среднего набора.
- Количество возвращаемых записей из среднего набора.

Меняя эти три параметра, можно добиться возвращения любого количества записей, в том числе ни одной.

Наконец, следует отметить, что все найденные при поиске записи сервер должен сохранить в сессионном блоке для последующего использования. Если сервер допускает опцию присвоения имени результату поиска, этой

сохраненной совокупности может быть присвоено имя, если нет, совокупность сохраняется неименованной и переписывается при последующем поиске. Именованные результирующие наборы, которые сохраняются на сервере, можно использовать в последующих RPN-запросах, где они выступают такими же операндами, как блоки APT.

#### 4. МОДЕЛЬ ИЗВЛЕЧЕНИЯ ИНФОРМАЦИИ

После того как произведен поиск и создан результирующий набор на сервере, записи из результирующего набора можно извлекать и просматривать. Для того чтобы запросить записи, необходимо задать: имя результирующего набора; номер записи в результирующем наборе, с которой начинается извлечение; количество извлеченных записей; формат представления данных и максимальную длину записи, доступную для извлечения.

Рассмотрим модель извлечения и представления данных, принятую в Z39.50. Идеология максимального абстрагирования от структур реальных баз данных приводит к весьма изощренной схеме извлечения данных, описанной в стандарте Z39.50. Каждой базе данных приписывается одна или несколько схем.

##### Схема данных и абстрактная структура записи

*Схема базы данных* представляет собой взаимное соглашение между клиентом и сервером об информации, содержащейся в записях базы данных, которое позволяет в дальнейшем отбирать часть этой информации в соответствии со спецификацией элемента.

Схема определяет *абстрактную структуру записи*. Это первичный элемент схемы базы данных, представляющий собой дерево элементов, специфицируемых тэгами (*tag*) из стандартных наборов тэгов (*tagSet*). При применении абстрактной структуры записи к записи базы данных получается абстрактная запись базы данных.

*Абстрактная запись базы данных* — абстрактное представление информации, содержащейся в записи базы данных. Для формирования абстрактной записи нужно применить определенную в схеме абстрактную структуру записи к записи базы данных.

*Спецификация элемента* преобразует абстрактную запись базы данных в другой экземпляр абстрактной записи (преобразование может быть нулевым). По спецификации элемента из абстрактной записи отбираются эле-

менты, а также, возможно, указываются формы варианта для этих элементов.

Сервер применяет к этой абстрактной записи *синтаксис записи* для представления извлеченных записей, в результате чего получается структура, подлежащая пересылке (извлеченная запись).

Теперь на конкретном примере рассмотрим вышеопределенные понятия схемы базы данных и абстрактной структуры записи. Предположим, существуют два музея (музейных информационных систем), в которых независимо ведется учет музейных предметов. Учет ведется в различных СУБД, информация классифицируется по следующим полям.

Таблица 11

### Список полей баз данных двух музейных систем

Museum1	Museum2
Identifier	ObjectID
Title	ObjectTitle
Type	ObjectType
Creator	CreatorName
	CreatorDateOfBirth
	CreatorDateOfDeath
	CreatorRole
Date	DateOfOrigin
Place	PlaceOfOrigin
Material	
Technique	
Dimensions	
Period	StylePeriod
Keywords	Subject
Description	

Очевидно, что происходит накопление однотипной информации, но структуры данных в системах отличаются. Построим некую обобщающую модель данных, в которой зафиксируется суть накапливаемой информации. Эта модель должна включать позиции, представленные в табл. 12.



Таблица 12

**Обобщающая модель данных**

Идентификационный номер предмета
Название предмета
Типология
Информация о создателе (имя, дата рождения, дата смерти, роль)
Дата создания
Место создания
Материал
Техника
Размеры
Культурный период
Ключевые слова
Краткое описание предмета

В иерархической схеме это будет выглядеть следующим образом:

- Идентификатор
- Название
- Типология
- Создатель
  - Имя
  - Дата рождения
  - Дата смерти
  - Роль
- Дата создания
- Место создания
- Материал
- Техника
- Размеры
- Культурный период
- Ключевые слова
- Краткое описание

К этой схеме нужно добавить дополнительные характеристики, такие как обязательность, повторяемость полей и поисковые атрибуты. Пример приведен в табл. 13.

Таблица 13

## Схема данных

Название поля	Обязательность	Повторяемость	Поисковые атрибуты
Идентификатор	обязательно	не повторяется	2024
Название	обязательно	не повторяется	4, 2033, 2047
Типология	обязательно	не повторяется	1031, 2032
Создатель	обязательно	не повторяется	1003, 2041
Имя	обязательно	не повторяется	2035, 2046
Дата рождения	не обязательно	не повторяется	2036
Дата смерти	не обязательно	не повторяется	2037
Роль	не обязательно	не повторяется	2014
Дата создания	обязательно	не повторяется	58, 2022, 2048
Место создания	обязательно	не повторяется	2023, 2049
Материал	не обязательно	повторяется	2008
Техника	не обязательно	повторяется	2012
Размеры	не обязательно	не повторяется	2074
Культурный период	обязательно	повторяется	2017
Ключевые слова	обязательно	повторяется	2040
Краткое описание	не обязательно	не повторяется	62

В последнем столбце проставлены соответствующие значения поисковых Use атрибутов по СИМІ-1. Таким образом, мы определили модель данных, которая называется в Z39.50 *схемой данных*.

Следующий шаг — определение *абстрактной структуры записи*. Для этого воспользуемся наборами тэгов Generic Tag Set (2), набор элементов которого приведен ниже в таблице 14, и СИМІ Tag Set (5), набор элементов которого будет приведен позже при рассмотрении профиля СИМІ в таблице 19.

Таблица 14

**Набор тэгов Generic Tag Set (TagSet-G)**

ТЭГ	элемент	ТЭГ	элемент	ТЭГ	элемент	ТЭГ	элемент
1	title	11	postalAddress	21	subject	31	publisher
2	author	12	networkAddress	22	resourceType	32	contributor
3	publication-Place	13	eMailAddress	23	city	33	source
4	publicationDate	14	phoneNumber	24	stateOrProvince	34	coverage
5	documentId	15	faxNumber	25	zipOrPostalCode	35	private
6	abstract	16	country	26	cost	36	data-baseName
7	name	17	description	27	format	37	recordId
8	dateTime	18	time	28	identifier		
9	displayObject	19	DocumentContent	29	rights		
10	organization	20	language	30	relation		

В таблице 15 приведены элементы и тэги, используемые в абстрактной структуре записи.

Таблица 15

**Используемые элементы и тэги**

(2,7) - Name	(5,5) - MaterialMedium	(5,11) - PlaceOfOrigin	(5,31) - ObjectName
(2,17) - Description	(5,8) - DateOfBirth	(5,12) - ProcessTechnique	(5,32) - ObjectTitle
(5,2) - Subject	(5,9) - DateOfDeath	(5,13) - Dimensions	(5,36) - CreatorInfo
(5,3) - ObjectID	(5,10) - Role	(5,14) - StylePeriod	(5,45) - DateOfOrigin

Определим меточную абстрактную структуру записи.

Таблица 16

**Абстрактная структура записи**

Тэговый путь	Элемент
(5,3)	ObjectID
(5,32)	ObjectTitle
(5,31)	ObjectType
(5,36)	Creator

(5,36) / (2,7)	CreatorName
(5,36) / (5,8)	CreatorDateOfBirth
(5,36) / (5,9)	CreatorDateOfDeath
(5,36) / (5,10)	CreatorRole
(5,45)	DateOfOrigin
(5,11)	PlaceOfOrigin
(5,14)	Period
(5,5)	Material
(5,12)	Technique
(5,13)	Dimensions
(5,2)	Keywords
(2,17)	Description

Первая цифра в скобках означает номер набора тэгов, а вторая — номер тэга в наборе. Каждая строчка в определении абстрактной структуры записи имеет тэговый путь (TagPath), именована и называется элементом. Элементы могут быть простыми (ObjectTitle) и сложными (Creator).

Теперь вернемся к двум музейным системам, которым требуется на уровне извлеченных записей объединить свои базы данных. Для этого им необходимо просто выдавать записи в соответствии с определенной схемой. В каждом музее потребуется установление соответствия элементов абстрактной структуры полям реальных баз данных и соответственное их наполнение.

### Формат GRS-1

Рассмотренные схема данных и абстрактная структура записи определяют лишь логическую структуру записи. Физическая реализация этой структуры остается неопределенной. Стандарт Z39.50 требует, чтобы записи из внутреннего представления отображались во внешние структуры, которые и пересылаются по сети. Для внешних представлений, которые стандартизируются в классе recordSyntax {Z39.50 5}, физическая реализация становится определяющей.

Существует всего один универсальный формат внешнего представления (recordSyntax), который однозначно отображает абстрактную структуру записи в экспортируемую внешнюю физическую структуру. Он называется GRS-1 (Generic Record Syntax) OID {Z39.50 5 105}.

Полное ASN.1 определение записи в формате GRS-1 можно найти в приложении к документации по Z39.50 [5,6]. Если в этом определении оста-

вить основное, исключив метаинформацию (metaData) и варианты (applied-Variant), то останется:

```

GenericRecord ::= SEQUENCE OF TaggedElement
TaggedElement ::= SEQUENCE {
    tagType          [1] IMPLICIT INTEGER OPTIONAL,
    tagValue         [2] StringOrNumeric,
    tagOccurrence    [3] IMPLICIT INTEGER OPTIONAL,
    content          [4] ElementData,
    metaData         [5] IMPLICIT ElementMetaData OP-
TIONAL,
    appliedVariant   [6] IMPLICIT Variant OPTIONAL}

ElementData ::= CHOICE{
    octets           OCTET STRING,
    numeric          INTEGER,
    date             GeneralizedTime,
    ext              EXTERNAL,
    string           InternationalString,
    trueOrFalse     BOOLEAN,
    oid              OBJECT IDENTIFIER,
    intUnit          [1] IMPLICIT IntUnit,
    elementNotThere [2] IMPLICIT NULL, -- element
requested but not there
    elementEmpty    [3] IMPLICIT NULL, -- element
there, but empty
    noDataRequested [4] IMPLICIT NULL, -- variant
request said 'no data'
    diagnostic      [5] IMPLICIT EXTERNAL,
    subtree         [6] SEQUENCE OF TaggedElement
}

```

Итак, упрощенно можно полагать, что запись в формате GRS-1 — это последовательность элементов (taggedElement), каждый из которых содержит тип тэга (tagType, т. е. номер tagSet), значение тэга (tagValue), вхождение тэга (tagOccurrence) и данные (content). Значение тэга может быть числовым или текстовым. Данные — это или значение, или вложенный элемент (taggedElement) с описанной структурой (определение subtree ничем не отличается от определения GenericRecord). Таким образом, GRS-1 представляет собой последовательность элементов со структурой, определенной в терминах, аналогичных определению абстрактной структуры записи и схемы данных, с бесконечным количеством вложений.

## 5. ПРОФИЛЬ СИМІ

Теперь рассмотрим более подробно, как происходит доступ к информации о культурном наследии по протоколу Z39.50 в соответствии с профилем СИМІ (OID 1.2.840.10003.3.8.) [4,10].

*Профиль СИМІ* определяет подмножество характеристик Z39.50, опций и параметров, необходимых для поддержки функциональности и требований пользователя при поиске и извлечении информации о культурном наследии. Профиль СИМІ оперирует со следующими объектами Z39.50.

Таблица 17

### Объекты Z39.50 профиля СИМІ

Наборы поисковых атрибутов:	Bib-1	1.2.840.10003.3.1
	СИМІ-1	1.2.840.10003.3.8
Диагностика:	Bib-1	1.2.840.10003.4.1
Форматы:	GRS-1	1.2.840.10003.5.105
	SUTRS	1.2.840.10003.5.101
	Usmarc	1.2.840.10003.5.10
Схемы:	Digital Collections Schema	1.2.840.10003.13.3
	СИМІ Schema	1.2.840.10003.13.5
Наборы тэгов:	TagSet-M (Metadata TagSet)	1.2.840.10003.14.1
	TagSet-G (Generic TagSet)	1.2.840.10003.14.2
	Collections TagSet	1.2.840.10003.14.5
	СИМІ TagSet	1.2.840.10003.14.6

На основе этих объектов определены модели поиска и извлечения данных с музейной информацией. Полное описание профиля СИМІ достаточно объемно и здесь приводиться не будет [10]. Ниже будут отмечены только некоторые аспекты, характерные для этого профиля.

### Поиск

Для проведения пользователем поиска во многих базах данных на нескольких серверах необходимо стандартизировать поисковое выражение таким образом, чтобы клиент и сервер могли обмениваться информацией недвусмысленным образом. Эта задача решается путем определения набора атрибутов, который определяет множество пунктов доступа и дополнитель-

ную информацию, используемую для характеристики поисковых термов, а также путем представления поисковой строки стандартным образом.

### *Набор атрибутов CIMI-1*

Профиль CIMI определяет стандартный набор атрибутов CIMI-1 для поиска информации о культурном наследии, поддержка которого является обязательной для CIMI клиентов и серверов. Набор CIMI-1 использует шесть типов атрибутов, определенных в наборе Bib-1 (типы атрибутов Use, Relation, Position, Structure, Truncation и Completeness) [7]. CIMI-1 также определяет два новых типа атрибутов: тип 101 — Authority (авторитетный источник) и тип 102 — Charset (набор символов).

Тип атрибутов *Authority* содержит значения, идентифицирующие авторитетный источник, из которого взят термин. В качестве примера значений Authority атрибутов можно перечислить следующие: Non-authoritative, Local-to-server, USMARC, LCSH (Library of Congress Subject Headings), AAT (Art & Architecture Thesaurus), Dictionarium Museologicum, ISO Documentation и так далее.

Тип атрибутов *Charset* включает значения, идентифицирующие набор символов, используемый для кодировки термина. В качестве примера значений Charset атрибутов можно привести следующие: 7-bit US-ASCII, ISO 8859-1 (Latin-1), ISO 8859-5 (Cyrillic), ISO 8859-6 (Arabic) и так далее.

Что касается Use атрибутов, определенных в CIMI-1, то они частично получены из пунктов доступа CIMI (CIMI Access Points) [8], а также из других руководств и стандартов по информации о культурном наследии. Использование Use атрибутов CIMI-1 позволяет клиенту расширить поиск музейной информации по специфическому содержанию (например, по названию и типологии объекта, сведениям о создателе, дате и месте происхождения объекта, культурному периоду, материалу и технике, размерам, данным о хранении и т.д.) стандартным путем, который может быть понят сервером. CIMI-1 также содержит небольшое подмножество (14 значений с номерами в промежутке от 4 до 1032) Use атрибутов набора Bib-1 (title, ISBN, ISSN, date of publication, abstract, author и т.д.). Эти атрибуты включены для обеспечения CIMI клиентам базисного поиска (автор–название–ключевые слова) в библиографических базах данных, а также для поддержки поиска на CIMI серверах библиографическими клиентами Z39.50, не поддерживающими профиль CIMI.

Набор атрибутов CIMI-1 был разработан консорциумом CIMI при изучении существующих стандартов и систем после проведения анализа запросов пользователей к музейным коллекциям. Он отражает в себе соглашения

с широким музейным сообществом о наборе пунктов доступа, которые должна поддерживать система. Таким образом, набор CIMI-1 предоставляет механизм совместного понимания при поиске для клиента и сервера. Например, когда пользователь отправляет запрос на извлечение информации о «происхождении произведения», база данных сервера может иметь или не иметь «происхождение произведения» как отдельный пункт доступа. В этом случае необходимо, чтобы такой запрос был корректно отображен сервером на соответствующие поля или индексы локальной базы данных. Сервер, поддерживающий этот профиль, может понять, когда он получает запрос на «происхождение произведения», поскольку запрос представлен и выражен в стандартной форме набора атрибутов CIMI-1.

Для переадресации запросов профиль использует набор элементов метаданных Дублинского ядра (Dublin Core Metadata) [11]. Use атрибуты, ассоциированные с элементами Дублинского ядра (DC-title, DC-creator, DC-subject, DC-description и т.д.), могут выразить запрос на поиск в терминах пунктов доступа, представленных или охарактеризованных элементами Дублинского ядра.

Локальные реализации могут определять Use атрибуты для локального использования и присваивать им значения, находящиеся в промежутке 5000-7999.

Полная спецификация набора атрибутов CIMI-1 приведена в приложении к документации по CIMI профилю (APPENDIX A: CIMI-1 Attribute Set). Ниже приведен список Use атрибутов CIMI-1, информация о семантике которых доступна в приложении к документации (APPENDIX C: Semantics for Use Attributes and Schema Elements) [10].



Таблица 18

Use атрибуты набора CIMI-1

Значение	Название атрибута	Значение	Название атрибута
4	title	2041	creatorGeneral
7	ISBN	2042	associationGeneral
8	ISSN	2043	objectLanguage
31	date of publication	2044	condition
32	date of acquisition	2045	physicalDescription
54	code language	2046	who
58	name geographic	2047	what
62	abstract	2048	when
1003	author	2049	where
1004	personal author	2051	DC-title
1016	any	2052	DC-creator
1018	publisher	2053	DC-subject
1031	material type	2054	DC-description
1032	doc-id	2055	DC-publisher
2000	award	2056	DC-contributors
2002	collection	2057	DC-date
2004	copyrightRestriction	2058	DC-type
2005	creditLine	2059	DC-format
2007	inscriptionMark	2060	DC-identifier
2008	materialMedium	2061	DC-source
2009	creatorNationalityCultureRace	2062	DC-language
2012	processTechnique	2063	DC-relation
2014	creatorRole	2064	DC-coverage
2017	stylePeriod	2065	DC-rights
2020	image	2070	fieldCollector
2022	dateOfOrigin	2071	dateCollected
2023	placeOfOrigin	2072	agePeriod
2024	objectID	2073	typeSpecimen
2026	owner	2074	dimensions
2027	repositoryName	2075	quantity
2028	repositoryPlace	2076	relatedObjects
2029	provenance	2077	resource
2030	contentGeneral	2078	wallTextLabel
2032	objectName	2079	administrativeEventGeneral
2033	objectTitle	2080	administrator
2034	relatedTextualReferences	3000	protectionStatus
2035	creatorName	3001	protectionDate
2036	creatorDateOfBirth	3003	spatialReferencingSystem
2037	creatorDateOfDeath	3004	x-coordinateInReferencingSystem
2038	contextHistorical	3005	y-coordinateInReferencingSystem
2039	contextArchaeological	3007	Address
2040	subjectContent	3009	PeriodName

CIMI сервера должны предусматривать ситуацию, когда поисковый запрос содержит только один поисковый терм, а атрибуты не присутствуют в запросе. В этом случае в CIMI профиле рекомендуется использовать следующие значения по умолчанию для атрибутов каждого типа. Если запрос содержит только поисковый терм без указания Use атрибута, для Use атрибута принимается значение по умолчанию, равное «Any» (1016). Что касается атрибутов типов 2-6, следующие значения рекомендуются по умолчанию: для типа Relation — «equal» (3), типа Position — «any position in field» (3), типа Structure — «word» (2), типа Truncation — «do not truncate» (100) и типа Completeness — «complete field» (3). Тип Authority не имеет значения по умолчанию, сервер интерпретирует его отсутствие в запросе клиента просто как отсутствие данных по этому поводу и обрабатывает запрос по своему усмотрению. Поддержка в запросе логических операторов AND и OR для CIMI клиентов и серверов является обязательной.

### Выбор

Для проведения поиска информации во многих базах данных на нескольких серверах необходимо обеспечить выполнение следующих двух требований. Во-первых, клиенты и серверы должны быть способны обмениваться записями из баз данных (или элементами записей) в форматах, которые они могут распознавать и обрабатывать. Во-вторых, клиенты и серверы должны иметь одинаковое понимание элементов в этих базах данных и возможность идентифицировать эти элементы недвусмысленно для выбора информации, которую необходимо получить. Спецификации выбора позволяют клиенту и серверу обмениваться информацией о записях в базе данных для извлечения полной записи или определенных единиц информации (т.е. одной или более групп полей базы данных).

Профиль CIMI реализует эту возможность путем определения набора тэгов CIMI (CIMI Tag Set), схемы CIMI (CIMI Schema) и абстрактной структуры записи (Abstract Record Structure — ARS) для извлечения записей. Аналогично, как и для набора атрибутов CIMI-1, обсужденных выше, схема CIMI и соответствующая абстрактная структура записи служат «языком взаимопонимания» для обмена между клиентом и сервером при извлечении информации.

#### *Схема CIMI и набор тэгов CIMI Tag Set*

*Схема CIMI* абстрактно определяет единицы информации, которые могут присутствовать в записях локальных баз данных объектов, изображений

с присоединенным текстом и каталожных записях. Схема не указывает, как поле названо в реальной базе данных, а предлагает стандартный путь для именованя этих элементов или полей. Всякая локальная база данных применяет свои термины для обозначений полей базы данных и их структуры. Схема обеспечивает абстрактное представление этих баз данных: в этом абстрактном виде поля базы данных пронумерованы как элементы схемы, каждый элемент которой имеет уникальное имя, уникальную цифровую метку (тэг) и определение. Схема также показывает структурную организацию этих элементов в абстрактную структуру записи.

Например, схема CIMI определяет элемент *dateOfOrigin*. Локальная база данных может иметь одно или более полей, относящихся к «дате создания объекта». Поскольку семантика предлагается для каждого элемента схемы CIMI, администратор базы данных знает, что в случае, если клиентом запрошен элемент *dateOfOrigin*, то должна быть возвращена информация, относящаяся к «дате создания объекта».

CIMI схема использует элементы, определенные в наборе тэгов CIMI Tag Set, так же как и элементы из других зарегистрированных наборов тэгов.

Ниже приведен набор тэгов CIMI Tag Set (OID=1.2.840.10003.14.6), содержащий значения тэгов и названия элементов. Семантика этих элементов приведена в приложении к документации по профилю CIMI (Appendix C -- Semantics for Use Attributes and Schema Elements).

Таблица 19

**Набор тэгов CIMI Tag Set**

Тэг	Элемент	Тэг	Элемент	Тэг	Элемент
1	RepositoryName	25	content	50	associationGeneral
2	Subject	26	repositoryPlace	51	objectLanguage
3	ObjectID	28	mrObject	52	condition
4	nationalityCultu- reRace	29	rendition	53	physicalDescription
5	MaterialMedium	30	resource	54	wallTextLabel
7	CreditLine	31	objectName	55	protectionStatus
8	DateOfBirth	32	objectTitle	56	protectionDate
9	DateOfDeath	33	bibliographicTitle	57	spatialReferencing- System
10	Role	35	relatedTextual- References	58	xCoordinateInSpatial- ReferencingSystem
11	PlaceOfOrigin	36	creatorInfo	59	yCoordinateInSpatial- ReferencingSystem
12	ProcessTechnique	38	owner	60	fieldCollector
13	Dimensions	39	contentGeneral	61	dateCollected

14	StylePeriod	41	place	62	agePeriod
15	Provenance	42	event	63	typeSpecimen
16	RelatedObjects	43	activity	64	address
17	Quantity	45	dateOfOrigin	65	periodName
18	Award	46	contextHistorical	66	administrativeEvent
20	Collection	47	contextArchaeological	67	administrativeEvent- Type
22	InscriptionMark	48	copyrightRestriction	68	administrativeEvent- General
24	Association	49	creatorGeneral	69	administrator

CIMI схема определяет также пять структурных типов данных (элементов): *CreatorInfo*, *MrObject*, *Rendition*, *MoreInfo*, *AdministrativeEvent*. Схема позволяет извлекать одно или более «изображений», ассоциированных с записью об объекте (где под «изображением» понимается любой тип электронного ресурса, включая видео, аудио, графику). Каждому экземпляру «изображения» соответствует элемент схемы *mrObject*. Поскольку один и тот же электронный ресурс может быть представлен в нескольких вариантах (например, изображения различных размеров), схема содержит понятие о «представлении» (*Rendition*), которое является специфической версией «изображения». Каждому имеющемуся варианту «изображения» соответствует элемент *Rendition*, подэлемент элемента *mrObject*. Элемент *Rendition* включает подэлемент *Resource*, который содержит или URL ресурса, или непосредственно сам электронный ресурс. Таким образом, сервер может вернуть клиенту одно или более «изображений», так же как и одно или более видов (представлений) каждого «изображения». Более того, для каждого изображения и наглядного представления может быть извлечена специфическая описательная информация.

#### *Абстрактная структура записи CIMI*

CIMI схема описывает *абстрактную структуру записи*, определяющую размещение и порядок элементов в извлеченной записи. Поскольку CIMI профиль является дополнительным к профилю для доступа к цифровым коллекциям (Digital Collections Profile) [9], схема данных CIMI и абстрактная структура записи основаны на использовании схемы Digital Collections. Абстрактная структура записи CIMI поддерживает структуру записи Digital Collections и вложена в элемент *actualDO* записи Digital Collections (*object descriptive record*).

Термин абстрактная структура записи обычно используется в контексте специфической схемы. В CIMI профиле абстрактная структура записи вы-

ходит за пределы CIMI схемы. Абстрактная структура записи специфицирует использование элементов из наборов тэгов tagSet-M, tagSet-G, Collections Tag Set и CIMI Tag Set. Следующие типы тэгов (1-5) используются в ней для извлеченной записи.

Таблица 20

### Типы тэгов в абстрактной структуре записи CIMI

Тип тэгов	Определение
1	Элементы из tagSet-M. Сервер может включать их по своему выбору, а клиент может их игнорировать (кроме элемента <i>schemaIdentifier</i> ).
2	Элементы из tagSet-G. Сервер может включать элементы, не перечисленные в абстрактной структуре записи, а клиент может их игнорировать.
3	Зарезервирован для тэгов, локально определенных сервером. Серверы, посылающие строковые тэги для локально определенных элементов, должны использовать 3 тип тэгов для их идентификации. Строковые тэги следует использовать только тогда, когда имеющиеся в наличии элементы из tagSet-M, tagSet-G, Collections TagSet и CIMI TagSet не адекватны.
4	Элементы из Collections Tag Set.
5	Элементы из CIMI Tag Set.

Абстрактная структура извлеченной записи, определенная в CIMI профиле, поддерживает три уровня семантической интероперабельности путем разделения записи на три уровня элементов: общий уровень (Generic Level), уровень схемы цифровых коллекций (Digital Collections Schema Level) и уровень CIMI схемы (CIMI Schema Level). Это разделение позволяет обеспечить семантическую интероперабельность между CIMI серверами и Z39.50 клиентами, не поддерживающими CIMI профиль.

Чтобы обеспечить эти три уровня семантической интероперабельности, необходимо добавить элемент *schemaIdentifier* (из tagSet-M) в извлеченную запись. На общем уровне семантической интероперабельности абстрактная структура записи предполагает использование тэгов типов 1, 2 и 3. На уровне Digital Collections — использование тэгов типов 1, 2, 3 и 4, а на уровне CIMI — тэгов типов 1, 2, 3, 4 и 5.

В нижеприведенных таблицах используются следующие обозначения для вхождения элементов: [0,1] (не обязательно, не повторяется), 0+ (не обязательно, повторяется), 1 (обязательно, не повторяется), 1+ (обязательно, повторяется).

**1. Общий уровень (Generic Level)** семантической интероперабельности: в начале извлеченной записи могут присутствовать элементы из tagSet–M и tagSet–G. Серверы и клиенты Z39.50 могут распознавать и обрабатывать эти элементы, даже если они не поддерживают ни одной специализированной схемы. Таким образом, включение общих элементов на верхнем уровне извлеченной записи позволяет любым клиентам Z39.50 производить поиск в базах данных, содержащих описательные записи коллекций и объектов, и частично их обрабатывать.

Таблица 2 1

### Общий уровень абстрактной структуры записи

Тэговый путь	Элемент	Вхождение
(1, 14)	localControlNumber	1
(2, 1)	title	0+
(2, 2)	creator	0+
(2, 32)	contributor	0+
(2, 8)	date	0+
(2, 170)	description	0+
(2, 28)	identifier	0+
(2, 22)	type	0+
(2, 20)	language	0+
(2, 21)	subject	0+
(2, 31)	publisher	0+
(2, 27)	format	0+
(2, 33)	source	0+
(2, 30)	relation	0+
(2, 34)	coverage	0+
(2, 29)	rights	0+

**2. Уровень схемы цифровых коллекций (Digital Collections Schema Level)** семантической интероперабельности: принимается схема Digital Collections (указывается с помощью schemaIdentifier), и поддерживающие ее клиенты Z39.50 могут распознавать и обрабатывать эти элементы. Таким образом, клиенты, поддерживающие схему Digital Collections, могут выполнять поиск в базах данных, предоставляемых CIMI серверами. Они могут извлекать эти записи, но не могут обрабатывать их полностью.

Таблица 22

Тэговый путь	Элемент	Вхождение
(1, 1)	schemaIdentifier	1
(4, 1)	typeOfDescriptiveRecord	1
(4, 4)	objectInfo	1
(4, 4) (4, 12)	typeOfObject	1
(4, 4) (4, 13)	categoryOfObject	[0, 1]
(4, 4) (4, 14)	digitalObject	1
(4, 4) (4, 14) (4, 29)	actualDO	1

Табл. 22. Уровень Digital Collections абстрактной структуры записи

**3. Уровень схемы CIMI (CIMI Schema Level)** семантической interoperability: принимается схема CIMI (указывается с помощью *schemaIdentifier*), и Z39.50 клиенты, ее поддерживающие, могут распознавать и обрабатывать остальные элементы записи (т.е. полностью обрабатывать целую извлеченную запись).

Таблица 23

**Уровень CIMI абстрактной структуры записи**

Тэговый путь	Элемент	Вхождение
(4, 4) (4, 14) (4, 29) (1, 1)	SchemaIdentifier	1
(4, 4) (4, 14) (4, 29) (5, 31)	ObjectName	0+
(4, 4) (4, 14) (4, 29) (5, 32)	ObjectTitle	1+
(4, 4) (4, 14) (4, 29) (5, 33)	bibliographicTitle	1+
(4, 4) (4, 14) (4, 29) (5, 49)	creatorGeneral	0+
(4, 4) (4, 14) (4, 29) (5, 36)	creatorInfo	0+
(4, 4) (4, 14) (4, 29) (5, 60)	fieldCollector	0+
(4, 4) (4, 14) (4, 29) (5, 1)	repositoryName	0+
(4, 4) (4, 14) (4, 29) (5, 26)	repositoryPlace	0+
(4, 4) (4, 14) (4, 29) (5, 38)	owner	0+
(4, 4) (4, 14) (4, 29) (5, 7)	creditLine	0+
(4, 4) (4, 14) (4, 29) (5, 2)	subject	0+
(4, 4) (4, 14) (4, 29) (5, 3)	objectID	0+
(4, 4) (4, 14) (4, 29) (5, 5)	materialMedium	0+
(4, 4) (4, 14) (4, 29) (5, 12)	processTechnique	0+
(4, 4) (4, 14) (4, 29) (5, 13)	dimensions	0+
(4, 4) (4, 14) (4, 29) (5, 11)	placeOfOrigin	0+
(4, 4) (4, 14) (4, 29) (5, 45)	dateOfOrigin	0+

(4, 4) (4, 14) (4, 29) (5, 61)	dateCollected	0+
(4, 4) (4, 14) (4, 29) (5, 62)	agePeriod	0+
(4, 4) (4, 14) (4, 29) (5, 63)	typeSpecimen	0+
(4, 4) (4, 14) (4, 29) (5, 14)	stylePeriod	0+
(4, 4) (4, 14) (4, 29) (5, 65)	periodName	0+
(4, 4) (4, 14) (4, 29) (5, 15)	provenance	0+
(4, 4) (4, 14) (4, 29) (5, 17)	quantity	0+
(4, 4) (4, 14) (4, 29) (5, 18)	award	0+
(4, 4) (4, 14) (4, 29) (5, 20)	collection	0+
(4, 4) (4, 14) (4, 29) (5, 22)	inscriptionMark	0+
(4, 4) (4, 14) (4, 29) (5, 51)	objectLanguage	0+
(4, 4) (4, 14) (4, 29) (5, 52)	condition	0+
(4, 4) (4, 14) (4, 29) (5, 53)	physicalDescription	0+
(4, 4) (4, 14) (4, 29) (5, 55)	protectionStatus	0+
(4, 4) (4, 14) (4, 29) (5, 56)	protectionDate	0+
(4, 4) (4, 14) (4, 29) (5, 57)	spatialReferencingSystem	0+
(4, 4) (4, 14) (4, 29) (5, 58)	xCoordinateInSpatialReferencingSystem	0+
(4, 4) (4, 14) (4, 29) (5, 59)	yCoordinateInSpatialReferencingSystem	0+
(4, 4) (4, 14) (4, 29) (5, 64)	address	0+
(4, 4) (4, 14) (4, 29) (5, 16)	relatedObjects	0+
(4, 4) (4, 14) (4, 29) (5, 35)	relatedTextualReferences	0+
(4, 4) (4, 14) (4, 29) (5, 50)	associationGeneral	0+
(4, 4) (4, 14) (4, 29) (5, 24)	association	0+
(4, 4) (4, 14) (4, 29) (5, 39)	contentGeneral	0+
(4, 4) (4, 14) (4, 29) (5, 25)	content	0+
(4, 4) (4, 14) (4, 29) (5, 46)	contextHistorical	0+
(4, 4) (4, 14) (4, 29) (5, 47)	contextArchaeological	0+
(4, 4) (4, 14) (4, 29) (5, 48)	copyrightRestriction	0+
(4, 4) (4, 14) (4, 29) (5, 54)	wallTextLabel	0+
(4, 4) (4, 14) (4, 29) (2, 9)	displayObject	0+
(4, 4) (4, 14) (4, 29) (5, 68)	administrativeEventGeneral	0+
(4, 4) (4, 14) (4, 29) (5, 66)	administrativeEvent	0+
(4, 4) (4, 14) (4, 29) (5, 69)	administrator	0+
(4, 4) (4, 14) (4, 29) (5, 28)	mrObject	0+

### Наборы элементов b, mb, f

Сервер создает *результатирующий набор* в ответ на запрос. Результатирующий набор — это множество указателей на записи в одной или более базах



данных. *Именованные наборы элементов* идентифицируют группы элементов в записи базы данных, возвращаемой клиенту сервером. CIMI профиль определяет следующие три именованные набора элементов для абстрактной структуры извлеченной записи: *b*, *mb*, *f*.

**Набор элементов b (brief record)** предназначен для извлечения краткой формы записи базы данных, составленной из элементов tagSet-G верхнего уровня абстрактной структуры извлеченной записи, который соответствует основному (общему) уровню поиска и извлечения при использовании элементов Dublin Core Metadata. Что касается элементов, составляющих набор элементов **b**, они в точности совпадают с набором элементов общего уровня абстрактной структуры записи и уже были приведены выше в табл. 21. Сервер может также включать в запись другие элементы из tagSet-G и tagSet-M.

**Набор элементов mb (museum brief record)** предназначен для извлечения краткой формы записи базы данных согласно принятым в музейных системах стандартам. Цель этой краткой формы — предоставить CIMI клиентам достаточное количество элементов для построения краткой записи (*tombstone record*). Набор элементов **mb** включает элементы уровней Digital Collections и CIMI абстрактной структуры записи, которые приведены в таблицах 22 и 23. Сервер может также включать в запись элементы из tagSet-G и tagSet-M.

**Набор элементов f (full record)** включает все имеющиеся элементы записи базы данных. Элементы записи базы данных, которые могут быть помечены имеющимися тэгами из tagSet-G, tagSet-M, tagSet-Collections и tagSet-CIMI, должны быть помечены соответствующим образом. Дополнительные элементы, возвращаемые сервером, должны быть или включены в элемент *displayObject*, или должны использоваться строковые тэги из третьего типа тэгов для локально определенных элементов. Строковые тэги и содержащиеся в *displayObject* данные клиент не обрабатывает, а предоставляет пользователю в неизменном виде.

## Передача

Схема CIMI и ассоциированная с ней абстрактная структура записи описывают, как должны быть однозначно помечены сервером элементы/поля базы данных. Передача элементов от сервера к клиенту требует еще одного набора спецификаций. Z39.50 использует понятие синтаксиса записей для определения того, каким образом сервер упаковывает элементы базы данных для отсылки клиенту. Синтаксис записи предписывает серверу, каким

образом должны быть отформатированы элементы/поля базы данных перед отсылкой их клиенту.

Общий синтаксис записи GRS-1 (Generic Record Syntax) Z39.50 позволяет серверу использовать произвольно структурированные данные. Профиль CИMІ требует поддержки только синтаксиса записи GRS-1, обязательной является поддержка следующих параметров GRS-1: tagType, tagValue, tagOccurrence, content, appliedVariant.

Для обеспечения интероперабельности между библиотеками и музеями сервера могут поддерживать другие синтаксисы записи, включая USMARC и SUTRS (Simple Unstructured Text Record Syntax).

Схема CИMІ может быть использована и вне протокола Z39.50. Тогда, когда работа происходит за рамками профиля CИMІ, можно создавать и передавать записи баз данных, соответствующих схеме CИMІ, в других форматах, таких как, например, XML.

## **6. ПРИМЕР РАСПРЕДЕЛЕННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ НА БАЗЕ Z39.50**

В заключение приведем пример распределенной информационной системы на базе Z39.50, схема которой приведена на рис. 2.

Центральное место в информационной системе занимают несколько серверов Z39.50, каждый из которых обеспечивает доступ к своим базам данных. Все серверы Z39.50 связаны механизмом перенаправления запросов.

В качестве клиентского программного обеспечения для простого доступа к информационной системе можно использовать шлюз Z39.50-WWW, представляющий собой CGI-приложение с функциями клиента Z39.50. Для этого система должна содержать WWW-сервер, на котором исполняется программа шлюза WWW-Z39.50.

Доступ пользователей к информационной системе может осуществляться из обычных WEB-браузеров (клиент WWW) через шлюз WWW-Z39.50 или из специализированных рабочих мест по протоколу Z39.50 (клиент Z39.50).

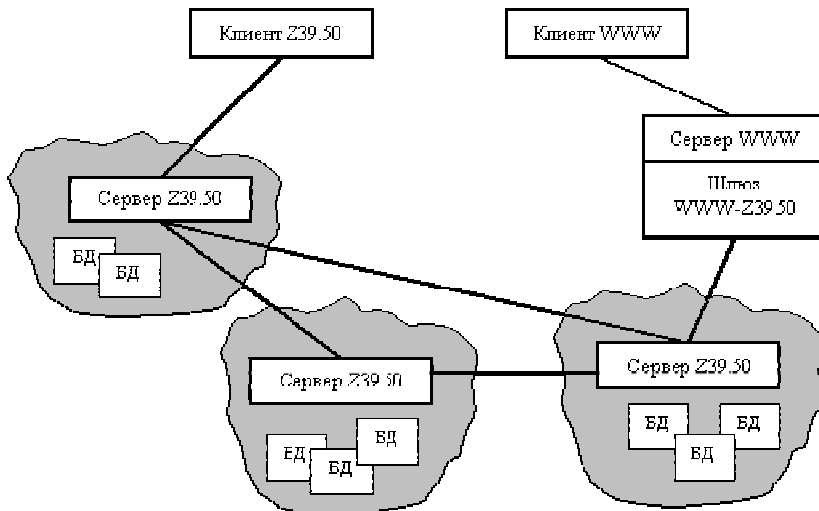


Рис. 2. Схема распределенной информационной системы на базе Z39.50

## ЗАКЛЮЧЕНИЕ

Профиль CIMI отражает набор спецификаций при использовании Z39.50 для поиска и извлечения информации о культурном наследии. Он также предоставляет две значительных области стандартизации, которые могут быть полезны вне области применения Z39.50.

Во-первых, набор атрибутов CIMI-1 определяет большой набор пунктов доступа, который может быть использован для поиска информации о культурном наследии. Поскольку этот набор пунктов доступа был получен в результате эмпирических исследований и обсуждений с членами музейного сообщества, он может рассматриваться как представление общего набора пунктов доступа, полезных в области информации о культурном наследии.

Во-вторых, схема CIMI предоставляет стандартный список элементов баз данных и организации этих элементов для обмена информацией о культурном наследии. Стандартный список может быть использован как конвертор или метаязык, для того чтобы пометить элементы локальной базы данных и обмена этими элементами с другими системами.

Z39.50 как протокол обмена <компьютер-компьютер> использует эти структуры, для того чтобы сделать возможным интероперабельный поиск и

извлечение информации. В контексте приложения к информации о культурном наследии профиль CIMI определяет использование взаимопонятных атрибутов и элементов схемы для стабильного извлечения информации посредством Z39.50.

Возможно, в скором времени и музейное сообщество России подключится к программе предоставления доступа к своим информационным ресурсам по протоколу Z39.50 с использованием профиля CIMI.

### СПИСОК ЛИТЕРАТУРЫ

1. Жижимов О.Л. Введение в Z39.50. — Новосибирск: Изд-во НГОНБ, 2000.
2. Жижимов О.Л., Мазов Н.А. О доступе к информационным ресурсам по культурному наследию по протоколу Z39.50 // Матер. конф. EVA'2000. «Электронная конвергенция: новые технологии в музеях, галереях, библиотеках и архивах», 30 окт.–3 нояб. 2000 г. — М.: Центр ПИК Минкультуры РФ, 2000. — 08-2-1–08-2-2.
3. Касьянов В.Н., Несговорова Г.П., Волянская Т.А. Виртуальный музей истории информатики в Сибири. // Современные проблемы конструирования программ. — Новосибирск, 2002. — С. 169–181.
4. Мазов Н.А., Жижимов О.Л. Профиль Z39.50-CIMI как основа интеграции информационных ресурсов по культурному наследию // Матер. конф. EVA'2003. «Информация для всех: культура и технологии информационного общества», 1–5 дек. 2003 г. — М.: Центр ПИК Минкультуры РФ, 2003.
5. ANSI/NISO Z39.50-1995. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. // Z39.50 Maintenance Agency Official Text for Z39.50-1995. — 156 p. <<http://lcweb.loc.gov/z3950/agency/1995doce.html>>
6. ANSI/NISO Z39.50-2003. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. // NISO Press, Bethesda, Maryland, U.S.A. — 276 p. <<http://www.niso.org/standards/resources/Z39-50-2003.pdf>>
7. Attribute Set Bib-1 (Z39.50-1995): Semantics. (1995, September). <<ftp://ftp.loc.gov/pub/z3950/defs/bib1.txt>>.
8. Janney, Kody and Sledge, Jane. (1995, September). A User Model for CIMI Z39.50 Application Profile. <[http://www.cimi.org/documents/Z3950\\_app\\_profile\\_0995.html](http://www.cimi.org/documents/Z3950_app_profile_0995.html)>.
9. Library of Congress. (1996, May). Z39.50 Profile for Access to Digital Collections. <<http://lcweb.loc.gov/z3950/agency/profiles/collections.html>>.
10. The CIMI Profile Release 1.0H. A Z39.50 Profile for Cultural Heritage Information. (1998, November). <<http://www.cimi.org/documents/HarmonizedProfile/HarmonProfile1.htm>>
11. Weibel, S., Kunze, J., Lagoze, C., Wolf, M. (1998, September). RFC 2413: Dublin Core Metadata for Resource Discovery. <<ftp://ftp.isi.edu/in-notes/rfc2413.txt>>