Информационное обеспечение поддержки семантической интероперабельности

Д.В. Жевнерчук, В.В. Магафуров

Аннотация. Предложена модель описания интерфейсов приложений, которая может быть использована при построении комплексных диалоговых систем, обладающих свойством интероперабельности своих компонент.

Ключевые слова: гетерогенная среда, семантическая интероперабельность, модели, гипервизор, виртуализация.

Введение

При создании информационных систем с повышенными требованиями к свойствам открытости часто возникает проблема включения программных компонент с низким коэффициентом семантической интероперабельности [1]. В работах [2,3] описаны подходы к повышению способности компонент к взаимодействию с внешним окружением. Отмечается, что, несмотря на дорогостоящие проекты, компаниями IBM, Microsoft, Oracle, SAP и др. цели эффективной семантической интероперабельности не были достигнуты. Одним из перспективных направлений является интероперабельность межсистемных сообщений. В этом случае информационная система может быть рассмотрена с точки зрения совокупности интерфейсов взаимодействия с внешним программным обеспечением, программно-аппаратной платформой и пользователями. Передачу данных и управляющих сигналов программам выполняет в конечном итоге операционная система.

Постановка задачи

Рассмотрим систему, состоящую из некоторого множества вычислительных узлов, функционирующих под управлением различных программно-аппаратных платформ. На каждом узле выполняются информационные системы со своим набором интерфейсов.

Требуется разработать технологию построения среды поддержки семантической интероперебельности, которая обладает механизмами включения новых программно-аппаратных комплексов и построения на их основе сервис-ориентированных диалоговых систем. В круг задач входит разработка форматов хранения информации об интерфейсах приложений, включая пользовательский интерфейс, и о доступных функциях системы.

Информационное обеспечение и потоки сообщений в системе

На Рис.1 представлена схема передачи управляющих сигналов и данных между компонентами среды поддержки семантической интероперабельности.

В состав среды входит 4 подсистемы:

– клиент,

– сервер адресов, на котором располагаются сведения о существующих информационных системах,

– сервер ресурса,

– ресурс, отдельная информационная система с поддержкой CLI.

Рис. 1. Схема передачи управляющих сигналов и данных

На узлах хранится информация о поддерживаемых интерфейсах и функциях, которые доступны через этот интерфейс. На специально выделенном сервере адресов хранится информация о существующих в системе приложениях с указанием адреса узла. Запрос к системе выполняется с удаленного клиентского приложения, например, Web-браузера. Предварительно запрашиваются сведения о доступных приложениях или отдельных сервисах, построенных на базе приложений. Клиенту передается список приложений и сервисов. Идентификационные данные выбранного приложения или сервиса передаются содержащему их узлу. После установления соединения, клиенту передается информация о доступных интерфейсах, строится пользовательский интерфейс. Далее клиент взаимодействует с сервером ресурса, который осуществляет прием кода запрашиваемого сервиса и входные данные, преобразует их в инструкции, интерпретируемые информационной системой, и инициирует выполнение этих инструкций. После обработки запроса результат возвращается от ресурса к серверу, где он передается клиенту в доступном для него формате.

Клиент содержит список технологий, которые могут быть использованы для построения пользовательского интерфейса (карта UI), сервер адресов содержит идентификационные сведения узлов, на которых расположены ресурсы, а также список сервисов, доступных у ресурсов. Сервер ресурса включает описание интерфейсов ресурса и поддерживаемые ресурсом форматы запросов.

В рамках статьи рассматриваются системы с поддержкой интерфейса командной строки.

Модель интерфейсов CLI информационной системы

Были проанализированы CLI ряда программных систем, работающих под управлением ОС Windows и *nix. Любая команда CLI может быть описана с помощью шаблона:

[символ_начала_команды]имя_команды
[параметр_1 [параметр_2 […]]] (1)

Символ начала команды может быть самым разным, однако чаще всего для этой цели используется косая черта (/). Если строка вводится без этого символа, выполняется некоторая базовая команда. Если же такой базовой команды нет, символ начала команды отсутствует вообще (как, например, в DOS). Параметры команд могут быть представлены в самых разных форматах. В основном применяются следующие правила:

– параметры разделяются пробелами (и отделяются от названия команды пробелом)

– параметры, содержащие пробелы, обрамляются кавычками-апострофами (') или двойными кавычками (")

– если параметр используется для обозначения включения какой-либо опции, выключенной по умолчанию, он начинается с косой черты (/) или дефиса (-)

– если параметр указывает действие из группы действий, назначенных команде, он не начинается со специальных символов

– если параметр указывает объект, к которому применяется действие команды, он не начинается со специальных символов

– если параметр указывает дополнительный параметр какой-либо опции, то он имеет формат /опция:дополнительный_параметр (вместо косой черты также может употребляться дефис).

Примеры приложений, предоставляющих CLI:

– САПР AutoCAD (http://www.autodesk.ru);

– система моделирования общего назначения GPSS World (http://www.minutemansoftware.com);

– свободный режимный текстовый редактор VIM (http://www.vim.org/).

Предлагается описание формата представления CLI приложения с помощью XSD [4], так как в отличие от большинства языков описания XSD был разработан для использования в создании программного обеспечения для обработки документов XML, а также:

– обладает мощными средствами для определения сложных структур данных,

– обеспечивает понятный способ описания грамматики языка,

– способен легко модернизироваться и расширяться.

XML-описание входящего сообщения состоит из общего описания формата и описания команды.

```
<xs:element name="inMessage">
  <xs:complexType>
        <xs:element name="command" type="command" minOccurs="1" use="required"/>
        <xs:attribute name="number" type="xs:string"/>
        <xs:attribute name="service" type="xs:string"/>
  <xs:complexType>
</xs:element>
```

Тег command является контейнером для описания команд, подаваемых на вход CLI.

```
<xs:complexType name="command">
  <xs:sequence>
     <xs:element name="startSymbol" type="startSymbol" minOccurs="0" maxOccurs="1"/>
    <xs:element name="comName" type=" tName " use="required" minOccurs="1" maxOccurs="1"/>
    <xs:element name="comParams" type="comParams" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:ComplexType>
```

Тег startSymbol служит для описания символа начала команды, если в этом есть необходимость.

```
<xs:simpleType name="startSymbol">
        <xs:restriction base="xs:string">
                <xs:enumeration value="/"/>
                <xs:enumeration value="-"/>
                <xs:enumeration value=":"/>
                <xs:whiteSpace value="collapse"/>
        </xs:restriction">
</xs:simpleType>
```

Тег comName определяет имя команды.

```
<xs:simpleType name=" tName ">
        <xs:restriction base="xs:string">
```

```
                <xs:whiteSpace value="collapse"/>
                <xs:pattern value="([a-zA-Z])*"/>
        </xs:restriction>
</xs:simpleType>
```

Тег comParams является контейнером описаний параметров команды.

```
<xs:complexType name="comParams ">
  <xs:sequence>
    <xs:element name="pSeparator" type="pSeparator" use="required" maxOccurs="1"/>
    <xs:element name="paramKey" type="pKey" minOccurs="0" maxOccurs="1"/>
    <xs:element name="pName" type="tName"  maxOccurs="1"/>
    <xs:element name="addParam" type=" addParam " minOccurs="0" maxOccurs="1"/>
    <xs:element name="pSeparator" type="pSeparator" minOccurs="0"  maxOccurs="1"/>
    <xs:element name="pValue" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

Тег pSeparator описывает разделитель параметров.

```
<xs:simpleType name="pSeparator">
        <xs:restriction base="xs:string">
                <xs:enumeration value=" "/>
                <xs:enumeration value="_"/>
        </xs:restriction>
</xs:simpleType>
```

Тег paramKey описывает элемент включения/выключения опции.

```
<xs:simpleType name="pKey">
        <xs:restriction base="xs:string">
                <xs:enumeration value=" /"/>
                <xs:enumeration value="-"/>
        </xs:restriction>
</xs:simpleType>
```

Тег addParam описывает дополнительные параметры в случае их наличия.

```
<xs:simpleType name="addParam">
  <xs:sequence>
    <xs:restriction base="xs:string">
        <xs:element name="colon" type="xs:string" use="required" fixed=":" maxOccurs="1"/>
        <xs:element name="addParName" type="tName"  maxOccurs="1"/>
    </xs:restriction>
  </xs:sequence>
</xs:simpleType>
```

Ниже представлен пример входящего CLI – сообщения, сформированного в соответствии с предлагаемым форматом и использующего механизм макроподстановок для передачи пользовательских данных.

```
<inMessage>
  <command>
        <startSymbol>@usSymbol</startSymbol>
        <comName>@usCom</comName>
        <comParams>
                <pSeparator >@usSep</pSeparator >
                <pName>@usPName</pName>
         </comParams>
```

```
        <comParams>
                <pSeparator >@usSep</pSeparator >
                <pName>@usPName</pName>
                <addParam>
                        <colon>:</colon>
                        <addParName>>@usAddPName </addParName>
                </addParam>
                <pValue>@usPValue</pValue>
         </comParams>
        </command>
</inMessage>
```

Исходящее CLI-сообщение рассматривается как совокупность некоторых текстовых данных.

```
<xs:element name="outMessage">
        <xs:annotation>
                <xs:documentation>OutMessage</xs:documentation>
        </xs:annotation>
        <xs:element name="data" type="xs:string"/>
</xs:element>
```

Карта сервисов

Сервис представляет собой функцию, которая может быть запрошена у приложения. Для доступа к функции необходимо передать операционной системе, под управлением которой функционирует приложение, информацию о параметрах CLI для последующего ее выполнения. С клиентской системы передается идентификатор запрашиваемого сервиса и пользовательские данные для обработки. Для каждого приложения поддерживается xml файл с информацией о доступных сервисах и шаблонах команд, именуемый далее картой сервисов. Шаблон команды включает наименование команд, список ключей, параметров, а также макроподстановок для пользовательских данных.

Экспериментальная система

Разработана система управления диалоговыми программами, поддерживающими интерфейс командной строки. Схема системы приведена на Рис. 2.

Для решения проблемы переносимости интерфейса пользователя использован механизм виртуализации. Суть его состоит в следующем. Для всех серверных информационных систем создается формальное описание интерфейса командной строки CLI и пользовательского интерфейса. Строится карта графического интерфейса пользователя, которая содержит информацию о дереве форм и элементах форм. Дополнительно формируется список доступных сервисов, которые могут быть запрошены пользователем. С каждым сервисом связываются интерфейсы CLI и подмножество элементов GUI, необходимые для взаимодействия с пользователем. По коду сервиса сервер определяет дерево и состав форм, формирует пользовательский интерфейс в javascript коде.

Для обеспечения переносимости пользовательского интерфейса были разработаны:

1. формат описания интерфейсов (карта интерфейсов) подключаемой информационной системы;
2. формат описания средств построения/воспроизведения UI клиентской платформы;
3. формат описания функциональных сервисов (карта сервисов) произвольных информационных систем.

Клиентское приложение – это браузер, в котором формируется удаленный GUI на основе передаваемого с сервера javascript кода. Для повышения эффективности разработки и обеспечения независимости от браузера была использована библиотека jquery. Сервер ресурсов включает группу php модулей, выполняющих следующие функции:

– прием запросов сервисов от клиентских систем в асинхронном режиме;

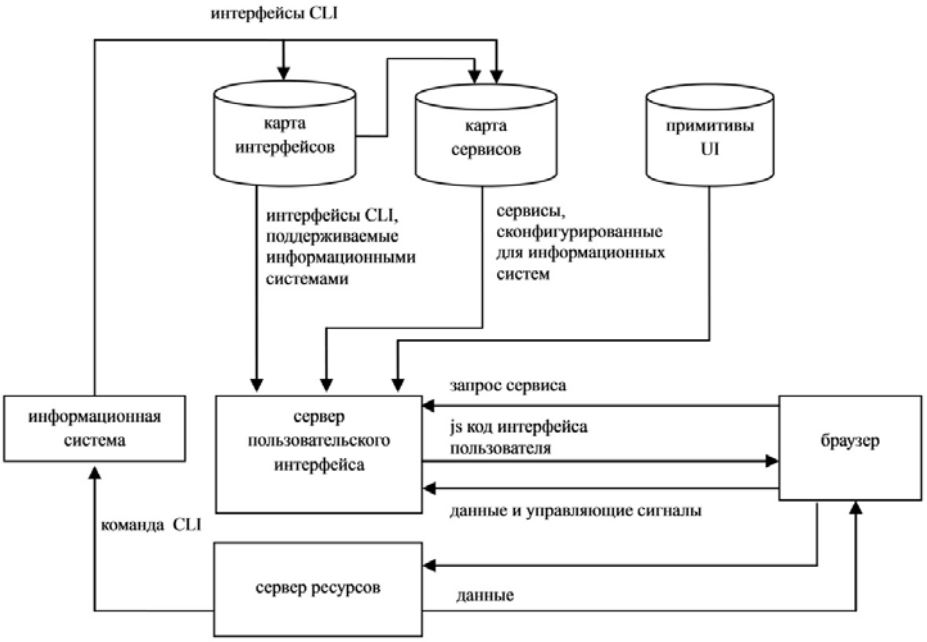

Рис. 2. Схема системы управления диалоговыми программами

– формирование команд на основе поддерживаемых шаблонов инструкций CLI;
– передача команд конечным приложениям;
– прием результатов обработки;
– передача результатов клиентским системам.

Проведено тестирование системы для приложений GPSS World Student (платформа win32) и Midnight Commander (платформа Linux).

Заключение

Разработана архитектура среды, расширяемой по приложениям на основе разных API операционных систем, поддерживающим интерфейс командной строки. Для обеспечения независимости от программной платформы применялись гипервизоры. Предложен кросс-платформенный формат хранения шаблонов инструкций CLI, на основе которого построены механизмы взаимодействия клиентских систем и серверного программного обеспечения с CLI.

Методика может быть применена при создании систем с поддержкой семантической интероперабельности программных компонент, обладающих другими интерфейсами.

Литература

1. В.К. Батоврин, А.С. Королев, Способ количественной оценки интероперабельности, Информационные технологии и вычислительные системы, 5/2009.
2. Ю.В. Бородакий, Ю.Г. Лободинский, К проблеме обеспечения интероперабельности, Информационные технологии и вычислительные системы, 5/2009.
3. В.К. Батоврин, Ю.В. Гуляев, А.Я. Олейников, Обеспечение интероперабельности – основная тенденция в развитии открытых систем, Информационные технологии и вычислительные системы, 5/2009.
4. Официальный сайт консорциума W3C [Электронный ресурс]. - Режим доступа к ресурсу: http://www.w3.org.

Жевнерчук Дмитрий Валерьевич. Доцент Чайковского технологического института ФГБОУ ВПО «Ижевский государственный технический университет». Окончил Чайковский технологический институт ФГБОУ ВПО «Ижевский государственный технический университет» в 2004 году. Кандидат технических наук, доцент. Автор 25 печатных работ и двух монографий. Область научных интересов: имитационное моделирование открытых информационных систем. E-mail: drevnigeck@yandex.ru

Магафуров Вадим Вильданович. Аспирант Чайковского технологического института ФГБОУ ВПО «Ижевский государственный технический университет». Окончил Чайковский технологический институт ФГБОУ ВПО «Ижевский государственный технический университет» в 2011 году. Автор трех печатных работ. Область научных интересов: открытые системы, проектирование программных систем, методики оценки свойств систем. E-mail: vadimag59@gmail.com