

УДК 004.822

ОНТОЛОГИЧЕСКИЙ ПОДХОД К ПРОЕКТИРОВАНИЮ БИЛЛИНГОВЫХ СИСТЕМ

Д.В. АНЦИБОР, М.Ю. ТЕРНОВОЙ, Д.В. ШУНКЕВИЧ

*Национальный технический университет Украины
«Киевский политехнический институт»,
пр. Победы, 37, Киев, 03056, Украина*

*Белорусский государственный университет информатики и радиоэлектроники
П. Бровки, 6, Минск, 220013, Беларусь*

Поступила в редакцию 20 января 2015

Рассмотрены принципы и подход к применению онтологий в биллинговых системах и других системах OSS/BSS, выделены преимущества такого подхода. Построена онтология системы биллинга, выделены классы и отношения. Разработанная модель онтологии реализована средствами Protégé. Осуществлена проверка разработанной онтологии на корректность и непротиворечивость.

Ключевые слова: онтология, OSS/BSS системы, Protégé, база знаний, биллинговые системы.

Введение

В связи с ростом конкуренции среди поставщиков телекоммуникационных услуг, вопрос повышения эффективности их работы с каждым днем становится все более актуальным. Улучшение методов доступа к информации, анализа, представления и систематизации данных играют ключевую роль в развитии рынка телекоммуникаций. В том числе методы доступа и обработки информации, хранящейся в системах биллинга и других системах OSS/BSS [1]. Существующие стандарты разработки OSS/BSS систем TM Forum [2] позволяют частично решить проблему доступа и обмена информацией. Но системы, которые были внедрены ранее, а также новые системы, не соответствующие стандартам TM Forum, усложняют интеграцию информации из различных систем OSS/BSS в телекоммуникационных компаниях. Существуют альтернативные подходы к интеграции информации OSS/BSS систем с использованием технологий Semantic Web [3], но такие подходы реализованы в виде прикладных программ и являются коммерческой тайной. Использование онтологий в информационно-телекоммуникационных системах имеет ряд существенных преимуществ, таких как возможность логического вывода, интеграция доступной в сети информации, сокращение концептуального несоответствия, повторное использование знаний (уже определенных в других онтологиях понятий) и использование формальной логики [4–6]. Формальная структура онтологий упрощает их компьютерную обработку, что является преимуществом онтологий в качестве способа представления данных [7]. В вычислительном плане использование онтологий также позволяет достичь сокращения времени вычислений и поиска [8]. Возможность логического вывода позволяет получать новые знания, а также выполнять проверку уже имеющихся данных на целостность.

В данной работе рассмотрен подход к построению и реализации онтологии системы биллинга, которая является составной частью общей онтологии для OSS / BSS систем. Были определены этапы построения онтологии, этапы реализации и получения логического вывода по онтологии.

Разработка онтологии

При описании онтологии определяются общие знания о классах и отношениях,

называемыми также TBox. Также выделяют знания о конкретных экземплярах: к какому классу они принадлежат, какими отношениями они связаны друг с другом, эта часть знаний называется ABox. Существуют различные методы построения онтологий, предложенные различными исследовательскими группами [9–12].

Для построения онтологии системы биллинга в данной работе были выделены следующие этапы, присущие большинству разработанных современных методов создания онтологий:

- определение классов и создание иерархии между ними (относится к TBox);
- определение отношений между классами (относится к TBox);
- добавление ограничений и свойств для отношений (относится к TBox);
- определение необходимой для онтологии дескрипционной логики;
- создание экземпляров классов и экземпляров отношений (относится к ABox).

Определение классов и создание иерархии. Классы в создаваемой онтологии должны быть близки к физическим или логическим объектам, а связи между ними к связям между этими объектами.

Существует несколько подходов к созданию иерархии классов [13]:

- сверху-вниз;
- снизу-вверх;
- комбинация первых двух методов.

В данной работе использовался подход к построению иерархии сверху-вниз. Были выделены 6 основных классов: «Провайдер», «Услуга», «Тарифный план», «Учетная запись», «Пользователь», «Счет». Также выделено 3 подкласса для класса «Услуга»: «Телефония», «Интернет» и «Телевидение» и 3 подкласса для класса «Счет»: «Оплаченный счет (receipt)», «Баланс», «Счет к оплате (invoice)». Ниже детальнее описано назначение классов.

Класс «Провайдер» содержит информацию о провайдере услуг.

Класс «Услуга» содержит информацию об услугах, предоставляемых провайдером, их описание. Данный класс является родительским для подклассов услуг «Интернет», «Телевидение» и «Телефония».

Класс «Тарифный план» содержит информацию о тарифных планах провайдера.

Класс «Учетная запись» – это учетная запись пользователя.

Класс «Пользователь» содержит непосредственно информацию о человеке, который имеет учетную запись (его ФИО, адрес и т.д.).

Класс «Счет» содержит информацию о денежных счетах, которые связаны с данной учетной записью. Этот класс является родительским для классов «Оплаченный счет (receipt)», «Баланс», «Счет к оплате (invoice)».

Определение отношений. В онтологиях существует два типа отношений: отношение между классами и отношение типов данных. В онтологии системы биллинга выделяются следующие отношения между классами.

Класс «Провайдер» связан с классом «Услуга» отношением «предоставляет Услугу», с классом «Тарифный план» отношением «имеет Тарифный план» и с классом «Пользователь» отношением «имеет отношение».

Класс «Тарифный план» связан с классом «Услуга» отношением «имеет Услугу».

Класс «Учетная запись» связан с классами «Счет» и «Тарифный план» отношениями «имеет Счет» и «пользуется Тарифным планом» соответственно.

Класс «Пользователь» связан с классом «Учетная запись» отношением «имеет Учетную запись», с классом «Услуга» отношением «пользуется Услугой» и с классом «Провайдер» отношением «имеет отношение».

Класс «Счет» связан с классом «Тарифный план» отношением «имеет Тарифный план».

Отношения типов данных, имеющихся в онтологии системы биллинга, описаны ниже.

Класс «Провайдер» имеет отношения типов данных:

- «Называется» со строчным типом данных (название провайдера);
- «Имеет дату создания» с типом данных дата/время (дата создания провайдера);
- «Имеет описание» со строчным типом данных (дополнительное описание провайдера).

Класс «Услуга» имеет такие отношения типов данных:

- «Называется» со строчным типом данных (название услуги);
- «Имеет описание» со строчным типом данных (описание услуги).

Класс «Тарифный план» имеет следующие отношения типов данных:

- «Называется» со строчным типом данных (название тарифного плана);
 - «Имеет описание» со строчным типом данных (описание тарифного плана);
 - «Имеет стоимость» с числовым типом данных (стоимость тарифного плана).
- Класс «Учетная запись» содержит следующие отношения типов данных:
- «Имеет login» со строчным типом данных (login учетной записи);
 - «Имеет пароль» со строчным типом данных (пароль учетной записи);
 - «Имеет e-mail» со строчным типом данных (зарегистрированный e-mail учетной записи);
- В описании класса «Пользователь» присутствуют следующие отношения типов данных:
- «Имеет ФИО» со строчным типом данных (ФИО пользователя);
 - «Имеет паспортные данные» со строчным типом данных (паспортные данные пользователя);
 - «Имеет адрес» со строчным типом данных (адрес пользователя).
- Класс «Оплаченный счет (receipt)» имеет следующие отношения типов данных:
- «Имеет уплаченную сумму» с числовым типом данных (уплаченная сумма);
 - «Имеет дату оплаты» с типом данных дата/время (дата оплаты по данному счету).
- Класс «Баланс» имеет следующие отношения типов данных:
- «Имеет сумму» с числовым типом данных (сумма денег на балансе)
 - «Имеет дату баланса» с типом данных дата/время (дата, за которую смотрят информацию о балансе).
- Класс «Счет к оплате (invoice)» имеет следующие отношения типов данных:
- «Имеет дату выставления» с типом данных дата/время (дата выставления счета);
 - «Имеет сумму» с числовым типом данных (сумма, указанная в счете);
- Основные классы онтологии и основные отношения между ними показаны на рис. 1.

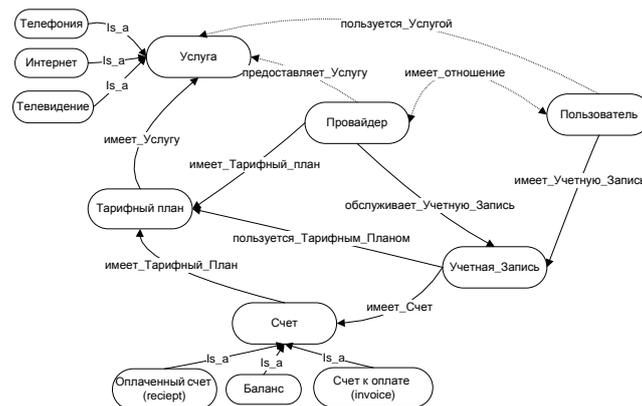


Рис. 1. Схематическое изображение онтологии биллинга

Ограничение и свойства отношений. Определение дескрипционной логики. Ограничения на отношения позволяют описать допустимые значения, тип, количество значений и другие особенности, которыми может обладать отношение данного класса. Отношения могут быть симметричными, рефлексивными, транзитивными, обратными, могут иметь функциональные ограничения, или же быть иерархическими [14]. Также определение типов отношений необходимо для определения дескрипционной логики и соответственно языка написания онтологии, который необходим для описания данной онтологии.

Ограничения и свойства отношений в данной онтологии. Инверсия отношений $R \equiv S^{-}$. Многие отношения в онтологии имеют явно выраженные обратные отношения. Примерами таких отношений являются «имеет Счет» и «является Счетом для», «имеет Учетную запись» и «является Учетной записью для», «имеет Услугу» и «является Услугой для», «имеет Тарифный план» и «является Тарифным планом для». Обратные отношения удобно использовать для избежания ошибок при заполнении информации. Такие отношения добавляются автоматически с помощью машины вывода, если одно из них присутствует в онтологии. Также возможно использование неявно заданных обратных отношений.

Иерархия отношений $R \subseteq S$. В общем случае эти отношения соответствуют иерархии классов. Отношение «имеет Услугу Интернет», «имеет Услугу Телефония» и «имеет Услугу Телевидение» являются дочерними к отношению «имеет Услугу». Отношение «имеет Баланс», «имеет Счет к оплате» и «имеет Оплаченный счет» – дочерние от отношения «имеет Счет». В результате вывода, при наличии в онтологии дочернего отношения, машиной вывода будет

добавлено отношение и для родительского отношения.

Симметричность $R \sqsubseteq (R^-)$ и асимметричность отношений $R \sqsubseteq \text{not}(R^-)$. Отношение «имеет отношение» между классами «Пользователь» и «Провайдер» является симметричным. Асимметричными является большинство отношений, например отношения «имеет Тарифный план», «обслуживает Учетную запись» и другие. Для симметричных отношений автоматически добавляется отношение для обратной пары при логическом выводе. Асимметричность для отношений указывается для предотвращения ошибок при заполнении информации в онтологии.

Функциональность отношений ($\leq 1R$). Большинство отношений типов данных являются функциональными, например отношение «имеет пароль» для класса «Учетная запись» или «имеет дату создания» для класса «Провайдер».

Иррефлексивность отношений. При отсутствии рефлексивных отношений, все отношения данной онтологии являются иррефлексивными. Это свойство отношений добавляется для предотвращения ошибок, которые могут возникнуть при заполнении информации в онтологии.

Композиция отношений $R \circ S$. Используется для определения отношений «имеет отношение», определенного как композиция отношений «имеет Учетную запись» и «inverse обслуживает Учетную запись», а также для определения отношения «пользуется Услугой», определенного как композиция отношений «имеет Учетную запись», «пользуется Тарифным планом» и «имеет Услугу». Также композиция используется для определения отношения «предоставляет Услугу». Эти отношения добавляются машиной вывода, если в онтологии будет присутствовать необходимая цепочка отношений.

Отношение типов данных. Для большинства классов онтологии определены отношения типов данных, например свойство «Имеет пароль» класса «Учетная запись», свойство «Имеет дату создания» класса «Провайдер».

Исходя из выделенных выше свойств отношений, можно сказать, что для данной онтологии достаточно выразительности логики SRIF(D), где буквы SR – означают наличие транзитивности, композиции и характеристики (симметричность, рефлексивность и т.д.) ролей, I – инверсию, F – функциональность, (D) – отношение типов данных. Так как в онтологии используются композиции отношений, а они были добавлены только в версии языка описания онтологий OWL 2, то для дальнейшего построения онтологии был выбран этот язык. На рис. 2 приведена часть TBox для онтологии системы биллинга.

```
TBox:
 $\exists$ имеет_отношение. T  $\sqsubseteq$  имеет_Учетную_запись o inverse (обслуживает_Учетную_запись),
Symmetric: имеет_отношение;
 $\exists$ пользуется_Услугой. T  $\sqsubseteq$ 
    имеет_Учетную_запись o пользуется_Тарифным_планом o имеет_Услугу;
Провайдер p ( $\exists$ предоставляет_Услугу.Телевидение)  $\sqsubseteq$  Провайдер_Телевидения;
Провайдер p ( $\exists$ предоставляет_Услугу.Телефония)  $\sqsubseteq$  Провайдер_Телефонии;
Провайдер p ( $\exists$ предоставляет_Услугу.Интернет)  $\sqsubseteq$  Провайдер_Интернета;
Пользователь p ( $\exists$ пользуется_Услугой.Интернет)  $\sqsubseteq$  Пользователь_Интернета;
Пользователь p ( $\exists$ пользуется_Услугой.Телевидение)  $\sqsubseteq$  Пользователь_Телевидения;
Пользователь p ( $\exists$ пользуется_Услугой.Телефония)  $\sqsubseteq$  Пользователь_Телефонии;
 $\exists$ предоставляет_Услугу. T  $\sqsubseteq$  имеет_Тарифный_план o имеет_Услугу;
является_Учетной_записью o не  $\sqsubseteq$  имеет_Учетную_запись;
```

Рис. 2. Часть TBox для онтологии системы биллинга

Создание экземпляров класса. Во время этого этапа проводится создание конкретных экземпляров классов, а также добавление отношений для этих экземпляров. Другими словами, происходит наполнение ABox. Данные для ABox (факты) в основном берутся из баз данных существующих систем. На рис. 3 приведена часть ABox для онтологии системы биллинга. Отметим, что в базах данных (БД) ложно любое утверждение, о котором не известно, что оно верно (т.е. используется предположение о замкнутости мира). В подходах с использованием онтологий используется предположение об открытости мира, т.е. если об утверждении неизвестно, что оно верно, то говорится, что это утверждение является неопределенным (оно может быть как истинным так и ложным) [15]. Используя факты, полученные из баз данных (ABox) и общие знания о классах

и отношениях онтологий (TBox), можно получить новые знания путем логического вывода. Объединение TBox и ABox называется базой знаний (БЗ).

АВох:
 Зайченко: **Пользователь**, Петренко: **Пользователь**, Иваненко: **Пользователь**,
 Kyiv TV: **Провайдер**, MobiNet: **Провайдер**, On-net: **Провайдер**,
 Смотри дома: **Тарифный план**, (Зайченко, Zaic1989): **имеет_Учетную_запись**,
 (Петренко, Petr1992): **имеет_Учетную_запись**, (Иваненко, Ivan1989): **имеет_Учетную_запись**,
 (Ivan1989, Смотри дома): **пользуется_Тарифным_планом**,
 (Zaic1989, На связи): **пользуется_Тарифным_планом**,
 (Zaic1989, Свободный): **пользуется_Тарифным_планом**,
 (MobiNet, Zaic1989): **обслуживает_Учетную_запись**, (Kyiv TV, Смотри дома): **имеет_Тарифный_план**,
 (MobiNet, На связи): **имеет_Тарифный_план**, (MobiNet, Свободный): **имеет_Тарифный_план**,
 (On-net, Эконом+): **имеет_Тарифный_план**, (Свободный, Мобильная связь): **имеет_Услугу_Телефония**,
 (Смотри дома, Аналоговое телевидение): **имеет_Услугу_Телевидение**,
 (Эконом+, Цифровое телевидение): **имеет_Услугу_Телевидение**,
 (Эконом+, Домашний интернет): **имеет_Услугу_Интернет**,
 (На связи, Мобильный интернет): **имеет_Услугу_Интернет**.

Рис. 3. Часть АВох для онтологии системы биллинга

Ниже приведена сравнительная таблица запросов к базе данных и базе знаний. Таким образом, с помощью использования правил и дополнительных взаимоотношений, указанных в TBox части онтологии, запросы к базе знаний вернули более полные результаты, чем запросы к базе данных.

Сравнение запросов к БД и к БЗ

Запрос	БД	АВох	БЗ
Провайдер (On-net)	Да	Да	Да
Пользователь_Телевидения (Иваненко)	Нет	Неизвестно	Да
∃имеет_отношение.Провайдер (Зайченко)	Нет	Неизвестно	Да
¬Пользователь_Интернета (Петренко)	Да	Неизвестно	Нет
Тарифный_план (Смотри дома)	Да	Да	Да
∃пользуется_Услугой.Телефония (Зайченко)	Нет	Неизвестно	Да
Провайдер_Интернета ∩ Провайдер_Телефонии (MobiNet)	Нет	Неизвестно	Да
Тарифный_план (Зайченко)	Нет	Неизвестно	Нет
∃предоставляет_Услугу_Телевидение (Kyiv TV)	Нет	Неизвестно	Да
∃является_Учетной_записью.Пользователь (Petr1992)	Нет	Неизвестно	Да
Пользователь_Телефонии ∩ Пользователь_Интернет (Зайченко)	Нет	Неизвестно	Да
Провайдер_Интернета (On-net)	Нет	Неизвестно	Да
Провайдер ∩ ¬Провайдер_Интернета (Kyiv_TV)	Да	Неизвестно	Неизвестно

Построение онтологии. В работе [16] проведен подробный анализ существующих инструментов для построения онтологий. Редактор онтологий использовался для построения онтологии и фреймворк для построения баз знаний Protégé. Данный редактор поддерживает выбранный язык описания онтологий OWL 2. На основе сформированной онтологии с помощью программного модуля Hermit, встроенного в Protégé, были произведены логические выводы, что позволило выявить новые отношения для экземпляров, получить более полные результаты на запросы и осуществить проверку онтологии на непротиворечивость. На рис. 4 представлены иерархия классов онтологии, отношения и экземпляры, построенные в Protégé.

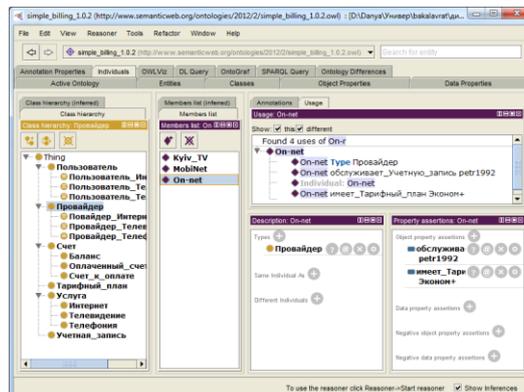


Рис. 4. Построенная онтология в редакторе Protégé

Заключение

Предложена онтология биллинга, как составляющая часть OSS/BSS систем. В онтологии использовались типы свойств и ограничения, соответствующие дескрипционной логике SRIF(D). Онтология была построена в графическом редакторе онтологий Protégé. Также был проведен логический анализ, что позволило получить новые знания. В дальнейшем планируется построение онтологий остальных составных частей OSS/BSS и общей объединенной онтологии, которые станут основой системы интеграции информации.

ONTOLOGY-BASED APPROACH TO BILLING SYSTEMS DESIGN

D.V. ANTSYBOR, M.Y. TERNOVOY, D.V. SHUNKEVICH

Abstract

The principles and approach to ontologies application in billing and other OSS/BSS systems, advantages of such an approach are described. Ontology of billing system is designed, classes and relations are described. Designed model was implemented using Protégé, and was verified on the correctness and consistency.

Список литературы

1. Kot T., Reverchuk A., Globa L. et. al. // Lecture notes in business information processing. 2012. Vol. 117. P. 142–152.
2. TM Forum Framework. [Электронный ресурс]. – Режим доступа: <http://www.tmforum.org/TMForumFramework/1911/home.html>. – Дата доступа: 10.01.2015.
3. Ontology. [Электронный ресурс]. – Режим доступа: <http://ontology.com>. – Дата доступа: 10.01.2015.
4. Голиков Н.В. Применение онтологий. [Электронный ресурс]. – Режим доступа: <http://www.ict.nsc.ru/ws/YM2006/10628/golikov.html>. – Дата доступа: 10.01.2015.
5. Tim Menzies, Cost-Benefits of Ontologies. [Электронный ресурс]. – Режим доступа: <http://menzies.us/pdf/99sigart.pdf>. – Дата доступа: 10.01.2015.
6. Hans-JörgHappel, Stefan Seedorf, Applications of Ontologies in Software Engineering. [Электронный ресурс]. – Режим доступа: https://km.aifb.kit.edu/ws/swese2006/final/happel_full.pdf. – Дата доступа: 10.01.2015.
7. Daniel Oberle, How Ontologies Benefit Enterprise Applications. [Электронный ресурс]. – Режим доступа: http://www.semantic-web-journal.net/system/files/swj212_2.pdf. – Дата доступа: 10.01.2015.
8. Предпосылки использования онтологий. [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/1078/270/lecture/6849>. – Дата доступа: 10.01.2015.
9. И.В. Ефименко, В.Ф. Хорошевский. Онтологическое моделирование: подходы, модели, методы, средства, решения. [Электронный ресурс]. – Режим доступа: http://www.hse.ru/data/2011/12/22/1261631357/WP7_2011_08_1.pdf. – Дата доступа: 10.01.2015.
10. Lenat D., Guha R.V. Building large knowledge based systems: representation and inference in the Cyc project. Massachusetts, 1989.
11. Fernandez M., Gomez-Perez A., Juristo N. Methontology: from ontological art towards ontological engineering. AAAI Technical Report SS-97-06, 2006.
12. TOVE Ontology Project. [Электронный ресурс]. – <http://www.eil.utoronto.ca/enterprise-modelling/tove/>. – Дата доступа: 10.01.2015.
13. N. Noy, D. McGuinness. Ontology Development: A Guide to Creating Your First Ontology. [Электронный ресурс]. – http://Protégé.stanford.edu/publications/ontology_development/ontology101.pdf. – Дата доступа: 10.01.2015.
14. OWL Web Ontology Language Reference. [Электронный ресурс]. – <http://www.w3.org/TR/owl-ref/#Property>. – Дата доступа: 10.01.2015.
15. Juan Sequeda. Introduction to: Open World Assumption vs Closed World Assumption. [Электронный ресурс]. – <http://semanticweb.com/introduction-to-open-world-assumption-vs-closed-world-assumption/>. – Дата доступа: 10.01.2015.
16. О.М. Овдей, Г.Ю. Проскудина. Обзор инструментов инженерии онтологий. [Электронный ресурс]. – <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part4/op>. – Дата доступа: 10.01.2015.