

УДК 004.021:004.822

В.В. Грачев, В.В. Разин, А.Ф. Тузовский

Анализ ситуаций в телекоммуникационных сетях с применением семантических технологий

Рассматривается подход с использованием технологий Semantic Web для анализа ситуаций в коммуникационных сетях. Предлагается онтология ситуаций в сети коммуникаций, набор правил для выявления интересующих пользователя ситуаций.

Ключевые слова: онтология, OWL, ситуационное управление, телекоммуникационная сеть, анализ ситуаций.

В последние десятилетия онтологии получили интенсивное развитие в исследованиях проблем искусственного интеллекта и в инженерии знаний. Они рассматриваются в качестве важного компонента реализации концепции Semantic Web [1]. В данной статье онтологические модели используются для анализа ситуаций, возникающих в коммуникациях между группами объектов. По мере развития мобильных и стационарных систем и телекоммуникаций все чаще возникает необходимость и возможность осуществлять поиск отдельных лиц или групп, в том числе находящихся в удаленных и труднодоступных местах отдыха, туризма, а также при возникновении различных природных аномалий (ураганы, наводнения, лесные пожары и др.). Представляет значительный интерес осуществление поиска различного типа групп и отдельных ее членов путем анализа их взаимодействий в телекоммуникационном пространстве.

Предлагаемый подход основывается на применении ранее разработанного подхода к анализу ситуаций [2, 3] путем использования методов разработки онтологий на языке OWL [4] и правил на языке SWRL [5]. Реализация данного подхода включает компоненты, работающие в реальном времени, для сбора данных о событиях, их хранения и выполнения запросов к ним, а также для отслеживания подходящих отношений и просмотра результатов, используя графический интерфейс пользователя и анализ телекоммуникационного трафика.

Постановка задачи. Решаемая задача может быть сформулирована следующим образом: **Имеется** массив информации о контактах между абонентами некоторой телекоммуникационной сети. Каждый абонент представлен уникальным идентификационным номером ID_i (например, IP адресом). Связь представлена кортежем вида $\langle ID_s, ID_r, t \rangle$, где ID_s – номер отправителя, ID_r – номер получателя, t – время установки связи. **Требуется** определить группы общения пользователей, которые имеют некоторые специальные свойства, например являются замкнутыми или иерархически организованными.

При этом искомые замкнутые группы должны обладать следующими свойствами: количество участников не превышает N ; время существования группы (между первым и последним контактом) не менее периода времени T ; количество контактов не менее M . Значения N , M и T должны задаваться пользователем системы.

Вначале данные о контактах заносятся в базу данных (БД), в которой описаны триггеры, вызывающие на выполнение компоненты, осуществляющие обработку этих данных.

Онтология задачи. Для решения поставленной задачи с использованием разработанного подхода использовалась базовая онтология описания ситуаций, предложенная в [2], на основе которой была разработана онтология коммуникации (рис. 1), расширяющая базовые онтологии ситуации новыми понятиями и отношениями.

В онтологию коммуникации были введены такие новые понятия, как: Subscriber (абонент, участник обмена телекоммуникационными сообщениями); Contact (контакт, история сеансов связи между абонентами); Group (группа). Для понятия Group заданы производные понятия IsolatedGroup и HierarchicalGroup, соответствующие интересующим видам групп абонентов. Абонент определяется внутри телекоммуникационной сети некоторым уникальным идентификатором (name). Для облегчения логического вывода ему задано литеральное булево свойство inGroup.

Контакт определяет связь двух абонентов и хранит в себе импортируемую с датчиков (из некоторой БД) информацию о сеансах связи между пользователями. Сеансы внутри контакта различа-

ются временем установки сеанса (свойство time). Также контакт хранит дату первого установления связи (firstContact) между пользователями для облегчения дальнейшего логического вывода.

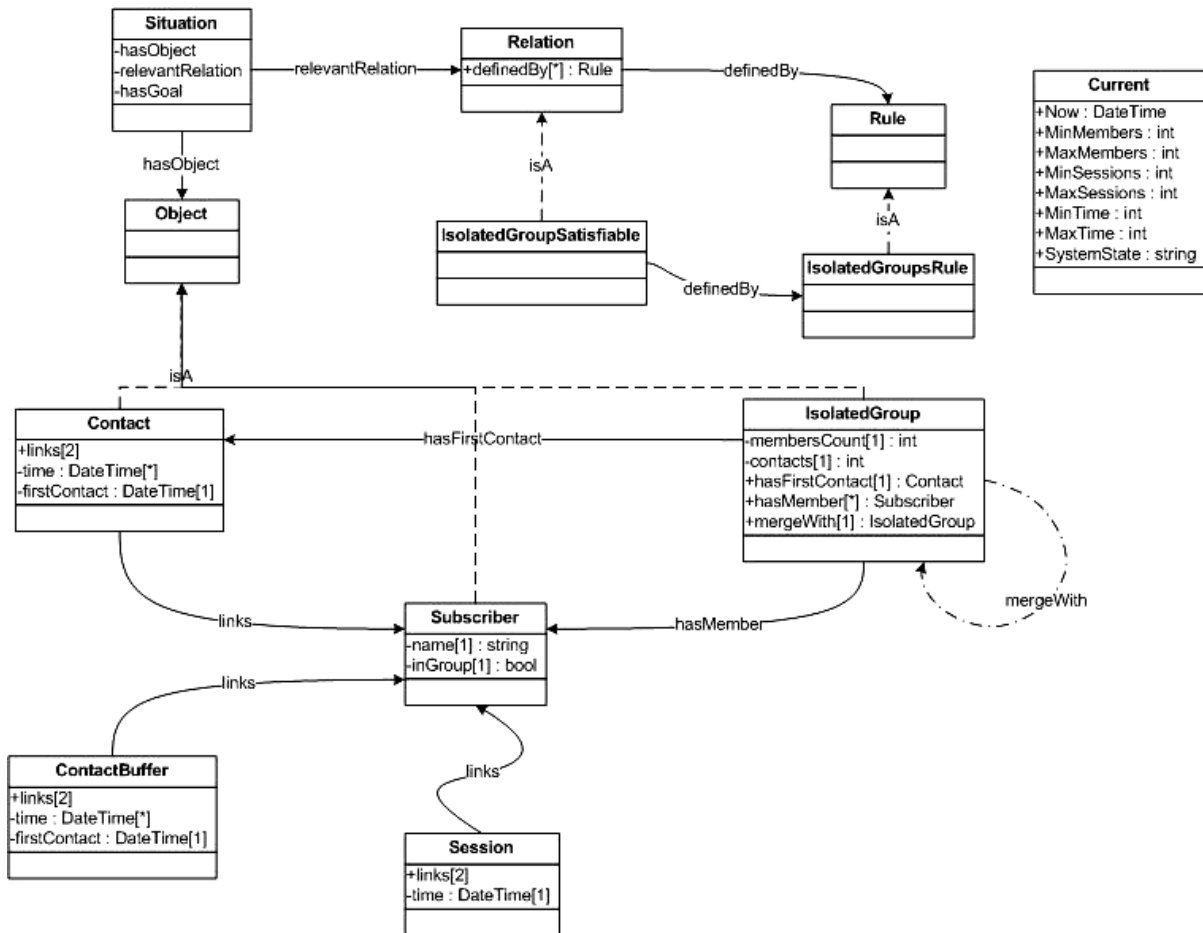


Рис. 1. Предлагаемая онтология коммуникации

Группа характеризуется, в первую очередь, входящими в неё пользователями (hasMember). Кроме того, у группы определено свойство mergeWith для указания слияния данной группы с другой группой. Для облегчения логического вывода также введены литеральные свойства: количество участников в группе (membersCount) и количество контактов между ними (contacts), а также объектное свойство firstContact, указывающее на первый контакт членов группы. Отслеживаемым системой отношением является GroupSatisfiable и, соответственно, его понятия – потомки IsolatedGroupSatisfiable и HierarchicalGroupSatisfiable.

Обработка поступающих данных о сеансах связи осуществляется компонентом системы Transceiver, отслеживающим изменения в БД, в которую поступают данные о контактах. Данный компонент при необходимости добавляет новые экземпляры Subscriber и Contact в онтологию, а также инициирует обработку правил логического вывода. Для инициализации логического вывода, обеспечивающего проверку отношений класс-подкласс GroupSatisfiable, был разработан специальный компонент Watcher, отслеживающий изменения в составе групп и контактов в онтологии.

Вспомогательными понятиями, заданными для минимизации алгоритмической обработки данных, являются Session, ContactBuffer и Current. Понятие Session является пулом экземпляров для представления загружаемых из БД компонентом Transceiver данных о контактах. Понятие ContactBuffer является пулом, сформированным на основе данных, хранящихся в экземплярах, а понятие Current хранит в единственном экземпляре общие переменные системы, такие как состояние обработки SystemState (хранит строку, равную либо 'Adding', либо 'Computing', либо 'Checking'), минимальное количество контактов в отслеживаемых группах и т.п. Они необходимы для разделения наборов правил для разных компонентов. Это связано с тем, что правила хранятся в онтологии и по умолчанию все применяются одновременно. Однако в рассматриваемой задаче требуется упо-

рядочное применение пакетов правил. После выполнения очередного набора правил при возвращении управления из метода, инициализировавшего вывод, происходит смена строки состояния, и вывод запускается снова. Для этого в начало каждого правила добавляется проверка соответствующего условия: к примеру, при внесении новых групп проверяется, равно ли `Current.SystemState` значению ‘Adding’, при пересчёте параметров групп – ‘Computing’, при проверке групп – ‘Checking’.

Правила логического вывода новых объектов и отношений. Вывод выполняется при помощи системы логического вывода Jess [6], позволяющей осуществлять логический вывод на основе набора продукционных правил, записанных на языке SWRL [5].

Для выявления требуемых групп и отслеживания изменения их параметров составлен набор продукционных правил. Пример правил, обеспечивающих выявление изолированных групп на языке SWRL, показан ниже. Правила упорядочены по событиям, происходящим в системе, которые вызывают их выполнение.

1. Событие: внесение массива новых данных о сеансах связи из БД в онтологию. Выполнение правил инициируется тем же компонентом, что переносит данные из БД в онтологию.

1.1. Добавить нового пользователя в группу:

```
IsolatedGroup(?group) ^ hasMember(?group, ?a) ^ inGroup(?b, false) ^ Contact(?c) ^ links(?c, ?a) ^ links(?c, ?b) -> hasMember(?group, ?b) ^ inGroup(?b, true)
```

1.2. Пометить две изолированные группы для слияния при обнаружении связи между их членами:

```
IsolatedGroup(?group1) ^ IsolatedGroup(?group2) ^ hasMember(?group1, ?a) ^ links(?a, ?b) ^ hasMember(?group2, ?b) -> mergeWith(?group1, ?group2)
```

1.3. Создать новую группу для двух не состоящих в группах связанных абонентов:

```
Contact(?c) ^ links(?c, ?a) ^ links(?c, ?b) ^ inGroup(?a, false) ^ inGroup(?b, false) ^ IsolatedGroup(?G) ^ membersCount(?g, ?n) ^ swrlb:equal(?n, 0) -> hasMember(?G, ?a) ^ hasMember(?g, ?b) ^ swrlb:add(?n, ?n, 2)
```

Следует отметить, что язык SWRL не позволяет создавать новые экземпляры, так что имеет смысл просто подготавливать пул пустых групп и добавлять в них новых пользователей.

1.4. Объединение двух помеченных для слияния групп (практически, хранимая процедура):

```
IsolatedGroup(?group) ^ IsolatedGroup(?group2) ^ mergeWith (?group1, ?group2) ^ hasMember(?group1, ?a) ^ contacts(?group1, ?n) -> hasMember(?group2, ?a) ^ swrlb:subtract(?n, ?n, 1) ^ processing(?a, true)
```

2. Событие: изменение состава изолированной группы. Выполнение правил инициируется отдельным компонентом, выполняющимся в отдельном процессе, который запускается через заданные интервалы времени.

2.1. Пересчёт количества контактов в группе:

а) сосчитать связи только что добавленных пользователей с состоявшими ранее в группе:

```
hasMember(?G, ?a) ^ hasMember(?G, ?b) ^ links(?c, ?a) ^ links(?c, ?b) ^ processing(?a, true) ^ processing(?b, false) ^ contacts(?G, ?n) -> swrlb:add(?n, ?n, 1)
```

б) сосчитать связи только что добавленных пользователей друг с другом:

```
hasMember(?G, ?a) ^ hasMember(?G, ?b) ^ links(?c, ?a) ^ links(?c, ?b) ^ processing(?a, true) ^ processing(?b, true) ^ contacts(?G, ?n) -> swrlb:add(?n, ?n, 1)
```

в) завершить подсчёт контактов:

```
processing(?a, true) -> processing(?a, false)
```

Отметим, что служебное литеральное свойство `processing`, не показанное на схеме, указывает на то, что пользователь в группу добавлен, но его контакты ещё не добавлены к числу контактов группы.

2.2. Проверка изолированной группы на соответствие условиям:

```
//выбираем оцениваемые параметры группы и кортеж свойства IsolatedGroupSatisfiable, проверяя его на значение – группы, уже подходящие под условие, не используются
IsolatedGroup(?g) ^ contacts(?g, ?m) ^ membersCount(?g, ?n) ^ hasFirstContact(?g, ?c) ^ firstContact(?c, ?t) ^ onObject(?tuple, ?g) ^ hasRelationValue(?tuple, ?value) ^ swrlb:equal(?value, false)
^
//забираем заданные пользователем значения и текущее значение времени
maxMembers(?current, ?N) ^ minContacts(?current, ?M) ^ maxTime(?current, ?T) ^ Time(?current, ?curtime) ^
//сравниваем
swrlb:subtract(?totaltime, ?curtime, ?t) ^ swrlb:greaterThan(?T, ?totaltime) ^ swrlb:greaterThan(?N, ?n) ^
```

`swrlb:greaterThan(?m, ?M) -> swrlb:equal(?value, true)`

2.3. Проверка выявленной ранее удовлетворяющей условиям изолированной группы на превышение максимального количества участников:

`IsolatedGroup(?g) ^ membersCount(?g, ?n) ^ onObject(?tuple, ?g) ^ hasRelationValue(?tuple, ?value) ^ swrlb:equal(?value, true) ^ maxMembers(?current, ?N) ^ swrlb:greaterThan(?n, ?N) -> swrlb:equal(?value, false)`

Таким образом, группы, более не соответствующие критериям (превышено максимальное количество членов или превышено максимальное время последнего контакта), также остаются в «поле зрения» системы и доступны пользователю для просмотра. Причина выхода группы из состава актуальных сохраняется в поле `rdf:Description`.

Эти наборы правил определяются в онтологии при помощи набора экземпляров `GroupsRule`, связанных отношением `definedBy` с экземпляром соответствующего класса (для `IsolatedGroup` – `IsolatedGroupSatisfiable`). При изменении пороговых значений `M`, `N` и `T` порождаются соответствующие им экземпляры правил (`GroupsRule`).

Апробация предложенного метода. Для апробации предложенного метода был разработан прототип приложения на языке `C#` в среде `Microsoft Framework 3.5`. На рис. 2 показан графический интерфейс данного прототипа.

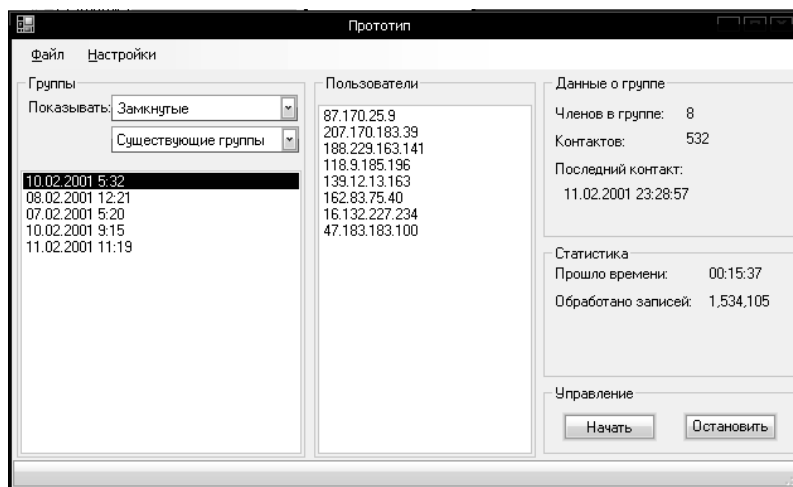


Рис. 2. Окно просмотра результатов работы системы – показаны выделенные группы

В проведенных вычислительных экспериментах проверялся поиск замкнутых групп. В качестве входных данных использовалась выборка реальных данных IP-трафика объемом более миллиона записей. В исходных данных замкнутых групп выявлено не было.

Для проверки работы алгоритма тестовых данных были внесены изменения путем добавления данных о сеансах связи, соответствующих 9 «контрольным изолированным группам» объемом от 5 до 20 участников. В результате обработки были выявлены все контрольные изолированные группы.

Заключение. Описанный метод и алгоритм выявления «замкнутых групп общения» могут быть сведены к набору продукционных правил-триггеров при использовании онтологии анализа ситуаций. Описанные структуры, несмотря на упрощенность модели, для которой они разрабатывались, позволяют обобщить правила для работы с ориентированными потоками информации (и, следовательно, с ориентированным графом), а также расширить возможности системы до обработки различных типов абонентов (к примеру, абонент, веб-ресурс, IRC-сервер и т.д.). Однако ряд описанных выше пороговых значений может быть определен только практическим путем на основании обучающих выборок реальных данных.

Литература

1. Allemang D. Semantic Web for the Working Ontologist: Modeling in RDF, RDFS and OWL / D. Allemang, J. Hendler. – N.Y.: Morgan Kaufmann Publishers, 2011. – 364 p.
2. An Application of Semantic Web Technologies to Situation Awareness / Christopher J. Matheus, Mieczyslaw M. Kokar, Kenneth Baclawski, Jerzy A. Letkowski [Электронный ресурс]. – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.1270&rep=rep1&type=pdf> (дата обращения: 14.02.2012).
3. Разин В.В. Анализ ситуаций с применением семантических технологий / В.В. Разин, А.Ф. Тузовский // Доклады ТУСУРа. – 2010. – № 2 (22), ч. 2. – С. 238–240.

4. Web Ontology Language OWL [Электронный ресурс]. – Режим доступа: <http://www.w3.org/2004/OWL/> (дата обращения: 14.02.2012).
 5. SWRL: A Semantic Web Rule Language Combining OWL and RuleML [Электронный ресурс]. – Режим доступа: <http://www.w3.org/Submission/SWRL/> (дата обращения: 14.02.2012).
 6. Jess: the Rule Engine for the Java Platform [Электронный ресурс]. – Режим доступа: <http://herzberg.ca.sandia.gov/> (дата обращения: 14.02.2012).
-

Грачев Владимир Викторович

Канд. техн. наук, вед. сотрудник ООО «Инком»

Тел.: +7-916-158-15-40

Эл. почта: income@cc.tpu.edu.ru

Разин Владимир Викторович

Аспирант каф. оптимизации систем управления Института кибернетики НИТПУ

Тел.: (382-2) 46-52-23

Эл. почта: moonbreeze@sibmail.com

Тузовский Анатолий Федорович

Д-р техн. наук, профессор каф. оптимизации систем управления Института кибернетики НИТПУ

Тел. (382-2) 42-14-85

Эл. почта: tuzovskyaf@tpu.ru

Grachev V.V., Razin V.V., Tuzovsky A.F.

An analysis of situations in telecommunication network by using of semantic technologies

In the paper, an application of Semantic Web technologies for situation analysis in telecommunication network is considered. A communication ontology and SWRL rules for identification of close communicated groups are suggested.

Keywords: ontology, OWL, situational control, telecommunication network, situation analysis.
