

УДК 004.042+004.457

Алгоритм замещения агентов dataflow-сети на платформе Smart-M3

Васильев А.М., Парамонов И.В., Лагутина Н.С., Мамедов Э.И. ¹

*Ярославский государственный университет им. П. Г. Демидова
150000 Россия, г. Ярославль, ул. Советская, 14*

*e-mail: vamonster@gmail.com, ilya.paramonov@fruct.org,
lagutinans@gmail.com, eldar.mamedov@list.ru*

получена 27 августа 2013

Ключевые слова: замещение агентов, dataflow-сеть, целостность информационных потоков, платформа Smart-M3

Рассматривается алгоритм замещения агента dataflow-сети, реализованной на платформе Smart-M3. Такое замещение позволяет перенести управление и контекст вычислений от преждевременно отключившегося агента к программируемому агенту-заместителю на время отсутствия первого агента в сети. При этом гарантируется целостность информационных потоков, то есть функционирование всех зависимых сервисов не нарушается при отключении агента. При возвращении агента в сеть происходит обратное замещение также с сохранением целостности всех информационных потоков.

Приведено описание реализации dataflow-сети и структуры механизма замещения агентов для платформы Smart-M3. Дано детальное описание алгоритма замещения, включающее процедуры инициализации, регистрации и двустороннего замещения агентов. Предложенный алгоритм замещения реализован авторами в механизме замещения в брокере семантической информации RedSIB на платформе Smart-M3.

Введение

Интернет вещей охватывает множество взаимодействующих элементов, соединённых каналами связи и предоставляющих различные сервисы конечным пользователям [1]. Достижение данной цели осуществляется за счёт эффективного сбора, преобразования и интерпретации собранных данных в соответствии с потребностями пользователя. Источниками информации в интернете вещей являются различные интеллектуальные устройства, в том числе сенсоры, встроенные в окружающие предметы («вещи»). Примерами таких устройств могут быть сенсоры движения

¹Работа выполнена при поддержке Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы». Контракт № 14.В37.21.0876.

и положения в посуде [2], сенсоры энергопотребления бытовой электроники [3] и устройства обнаружения присутствия людей в комнате на основе инфракрасного датчика [4]. Самыми распространёнными обрабатывающими устройствами в интернете вещей являются мобильные устройства (смартфоны и планшеты), а также бытовая электроника.

Многие устройства интернета вещей, например, смартфоны или сенсорные сети, имеют маломощные источники питания и связаны друг с другом ненадёжными беспроводными каналами. Данные факторы могут приводить к временному отсоединению устройства от сети или прекращению его функционирования, что может повредить работе зависимых компонентов сети, вызывая снижение качества предоставляемых сервисов или нарушение их целостности [5].

Основой для сервисов интернета вещей служит информация, поступающая от различных сенсоров и служб. Одной из удобных моделей для обработки данных от сенсоров и разработки приложений в данной парадигме представляется dataflow-сеть [6], состоящая из вычислительных агентов. Агенты получают информацию, обрабатывают её и передают результат вычислений следующим агентам.

В работе [7] нами была предложена идея замещения вычислительного агента dataflow-сети, основанная на платформе Smart-M3, являющейся промежуточным программным обеспечением для разработки приложений в рамках парадигмы интеллектуальных пространств [8]. Механизм замещения представляет собой специальный модуль одного из ключевых элементов платформы. Его задача — отслеживание состояния вычислительных агентов и автоматическая замена вышедших из строя агентов другими агентами (агентами-заместителями) с сохранением контекста обработки информации.

В данной работе приводится детальное описание разработанного нами алгоритма замещения активного агента в случае его добровольного или вынужденного отсоединения от сети, а также алгоритма обратного замещения, используемого при возвращении агента в сеть. Данные алгоритмы используются в реализации механизма замещения в брокере семантической информации на платформе Smart-M3.

1. Реализация dataflow-сети на платформе Smart-M3

Платформа Smart-M3 предоставляет инфраструктуру для создания приложений в соответствии с парадигмой интеллектуальных пространств, в том числе для интернета вещей [8]. Главным компонентом платформы является брокер семантической информации (Semantic Information Broker, SIB), предоставляющий агентам — «обработчикам знаний» (Knowledge processors, KP) — совместный доступ к информации. Данная информация хранится в SIB в соответствии с моделью RDF [9], использующей представление информации в виде троек вида — «субъект», «предикат», «объект».

Dataflow-сеть состоит из узлов, связанных между собой коммуникационными каналами, позволяющими осуществлять многоступенчатую обработку информации [6]. Модель агента dataflow-сети, используемая в настоящей работе, изображена на рис. 1.

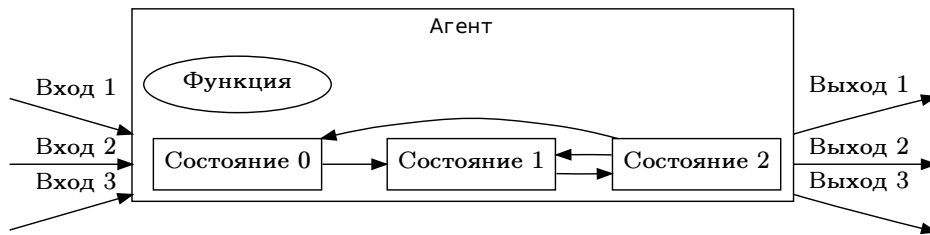


Рис. 1. Модель агента dataflow-сети

Обработка информации узлами сети осуществляется следующим образом. Каждый узел имеет некоторое внутреннее состояние, которое определяет контекст вычислений. Определение нового состояния осуществляется при поступлении информации по входным каналам на основе текущего состояния узла и полученной информации. При этом также вычисляется результирующее значение, которое затем передаётся по выходным каналам другим узлам сети.

Реализация агента dataflow-сети на платформе Smart-M3 подразумевает создание КР, следующего определённым правилам. Входные каналы агента реализуются посредством механизма подписки таким образом, что при изменении значений определённых троек SIB отправляет КР уведомление, содержащее новые значения троек. При получении уведомления КР осуществляет обработку принятых данных с учётом своего внутреннего состояния и обновляет информацию в интеллектуальном пространстве. Обновлению подлежат тройки, соответствующие выходным каналам (их состояние может отслеживаться другими агентами также с использованием механизма подписки), а также текущему состоянию агента.

2. Основная идея и компоненты механизма замещения

В основе предлагаемого подхода к обеспечению надёжности функционирования dataflow-сетей лежит идея постоянного слежения за состоянием подключения агентов к сети. В случае отключения агента от сети он временно замещается другим агентом, которому передаётся контекст вычислений. При возвращении агента в сеть контроль за вычислениями и контекст передаются в обратном направлении. Таким образом, замещение агентов производится без потери контекста вычислений и прозрачно для других элементов сети.

Основным элементом dataflow-сети является агент, выполняющий обработку поступающих данных и описываемый своим контекстом. Контекст вычислений агента включает в себя набор подписок (входные каналы), набор выходных троек (выходные каналы), набор троек, описывающих состояние агента, служебные тройки, описывающие контроль доступа, и программу обработки данных для агента.

заместителя. Все компоненты контекста вычислений присутствуют в SIB либо в виде троек, либо в виде внутренних данных.

В рамках нашей модели предполагаются две роли для вычислительных агентов: главный агент и агент-заместитель. Роль определяет особенности поведения агентов и взаимодействия с механизмом замещения. Первоначально в сети работают только главные агенты. При выходе из строя некоторых из них управление получают агенты-заместители.

Передачей вычислительного контекста между агентами управляет специализированный механизм. Для выполнения этой задачи ему необходимо эффективно координировать взаимодействие между агентами, управлять содержимым контекста и следить за подключением агентов к сети. Для успешного выполнения всех возложенных задач механизм реализован как часть брокера семантической информации.

3. Инициализация агентов dataflow-сети

Для функционирования в рамках механизма замещения агент должен разместить в интеллектуальном пространстве тройки экземпляра («individual» в терминах языка OWL — Web Ontology Language [10]) класса, описывающего его конфигурацию. Допустимыми классами являются *DataflowAgent* и *SubstituteDataflowAgent*. Описание свойств данных классов приведено ниже.

Класс *DataflowAgent* описывает конфигурацию главного агента. Свойство *Description* содержит URI документа, описывающего назначение агента, его программу по обработке данных. Данное свойство используется для идентификации агента внутри dataflow-сети механизмом замещения и должно быть уникальным для каждого главного агента. Свойство *SubstituteProgramType* определяет тип программы агента-заместителя. Свойство *SubstituteProgram* содержит программу агента-заместителя. Свойство *Active* указывает, активен агент (значение «yes») или нет (значение «no»).

Класс *SubstituteDataflowAgent* описывает конфигурацию агента-заместителя. Свойство *ProgramType* определяет тип программы, которую агент-заместитель может выполнять. Свойство *Substitutes* содержит либо ссылку на экземпляр класса, описывающий замещаемый агент, либо строковый литерал «None», который указывает, что агент в настоящий момент ничего не замещает. Свойство *Active* аналогично одноимённому свойству главного агента.

В последующем тексте для краткости изложения при ссылке на свойства агентов будут подразумеваться свойства соответствующих экземпляров класса конфигурации, описывающий данного агента.

Главный агент перед регистрацией должен проверить наличие экземпляра класса конфигурации в интеллектуальном пространстве. Если агент не обнаружил экземпляр класса, то он должен добавить настроенный экземпляр класса, в противном случае он должен следовать процедуре, описанной в секции 5. После регистрации главный агент должен установить защиту от внешнего изменения для экземпляра класса конфигурации, выходных троек и троек состояния. После этого он инициализирует входные каналы, устанавливая подписки на изменение необходимых троек. Если запросы подписок возвращают какие-либо данные, агент обрабатывает их и устанавливает выходные значения и состояние. После этого КР переходит в режим

ожидания оповещений механизма подписок, при получении которых ведёт себя в соответствии со спецификацией поведения агента dataflow-сети.

В отличие от главного агента агент-заместитель после установки защиты на экземпляр класса конфигурации устанавливает подписку на его свойство *Substitutes*. Механизм защиты устанавливает значение данного свойства для управления поведением агента-заместителя. Когда данное свойство изменяется, агент-заместитель получает оповещение, анализирует его содержание и выполняет соответствующее действие, описанное в следующих главах. После этого агент переходит в режим ожидания сообщений от механизма замещения.

Для отмены регистрации в механизме замещения агенты должны удалить экземпляры класса конфигурации из пространства и убрать защиту от изменения этих данных.

4. Процесс замещения агента

Диаграмма последовательности, описывающая процесс замещения активного агента агентом-заместителем, показана на рисунке 2. Процесс замещения начинается, когда агент теряет возможность правильно функционировать в случае разрыва TCP-соединения или поломки устройства. Ситуация отсоединения агента определяется либо брокером семантической информации, либо агентом самостоятельно. В первом случае сигналом для замещения служит поломка TCP-канала связи между КР и SIB. Во втором случае агент запрашивает замещение, изменяя свойство *Active* у собственного экземпляра класса конфигурации.

Процесс замещения агента начинается с выбора свободного агента-заместителя, способного выполнять программу, предоставленную главным агентом при регистрации. Выбранному агенту-заместителю предоставляются права на доступ к выходным тройкам и тройкам, описывающим состояние главного агента. Затем механизм замещения устанавливает ссылку на экземпляр класса конфигурации главного агента в качестве свойства *Substitutes* агента-заместителя.

Когда агент-заместитель получает оповещение подписки от SIB об изменении конфигурационного экземпляра, он запрашивает программу для замещения и подготавливает её для дальнейшего использования. После этого, в соответствии с полученной программой, он устанавливает подписки, которые совпадают с набором подписок вышедшего из строя агента. Когда агент заканчивает инициализацию, он сообщает об этом механизму замещения, устанавливая свойство *Active* агента-заместителя равным «yes».

С момента обнаружения ситуации отключения агента оповещения подписки по входным каналам помещаются во временное хранилище оповещений для предотвращения потери данных, передаваемых по входным каналам. Когда механизм замещения определяет, что агент-заместитель закончил инициализацию, он высылает оповещения, накопившиеся в модуле временного хранения, одно за другим. Накопитель выключается, когда все оповещения были высланы, и оповещения начинают приходить напрямую новому агенту. Агент-заместитель обрабатывает приходящие сообщения подписки с использованием полученной ранее программы.

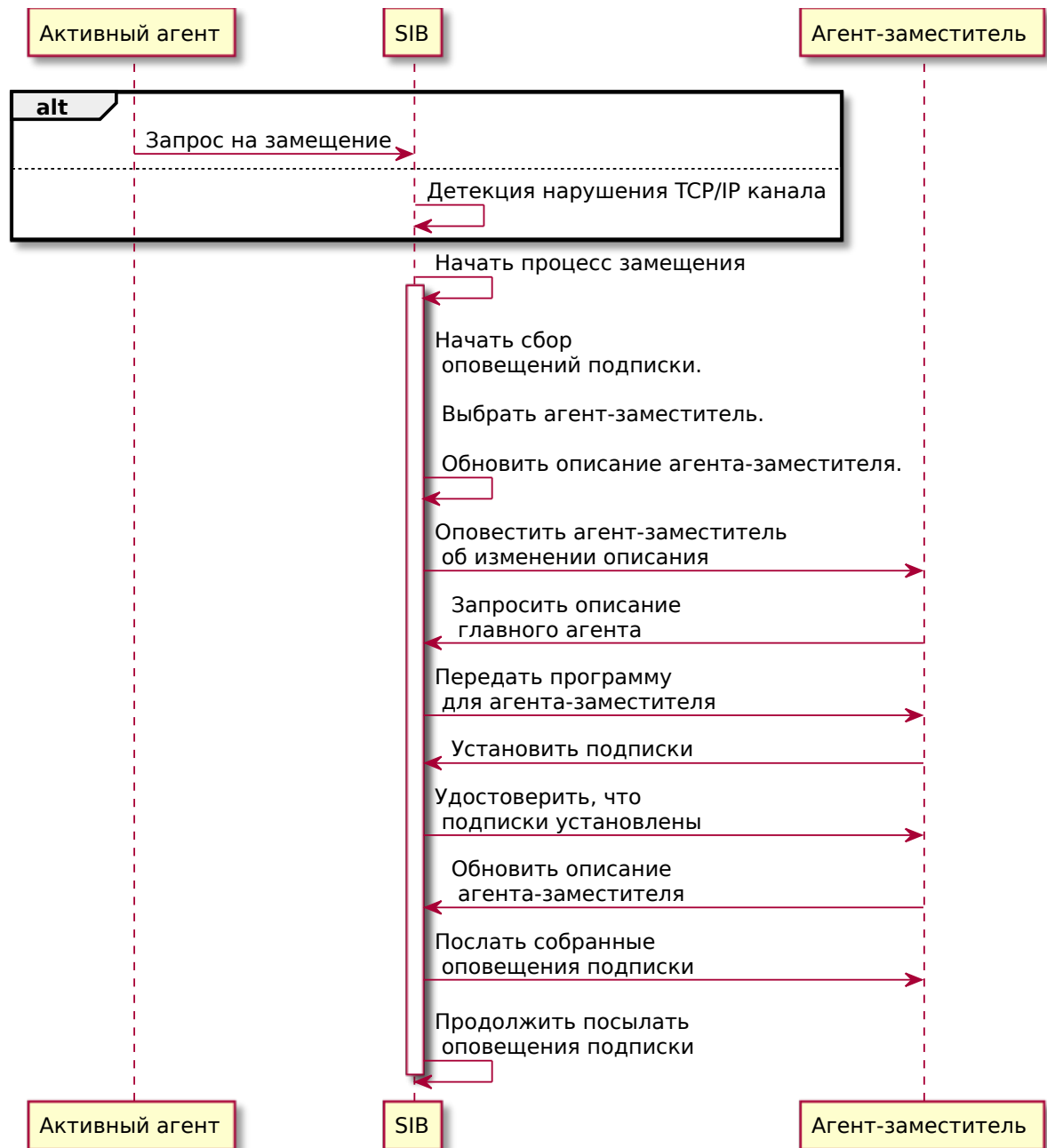


Рис. 2. Процесс замещения агента в случае отключения от сети

5. Обратное замещение агента-заместителя

Диаграмма последовательности, описывающая процесс обратного замещения, изображена на рис. 3. Данный процесс начинается, когда главный агент заново устанавливает связь с интеллектуальным пространством и обнаруживает экземпляр класса конфигурации, описывающий данный агент, что обозначает некорректное завершение предыдущей сессии.

В ходе обратного замещения главный агент сначала устанавливает подписки, соответствующие входным каналам. Следует отметить, что на данном этапе замещения вся информация, получаемая по каналам подписок, игнорируется главным агентом, так как она обрабатывается агентом-заместителем. Далее агент устанавливает свойство *Active* экземпляра класса конфигурации равным «yes», тем самым сообщая механизму замещения, что он готов обрабатывать данные.

Когда механизм замещения обнаруживает возвращение главного агента в сеть, он начинает процедуру отключения агента-заместителя. Сначала для всех активных входных каналов запускается механизм временного хранения оповещений подписки. Далее механизм устанавливает значение «None» свойству *Substitutes* агента-заместителя, сообщая о необходимости завершить свои операции. Когда агент-заместитель получает оповещение об изменении конфигурации, он отменяет подписку от входных троек и ожидает завершения внутренней обработки информации, если это необходимо. После завершения обработки данных и установки нужных выходных значений агент-заместитель устанавливает значение «no» свойству *Active* своего экземпляра класса конфигурации, чтобы обозначить завершение своих операций. Последним шагом он переходит в состояние ожидания следующей команды на замещение от механизма замещения.

После получения оповещения о прекращении функционирования агента-заместителя механизм замещения начинает посылать собранные сообщения подписок главному агенту и лишает отработавшего агента-заместителя прав на изменение выходных данных и троек, описывающих состояние, от отработавшего агента-заместителя. Когда все оповещения, сохранённые в модуле временного хранения, были отосланы, все последующие оповещения начинают напрямую посылаться главному агенту.

Заключение

В условиях изменчивого окружения интернета вещей, характеризуемого высокой степенью вероятности отключения источников информации и обрабатываемых элементов, обеспечение надёжности информационных потоков является критически необходимым для поддержания функционирования сервисов.

В данной работе представлен разработанный нами алгоритм замещения агентов, который позволяет временно заместить отключённый агент другим до момента возвращения отключённого агента в сеть. Во время процедуры замещения агент-заместитель получает контекст вычислений главного агента, что делает процедуру замещения прозрачной для зависимых сервисов и их пользователей.

Рассмотренный в работе алгоритм замещения агентов был успешно реализован в основе брокера семантической информации RedSIB [11]. В настоящий момент ведутся работы по включению изменений в исходные коды брокера. Дальнейшие направ-

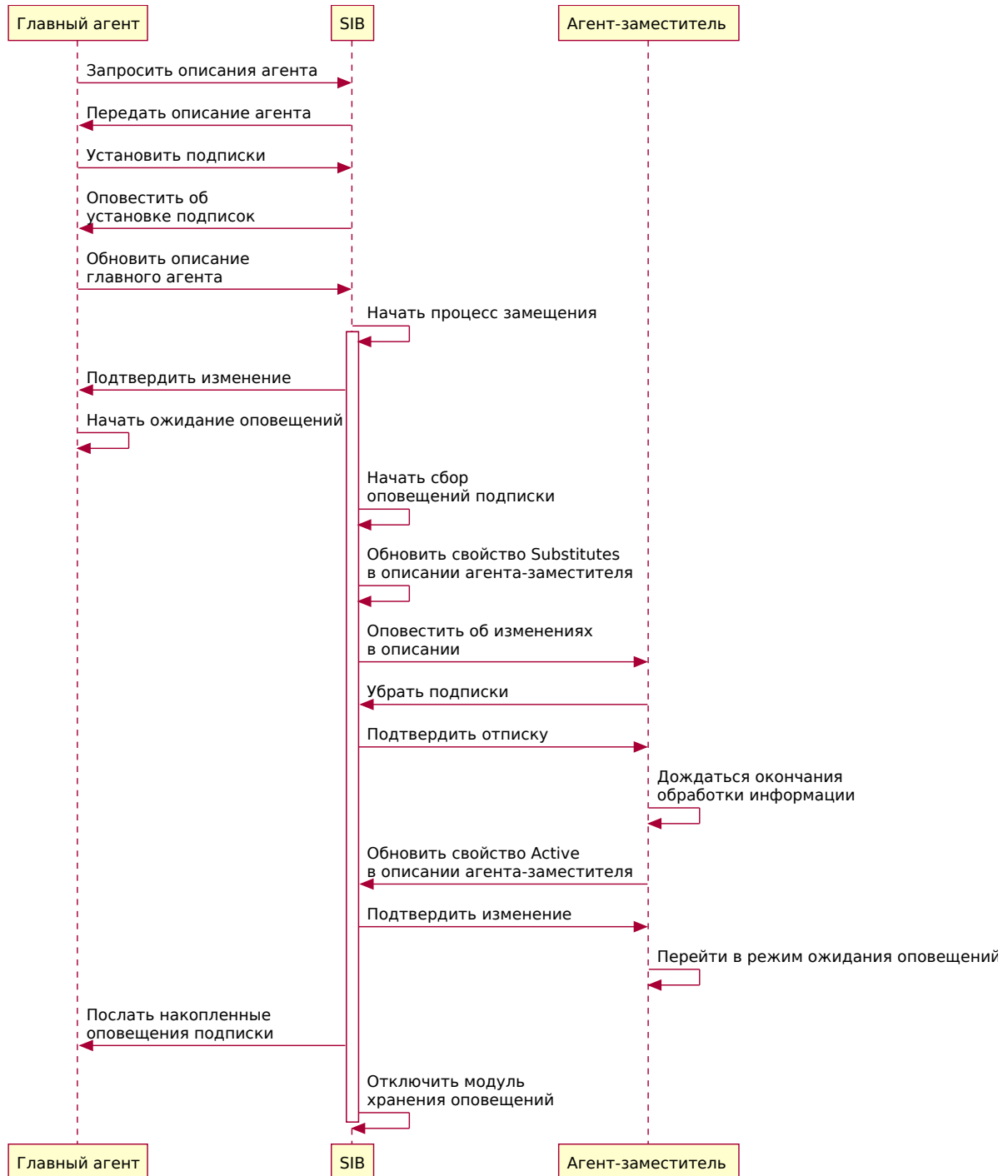


Рис. 3. Процесс обратного замещения агента

ления работы включают в себя исследование характеристик механизма замещения агентов, реализации прототипов сервисов, использующих предложенный механизм и анализа границ применимости данного механизма.

Список литературы

1. Atzori L., Iera A., Morabito G. The internet of things: A survey // *Computer Networks*. 2010. Vol. 54, No 15. P. 2787–2805.
2. Sensor-enabled rfid system for monitoring arm activity: Reliability and validity / Barman Joydip, Uswatte Gitendra, Ghaffari Touraj et al. // *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 2012. Vol. 20, No 6. P. 771–777.
3. The energy aware smart home / Marco Jahn, Marc Jentsch, Christian R. Prause et al. // *5th International Conference on Future Information Technology (FutureTech)*. 2010. P. 1–8.
4. Song J.-h., Hou S.-f. Infrared application in smart home system—based on intelligent air conditioning design // *Proceedings of 3rd International Asia Conference on Industrial Engineering and Management Innovation (IEMI2012)* / Ed. by Runliang Dou. Springer Berlin Heidelberg, 2013. P. 721–728.
5. Chen Y.-K. Challenges and opportunities of internet of things // *17th Asia and South Pacific Design Automation Conference (ASP-DAC)*. 2012. February. P. 383–388.
6. Synthesizing hardware from dataflow programs / Jörn W Janneck, Ian D Miller, David B Parlour et al. // *Journal of Signal Processing Systems*. 2011. Vol. 63, No 2. P. 241–249.
7. Vasilev A., Paramonov I., Balandin S. Mechanism for robust dataflow operation on smart spaces // *Proceedings of the 12th Conference of Open Innovations Association FRUCT and Seminar on e-Travel*. Oulu, Finland, November 5-9, 2012. St.-Petersburg : SUAI, 2012. P. 154–164.
8. Smart-M3 information sharing platform / J. Honkola, H. Laine, R. Brown, O. Tyrkkö // *IEEE Symposium on Computers and Communications (ISCC)* / IEEE. 2010. P. 1041–1046.
9. Klyne G., Carroll J., McBride B. Resource description framework (rdf): concepts and abstract syntax. 2004. February. URL: <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>.
10. Schneider M., Carroll J., Herman J., Patel-Schneider P. Owl 2 web ontology language rdf-based semantics (second edition). 2011. December. URL: <http://www.w3.org/TR/2012/REC-owl2-rdf-based-semantics-20121211/>.
11. RedSib: a Smart-M3 semantic information broker implementation / F. Morandi, L. Roffia, A. D’Elia et al. // *Proceedings of the 12th Conference of Open Innovations Association FRUCT and Seminar on e-Tourism*. Oulu, Finland. St.-Petersburg : SUAI, 2012. November. P. 86–98.

A Substitution Algorithm for Dataflow Network Agents on Smart-M3 Platform

Vasilev A. M., Paramonov I. V., Lagutina N. S., Mamedov E. I.

P.G. Demidov Yaroslavl State University, Sovetskaya str., 14, Yaroslavl, 150000, Russia

Keywords: agent substitution, dataflow network, information flow integrity, Smart-M3 platform

The paper presents an agent substitution algorithm for a dataflow network implemented on the Smart-M3 platform. Such a substitution allows to transfer control and computational context from an unexpectedly disconnected agent to a programmable substitute agent for the period of absence of the first agent in the network. It also guarantees integrity of the information flow, i.e. the functioning of all dependent services is not disrupted after the agent disconnection. When the agent returns to the network the reverse substitution occurs also with keeping integrity of the information flow.

The paper gives a description of the dataflow network implementation and substitution mechanism structure on the Smart-M3 platform. The detailed description of the substitution algorithm including initialization, registration, and bidirectional substitution phases is given. The proposed substitution algorithm was implemented by the authors in the substitution mechanism as a part of the RedSIB semantic information broker on the Smart-M3 platform.

Сведения об авторах:

Васильев Андрей Михайлович,

Ярославский государственный университет им. П.Г. Демидова,
аспирант каф. теоретической информатики.

Парамонов Илья Вячеславович,

Ярославский государственный университет им. П.Г. Демидова,
канд. физ.-мат. наук, доцент каф. компьютерных сетей.

Лагутина Надежда Станиславовна,

Ярославский государственный университет им. П.Г. Демидова,
канд. физ.-мат. наук, доцент каф. вычислительных и программных систем.

Мамедов Эльдар Интизамович,

Ярославский государственный университет им. П.Г. Демидова,
аспирант каф. компьютерных сетей