

# МЕДИАТОРЫ ОНТОЛОГИИ ДЛЯ ПОВСЕМЕСТНЫХ ВЫЧИСЛЕНИЙ

**Э.А. Бабкин,**  
профессор Нижегородского филиала ГУ-ВШЭ,  
babkin@hse.nnov.ru

**Н.В. Асеева,**  
старший преподаватель Нижегородского филиала ГУ-ВШЭ,  
zubov@hse.nnov.ru

*В статье рассматриваются архитектура и методы проектирования устройств семантической интеграции гетерогенных сетей датчиков, названных «Медиатор Онтологий». Для специализации таких устройств и ускорения разработки предлагается использовать модельно-ориентированный подход и принципы трансформации знаний на основе онтологий.*

## Введение

За последние годы миниатюризация, низкая стоимость компонентов и современные информационные технологии открыли практические возможности для проектирования, реализации, производства и развёртывания распределённых информационных систем, объединяющих тысячи датчиков и программное обеспечение для их мониторинга. Ожидается, что в ближайшее время пользователи будут ежедневно и повсеместно взаимодействовать с системами обработки и визуализации данных от датчиков во всех областях жизнедеятельности человека. Этот новый класс программно-аппаратных систем получил особое название – Ubiquitous Computing Environment (UCE, Вычислительная среда повсеместных вычислений) [1, 2] или даже «Интеллектуальное повсеместное пространство», Ubiquitous Smart Space [3, 4].

Многочисленные элементы UCE, использующие беспроводные соединения для обмена информацией, обеспечивают пользователю широчайшие возможности для получения информации от датчиков в реальном масштабе времени. Причём они позволяют пользователю получать информацию с учётом контекста использования и предпочтительных методов визуализации. В результате у человека появляются абсолютно новые возможности взаимодействия со средой обитания.

Много исследователей внесли свой вклад в дело развития теории и практики построения сетей датчиков и актуаторов, которые без сомнений являются основой UCE и носят обобщающее название SANET (Sensors and Actuators Networks). В результате сегодня доступны миниатюрные устройства, используемые для построения SANET, встраиваемые операционные системы, протоколы для беспроводных сетей и алгоритмы для эффективного управления энергопотреблением SANET [7,8,9,10].

В последнее время все более возрастает интерес к решению следующей важной проблемы, с которой столкнутся разработчики и пользователи UCE в недалеком будущем. Это – проблема совместного взаимодействия SANET, существующей IT-инфраструктуры и человеческого сообщества. Последние результаты [6, 11-15] показывают возможность применения подхода, основанного на использовании программного обеспечения промежуточного уровня (так называемого Middleware), для решения этих проблем, и предоставляют возможность интеграции SANET на прикладном уровне корпоративных систем. Этот подход позволяет использовать единый язык для взаимодействия различных частей распределённой системы в контексте общей модели мира. Однако в случае ширококомасштабных проектов по построению UCE большое количество SANET разворачивается в различных административных, корпоративных и социальных кругах. Этот простой

факт разрушает предположение «замкнутого мира» и требует повышенного внимания исследователей к таким аспектам, как явное представление мета-данных и формальное моделирование свойств и интерфейсов системы для достижения семантического взаимодействия распределенных компонент.

В ходе собственных научных исследований мы проанализировали пути улучшения существующих решений по интеграции различных компонентов SANET в составе UCE на основе middleware. Основой для наших собственных решений стала концепция медиатора — специального устройства, которое манипулирует мета-данными для выполнения «гладкой» интеграции различных информационных моделей (концепций, атрибутов и взаимосвязей реального мира) с целью удовлетворения информационных нужд отдельного пользователя или небольшой группы пользователей. Полученное решение носит название «Медиатор Онтологий», что отражает взаимосвязь нашего подхода с другими проектами по интеграции на основе медиаторов, в частности — с подходом «Mediator-wrapped Architecture» [5]. Некоторые важные части нашего исследования соприкасаются также с работами по интеграции знаний и инженерии онтологий в области проникающих вычислений [18, 19, 20].

Нужно отметить, что практическая реализация решения «Медиатор Онтологий» в глобальном контексте UCE невозможна без применения согласованной методологии разработки алгоритмов медиации и реконфигурируемой программно-аппаратной архитектуры устройств медиации. При этом мы разделяем мнение в том, что представление профессиональных знаний по разработке программного обеспечения, его преобразование и использование для автоматического проектирования сложных систем должны проводиться с применением формальных методов построения и анализа иерархии объектно-ориентированных моделей и онтологий.

В этой работе мы представляем наиболее важные характеристики предлагаемой методологии разработки и специализации устройств «Медиатор Онтологий», а также основные функции оригинальной среды разработки IRWIN-NG. Все эти компоненты могут использоваться для быстрой разработки высоко специализированных решений «Медиатор Онтологий» для UCE.

### Что такое онтологии?

Широкое распространение распределённых и агентно-ориентированных методологий разработки интеллектуальных систем привело к необходимости дальней-

ших исследований в области эффективного представления знаний о предметной области. При этом фокус внимания исследований сместился с понимания высказываний на естественном языке к выработке общих принципов обмена разнородными данными и знаниями о разных предметных областях между слабо связанными программными сущностями (например, агентами). Наиболее важными, таким образом, стали вопросы интеграции и совместного использования знаний. В этом контексте эксперты в области ИИ, в частности Том Грубер, предлагают обозначить формализмы представления и концептуализации понятий новым термином — онтологией. Под онтологией в искусственном интеллекте понимается формальное описание понятий (концептов) их взаимосвязей и других сущностей, которые существуют в предметной области и интересны для отдельного агента, или их сообществ. Это определение отличается от привычного философского значения термина онтология, где он обозначает систематическое описание всего существующего.

С точки зрения программных систем существует лишь то, что может быть представлено в формальной форме. Когда знания о предметной области представляются с помощью некоторого декларативного формализма, то множество сущностей, которые можно представить называются универсумом общения (the universe of discourse). Это множество сущностей и описанные взаимосвязи между ними имеют отражение в словаре, с помощью которого в программе представляются знания. Таким образом в контексте ИИ мы можем описать онтологию программы явно выразив термины представления. В такой онтологии определения связывают имена сущностей в универсуме общения (например, классы, отношения, функции или другие понятия) с текстовыми описаниями в форме, понятной человеку, дающими информацию о том, что означают имена, а также с формальными аксиомами, которые определяют ограничения на возможные интерпретации структур из введенных понятий. Формально онтология является выражением некоторой логической теории.

С прагматической точки зрения онтология определяет некоторый терминологический словарь; в его терминах выражаются запросы и утверждения, которыми обмениваются между собой различные агенты в ходе взаимодействия. Введенные в онтологии аксиомы выражают общепринятые соглашения по использованию единого словаря когерентным и согласованным образом. Отметим, что агенты, совместно использующие одну и ту же онтологию, не обязательно должны обладать совпадающей базой знаний (фактическими знаниями). Каждый из них может владеть знаниями, неизвестными другим, или не иметь

возможности отвечать на все вопросы, которые можно сформулировать на основании онтологии.

Сегодня онтологии широко применяются не только для теоретических исследований в ИИ. Они также применяются при создании информационных систем на этапах сбора требований и концептуального проектирования, объектно-ориентированном дизайне и разработке, компонентном программировании, разработке пользовательского интерфейса и т.п..

Можно отметить ряд положительных эффектов разработки и использования на деятельность распределенных организаций со сложной структурой. Прежде всего процесс онтологического анализа является процессом открытий и служит лучшему пониманию предметной области. Решения, полученные в ходе разработки онтологии, полезны для:

- ◆ определения проблем (диагностика);
- ◆ определения причин возникновения проблемы (причинный анализ);
- ◆ определения альтернативных методов решения проблемы;
- ◆ выработки единой точки зрения на стоящие задачи всеми членами группы;
- ◆ совместного использования и повторного использования знаний.

После завершения разработки онтологий они полезны

- ◆ в ходе разработки информационной системы, обеспечивая основу для интеграции с другими системами;

- ◆ для координации и контроля за различными процессами производства;
- ◆ для бизнес-реинжиниринга, позволяя определять наиболее важные элементы организационной структуры, которые нужно реструктурировать, а также последовательность действий по реструктуризации.

### Описание методологии разработки

На рис. 1 проиллюстрированы общие принципы предложенной методологии дизайна специализуемых устройств. В нашем подходе спецификация онтологии предметной области предшествует созданию нового устройства, реализующего функции «Медиатор Онтологий».

Эта онтология включает описание структуры и поведения пользовательских знаний о предметной области, представленных в форме соответствующих UML-концепций (классов, автоматов состояний и логических ограничений). Процесс разработки онтологии представляет собой диалог между основными заказчиками и инженером по знаниям. Для выявления концепций и взаимосвязей инженер может использовать различные инструменты UML-моделирования общего назначения или редакторы онтологий, а также специфические графические инструментальные среды для конкретной предметной области.

Как только пользователи определяют содержание и логические ограничения онтологии, их преобразуют

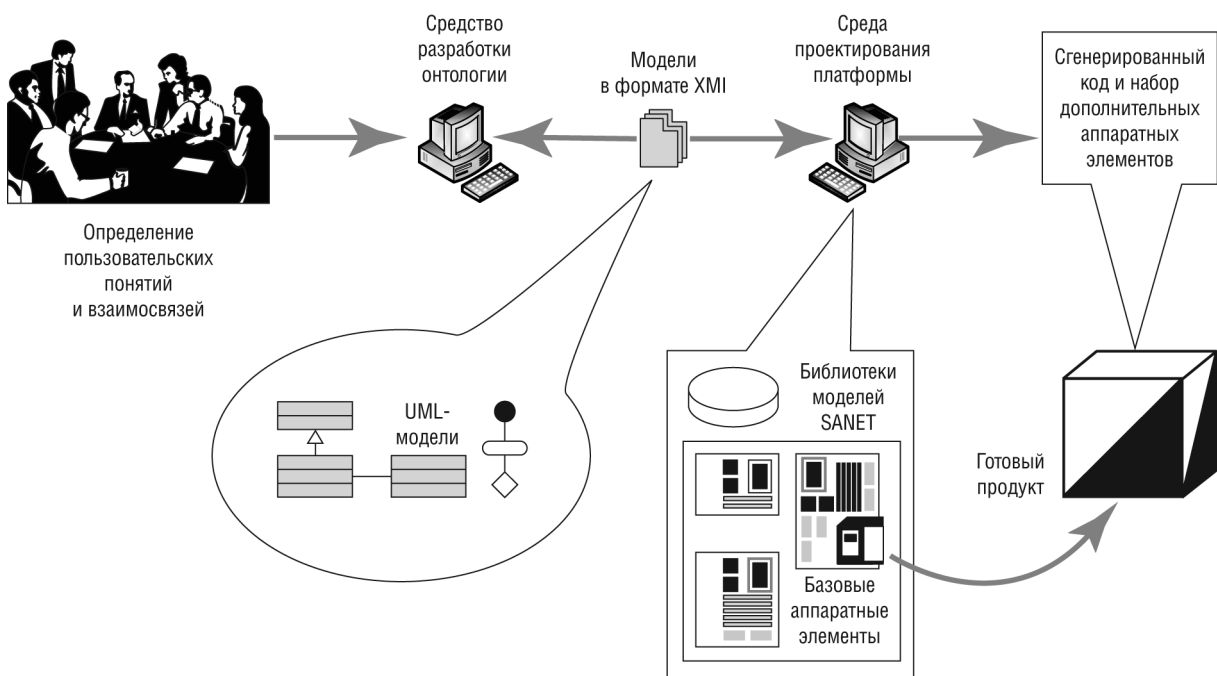


Рис. 1. Этапы проектирования и реализации в процессе создания устройства "Медиатор Онтологий"

в переносимый формат XMI для дальнейшего использования другими инструментами. Созданный XMI-документ передается в среду проектирования платформы (Platform Design Studio), которая является основным инструментом работы дизайнеров аппаратной платформы. Их первая задача заключается в создании платформно-зависимых UML-моделей, определяющих программные и аппаратные компоненты создаваемого устройства «Медиатор Онтологий». Для успешного выполнения этой задачи среда проектирования платформы предоставляет дизайнеру специализированный графический редактор, симулятор и библиотеку моделей SANET. Графический редактор помогает разработчику платформы в определении правил отображения между пользовательскими онтологиями и моделями SANET. Симулятор тестирует корректность отображения и помогает получить оценки производительности. Как только платформно-зависимая модель была создана, разработчик платформы запускает генератор артефактов, который генерирует исходный код, список дополнительных аппаратных элементов и рекомендации для выбора подходящих базовых аппаратных модулей. В процессе окончательной компоновки разработчики решения «Медиатор Онтологий» создают небольшую часть связующего кода, а разработчики аппаратного обеспечения добавляют к базовым модулям дополнительные аппаратные элементы. Этот этап завершает процесс создания специализированного решения «Медиатор Онтологий» и готовый к использованию продукт поставляется заказчику.

### Текущие результаты

Изучение нескольких предметных областей, предпочтительных для использования «Медиаторов Онтологий», определило основные требования, которые должны быть учтены при разработке архитектуры нашего решения. Аппаратное устройство Медиатора должно иметь такие же свойства, какие имеют компоненты, используемые для построения сетей датчиков: низкая стоимость, дружелюбный интерфейс для развертывания и инсталляции. Это позволяет сохранить уникальные особенности сетей датчиков и размещать «Медиатор Онтологий» в различных местах в соответствии с изменяющимися потребностями конечного пользователя. В то же время наше устройство должно обеспечивать широкий диапазон сценариев медиации, и, по нашему мнению, аппаратное обеспечение принципиально должно поддерживать связь посредством большинства популярных беспроводных или проводных протоколов. Удовлетворение последнего требования приводит

к большому количеству вариантов архитектуры аппаратуры и программного обеспечения. В соответствии с предлагаемой методологией разработки, управляемой моделями (Model-Driven Architecture) мы создали оригинальную среду разработки IRWIN-NG для эффективной поддержки процесса проектирования и разработки специализированных устройств «Медиатор Онтологий» по требованиям различных типов пользователей. Специально разработаны два аппаратных модуля, являющихся ядром этой среды.

1. Вычислительный Модуль (Processing Unit) – немодифицируемая часть устройства «Медиатор Онтологий», состоящая из процессора и основных интерфейсов для обмена данными.

2. Плата Расширений (Extention Board) – специализируемая в процессе разработки плата, которая может быть расширена различной аппаратурой и дополнительными аппаратными элементами в соответствии со специфическими требованиями.

Были изучены различные дизайнерские решения, созданы прототипы для нахождения наиболее подходящего набора функций и стоимости. Текущая экспериментальная реализация Вычислительного Модуля имеет следующие свойства:

- ◆ MCU: LPC2294 16/32 bit ARM7TDMI-S™t with 256K Bytes Program Flash, 16K Bytes RAM, 4x 10 bit ADC, 2x UARTs, 4x CAN, I2C, SPI, up to 60MHz operation;
- ◆ 4MB SRAM 4x K6X8008T2B-F/Q SAMSUNG;
- ◆ 4MB TE28F320C3BD90 C3 INTEL FLASH;
- ◆ 10Mb TP Ethernet (CS8900A);
- ◆ OS: RTOS ECOS 2.0;
- ◆ Connectors: Power supply, RJ45, HDR26F (for connection to the extension board).

Алгоритмы медиации, внутреннее управление и функции взаимодействия были написаны на языках C и C++. В настоящий момент завершен первый этап портирования виртуальной машины языка Java на Вычислительный модуль. За счёт этого устройство «Медиатор Онтологий» имеет доступ к внешним базам данных через интерфейс JDBC. В наших дальнейших планах реализация на языке Java таких дополнительных возможностей, как :

- ◆ специализированное ядро трансформирования онтологий с поддержкой языка RDF;
- ◆ динамическая реконфигурация и адаптация алгоритмов медиации (платформа мобильных агентов).

В экспериментах со средой разработки IRWIN-NG наше внимание изначально было сконцентрировано на алгоритмах медиации. Поэтому использовались проводные сенсоры чтобы снизить стоимость

реализации. Большинство экспериментов с алгоритмами медиации выполнено на базе двухпроводных сетей датчиков, напрямую подключенных к устройству «Медиатор Онтологий» через интерфейс RS485. Первая тестовая сеть SANET, имеющая топологию кольца, состоит из 70 датчиков и актуаторов, подключённых к модели железной дороги. Алгоритмы, реализованные в этой специализации устройства «Медиатор Онтологий», представляют пользователю возможность наблюдать в динамике состояние железной дороги в терминах оператора и позволяют выполнять основные операции при помощи дружественного web-интерфейса (Java applets).

Вторая сеть SANET имеет топологию звезды с восемью лучами. К каждому лучу подключается до 30 датчиков. Сеть была развернута на площадке крупного оператора телефонной связи для контроля в условиях эксплуатации телекоммуникационного оборудования. В этом случае устройство «Медиатор Онтологий» обеспечивает оператора информацией об условиях эксплуатации, а также прогнозирует снижение производительности технологического оборудования. Для реализации этих специфических особенностей специализированное устройство «Медиатор Онтологий» включает плату расширения, оснащённую жидкокристаллическим монитором и клавишами управления, разработанные модели и алгоритмы изоморфизма понятий низкого уровня (значения температуры, влажности, концентрация газов и т.п.) и понятий высокого уровня абстракции, которые использует оператор.

К настоящему моменту разработанные Платы Расширений поддерживают как проводные, так и беспроводные протоколы: I2C, CAN, 8x30 сенсоров, а также слот для установки беспроводного трансивера.

Полученные нами решения и накопленный опыт позволяет провести обобщение разработанной архитектуры Платы Расширения и её перепроектирование с использованием концепций эволюционирующего аппаратного обеспечения (evolvable hardware paradigm). В результате перепроектирования Плата Расширения будет включать в себя полностью перенастраиваемый и эволюционирующий блок на основе FPGA и несколько слотов для установки драйверов и трансиверов.

## Заключение

В этой работе мы рассмотрели принципы семантического взаимодействия различных компонент распределенной системы в контексте UCE. Авторами предложена концепция специализируемого устройства «Медиатор Онтологий», назначение которого — сбор данных, их трансформирование и представление в различных формах, соотнесенных с понятиями реального мира.

В сравнении с известными подходами к созданию промежуточного ПО для сетей датчиков, новое решение использует механизм онтологий, работает на уровне мета-данных и служит связующим звеном между объектно-ориентированными моделями. К тому же, имея большой набор интерфейсов с различными коммуникационными протоколами, устройство «Медиатор Онтологий» может быть использовано для координации взаимодействия гетерогенных Служб, в том случае, когда распределенная система состоит из различных сетей датчиков и приложений корпоративного уровня. Возросший коэффициент повторного использования кода и проектных решений является еще одним привлекательным свойством нашего подхода. В процессе создания очередного специализированного варианта устройства «Медиатор Онтологий» разработчик формализует знания о конкретной предметной области, а также о паттернах поведения и потребностях различных пользователей. Это позволяет постоянно расширять библиотеку доступных моделей, и, кроме того, разработчики и команды разработчиков могут совместно использовать эти библиотеки. В то же время применение одной метамодели для создания различных моделей позволяет упростить трансформацию и интеграцию моделей.

В дополнение к использованию таких же принципов семантической интеграции на основе общей онтологии, какие были предложены в [18], мы предлагаем оригинальную методику разработки и среду разработки IRWIN-NG, которые вместе позволяют реализовать решения различного масштаба, а также выполнять имитационные эксперименты по трансформации онтологий и предварительное тестирование. ■

*Данная работа выполнена при частичной финансовой поддержке Научного Фонда НФ ГУ-ВШЭ, проект № 2006–09.*

## Литература

1. Abowd G.D., Mynatt E.D. Charting Past, Present, and Future Research in Ubiquitous Computing // ACM Transaction on Computer-Human Interaction, vol. 7, No 1, pp.29–58, 2000.

2. Niemelä E., Latvakoski J. Survey of requirements and solutions for ubiquitous software // Proc. of the 3rd international conference on Mobile and ubiquitous multimedia, College Park, Maryland, 2004, pp. 71–78.
3. Jeng T. Designing a Ubiquitous Smart Space of the Future: The Principle of Mapping // Proc. of International Conference of Cognition and Computation (DCC '04), MIT, Cambridge, 19–21 July, 2004.
4. Kawahara Y., Minami M., Saruwatari S., Morikawa H., and Aoyama T. Challenges and Lessons Learned in Building a Practical Smart Space // Procs. of the 1st Annual International Conference on Mobile and Ubiquitous Systems:Networking and Services (MobiQuitous '04), Boston, Massachusetts, August 22–25, 2004 , pp. 213–222.
5. Wiederhold G. Mediation in Information Systems, ACM Computing Surveys, vol. 27, No 2, June 1995, pp. 265–267.
6. Branch J. W., Davis J., Sow D., and Bisdikian C. Sentire: A Framework for Building Middleware for Sensor and Actuator Networks // Proc. of the 1st IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing PerSeNS'05), Kauai Island, Hawaii, March 8–12, 2005, pp. 396–400.
7. Misc.Tinyos:A component-based os for the networked sensor regime. <http://webs.cs.berkeley.edu/tos/>.
8. Feng J., Koushanfar F., Potkonjak M. System-Architectures for Sensor Networks Issues, Alternatives, and Directions // Proc. of the IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD'02), Rio de Janeiro, Brazil, September 16–18, 2002, pp. 226.
9. Tilak S., Abu-Ghazaleh N., Heinzelman W. A Taxonomy of Wireless Micro-Sensor Network Models. ACM Mobil Computing and Communications Review, vol. 6, No.2, 2002, pp. 28–36.
10. Crossbow Technology, Inc., In <http://www.xbow.com/Products/products.htm>.
11. Curino C., Giani M., Giorgetta M., Giusti A., Murphy A.L., and Picco G.P. Tiny LIME :Bridging Mobile and Sensor Networks through Middleware // Proc. of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005), Kauai Island, Hawaii, March 8–12, 2005. pp. 61–72.
12. Tsetsos V., Alyfantis G., Hasiotis T., Sekkas O., and Hadjiefthymiades S. Commercial Wireless Sensor Networks: Technical and Business Issues // Proc. of the Second Annual Conference on Wireless On-demand Network Systems and Services (WONS'05), St. Moritz, Switzerland, January 2005, pp. 166–173.
13. Ahamed S.I., Vyas A., Zulkernine M. Towards Developing Sensor Networks Monitoring as a Middleware Service // in Proc. of the 2004 International Conference on Parallel Processing Workshops (ICPPW '04), Montréal, Québec, Canada, August 15–18, 2004, pp. 465–471.
14. Tokunaga E., Zee A. van der, Kurahashi M., Nemoto M., and Nakajima T. A Middleware Infrastructure for Building Mixed Reality Applications in Ubiquitous Computing Environments // Procs. of the 1st Annual International Conference on Mobile and Ubiquitous Systems:Networking and Services (MobiQuitous '04), Boston, Massachusetts, August 22–25, 2004 , pp. 382–391.
15. Chan E., Bresler J., Al-Muhtadi J., and Campbell R. Gaia Microserver:An Extendable Mobile Middleware Platform // Proc. of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005), Kauai Island, Hawaii, March 8–12, 2005. pp. 309–313.
16. Oliver. Applying UML and MDA to Real Systems Design. // Proc. of the conference on Design, Automation and Test in Europe (DATE'05), Munich, Germany, 7–11 March, 2005, vol. 1, pp.70–71.
17. Volgyesi P., Ledeczi A. Component-Based Development of Networked Embedded Applications // Proc. of the 28th Euromicro Conference (EUROMICRO'02), Dortmund, Germany, 4–6 September, 2002, pp. 68.
18. Chen H., Perich F., Finin T., and Joshi A. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications // Procs. of the 1st Annual International Conference on Mobile and Ubiquitous Systems:Networking and Services(MobiQuitous '04), Boston, Massachusetts, August 22–25, 2004, pp. 258–267.
19. Zhou Y., Zhao Q., Perry M. Reasoning over Ontologies of On Demand Service // Proc. of 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05), Hong Kong, 29 March-1 April, 2005, pp. 381–384.
20. Hönle N., K?ppeler U.-P., Nicklas D., Schwarz T., Grossmann M. Benefits of Integrating Meta Data into a Context Model // Proc. of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005), Kauai Island, Hawaii, March 8–12, 2005, pp. 25–29.