

Ю. Вальченко, А.М. Кашевник

СОВРЕМЕННЫЕ ПОДХОДЫ К ПОСТРОЕНИЮ КОНТЕКСТНО-ОРИЕНТИРОВАННЫХ СИСТЕМ В ИНТЕЛЛЕКТУАЛЬНЫХ ПРОСТРАНСТВАХ

Вальченко Ю., Кашевник А.М. Современные подходы к построению контекстно-ориентированных систем в интеллектуальных пространствах.

Аннотация. На сегодняшний день различные информационные технологии все больше и больше становятся неотъемлемой частью повседневной жизни человека. Офисы, квартиры, транспорт, дороги общего пользования и т.п., оснащаются всевозможными сенсорами, датчиками, камерами и др. приборами для наблюдения за изменяющейся окружающей средой вокруг них. Целью внедрения таких технических средств является обеспечение для человека более удобного и прозрачного использования техники и различных устройств в рамках его окружения. Такие окружения принято называть интеллектуальными пространствами. Действия человека фиксируются в контексте интеллектуального пространства, в котором он находится, и обрабатываются различными системами, использующими этот контекст, для того чтобы более адекватно реагировать на взаимодействие человека с техническими устройствами. В данной работе представлен анализ современных контекстно-ориентированных систем, в ходе которого были рассмотрены их архитектуры, проанализированы достоинства и недостатки рассмотренных архитектур, и выведены общие принципы их построения.

Ключевые слова: интеллектуальные пространства, семантический веб, распределенные системы управления, онтологии.

Valchenko J., Kashevnik A.M. Modern Approaches to Context-Based Systems Developing in Smart Spaces.

Abstract. Today different information technologies increasingly become an integral part of everyday life. Offices, apartments and vehicles are equipped with all sorts of sensors, cameras and other devices to monitor the changing environment around them. The goal of space outfit is to provide more convenient and transparent usage of devices in this space. These environments are named smart spaces. Human actions are recorded in the context of smart space around the person and processed by different informational systems that use this context in order to more adequately respond to human interaction with technical devices. This paper presents an analysis of modern context-oriented systems. Their architectures were considered, examined their advantages and disadvantages, and derived general principles for their construction.

Keywords: smart spaces, semantic web, distributed management systems, ontologies.

1. Введение. Современные исследования в области информационных и телекоммуникационных технологий нацелены на оптимизацию взаимодействия большого количества различного рода устройств с объектами реального мира путем объединения физических устройств и вычислительных ресурсов с использованием интеллектуальных пространств. Под интеллектуальным пространством понимается инфра-

структура, обеспечивающая возможность получать и применять знания об окружающей среде и ее участниках для совершенствования взаимодействия последних в рамках данной среды. На сегодняшний день все большую популярность приобретает оснащение домов, офисов, аудиторий, транспорта и т.д. сенсорами, датчиками, стационарными компьютерами, мобильными устройствами для сбора и обработки информации о местонахождении пользователей и их действиях. Информация, которая может быть использована, чтобы охарактеризовать ситуацию, в которой находится в данный момент некоторый объект (мобильное устройство, датчик, пользователь и т.п.), и полученная от этого объекта информация, называется контекстом. Сужение объемов анализируемой информации, поступающей от различных устройств из пространства, в котором работает информационная система, возможность быстрого реагирования системы на события, происходящие в пространстве, легкая масштабируемость сервисов системы и т.д. достигается за счет использования в данных системах онтологической модели контекста. Интеграция в такой контекст информации, получаемой от разнородных источников, делает его моделью, которая описывает текущую ситуацию в явном виде и содержит знания, на основании которых может быть сгенерировано множество возможных решений [19]. Системы, построенные в соответствии с таким подходом, называются контекстно-ориентированными.

Информационные системы, работающие в окружении различного рода устройств должны быть контекстно-ориентированными для возможности их быстрого реагирования на быстро меняющиеся условия окружающей среды, в которой функционируют пользователи [2]. Динамический характер интеллектуальных пространств создает ряд проблем в развивающихся контекстно-ориентированных информационных системах: поиск нужных сервисов интеллектуальных пространств, контекстно-управляемая композиция сервисов различных интеллектуальных пространств, восприятие сервисами контекста, адаптация систем к изменяющемуся во времени контексту, распространение контекстной информации сервисам и (или) компонентам систем [3]. Построение данных систем определяется несколькими ключевыми принципами: формализованное представление информации, гарантированный доступ к контексту, логический вывод контекста, масштабируемость системы и ее децентрализация.

Формализованное представление информации. Данные, поступающие с устройств разного рода из информационного окружения, должны быть представлены таким образом, чтобы все участники ин-

формационного взаимодействия могли свободно их использовать, т.к. они не могут быть обработаны приложениями без “понимания” их контекста. Интероперабельность интеллектуальных пространств требует явного представления семантики информации, циркулирующей в информационном окружении.

Гарантированный доступ к контексту. Интеллектуальное пространство аккумулирует контекст различных устройств и для приложений необходимо обеспечение выборочного доступа к разным частям данного контекста в любой промежуток времени.

Логический вывод контекста. Контекстно-ориентированные информационные системы, обрабатывая контекст с различных устройств, датчиков и сенсоров в информационном окружении, выводят информацию об окружении пользователя и его активности. Например, “Что происходит в комнате?”, “Чем занимается пользователь N?” и т.п.

Представление контекста – важная часть архитектуры интеллектуальных пространств, потому как контекстно-ориентированные приложения должны уметь быстро адаптироваться к изменениям окружающей среды, отраженным в контексте устройств из её окружения.

Масштабируемость системы вытекает из потребности пользователя заменять одни физические устройства в интеллектуальном пространстве на другие, добавлять новые и исключать старые.

Децентрализация системы определяется расположением её частей на различных устройствах в интеллектуальном пространстве.

Различные исследования в области интеллектуальных пространств по-разному описывают представление контекста в архитектуре контекстно-ориентированных систем.

2. Современные контекстно-ориентированные системы в интеллектуальном пространстве.

2.1. Многоагентная архитектура К. Вейджана, Ш. Юанчун и Суо Ю. Авторы данной архитектуры в своей работе [4] объединили онтологическую модель описания контекста и модель, основанную на логике, в модель вероятностной логики первого порядка (first-order probabilistic logic), чтобы описать базовую структуру контекста и спроектировать механизм вероятностного вывода. В базовой онтологии интеллектуального пространства они описывают семь базовых концептов: пользователь (user), местоположение (location), время (time), деятельность (activity), сервис (service), окружение (environment), платформа (platform). Части базовой онтологии они спроектировали, адаптируя такие широко используемые онтологии верхнего уровня,

как DAML-Time и OWL-S. Концепты интеллектуального пространства представлены следующими классами базовой онтологии.

User. Данная онтология определяет словари, описывающие информацию профиля пользователя, его контактную информацию, его предпочтения и состояние пользователя, которое непосредственно зависит от его текущих задач и деятельности.

Location, Time, Activity. Отношения между концептами местоположением, временем и деятельностью способствуют проверке противоречивой контекстуальной информации, когда она поступает от сенсоров с различной точностью.

Platform, Service. Онтология платформы описывает аппаратные средства и программную инфраструктуру в интеллектуальном пространстве. Онтология сервиса определяет многоуровневую спецификацию сервисов, обеспечивая их поиск и композицию. Сообщество Semantic Web использует OWL-S как стандартизованную структуру для описания семантики сервиса, в общем.

Environment. Онтология окружения определяет спецификацию контекста условий (состояния) физического окружения пользователя, например, уровень шума, освещенность, температура и влажность.

Архитектура контекстно-ориентированной системы авторов данной работы была спроектирована, основываясь на многоагентной модели системы [20], с целью поддержки приложений, использующих контекстуальную информацию в интеллектуальном пространстве. Данная модель архитектуры состоит из нескольких взаимодействующих агентов (рис. 1).

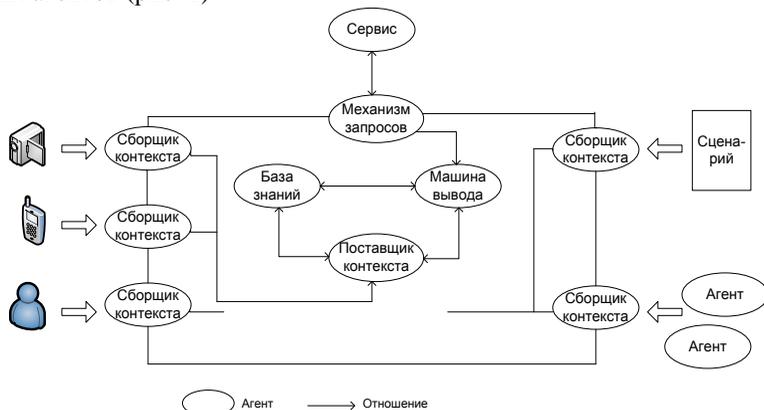


Рисунок 1. Многоагентная архитектура К. Вейджана, Ш. Юанчун и Суо Ю.

Context wrapper agent (сборщик контекста): получает различные типы данных с различных устройств и датчиков.

Context provider agent (поставщик контекста): преобразует данные сборщиков контекста в контекстуальную информацию, пригодную для дальнейшего использования в качестве знаний в интеллектуальном пространстве.

Inference engine agent (машина вывода): обеспечивает логический вывод контекстуальной информации высокого уровня (знаний) из низкоуровневых данных.

Knowledge base agent (база знаний): хранит правила вывода, наблюдаемые факты и онтологии.

Query filter agent (механизм запросов): обеспечивает интерфейс запросов приложениям и агентам для запросов и подписки на изменения контекстов сервисов интеллектуального пространства с поддержкой координации и управления данными сервисами на системном уровне.

Преимущества данного подхода заключаются в слабом связывании компонент системы, масштабируемости и прозрачности.

Изменения реализации компонент системы при слабом связывании их друг с другом происходят независимо в каждой из компонент и не влияют на реализацию других компонент. Слабое связывание обеспечивает возможность использования с минимальными затратами на интеграцию информационной системой нужных, включаемых в ее структуру, компонент для решения различных задач контекстно-ориентированных приложений по моделированию и выводу информации из контекста различного рода устройств и датчиков.

Данная архитектура за счет высокого уровня абстракции своих компонент и их инкапсуляции обеспечивает удобство и легкость в масштабировании контекстно-ориентированных сервисов. Настроив профиль сценария использования системы (*scenario profile*, рис. 1) и развертывая новые типы датчиков, сборщик контекста может собрать большое количество контекстуальной информации от различных источников, чтобы дать возможность системе наиболее близко адаптироваться под текущую ситуацию.

Разделяя функционал приложений и сервисов, промежуточный слой данной архитектуры использует механизм запросов для включения низкоуровневого функционала системы, который невидим для приложений.

2.2. Архитектура Семантического Пространства К. Ванга, Д.С. Донга, Ч. Чина и Р. Хетиараччи. Данную архитектуру разработали

авторы работ [2, 5] и предложили своё видение базовой онтологии *Семантического Пространства* (интеллектуального пространства) [2, 5, 6]. Данная архитектура Семантического Пространства состоит из нескольких взаимодействующих компонент (рис. 2): *wrappers* (сборщики), *aggregator* (агрегатор), *knowledge base* (база знаний), *query engine* (механизм запросов), *reasoner* (машина логического вывода).

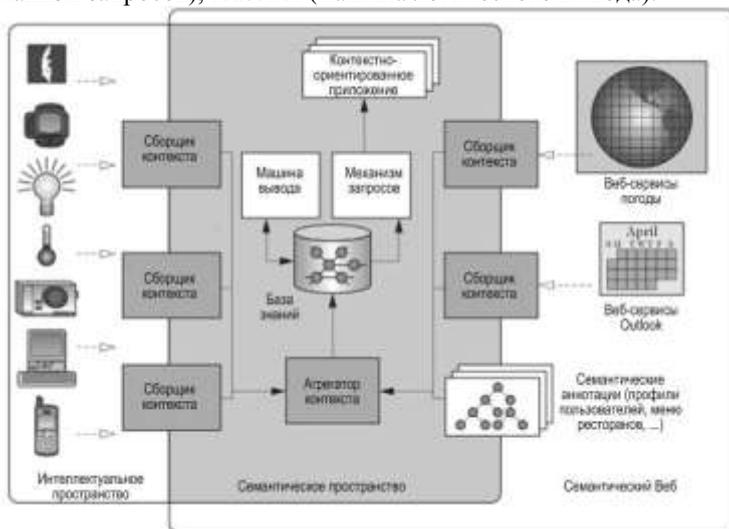


Рисунок 2. Архитектура семантического пространства

Context wrappers. Сборщики собирают данные от различных источников в интеллектуальном пространстве и преобразуют ее в контекстуальную информацию. Все сборщики являются автоматически настраиваемыми компонентами, поддерживающими унифицированный интерфейс для сбора данных от источников и преобразования их в контекстуальную информацию для приложений и агрегатора. Реализованы сборщики, как универсальные сервисы UPnP (Universal Plug and Play), с возможностью динамического подключения к интеллектуальному пространству, получением собственных IP-адресов и широковещательного оповещения о своем включении. Сборщики используют модель событий UPnP (General event notification Architecture – GENA) для оповещения об изменениях контекстуальной информации, на данные события могут подписываться приложения.

Context aggregator. Агрегатор реализован, как контрольная точка (control point) UPnP, он аккумулирует контекстуальную информацию со всех сборщиков и добавляет ее в базу знаний.

Context knowledge base. База знаний располагается в инфраструктуре каждого интеллектуального пространства и обеспечивает постоянное хранение контекстуальных знаний.

Context query engine. Механизм запросов обеспечивает абстрактный интерфейс доступа к контексту в базе знаний для приложений.

Context reasoner. Данный компонент выводит информацию высокого уровня абстракции из общего контекста. В семантическом пространстве явно представлен весь общий контекст, что позволяет обрабатывать его любым машинам логического вывода, и облегчает разработчикам реализовывать специфический для их приложений вывод, определяя эвристические правила.

В инфраструктуру семантического пространства включена онтология контекста верхнего уровня (*upper-level context ontology* – ULCO), определяющая набор базовых концептов, общих для разного рода устройств в окружении пользователя [2]. Среди разнообразного контекста авторы данных работ выделили три класса объектов из реального мира (пользователь, местоположение и вычислительный ресурс) и один класс концептуальных объектов (деятельность). Связанные отношениями эти классы формируют базовую онтологию контекста окружения пользователя. Авторы данной базовой онтологии делают упор на повторное использование знаний. Они интегрировали в неё открытые согласованные онтологии верхнего уровня, такие как friend-of-a-friend (FOAF), RCAL Calendar и FIPA Device Ontology. Таким образом, они смоделировали контексты пользователя, его деятельности и аппаратного обеспечения путем добавления недостающих концептов, характерных для интеллектуального пространства. Например, FOAF описывает простое отношение между людьми, – friendOf (является_другом), а авторы работы [2] расширили его до supervisorOf (является_руководителем), studentOf (является_студентом) и colleagueOf (является_коллегой). Для большей гибкости ULCO была спроектирована с возможностью дополнения описанных в ней классов необходимыми стороннему разработчику классами. Для этого достаточно “отнаследоваться” от определенного в ULCO класса и сформировать на его базе расширенную онтологию.

Данное онтологическое описание контекста позволяет инфраструктуре Семантического Пространства предоставлять контекстно-ориентированным приложениям доступ к контекстуальной информа-

ции посредством декларативных запросов и поддержке логического вывода информации высокого уровня абстракции из общего контекста. Каждая такая инфраструктура Семантического Пространства располагается в отдельном физическом пространстве и позволяет виртуально расширить границы данного пространства. Например, можно объединить несколько комнат в одно интеллектуальное пространство или разделить одну комнату на несколько интеллектуальных пространств.

2.3. Многоагентная архитектура и онтология COSE. Исследователи Вашингтонского государственного университета (Washington State University) Вемлингер З. и Холдер Л. разработали онтологию интеллектуального окружения (Casas Ontology for Smart Environments – COSE) для мультиагентной архитектуры интеллектуального пространства [20], основываясь на следующих исследованиях [2], [21-24]. Онтология COSE соответствует нотации OWL Lite, её базовыми концептами являются: здания, обитатели, датчики и человеческая деятельность [7]. Разработчики COSE определили отображение концептов COSE на онтологию OpenCyc [8], которая включает в себя полную онтологию Cyc [9], содержащую сотни тысяч выражений, миллионы утверждений, связывающих термины между собой. Данное отображение доступно в отдельной онтологии под нотацией OWL DL. На сегодняшний день у них в разработке находится онтология деятельности человека в повседневной жизни, которая доступна по следующей ссылке [10].

Контекстуальные данные, поступающие из различных источников в интеллектуальном пространстве, отображаются в базовой онтологии интеллектуального пространства, которая используется агентами и алгоритмами для управления устройствами в данном пространстве. Например, как это происходит согласно архитектуре EasyMeeting [22]. Информационный брокер предоставляет контекстуальные знания с помощью агента MajorDemo (рис. 3). Используя данные знания, MajorDemo выбирает и вызывает соответствующие сервисы из интеллектуального пространства, чтобы обеспечить необходимой информацией пользователей и другие сервисы.

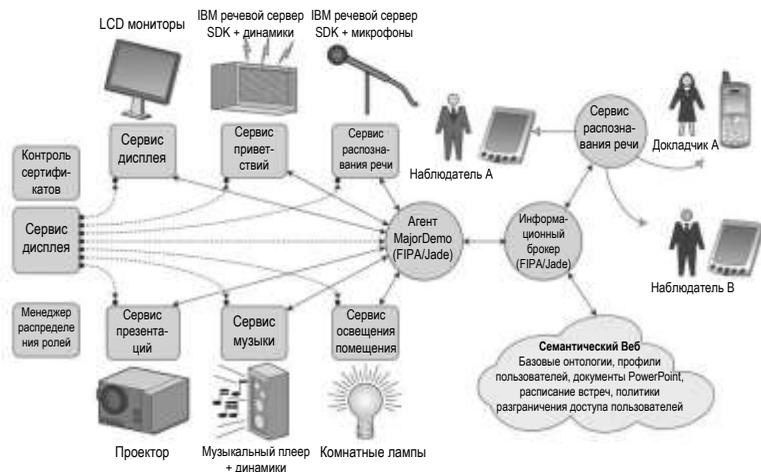


Рисунок 3. Архитектура EasyMeeting

Алгоритмы, распознающие деятельность (активность), обучаясь на данной онтологии, могут сфокусироваться только на семантике перемещения и взаимодействия независимо от набора датчиков и особенностей их использования. Онтологии верхнего уровня, такие как OpenCyc, DOLCE, SUMO и т.п., содержат знания здравого смысла (common sense) о моделируемом мире и формируют единую систему понятий для онтологий нижних уровней. Обработка онтологий верхнего уровня требует больших вычислительных ресурсов. Чтобы минимизировать вычислительные нагрузки авторы данной работы “наследуют” базовые онтологии интеллектуальных пространств от глобальных онтологий верхнего уровня. Используя данную иерархию, становится возможным уменьшить вычислительные требования к локальным ресурсам, не жертвуя необходимыми общими знаниями.

2.4. Архитектура COBRA. Гарри Чен, Тим Финин и Анупам Джоши из университета Мериленд округа Балтимор (США) разработали для поддержки контекстно-ориентированных систем архитектуру под названием COBRA (Context Broker Architecture), которая включает в себя онтологию интеллектуального пространства COBRA-ONT [11].

COBRA представляет собой брокер-ориентированную агентную архитектуру для поддержки контекстно-ориентированных систем. Центральной частью данной архитектуры является специализированный интеллектуальный агент – информационный брокер (context bro-

ker). Технически информационный брокер представляет собой сервер, запущенный на устройстве с высокими вычислительными мощностями и аккумулирующий контекст интеллектуального пространства.



Рисунок 4. Архитектура COBRA

На рисунке 4 изображена высокоуровневая модель информационного брокера и его связи с другими агентами в интеллектуальном пространстве. Все элементы интеллектуального пространства должны предопределять присутствие информационного брокера, а интеллектуальные агенты верхнего уровня должны уметь взаимодействовать с брокером по стандарту FIPA Agent Communication Language [12]. Модель информационного брокера состоит из четырех функциональных компонент:

1. *Context Knowledge Base*. Устойчивое хранилище контекстуальных знаний.
2. *Context Reasoning Engine*. Машина вывода, которая обрабатывает все хранимые контекстуальные знания.
3. *Context Acquisition Module*. Библиотека процедур для сбора контекстуальной информации на промежуточном уровне.
4. *Privacy Management Module*. Набор правил вывода, определяющий правила доступа различным элементам интеллектуального пространства к контекстуальной информации.

Все ресурсоемкие операции по обработке и выводу контекста проводит информационный брокер, а результат операций предоставляет маломощным устройствам в интеллектуальном пространстве. Трудности, связанные с обеспечением правил доступа и политикой безопасности, упрощаются при наличии централизованного менеджера. Но централизованный информационный брокер может стать узким местом в инфраструктуре интеллектуального пространства большого масштаба. Чтобы избежать этого, авторы данной архитектуры предлагают устойчивый к ошибкам подход, основанный на модели *устойчивой группы брокеров*, которая описана в работе С. Кумара и Филипа Р. Коена [13]. Информационные брокеры группируются вместе и формируют *федерацию брокеров*. Каждый брокер ответственен за управление определенной частью интеллектуального пространства (например, одной из аудиторий в университете). В федерации брокеры могут организовываться в иерархическую и одноранговую коммуникационные структуры. Они периодически изменяют и синхронизируют знания в интеллектуальном пространстве. Ключевым преимуществом данного подхода является возможность доступа к контексту интеллектуального пространства независимо от доступности отдельного брокера, участники федерации брокеров информируют участников интеллектуального пространства о доступности друг друга и перераспределяют нагрузку между собой.

COBRA-ONT классифицируется по четырем различным, но связным темам: онтология территории (*physical places context*), онтология агентов (человеческих и программных), онтология местоположения агентов (*agents' location context*), онтология деятельности агентов.

Онтология территории. Класс верхнего уровня данной онтологии называется Place и описывает абстракцию физического месторасположения. COBRA-ONT определяет два подкласса Place: AtomicPlace и CompoundPlace, которые имеют разные свойства *вместимости*. Свойство вместимости физического месторасположения определено как модель, способная включать в себя физические пространства (например, кампус включает в себя все здания, расположенные на его территории, которые в свою очередь состоят из комнат и т.д.). Оно представлено двумя инверсными классами свойства *spatiallySubsumes* и *isSpatiallySubsumedBy*. Для классов AtomicPlace мощность свойства *spatiallySubsumes* ограничена нулем, а диапазоном (*range*) свойства *isSpatiallySubsumedBy* является CompoundPlace. Таким образом, AtomicPlace не может быть составным, но может быть частью CompoundPlace. CompoundPlace может включать в себя AtomicPlace и са-

мого себя. Примерами подклассов AtomicPlace могут быть классы: комната, холл, гостиная, паркинг и т.д., подклассов CompoundPlace – классы: здание, кампус и т.д.

Онтология агентов. Классом верхнего уровня данной онтологии является класс Agent, также определены два его непересекающихся подкласса Person и SoftwareAgent, первый описывает человека в роли агента, а второй – программного агента. Каждый агент в COBRA-ONT имеет свою роль, зависящую от события (например, в процессе презентации (доклада) ролью человека будет «докладчик», после презентации роль человека меняется на «участник совещания»). Роль описана свойством fillsRole с диапазоном классов Role. Роли могут использоваться для характеристики участия агента в событии, что позволяет контекстно-ориентированной системе выводить возможные действия пользователя исходя из качества его участия в событии. Также в данной онтологии описаны возможные воздействия агента на других агентов с целью решения какой-либо задачи.

Онтология местоположения агентов. Местоположение агентов представлено в онтологии свойством locatedIn с диапазоном классов AtomicPlace и CompoundPlace. На данное свойство наложены следующие ограничения:

1. Один агент не может физически находиться в двух разных местах в один и тот же промежуток времени.
2. Один агент может физически находиться в двух разных местах в один и тот же промежуток времени, только в том случае, когда одно из данных мест включает в себя другое.

Данное ограничение представлено расширением свойства locatedIn двумя подклассами locatedInAtomicPlace и locatedInCompoundPlace. Агенты, находящиеся в разных местах, могут быть классифицированы исходя из их местоположения (например, для всех людей в здании определен класс PersonInBuilding).

Онтология деятельности агентов. Данная онтология описывает события, в которых участвуют агенты. Каждое событие имеет свое расписание. В COBRA-ONT класс PresentationSchedule описывает расписание презентации посредством таких свойств как время начала/конца презентации, её название, краткое содержание и местоположение. Определено не менее одного докладчика и один или более слушателей. Для описания текущего события был определен класс PresentationEventHappeningNow. Для объектов данного класса предполагается, что они имеют неявную ассоциацию с временным предикатом “now”. В COBRA-ONT обеспечена возможность логического вы-

вода из временных свойств людей и мест, связанных с событием презентации. Например, требуется вывести, кто участвует в совещании в настоящее время или в какой комнате проходит данное совещание. Для этого в данной онтологии определены: класс `RoomHasPresentationEventHappeningNow`, описывающий комнату в которой проходит текущее совещание; класс `SpeakerOfPresentationHappeningNow`, описывающий докладчика текущего совещания; и класс `AudienceOfPresentationHappeningNow`, описывающий слушателей на текущем совещании.

Использование онтологической модели описания контекста архитектуре COBRA дает следующие преимущества.

Базовые онтологии позволяют совместно использовать знания в открытых и динамических распределенных системах.

Онтологии с хорошо определенной семантикой обеспечивают интеллектуальным агентам возможность вывода новой контекстуальной информации из общего контекста.

Формализованное представление контекста в интеллектуальном пространстве повышает интероперабельность его элементов.

2.5. Архитектура SmartConference. Исследователями лаборатории интегрированных систем автоматизации Санкт-Петербургского института информатики и автоматизации РАН была разработана архитектура контекстно-ориентированной системы для проведения совещаний [15, 25], представленная на рисунке 5.

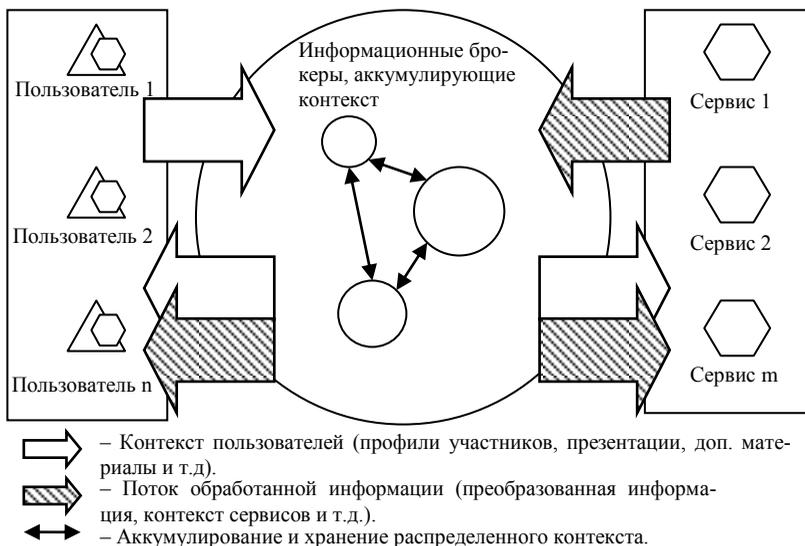


Рисунок 5. Архитектура SmartConference

Информационные брокеры представляют собой распределенные хранилища информации, способные организовываться в единое информационное пространство. Для описания контекста используется онтологическая модель. Сервисы посредством программного интерфейса информационного брокера получают доступ к необходимому контексту, обрабатывают его или добавляют новый в интеллектуальное пространство. Пользователи через графические интерфейсы используют сервисы в своих целях. Каждый сервис интеллектуального пространства имеет свой уникальный контекст, открытый для других элементов данного пространства. Для взаимодействия друг с другом каждый сервис интеллектуального пространства формирует у себя общий контекст путем отображения онтологий других элементов интеллектуального пространства на свою онтологию. Основной упор при поиске необходимого контекста и отображении онтологий делается на лексикографическое сходство концептов.

Преимуществом данного подхода является независимость контекста сервисов от базовой онтологии, что позволяет разработчикам данных сервисов проектировать их независимо от контекста интеллектуального пространства. Однако им следует включить в разрабатываемый сервис ресурсоемкий и сложный механизм объединения онтоло-

гий с обратной связью с пользователем для принятия решений по согласованию понятий из общего контекста.

3. Сравнительный анализ рассмотренных контекстно-ориентированных архитектур систем в интеллектуальном пространстве. Исходя из рассмотренных выше архитектур, был проведен сравнительный анализ контекстно-ориентированных информационных систем, по ряду критериев (см. таблицу).

Для обеспечения полной *децентрализации* и полной *масштабируемости* каждый сервис наделяется возможностью построения модели общего контекста, исходя из специфических моделей контекста сервисов в интеллектуальном пространстве. Использование базовой онтологии контекста интеллектуального пространства ограничивает масштабируемость сервисов данного пространства в его пределах, пока не будет найдено отображение данной онтологии на базовую онтологию контекста другого интеллектуального пространства. При таком отображении обеспечивается взаимодействие всех сервисов обоих интеллектуальных пространств. Так как в данном случае сервисы строятся вокруг центральной модели контекста, их децентрализация является частичной.

Большинство подходов использует в основе построения модели контекста базовую онтологию интеллектуального пространства. Это позволяет *систематизировать общий контекст* интеллектуального пространства, но обязывает разработчиков сервисов использовать специфическую для определенного интеллектуального пространства базовую онтологию контекста. Использование базовой онтологии не повышает *ресурсоемкость* контекстно-ориентированных приложений (не приводит к дополнительным вычислительным нагрузкам на устройствах). С другой стороны, отсутствие базовой онтологии контекста обязывает разработчиков контекстно-ориентированных сервисов включать в их реализацию сложные и вычислительно ёмкие механизмы систематизации контекста.

Систематизация общего контекста влияет на *сложность программной реализации* контекстно-ориентированных приложений. Отсутствие систематизации общего контекста приводит к необходимости усложнять реализацию приложений механизмами по отображению онтологических моделей контекстов друг на друга. Независимые друг от друга разработчики будут использовать варианты данных механизмов с различной эффективностью установления соответствий концептов, что приведет к различному “толкованию” приложениями состояния одних и тех же участников интеллектуального пространства.

Сравнительный анализ рассмотренных контекстно-ориентированных архитектур

	Масштабируемость	Децентрализация	Базовая онтология	Ресурсоемкость	Сложность реализации
Многоагентная архитектура К.Вейджана	частичная	частичная	да	низкая	низкая
Архитектура Семантического Пространства	частичная	частичная	да	низкая	низкая
Архитектура COBRA	частичная	частичная	да	низкая	низкая
Архитектура SmartConference	полная	полная	нет	высокая	высокая

4. Онтологическая модель описания контекста. Целью онтологической модели описания контекста является формализация структурированных контекстуальных элементов в интеллектуальном пространстве при помощи использования онтологической методологии, определяющей концепты и отношения элементов контекста. Онтологию контекста интеллектуального пространства можно разделить на базовую онтологию (*core ontology*), определяющую общие концепты интеллектуального пространства, и расширенную онтологию контекста (*extended ontology*), описывающую специфику окружающей среды.

Базовая онтология предоставляет глобальную и расширяемую (*extensible*) модель, в которую могут быть отображены и интегрированы данные из различных источников. Данная каноническая форма представляет собой единую базу знаний для кросс-доменных приложений и сервисов, ориентированных на поиск ресурсов, интеллектуальный анализ данных (*data mining*) и т.п. Полная и расширяемая онтология, описывающая базовые концепты общие для различных предметных областей, может обеспечить основополагающую модель для интеграции большего количества специализированных словарей и специфических для конкретной предметной области концептов. Такая онтология является обязательной для четкого отображения между специфичными знаниями из разных предметных областей и для дальнейшего проектирования и композиции сервисов, направленных на поиск информации из различных источников, интеллектуальный анализ данных и извлечение знаний (*knowledge extraction*) [14].

Существует разница между базовой онтологией и базовыми метаданными, такими, например, как *Dublin Core*. Базовые метаданные в общем случае созданы, отредактированы и представлены людьми, они зависят от многих человеческих факторов, включая ограничения на сложность, которые играют решающую роль при их проектировании.

Базовая онтология обеспечивает основополагающую формальную модель для приложений, которые интегрируют исходные данные от различных источников и предоставляют богатый набор дополнительных функций в автоматическом режиме. Уровень сложности верхних уровней базовой онтологии позволяет разработчикам сфокусироваться больше на полноте и логической правильности, чем на понимании концептов.

Диего Кальванезе и др. в своей работе [17] описали сценарий интеграции информации, в котором основополагающую роль играет базовая онтология. Центральным понятием в сценарии является *модель деятельности* (enterprise model) – “концептуальное представление глобальных концептов и отношений, которые интересны с точки зрения приложения”. Данная модель деятельности обеспечивает основу для интеграции набора открытых исходных данных, представленных независимыми сообществами из интернета или любого другого открытого информационного пространства. Отображение информации в данном подходе возможно благодаря:

1. Созданию *моделей исходных данных* (source models), обеспечивающих концептуальное представление для любой их категории, в объеме достаточном для использующих данные модели приложений.
2. Определению *модели предметной области* (domain model), которая описывает взаимосвязи между моделями исходных данных и моделью деятельности.
3. Формализации данных концептуальных моделей и отношений с помощью языков представления знаний, таких как дескрипционная логика.

Одной из реализаций, основанной на данной интеграционной модели, может быть сервис кросс-доменного поиска. Если интеграция информации прошла успешно, тогда запросы к модели деятельности должны возвращать такие же значения, как и запросы, адресованные непосредственным источникам информации.

Построение контекстно-ориентированных систем с использованием базовой онтологии контекста позволяет элементам системы взаимодействовать исходя из общепонятной структуры контекста. Независимым разработчикам необходимо знать общую структуру базовой онтологии контекста для создания приложений, способных взаимодействовать с любым сервисом из интеллектуального пространства. Также становится возможным гибко перераспределять вычислительную

нагрузку внутри данной системы, расширив соответствующие концепты базовой онтологии контекста.

Таким образом, независимый разработчик, исходя из специфики своей предметной области, создает свою *расширенную онтологию* контекста на основе базовой онтологии интеллектуального пространства, что делает понятной специфику контекстуальной информации разрабатываемого приложения для других приложений, использующих данную базовую онтологию. Данные с датчиков в интеллектуальном пространстве, преобразованные в контекстуальные сообщения на основе базовой онтологии, могут быть легко интерпретированы контекстно-ориентированными приложениями. Контекст реального мира обычно порождается различными источниками, что влияет на подходы к генерации контекстуальных сообщений. В работе [2] приведены следующие примеры.

В примере пользовательского сценария смартфона такой контекст, как имя пользователя, его связи и время выступления на конференции, имеет относительно медленный коэффициент изменения, пользователи часто дополняют данную информацию и обычно генерируют сообщения из данного контекста. На листинге 1 приведен пример контекстуального сообщения на языке OWL, которое описывает пользователя под именем Ross Geller (базовое пространство имен по умолчанию www.i2r.a-star.edu.sg/SemanticSpace#).

```
<User rdf:about="RossGeller">
  <name>Ross Geller</name>
  <mbox>ross@i2r.a-star.edu.sg</mbox>
  <homepage rdf:resource="www.i2r.a-star.edu.sg/~ross"/>
  <office rdf:resource="#Room209"/>
  <officePhone>1234</officePhone>
  <mobilePhone>6789</mobilePhone>
  <supervisorOfrdf:resource=
    "#JoeyTribbiani"/>
  <! -More properties not shown in this
    example->
</User>
```

Листинг 1. Пример контекста пользователя.

Датчики обычно представляют контекст, связанный с позиционированием пользователя, уровнем шума, освещенностью и т.п. Данный контекст меняется очень часто, и фиксировать его изменения должны

специальные программы в автоматическом режиме. Например, когда Ross Geller вошел в комнату №209 (Room 209), специальный датчик определил его присутствие и сгенерировал следующее контекстуальное сообщение (см. листинг 2).

```
<User rdf:about="#RossGeller">  
<locatedIn rdf:about="#Room209"/> </User>
```

Листинг 2. Пример контекстуального сообщения датчика.

Каждый объект OWL имеет уникальный идентификатор URI, и контекстуальные сообщения могут ссылаться на дополнительное описание объекта посредством данных URI. Например, URI www.i2r.a-star.edu.sg/SemanticSpace#RossGeller ссылается на пользователя, которого мы определили выше, а URI www.i2r.a-star.edu.sg/SemanticSpace#Room209 ссылается на конкретную комнату, определенную где угодно в контекстно-ориентированной системе.

Для композиции контекстно-ориентированных сервисов различных интеллектуальных пространств необходимо провести отображение их базовых онтологий, после чего все сервисы в обоих интеллектуальных пространствах смогут взаимодействовать между собой. Вычислительная нагрузка при отображении концептов из разного контекста может быть гибко распределена между сервисами объединяющихся интеллектуальных пространств.

5. Принципы построения контекстно-ориентированных систем в интеллектуальных пространствах. На основе проведенного анализа можно выделить следующие основные принципы проектирования контекстно-ориентированных систем в интеллектуальном пространстве.

Формализованное представление контекста подразумевает представление данных от различных источников в форме, доступной для “понимания” всеми участниками интеллектуального пространства.

Гарантированный доступ к контексту должен быть обеспечен с помощью инфраструктуры интеллектуального пространства или самими источниками контекста, таким образом, чтобы каждый авторизованный участник интеллектуального пространства имел доступ к контекстам других участников независимо от степени своей нагрузки или физической доступности (если участник вышел из интеллектуального пространства, это событие должно отразиться в его контексте).

Без *систематизации общего контекста* интеллектуального пространства невозможно формализованное представление контекста для всех его участников, что делает невозможным быстрое реагирование системы на изменения контекстов участников пространства. Построение контекстно-ориентированных сервисов вокруг базовой модели контекста предметной области интеллектуального пространства обязывает независимых разработчиков использовать её при проектировании сервисов для данного интеллектуального пространства, при этом пропадает необходимость во включении в приложение механизмов поиска соответствий понятий контекстов различных источников. Вычислительная мощность данных механизмов прямо пропорциональна объему контекста в интеллектуальном пространстве и целиком ложится на использующие их устройства. Интеграция базовой онтологии контекста с онтологиями предметной области верхнего уровня повышает уровень абстракции базовой онтологии, а использование открытых и общепринятых онтологий верхнего уровня повышает уровень гармонизации данной базовой онтологии с онтологиями различных предметных областей и делает её более понятной для разработчиков.

Логический вывод знаний на основе онтологической модели контекста интеллектуального пространства расширяет варианты возможного реагирования на события, происходящие в интеллектуальном пространстве.

Децентрализация компонент контекстно-ориентированной системы заключается в их независимом функционировании и расположении, при котором отключение или перемещение одного из модулей не влияет на работу системы в целом. Полная децентрализация сервисов в интеллектуальных пространствах и свободная композиция их независимо от специфики интеллектуального пространства возможна лишь при условии, когда каждый сервис способен расширять свои знания о предметной области, используя знания других сервисов. Другими словами, это означает, что необходимо реализовать механизм отображения онтологий контекстов для каждого сервиса. На сегодняшний день не существует возможности полностью автоматизировать данный механизм и сделать его прозрачным. К тому же, за уровень систематизации контекста сервиса будут отвечать его разработчики, а увеличение количества сервисов в интеллектуальном пространстве в данном случае будет прямо пропорционально увеличению вычислительной сложности при формировании общего контекста на каждом из сервисов.

Масштабирование контекстно-ориентированной системы подразумевает расширение ее функциональных возможностей за счет до-

бавления в нее сервисов без изменения ядра системы. На сегодняшний день реализуемо масштабирование в пределах предметной области интеллектуального пространства, однако, при объединении базовых онтологий различных интеллектуальных пространств расширяется количество возможных для включения в систему сервисов.

6. Заключение. Проектирование контекстно-ориентированных систем предполагает формализованное представление контекста, гарантированный доступ к нему, его логический вывод, а также данная система должна быть масштабируемой и децентрализованной. Данные принципы являются основополагающими для построения контекстно-ориентированных систем в интеллектуальных пространствах: формализованное представление контекста предполагает его систематизацию, что дает возможность масштабировать систему и делать ее компоненты децентрализованными, гарантированный доступ к контексту и логический вывод на его основе обеспечивают возможность наиболее полно проанализировать ситуацию и отреагировать на нее приложениям. Интеллектуальные пространства охватывают широкий спектр устройств из разнообразного окружения (дома, офисы, аудитории, транспорт и т.п.). Для того чтобы не проектировать полную онтологическую модель для каждого вида контекста, большинство разработчиков фокусируются на создании базовой онтологии контекста, описывающей базовые концепты общие для различных предметных областей. Что делает систему более гибкой и адаптивной к различному физическому окружению и упрощает независимую разработку сервисов данной системы. Анализируя ряд исследований по гармонизации онтологий [14, 26, 27], можно сделать следующий вывод: независимо от вида модели предметной области, будь то объектно-ориентированная, логическая, фреймовая, модель сущность-связь или какая-либо другая модель, исходные онтологии (source ontologies) должны быть отображены на базовой онтологической модели, для которой дальнейшая работа по гармонизации предпочтительней. Построение контекстно-ориентированных сервисов вокруг центральной базовой модели предметной области позволяет избежать комбинаторного взрыва и усложнения приложения, что неотвратимо последует при отображении одних онтологий на другие.

Литература

1. *Stefan Poslad. Ubiquitous Computing: Smart Devices, Environments and Interactions.* Wiley, 2009. P. 30.

2. X. Wang, J.S. Dong, C.Y. Chin, S.R. Hettiarachchi, and D. Zhang. Semantic Space: An Infrastructure for Smart Spaces // IEEE Pervasive Computing. Vol. 3, No 3 (July-September, 2004). P 32–39.
3. Sergey Boldyrev, Ian Oliver, Jukka Honkola. A Mechanism for Managing and Distributing Information and Queries in a Smart Space Environment // The 1st International Workshop on Managing Data with Mobile Devices (MDMD 2009) (Milan, Italy, May 6-7, 2009).
4. QIN Weijun, SHI Yuanchun, SUO Yue. Ontology-based context-aware middleware for smart space // Journal of Tsinghua Science and technology. Vol. 12, No 6 (December, 2007). P 707-713.
5. X.H. Wang, D.Q. Zhang, T. Gu, and H.K. Pung. Ontology-Based Context Modeling and Reasoning using OWL // In Context Modeling and Reasoning Workshop (March, 2004). P. 18–22.
6. H. Liao and C. Tu. A RDF and OWL-based temporal context reasoning model for smart home // Information Technology Journal. Vol. 6, No. 8 (2007). P 1130–1138.
7. Zachary Wemlinger and Lawrence Holder. The COSE Ontology: Bringing the Semantic Web to Smart Environments // ICOST (June, 2011). P 205-209.
8. Cycorp, Inc. Overview of OpenCyc. <http://opencyc.org/cb/welcome> (дата обращения: 25.07.2011).
9. Cycorp, Inc. The Cyc knowledge server. <http://www.cyc.com/cyc/technology/whatscyc> (дата обращения: 25.07.2011).
10. Zachary Wemlinger, Lawrence Holder. Daily Living Activities Ontology. School of Electrical Engineering and Computer Science Washington State University. http://casas.wsu.edu/owl/activity_experiment_1.owl (дата обращения: 25.07.2011).
11. Chen H, Finin T, Joshi A. An ontology for context-aware pervasive computing environments // Knowledge Engineering Review: Special Issue on Ontologies for Distributed Systems. Vol. 18, No 3 (2004). P 197–207.
12. The Foundations for Intelligent Physical Agents // FIPA Abstract Architecture Specification, SC00001L edition (December, 2002).
13. Sanjeev Kumar, Philip R. Cohen, and Hector J. Levesque. The adaptive agent architecture: Achieving fault-tolerance using persistent broker teams // In Proceedings of the Fourth International Conference on Multi-Agent Systems (2000). P 159–166.
14. Martin Doerr, Carl Lagoze, Jane Hunter, Thomas Baker, et al. Building Core Ontologies // White Paper of the DELOS Working Group on Ontology Harmonization (April, 2003).
15. Кашевник А.М., Вальченко Ю., Ситаев М.М., Шилов Н.Г. Интеллектуальная система автоматизированного проведения конференций // Труды СПИИРАН. 2010. Вып. 14. С. 228–245.
16. Кудрявцев Д. В. Практические методы отображения и объединения онтологий // Труды 11-й Национальной конференции по искусственному интеллекту с международным участием (КИИ-08), семинар «Знания и Онтологии *ELSEWHERE*», 29 сентября-3 октября 2008. г. Дубна, Россия. Том 3. - М.: URSS, 2008. С. 164–173..
17. D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Description Logic Framework for Information Integration // Proc. of the 6th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'98) (June 2-5, 1998). P 2-13.
18. N. Guarino. Formal Ontology in Information Systems // Proceedings of Formal Ontology in Information Systems (Trento, Italy, 6-8 June, 1998). IOS Press, Amsterdam. P 3–15.
19. Смирнов А.В., Левашова Т.В., Пашикин М.П. Модели контекстно-управляемых систем поддержки принятия решений в динамических структурированных областях // Труды СПИИРАН. 2009. Вып. 9. СПб.: Наука, 2009. С. 116–147.

20. *Diane J. Cook*. Multi-agent smart environments // *Journal of Ambient Intelligence and Smart Environments*. Volume: 1, Issue: 1, Publisher: IOS Press, 2009. P 47-51.
21. *Dario Bonino, Fulvio Corno*. DogOnt - Ontology Modeling for Intelligent Domestic Environments // *In Proceedings of International Semantic Web Conference (2008)*. P 790–803.
22. *H. Chen, T. Finin, A. Joshi, L. Kagal, F. Perich, and D. Chakraborty*. Intelligent agents meet the semantic web in smart spaces // *Internet Computing, IEEE*. Vol. 8 (2004). P 69–79.
23. *L. Sommaruga, A. Perri, and F. Furfari*. DomoML-env: an ontology for Human Home Interaction // *Proceedings of SWAP (2005)*. P 14–16.
24. *E. Kim and J. Choi*. An Ontology-Based Context Model in a Smart Home // *Computational Science and Its Applications-ICCSA (2006)*. P 11–20.
25. *Dmitry G. Korzun, Ivan V. Galov, Alexey M. Kashevnik, Nikolay G. Shilov, Kirill Krinkin, Yury Korolev*. Blogging in the Smart Conference System // *9th FRUCT Conference*, 26-29 April 2011, Petrozavodsk, Russia
26. *L.A. Kalinichenko*. Method for Data Models Integration in the Common Paradigm // *Proceedings of the First East-European Symposium ADBIS'97, St. Petersburg, September 1997*.
27. *L.A. Kalinichenko*. Compositional Specification Calculus for Information Systems Development // *Proceedings of the Third East European Symposium on Advances in Databases and Information Systems (ADBIS'99), Maribor, Slovenia, September 1999, Springer Verlag, Lecture Notes in Computer Science*.

Вальченко Юрас — специалист по компьютерной безопасности, программист лаборатории интегрированных систем автоматизации Учреждения Российской академии наук Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Область научных интересов: семантический веб, экспертные системы, компьютерная безопасность, распределенные системы управления. Число научных публикаций — 3. juras.valcenko@gmail.com; СПИИРАН, 14-я линия В.О., д. 39, г. Санкт-Петербург, 199178, РФ; р.т. +7(812)328-3337, факс +7(812)328-4450.

Valcenko Juras — specialist in computer security, programmer, Laboratory of Computer Aided Integrated Systems, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: semantic web, expert systems, computer security, distributed control systems. The number of publications — 3. juras.valcenko@gmail.com; SPIIRAS, 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812)328-3337, fax +7(812)328-4450.

Кашевник Алексей Михайлович — канд. техн. наук; старший научный сотрудник лаборатории интегрированных систем автоматизации учреждения Российской академии наук Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Область научных интересов: управление знаниями, профилирование, онтологии, интеллектуальные пространства, логистические системы. Число научных публикаций — 87. alexey@iias.spb.su; СПИИРАН, 14-я линия, д.39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328-8071, факс +7(812)328-0685.

Kashevnik Alexey — Ph.D.; senior researcher of the laboratory of computer aided integrated systems institution of the Russian Academy of Sciences St. Petersburg Institute for Informatics and Automation of RAS (SPIIRAS). Research area: knowledge management, profiling, ontologies, smart-spaces, logistics systems. Number of publications — 87. alexey@iias.spb.su;

SPIIRAS, 14th Line V.O., 39, Saint-Petersburg, 199178, Russia; office phone +7(812)328-8071, fax +7(812)328-0685.

Поддержка исследований. В публикации представлены результаты исследований, выполненные при поддержке гранта РФФИ 10-07-00368-а, программой Президиума РАН «Интеллектуальные информационные технологии, математическое моделирование, системный анализ и автоматизация» (проект 213) и Министерством образования и науки в рамках ФЦП «Научные и научно-педагогические кадры инновационной России на 2009-2013 годы» (госконтракт № 14.740.11.0357).

Рекомендовано СПИИРАН, зав. лаб. ИСА Смирнов А.В., д.т.н., проф.
Статья поступила в редакцию 20.10.2011.

РЕФЕРАТ

Вальченко Ю., Кашевник А. **Современные подходы к построению контекстно-ориентированных систем в интеллектуальных пространствах.**

С каждым днем офисы, квартиры, транспорт, и подобные места присутствия человека оснащаются все большим количеством различных датчиков, сенсоров, камер и другой аппаратуры, облегчающей взаимодействия людей и техники, которую им необходимо использовать в данном физическом окружении. Такие окружения принято называть интеллектуальными пространствами. Для облегчения взаимодействия людей с различного рода устройствами в таком интеллектуальном пространстве необходимо спроектировать информационную систему, которая будет управлять этими устройствами в зависимости от потребностей пользователя и текущей ситуации. На сегодняшний день одной из передовых технологий для обработки текущей ситуации является технология управления контекстом, который представляет собой описание функционала каждого из устройств и информацию, получаемую от него. Системы, построенные на основе технологии управления контекстом, называются контекстно-ориентированными, они организуют корректное взаимодействие всех устройств в пространстве, осуществляют мониторинг окружающей среды и быстро реагируют на изменения в ней. Такая система может отслеживать предпочтения пользователя и предсказывать его дальнейшее поведение, тем самым, автоматизируя работу с устройствами.

На основе выполненного в работе анализа различных контекстно-ориентированных систем был выявлен ряд особенностей, выделяющий такие системы среди других распределенных систем. Очевидно, что данная система децентрализована, т.к. ее компоненты распределены на устройствах в интеллектуальном пространстве, и выход из строя одного из устройств не должен влиять на работу системы в целом. Она должна быть масштабируема: замена устройств и добавление новых – обычное дело для повседневной жизни человека. Масштабируемость и децентрализованность системы закладывается еще на этапе ее построения, данные принципы тесно связаны с тремя другими важными принципами её построения. Формализованное представление контекста, гарантированный доступ к нему и возможность логического вывода из него новой информации обеспечивают данным системам интероперабельность, свободную композицию компонент и своевременное реагирование на события в пространстве.

SUMMARY

Valcenko J., Kashevnik A. **Modern Approaches to Context-Based Systems Developing in Smart Spaces.**

Today offices, apartments, vehicles, and similar places of human presence are equipped with an increasing number of different sensors, cameras and other devices to facilitate interaction between people and equipment they use. Such environments are called smart spaces. In order to facilitate human interaction with various devices in smart space it's necessary to develop the information system that would manage these devices, depending on user needs and current situation. Today the context management technology is one of the most advanced technologies for the current situation processing. It describes functionality of each device and information generated by it. Systems based on management context technology are called context-oriented. They organize the correct interaction of all devices in smart space, monitor and react quickly to changes in the space. Such a system can track user preferences and predict his future behavior; thereby it automates the work with devices in smart space.

Based on the analysis of various context-oriented systems, some features were revealed that distinguishes these systems from other distributed systems. It is obvious that the system is decentralized, because its components are distributed on the devices in the smart space, and the failure of one of the devices should not affect on the system work in whole. This system must be scalable, because equipment replacement and adding are common occurrences in human living. Scalability and decentralization are loaded at the stage of the system design; these principles are closely related to three other important principles of the system design. Context explicit representation, context querying and context reasoning provide system interoperability, free composition of system components and system timely response to events in smart space.