НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

А.А. Карасёв, В.А. Старых

ИНТЕЛЛЕКТУАЛЬНАЯ ОБРАБОТКА И РАЗРЕШЕНИЕ ИНЦИДЕНТОВ В ИНФОРМАЦИОННЫХ СИСТЕМАХ: ОНТОЛОГИЧЕСКИЙ ПОДХОД И КЛАССИФИКАЦИЯ

Под редакцией
д.т.н., профессора Болотовой Л.С.

Москва
2015

УДК 004.89
ББК 32.813
К21

РЕЦЕНЗЕНТЫ:
Д-р техн. наук, профессор Данчул А.Н.
Д-р техн. наук, профессор Позин Б.А.

Карасёв А.А., Старых В.А.
К21 Интеллектуальная обработка и разрешение инцидентов в информационных системах: онтологический подход и классификация / Карасёв А.А., Старых В.А.; НИУ ВШЭ. – М., 2015. – 232 с., ил.

ISBN 978-5-9518-0680-2

В книге излагается описание процесса разработки интегрированного программного инструментария, позволяющего повысить эффективность, достоверность и скорость принятия решений по локализации и диагностике информационных систем (ИС), при проявлении дефектов в ходе эксплуатации, на основе использования знаний о нарушении функциональности ИС, представленных в виде инцидентов. Программный инструментарий разработан на основании функциональной интеграции системы классификации инцидентов для формирования базы знаний инцидентов, информации о причинно-следственных связях для инцидентов, вызванных какой-либо проблемой и механизма создания и поддержания в актуальном состоянии связей между записями в базах данных инцидентов, проблем и информационно-справочных базах знаний. Разработаны два независимых метода классификации и структурирования знаний об инцидентах в ИС. Первый базируется на структурном подходе, второй основан на эталонной модели взаимодействия открытых систем ISO/OSI. На основе предложенных методов классификации и математической модели представления знаний об инцидентах в ИС разработана онтология обработки инцидентов в ИС, позволяющая описать все доступное множество объектов предметной области, значимое для обработки инцидентов в ИС. Программный инструментарий приобретения и представления экспертных знаний об инцидентах в ИС реализован на основе свободно распространяемого пакета Thinkmap SDK. По результатам опытной эксплуатации выполнена оценка повышения эффективности наполнения базы экспертных знаний об инцидентах с использованием метрик – количественных мер измерения значимых параметров функционирования ИС.

Для специалистов, деятельность которых связана с решением задач управления и администрирования сложных аппаратно-программных комплексов ИС: разработчиков ИС, администраторов служб технической поддержки, руководителей служб эксплуатации ИС.

Книга может быть полезна студентам, магистрам, аспирантам специальностей, связанных с изучением построения прикладных интеллектуальных информационных систем и методами интеллектуализации информационных технологий, научным работникам, а также исследователям интересующимся проблемами построения и применения систем искусственного интеллекта в задачах управления и эксплуатации современных информационных систем.

УДК 004.89
ББК 32.813

ISBN 978-5-9518-0680-2

© А.А. Карасёв, В.А. Старых, 2015
© НИУ ВШЭ, 2015

Содержание

Предисловие 5
Введение 13

Глава 1
Разрешение инцидентов: программный инструментарий 16
1.1. Приобретение экспертных знаний 16
1.2. Подходы и программные инструменты 18
1.3. Существующие программные инструменты: анализ 22
1.3.1. IBM Tivoli Monitoring 22
1.3.2. IBM Tivoli Service Request Manager 24
1.3.3. Microsoft System Center Service Manager 26
1.3.4. HP Service Manager 29
1.3.5. BMC Remedy IT Service Management Suite 31
1.3.6. ManageEngine ServiceDesk Plus 33
1.3.7. OTRS ITSM 35
1.4. Результаты сравнительного анализа 37
1.5. Выводы 42

Глава 2
Формализация и классификация экспертных знаний об инцидентах в ИС: модели и методы 44
2.1. Онтологический подход в формализации экспертных знаний . . . 44
2.1.1. Классификация онтологий 46
2.1.2. Формальная модель онтологий 48
2.1.3. Онтологические системы 51
2.1.4. Назначение онтологий 52
2.2. Концептуальная модель представления знаний об инцидентах в ИС 53
2.2.1. Таксономия объектов 55
2.2.2. Классификация и структурирование знаний об инцидентах в ИС 62
2.3. Математическая модель представления знаний об инцидентах в ИС 72
2.4. Выводы 79

Глава 3
База экспертных знаний об инцидентах в ИС: реализация 81
3.1. Средства программной реализации системы приобретения и представления экспертных знаний 81
3.1.1. Функциональные возможности Thinkmap SDK 83
3.1.2. Компоненты Thinkmap SDK 84
3.2. Объекты и структуры данных онтологии обработки инцидентов в ИС 86
3.3. Принципы построения представлений 98
3.4. Компоненты графического интерфейса 103
3.5. Выводы 108

Глава 4
Апробация системы приобретения и представления экспертных знаний 110
4.1. Информационное наполнение базы знаний 110
4.2. Построение OWL-онтологии 111
4.3. Запросы к OWL-онтологии 116
4.4. Программа испытаний 122
4.5. Результаты апробации 123
4.5.1. Количественная оценка показателей с использованием метрик 124
4.5.2. Оценка эффективности 126
4.6. Выводы 130

Заключение 133
Список абревиатур 135
Глоссарий 137
Литература 140

Приложение А
Фрагменты исходных текстов системы приобретения и представления экспертных знаний 147

Приложение Б
Структуры данных базы экспертных знаний об инцидентах в ИС и фрагменты их информационного наполнения 159

Приложение В
Испытания системы приобретения и представления экспертных знаний: методика 218

Предисловие

Сегодняшние реалии в развитии современных информационных систем (ИС) таковы: они становятся всё сложнее, их эксплуатация всё дороже. Как следствие, всё острее встаёт проблема повышения эффективности, достоверности и скорости принятия решений по локализации и диагностике при проявлении дефектов ИС в ходе её эксплуатации, и, наконец, в скорости выработки управленческих воздействий на эксплуатируемую систему. Решение проблемы лежит в автоматизации наиболее рутинных и трудоёмких процессов обеспечения эксплуатации современных ИС. Но, говоря об автоматизации, мы вынуждены искать способы повышения её эффективности. Современные тенденции в этой области – это использование средств интеллектуализации обработки данных о функционировании ИС с целью повышения эффективности принятия решений в процессах эксплуатации ИС, и в частности, в использовании автоматизированных систем, основанных на знаниях. Любой сбой в ИС, любой факт нарушения её функциональности, зафиксированные и соответствущим образом оформленные специалистами службы эксплуатации, уже есть знания. Но, знания необходимо уметь извлекать, фиксировать, как-то систематизировать, хранить, обеспечивать к ним доступ и обрабатывать. И это является, по сути, целым набором технологических проблем, требующих отдельных специализированных решений, последовательно раскрываемых в данной книге. Применительно к рассматривамой проблеме повышения эффективности принятия решений по совершенствованию процессов эксплуатации современных ИС, любое решение основывается на знаниях об объекте ИС, которым, в зависимости от состава и характера решаемых задач, может являться системное или прикладное программное обеспечение или

отдельный модуль в их составе, сетевое оборудование ИС и его компоненты, структуры данных и сами данные. Такие знания, используемые в ходе эксплуатации ИС, делятся на две группы: *нормативные*, – описывающие штатный режим работы ее компонентов на основании различных справочников, а также технической и эксплуатационной документации, и *уникальные*, – получаемые в результате разрешения инцидентов[1], возникающих непосредственно в процессе эксплуатации. Использование нормативных знаний в большинстве случаев не позволяет учитывать и оперативно обрабатывать различные аспекты взаимодействия конкретных аппаратных и программных компонентов в составе действующей ИС. Уникальные знания, накопленные в процессе эксплуатации ИС, также называемые экспертными знаниями[2], обладают большей ценностью, однако, из-за отсутствия эффективных инструментальных средств их фиксации – приобретения, обработки и накопления с целью последующего использования, возможность их широкого применения ограничена. Таким образом, центральной проблемой создания таких автоматизированных систем, основанных на знаниях, остаётся разработка инструментарного прикладного программного обеспечения, обеспечивающего решение конкретных задач повышения эффективности принятия решений в процессах эксплуатации ИС. При этом эффективность таких инструментов определяется не столько их производительностью, сколько удобством их функционального развития, способностью к эволюции, и в частности, – устойчивостью к изменениям эксплуатируемой ИС в течение жизненного цикла. Сам факт перехода на новую технологическую парадигму повышения эффективности, достоверности и скорости принятия решений по локализации и диагностике при проявлении дефектов ИС в ходе её эксплуатации, с использованием автоматизированных систем, основанных на знаниях, закономерен, однако, ситуация при этом зачастую оказывается не вполне благоприятной для разработчиков подобных инструментальных систем и их потребителей в силу следующих основных причин.

---

[1] Под инцидентами здесь подразумеваются любые события, оказывающие отрицательное воздействие на нормальное (штатное) функционирование подсистемы или сервиса ИС [11].

[2] Основой их формирования являются, как правило, эксперты – высокопрофессиональные специалисты в области эксплуатации и поддержки функциональности современных ИС.

1. *Существование*. На рынке прикладного программного обеспечения, обеспечивающего решение целого ряда задач повышения эффективности принятия решений в процессах эксплуатации ИС, существует некоторый набор программных инструментальных средств, в разной степени использующих подход, основанный на использовании баз экспертных знаний, которые можно отнести к одному из трёх типов – инцидентов, проблем и их решений, и библиотек справочных и информационно-аналитических материалов. Каждый программный инструмент подразумевает использование базы знаний, как правило, одного типа.
2. *Отсутствие*. Эффективность использования таких инструментов имеет весьма низкий уровень, вследствии отсутствия оснований их функциональной интеграции, а именно: отсутствия внятной системы классификации инцидентов для формирования баз знаний *первого* типа, отсутствия информации о причинно-следственных связях для всех инцидентов, вызванных какой-то проблемой, – для формирования баз знаний *второго* типа, и наконец, – отсутствия механизма создания и поддержания в актуальном состоянии связей между записями в базах данных инцидентов и проблем, а также – в базах знаний *третьего* типа.
3. *Формализация*. Для создания такого инструментарного прикладного программного обеспечения, позволяющего интегрировать функциональность баз знаний трёх типов необходим соответствующий аппарат формализации и классификации экспертных знаний в данной предметной области –автоматизированные интеллектуальные системы повышения эффективности принятия решений в процессах эксплуатации ИС, основанные на знаниях.

Программные инструменты повышения эффективности принятия решений в процессах эксплуатации ИС, события и инциденты, базы экспертных знаний, онтологии экспертных знаний, классификация инцидентов составляют основное содержание книги. В книге не предполагается детально рассматривать конкретные реализации используемых подходов для практических реализаций в построении подобных программных инструментов и их компонентов. Подобный анализ явился предметом ряда аналитических материалов, опубликованных в России [2, 8, 15–17, 29, 33] и за рубежом [47, 48–50].

С учетом этого обстоятельства, в начале книги дана лишь общая характеристика баз экспертных знаний и инцидентов, как объектов их содержимого, экспертных баз проблем, их решений и библиотек справочных и информационно-аналитических материалов. Основная же часть книги посвящена описанию применения и проектирования онтологического подхода для формализации и организации баз экспертных знаний в предметной области обработки инцидентов в ИС. Структурирование экспертных знаний предметной области обработки инцидентов в ИС осуществляется с помощью двух независимых классификаторов инцидентов. Применение онтологического подхода и предложенных классификаторов инцидентов позволило разработать концептуальную и математическую модели представления знаний об инцидентах в ИС, на основе которых разработано прикладное программное обеспечение инструментария.

В книге принят следующий порядок изложения. В *первой главе* выполнен анализ программных инструментов организации баз экспертных знаний, полученных при обработке инцидентов в ИС. Предложены критерии сравнения существующих подходов и программных инструментов, включающие в себя используемый формат хранения данных, способы и источники наполнения баз знаний, наличие предустановленных шаблонов и записей о типовых инцидентах, наличие механизма создания и поддержания причинно-следственных связей между соответствующими записями в базах знаний, возможность интеграции с ресурсами сторонних поставщиков. На основании данных критериев проведен сравнительный анализ программных инструментов ряда производителей.

Типовым решением является раздельное хранение собранной информации в базах данных инцидентов, проблем и их решений, предназначенных преимущественно для специалистов службы технической поддержки (СТП). При этом подробное описание способа разрешения в записи об инциденте в большинстве случаев не приводится. Базы данных проблем, напротив, содержат подробное описание корневых причин возникновения инцидентов и способов их разрешения, но не содержат информации обо всех инцидентах, вызванных той или иной проблемой.

Другим стандартным инструментом являются базы знаний, представляющие собой библиотеки информационно-аналитических ста-

тей, содержащих слабоструктурированную информацию, как правило, разрозненных.

Общим недостатком существующих программных инструментов является отсутствие механизма создания и поддержания в актуальном состоянии связей между записями в базах данных инцидентов и проблем, а также в имеющихся информационных базах знаний. Существующие программные инструменты не позволяют администратору ИС видеть причинно-следственные связи между событиями, приводящими к нарушению функционирования ИС, или по запросу получать ответ на следующие группы вопросов: какие ранее возникшие инциденты и проблемы могли привести к возникновению данного инцидента, сколько зарегистрированных инцидентов связано с отдельно взятым объектом в составе ИС или к возникновению каких инцидентов может привести известная проблема.

Во *второй главе* проведено исследование возможности применения онтологического подхода для формализации и организации баз экспертных знаний в предметной области обработки инцидентов в ИС, при котором знания, накопленные в ходе эксплуатации ИС, организуются в виде таксономии, описывающей иерархическую систему понятий, связанных друг с другом отношениями с определенной семантикой, что позволяет структурно организовывать сущности в составе онтологии в виде графа. В общем случае таксономия объектов предметной области включает в себя следующие элементы:

- *понятия-сущности*, соответствующие значимым объектам рассматриваемой области деятельности;
- *понятия-отношения*, соответствующие связям между объектами;
- *понятия-свойства*, описывающие значимые параметры и характеристики объектов или их отношений.

В результате, использование двух независимых классификаторов для структурирования знаний предметной области обработки инцидентов в ИС позволяет, в зависимости от специфики решаемых задач:

- получать подмножество инцидентов, связанных с заданным объектом, и последовательно, в соответствии с распределением по уровням модели ISO/OSI, просматривать их;
- выбирать нужный уровень модели для указанной подсистемы или компонента ИС, после чего просматривать полученное подмноже-

ство инцидентов для нахождения связанных с искомым объектом.

Применение онтологического подхода и предложенных классификаторов инцидентов позволило разработать концептуальную и математическую модели представления знаний об инцидентах в ИС.

В зависимости от порядка просмотра уровней, модель позволяет получать подмножества инцидентов, связанных с заданным объектом (прямой порядок «сверху-вниз»), или определять подмножество объектов, связанных с выбранным инцидентом (обратный порядок «снизу-вверх»). База экспертных знаний, организованная с использованием предложенной математической модели, содержащей, в том числе, формальное определение инцидента, позволяет описывать все доступное множество объектов и инцидентов – содержит все необходимые знания для обработки инцидентов, возникающих в ИС.

В *третьей главе* сформулированы требования, предъявляемые к подобным программным инструментальным средствам в соответствии с полученными ранее моделями представления знаний об инцидентах в ИС. В их число входят как архитектурные – возможность реализации в виде standalone-приложения и Java-апплета, поддержка внешних источников данных, так и обусловленные спецификой используемого подхода – поддержка логической организации сущностей и отношений в соответствии с разработанными методами классификации и структурирования знаний об инцидентах в ИС.

Для реализации действующего прототипа выбран комплект средств разработки на основе свободного ПО Thinkmap SDK, предназначенный для создания прикладных программ, решающих задачу обработки и визуализации больших объемов данных.

В соответствии с описываемыми концептуальной моделью свойствами сущностей и отношений, определены соответствующие структуры данных. В книге они используются для описания в формате XML шести типов понятий-сущностей, четырех типов иерархических и двух типов специфических для предметной области понятий-отношений, включенных в ранее описанную таксономию объектов.

Все объекты, обрабатываемые и визуализируемые с помощью разработанной системы, представляются в виде графовой модели. В каждый момент времени пользователю доступно определенное подмножество объектов (фрагмент графовой модели) с возможно-

стью перехода к новому подмножеству при необходимости. Такое подмножество, выбранное на основе значений указанных параметров, рассматривается в данной книге как представление. Для построения представлений (отбора всех входящих в них объектов и связей между ними) предложены специальные условия обхода графа, реализованные с использованием инструментального средства разработки Thinkmap SDK.

С позиции онтологического подхода, данная операция аналогична поиску термина, представляющего интерес для решения определенной задачи, и формирования соответствующего контекста. Под контекстом здесь подразумевается часть (срез) онтологии, значимая для ее решения.

В *четвертой главе* приведены результаты апробации разработанного программного инструментария приобретения и представления экспертных знаний, выполненной в процессе эксплуатации программно-аппаратного комплекса Федерального центра информационно-образовательных ресурсов [53].

В книге приведены примеры записей в базе знаний, созданных средствами программного инструментария в ходе опытной эксплуатации, а также количественные показатели для таких параметров, как число зафиксированных инцидентов, описанных объектов и ряда других в соответствии с разработанной концептуальной моделью представления знаний об инцидентах в ИС.

Проверка целостности и непротиворечивости базы экспертных знаний после ее информационного наполнения средствами разработанной системы выполнена с использованием среды Protégé, для чего XML-описание разработанной ранее таксономии объектов было преобразовано в соответствии со стандартами языка описания онтологий OWL [67]. Дано описание основных структурных элементов полученной OWL-онтологии и их соответствие элементам таксономии объектов; приведены основные этапы построения OWL-онтологии; представлены примеры выполнения специальных запросов, построенных в соответствии с синтаксисом OWL к разработанной онтологии.

Для определения соответствия разработанной системы приобретения и представления экспертных знаний сформулированным функциональным требованиям приведена программа испытаний, включа-

ющая в себя различные сценарии ее использования. Состав, порядок и методика проведения испытаний приведены в приложении В.

Для оценки эффективности разработанного инструментария применен подход определения количественных мер измерения – метрик – значимых параметров, предлагаемых методологией ITSM. Оценка эффективности проводилась на основании таких метрик, как процент инцидентов, решенных на первой линии поддержки или проактивно, а также как среднее время разрешения и общее число инцидентов, разрешенных с использованием экспертных знаний, содержащихся в полученной базе знаний.

Общая оценка повышения эффективности выполнена на основе расчета интегрального показателя, учитывающего значения перечисленных метрик и приоритетов, определяющих степень их значимости в целом.

Увеличение значений метрик, определяющих положительные факторы при разрешении инцидентов в ИС с использованием разработанной базы знаний – точность первоначальной классификации, правильность и скорость их разрешения, а также уменьшение значений метрик, оценивающих негативные факторы, – невозможность определить корневую причину возникновения, неверную классификацию, и, как следствие – рост интегрального показателя в целом, подтверждают возможность использования системы приобретения и представления экспертных знаний для достижения поставленной цели.

В Заключении определены направления дальнейшего развития подхода, предлагаемого в данной книге.

Введение

Знания, используемые в ходе эксплуатации ИС, делятся на две группы: *нормативные*, описывающие штатный режим работы ее компонентов на основании технической и эксплуатационной документации, а также различных справочников, и *уникальные*, получаемые в результате разрешения инцидентов, возникающих непосредственно в процессе эксплуатации. Использование нормативных знаний в большинстве случаев не позволяет учесть различные аспекты взаимодействия конкретных аппаратных и программных компонентов в составе действующей ИС. Уникальные знания, накопленные в процессе эксплуатации ИС, также называемые экспертными знаниями, с этой точки зрения обладают большей ценностью. Однако, из-за отсутствия эффективных инструментальных средств фиксации – приобретения, обработки и накопления с целью последующего использования – возможность их применения ограничена.

В настоящее время на рынке прикладного программного обеспечения не существует интегрированных инструментов организации баз экспертных знаний и их наполнения, предоставляющих реализацию концепции единой точки доступа ко всему объему знаний, накопленных в ходе создания и эксплуатации информационной системы. Это приводит к снижению эффективности решений, принимаемых в ходе эксплуатации ИС, в том числе, при разрешении возникающих инцидентов. Релизация механизмов обеспечения интегрированности предполагает наличие механизмов, обеспечивающих ирформационную связность содержимого баз экспертных знаний трёх типов: инцидентов, проблем и их решений, библиотек справочных и информационно-аналитических материалов. Эффективность, при этом, зависит как от качества данной информациии (базы экспертных знаний, содержащие информацию об ицидентах), количества ис-

точников доступной информации (внутренние руководства аппаратного обеспечения, справочные системы программного обеспечения, справочные руководства по операциям, конфигурационные базы данных, системы обработки изменений и запросов пользователей, а также базы данных и библиотеки информационных статей службы технической поддержки, пособия системного администратора, документация по процессам), так и от квалификации специалистов. Но информация данного рода в большинстве случаев является закрытой, то есть охраняемой корпоративными интересами, поэтому практически не публикуется в открытых источниках. Библиотека ITIL [47–49] зачастую дает мало полезного в решении повседневных типовых задач сегодняшних служб технической поддержки, так как процессы ITIL описываются на более высоком уровне абстракции. С этой точки зрения данная книга является, с одной стороны, примером иллюстрации практического решения типовой задачи СТП – разрешения инцидентов, в части использования соответствующего программного инструментария, а с другой – её изложение построено на основе строгого системного подхода, в основе которого находятся классификация и онтология.

Целью монографии является описание процесса разработки и использования программного инструментария, позволяющего повысить эффективность, достоверность и скорость принятия решений по локализации и диагностике при проявлении дефектов ИС в ходе её эксплуатации на основе использования знаний о нарушении функциональности ИС, представленных в виде инцидентов. По сути, в терминологии ITIL, речь идёт об описании всего процесса создания и использования базы данных управления инцидентами IMDB (Incident Management Database).

В книге представлены решения следующих задач:

1. Разработка методов классификации и структурирования знаний об инцидентах в ИС (*на основе применения структурного подхода и эталонной модели взаимодействия открытых систем ISO/OSI)*.
2. Разработка математических моделей и методов формализации экспертных знаний об инцидентах в ИС.
3. Разработка онтологии инцидентов в ИС (*на основе предложенных методов и моделей)*.

4. Разработка программного инструментария для организации и наполнения базы экспертных знаний об инцидентах в ИС.
5. Апробация программного инструментария приобретения и представления экспертных знаний об инцидентах в ИС.
6. Определение количественных показателей повышения эффективности использования разработанного инструментария.

ГЛАВА 1

Разрешение инцидентов: программный инструментарий

1.1. Приобретение экспертных знаний

Задача обеспечения штатного режима функционирования информационной системы включает в себя обслуживание аппаратного и программного обеспечения, распределение информационных, временных и людских ресурсов. Практически все методологии, разработанные для обеспечения штатного режима функционирования ИС, базируются на подходе, описанном в библиотеке ITIL, содержащей набор рекомендаций, обобщающих лучшие практики для отрасли в целом.

Особое значение в ней играют задачи [47–49, 61], относящиеся к группе эксплуатации услуг (Service Operation), в том числе, задачи разрешения инцидентов и проблем. При этом инцидент определяется как любое событие, не являющееся частью стандартного функционирования ИС и способное привести к сбою или ухудшению параметров ее функционирования. Корневая причина возникновения одного или нескольких инцидентов, не описанная к моменту их возникновения, определяется как проблема. Основной организационной единицей, взаимодействующей с этой группой задач, является служба технической поддержки (СТП, Service Desk в терминологии ITIL).

Данная служба предназначена не только для поддержки клиентов или пользователей предоставляемых ИТ-услуг, но и для обеспечения бесперебойной работы сотрудников других подразделений организации и штатного функционирования собственных информацион-

ных систем. Именно этим объясняется наличие такой службы в том или ином виде, в составе подавляющего большинства организаций. К функциям СТП относится сбор и обработка данных – в том числе и в автоматизированном режиме – о возникающих инцидентах, поиск способов их разрешения и регистрация полученной информации.

Для выполнения этой функции сегодня широко используются интеллектуальные информационные системы [3, 7, 43], в том числе, экспертные системы и системы поддержки принятия решений. При этом большое значение имеет использование экспертных знаний об особенностях функционирования конкретной ИС, накопленных в ходе ее эксплуатации и являющихся для нее уникальными. Связано это, в первую очередь, с прогрессирующей тенденцией нехватки квалифицированных специалистов [57, 59–60].

Современные реалии в проблематике управления и администрирования ИС таковы, что существует реальная и постоянная проблема неэффективности решений, принимаемых при разрешении инцидентов в ИС. Основными причинами этого являются:

- недостаток полной и всеобъемлющей информации о программно-аппаратном комплексе – отсутствие единой точки доступа к центральной базе знаний, содержащей информацию о составе и компонентах программно-аппаратного комплекса ИС и доступных ресурсах;
- необходимость повторного многократного разрешения однотипных инцидентов вместо устранения корневой причины их возникновения;
- использование реактивной стратегии разрешения инцидентов (после их возникновения, приведшего к ухудшению параметров функционирования ИС) вместо проактивной, позволяющей заранее устранять корневые причины инцидентов на основе имеющейся информации;
- недостаток информации, приводящий к принятию решений на основе предположений, а не ранее зафиксированных данных и знаний;
- сильная зависимость организации от СТП – потеря значительного объема незафиксированных экспертных знаний в случае перемены места работы сотрудниками [21, 33 с. 36–37].

Таким образом, *эффективность обработки возникающих инцидентов в ИС зависит не только от уровня квалификации сотрудников, но и*

от возможности приобретения, структурирования и последующего использования экспертных знаний, полученных в процессе эксплуатации ИС[3] *с целью сокращения времени обнаружения, классификации и разрешения однотипных инцидентов и схожих проблем в дальнейшем* [22].

1.2. **Подходы и программные инструменты**

В настоящее время существует ряд механизмов, позволяющих накапливать информацию об особенностях построения и функционирования информационных систем. Так, для учета программно-аппаратных компонентов в составе ИС и доступных ресурсов используются конфигурационные базы данных (CMDB, Configuration Management Database в терминологии ITIL). Они позволяют хранить информацию о характеристиках и взаимосвязях конфигурационных единиц (configuration items), каждая из которых может рассматриваться в качестве самостоятельного объекта в составе ИС. В некоторых случаях данная информация может дополняться сведениями о возникавших проблемах, затронувших ту или иную конфигурационную единицу, или более точно для рассматриваемой проблематики – это использование IMDB (базы данных управления инцидентами).

Инструментом, предназначенным для хранения данных об известных событиях, приводящих к нарушению функционирования ИС и способах их разрешения, являются базы данных инцидентов и проблем IMDB. Данные об инцидентах могут поступать из различных источников: от пользователей предоставляемых услуг, сотрудников организации или специализированных систем мониторинга. Обнаруженный тем или иным способом инцидент должен регистрироваться сотрудниками СТП в базе данных инцидентов. В некоторых случаях средства системного мониторинга обеспечивают возможность автоматизированного создания записи об инциденте на основе предустановленных шаблонов. Запись об инциденте в базе данных включает в себя дату и время создания, имя и организационную единицу пользователя, создавшего запрос (в случае наличия), исходное описание, предоставленное пользователем, и ряд других полей, в том числе:

- уникальный идентификатор, ID;

[3] Здесь и далее в работе термины «фиксация» и «приобретение» употребляются как синонимы.

- класс – определяет, относится ли данный инцидент к программным или аппаратным сбоям, известным ошибкам или является запросом на обслуживание;
- категория – используется для определения подсистемы, компонента ИС или предоставляемой услуги, с которой связан данный инцидент. В зависимости от требуемой степени детализации может содержать, в качестве примера, версию или название модуля в составе программного компонента;
- приоритет – определяется влиянием на функционирование ИС и срочностью, с которой необходимо обеспечить его разрешение. Пример матрицы, позволяющей определить приоритет инцидента, приведен в табл. 1;
- объект или группа объектов (конфигурационных единиц) в составе ИС, предположительно являющиеся причиной возникновения инцидента;
- объект или группа объектов (конфигурационных единиц) в составе ИС, на которые влияет инцидент;
- ссылка на известную ошибку или проблему, приведшую к возникновению инцидента и зарегистрированную в базе данных проблем;
- для стандартных инцидентов или инцидентов, для которых установлена причина возникновения – необходимые сведения и рекомендации по разрешению [33, с. 120].

Табл. 1. Пример матрицы приоритетов инцидентов

Степень воздействия / Срочность	**Высокая**	**Средняя**	**Низкая**
Высокая	Приоритет: Критический Время устранения: <12 ч.	Приоритет: Высокий Время устранения: <24 ч.	Приоритет: Средний Время устранения: <48 ч.
Средняя	Приоритет: Высокий Время устранения: <24 ч.	Приоритет: Средний Время устранения: <48 ч.	Приоритет: Низкий Время устранения: <72 ч.
Низкая	Приоритет: Средний Время устранения: <48 ч.	Приоритет: Низкий Время устранения: <72 ч.	Приоритет: Низший Время устранения: по согласованию

С активным развитием интранет-технологий и корпоративных порталов организаций появился дополнительный способ внесения информации об инцидентах в базы данных. Он подразумевает возможность пользователей самостоятельно, без непосредственного взаимодействия с СТП, регистрировать возникающие инциденты. В зависимости от действующих правил и политик безопасности пользователи могут создавать записи об инцидентах (в том числе указывая связанные с ним объекты, определяя степень его воздействия и предлагая возможные пути разрешения[4]) и просматривать существующую базу знаний об инцидентах или ее общедоступную часть с целью поиска нужного решения. Подобный подход позволяет снизить нагрузку на сотрудников СТП и повысить эффективность работы пользователей – типовые инциденты могут быть разрешены ими самостоятельно значительно быстрее при отсутствии необходимости ожидания ответа СТП, которое в случае низкого приоритета может длиться достаточно долго. Набор зарегистрированных инцидентов, доступных пользователям, определяется каждой организацией индивидуально.

Запись в базе данных инцидентов не всегда содержит последовательность действий по его разрешению. В большинстве случаев данное поле заполняется только для типовых инцидентов. Для хранения информации о причинах возникновения и возможных способах разрешения используется база данных проблем [6, с. 67–69]. Она содержит не только подробное руководство по разрешению возникающих инцидентов на основе задокументированных в процессе эксплуатации ИС решений, в том числе и обходных, применяющихся для экономии времени и других ресурсов организации, но и рекомендации по устранению их корневых причин.

Проблема, рассматриваемая с одной стороны, как первопричина возникновения, а с другой – как результат, в большинстве случаев связана с несколькими различными инцидентами, оказывающими сходное влияние на функционирование ИС. Это означает, что для эффективного использования баз данных инцидентов и проблем они должны поддерживать механизм организации связей типа «многие

---

[4] При этом новая запись должна быть утверждена сотрудниками СТП как предлагающая корректное и эффективное решение ранее не зафиксированного инцидента, не приводящее к появлению других инцидентов.

ко многим» – запись об инциденте должна быть связана со всеми записями о проблемах, которые могли привести к его возникновению, а запись о проблеме – со всеми записями о вызванных ею инцидентах. Вместе с тем, запись о проблеме может рассматриваться независимо от связанных с ней записях об инцидентах. Определение причин возникновения проблемы может продолжаться после успешного разрешения, связанного с ней инцидента. С точки зрения сотрудников СТП, база данных проблем может рассматриваться как набор информации, организованный и хранимый таким образом, чтобы к нему можно было быстро обращаться с помощью простых ссылок по легко обнаруживаемым признакам новых инцидентов [33, с. 124]. Это позволяет говорить об объединении баз данных инцидентов и проблем в общую базу знаний.

Большое значение при регистрации записи об инциденте и добавлении необходимых связей с известными ошибками имеет классификация, как механизм понимания причины возникновения инцидента и связанных с ним объектов в составе ИС, что позволяет установить последовательность действий для его разрешения. В процессе эксплуатации ИС значительное количество инцидентов возникает регулярно, при этом действия для их разрешения хорошо известны. Для других требуется использование стандартной процедуры привязки классификационных данных инцидента к известным ошибкам и задокументированным проблемам. В случае обнаружения описания аналогичного инцидента в базе знаний выполняется набор ранее зафиксированных и строго регламентированных действий для его разрешения, что позволяет значительно упростить процесс.

На основании рассмотренных аспектов можно выделить следующие основные критерии сравнения подходов и программных инструментальных средств, использующихся для фиксации экспертных знаний в процессе эксплуатации ИС:

- формат хранения: определяет состав программных инструментальных средств, использующихся для создания, редактирования и поиска записей в базах знаний – текстовые редакторы, встроенные средства редактирования, СУБД;
- способы и источники наполнения баз данных инцидентов и проблем, возможность автоматизированного создания записей средствами системного мониторинга;

- возможность доступа к используемой базе знаний об инцидентах или ее части, для сотрудников организации, не относящихся к СТП, и для внешних пользователей;
- наличие предустановленных шаблонов и наборов типовых, для предметной области инцидентов с рекомендациями по их разрешению во встроенных базах знаний;
- наличие механизма создания и редактирования связей между записями в базах данных инцидентов и проблем;
- поддержка графического отображения данных об инцидентах и их взаимосвязях, а также наличие визуального редактора, позволяющего вносить необходимые изменения в графическом режиме;
- встроенный механизм поиска по ключевым словам, кодам классификаторов и связанными с инцидентами объектами;
- наличие интерфейсов, позволяющих интегрировать имеющиеся базы знаний с ресурсами сторонних поставщиков;
- поддержка русского языка в интерфейсе пользователя и записях в базах знаний.

1.3. **Существующие программные инструменты: анализ**

В результате проведенного исследования современного состояния рынка программных инструментальных средств, предназначенных для организации и наполнения баз экспертных знаний, полученных в процессе функционирования ИС, для сравнительного анализа выбраны программные средства таких производителей, как BMC, HP, IBM, Microsoft; свободное ПО ManageEngine ServiceDesk Plus Standard Edition и ПО с открытыми исходными текстами [19] OTRS ITSM. Сравнительный анализ проводился на основании критериев, определенных в предыдущем разделе.

1.3.1. **IBM Tivoli Monitoring**

Программное инструментальное средство IBM Tivoli Monitoring предназначено для отслеживания возникновения инцидентов в информационной системе на основе заданных значений параметров объектов в составе ИС и выполнения действий в соответствии с пользовательскими сценариями.

Одним из ключевых понятий Tivoli Monitoring является ситуация (Situation), аналог инцидента в терминологии ITIL. Tivoli

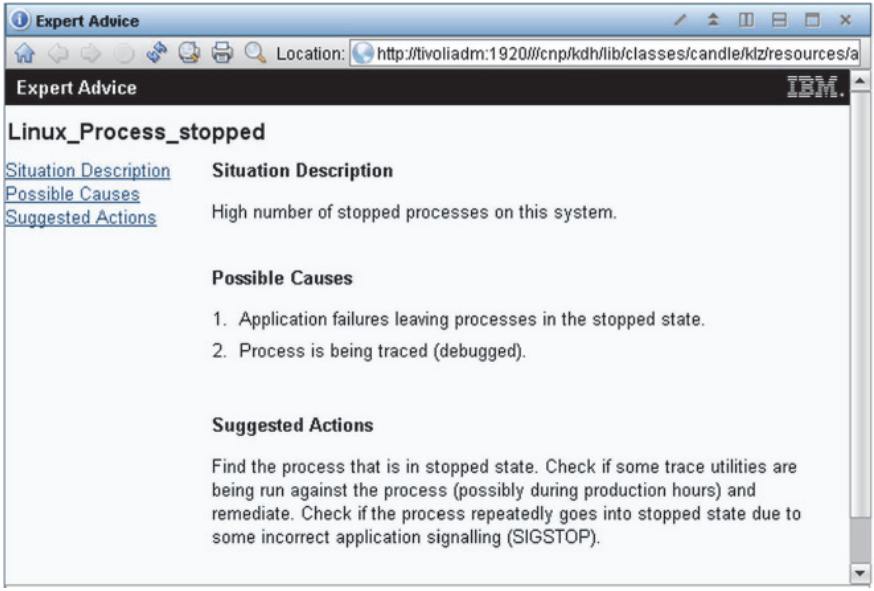

Рис. 1. Экспертная рекомендация IBM Tivoli Monitoring

Monitoring поставляется с предустановленным набором типовых ситуаций со стандартным описанием, включающим в себя возможные причины их возникновения для различных ИС, и рекомендуемые действия для их разрешения.

Встроенные инструментальные средства позволяют администратору создавать собственные ситуации на основе указанных параметров. Каждая ситуация может сопровождаться подробным описанием, содержащим перечень затронутых компонентов и возможные варианты разрешения. Соответствующая область рабочего пространства называется экспертной рекомендацией (Expert Advice), рис. 1.

Для создания подобной записи требуются права администратора с возможностью редактирования ситуаций, что не позволяет предоставить доступ к формируемой базе данных инцидентов внешним пользователям. Предустановленные или созданные администратором описания хранятся в виде отдельных .htm файлов в директории данных. Записи создаются путем редактирования соответствующих текстовых полей, либо внесением изменений непосредственно в .htm

файл с помощью произвольного редактора. Визуальный редактор ситуаций не предусмотрен.

Каждое описание может быть связано только с одной ситуацией. Отсутствие выделенных баз данных инцидентов и проблем не позволяет определить и задокументировать влияние и взаимосвязи различных инцидентов, проблем и связанных с ними объектов. Также отсутствует возможность интегрировать управляющую информацию, собранную несколькими экземплярами Tivoli Monitoring для различных информационных систем.

Описание ситуации доступно администратору только в том случае, если она является открытой на данный момент. Таким образом, администратор может достаточно быстро разрешить повторно возникшую ситуацию на основании ранее собранных данных. Однако, возможность провести сравнительный анализ корневых причин, приведших к возникновению текущих и успешно разрешенных закрытых ситуаций, не предусмотрена. Использование русского языка поддерживается не всеми компонентами (агентами) в составе IBM Tivoli Monitoring.

1.3.2. **IBM Tivoli Service Request Manager**

Программный инструмент семейства IBM Tivoli, Tivoli SRM предназначен для автоматизации задач группы эксплуатации услуг в терминологии ITIL [55].

Основными типами объектов, с которыми работает Tivoli SRM, являются инциденты, проблемы и запросы на обслуживание. Каждая запись включает в себя подробную информацию о соответствующем объекте, существующие взаимосвязи с другими объектами (например, для инцидента – проблему, ставшую причиной его возникновения, поступившие в связи с ним запросы на обслуживание, а также связанные объекты в составе ИС) и возможные варианты решения, а также служебную информацию: дату создания и время разрешения, приоритет и ответственного сотрудника СТП. Записи об объектах всех типов хранятся в единой базе данных (поддерживаются СУБД IBM DB2, Microsoft SQL Server, Oracle). Общая архитектура Tivoli SRM представлена на Рис. 2.

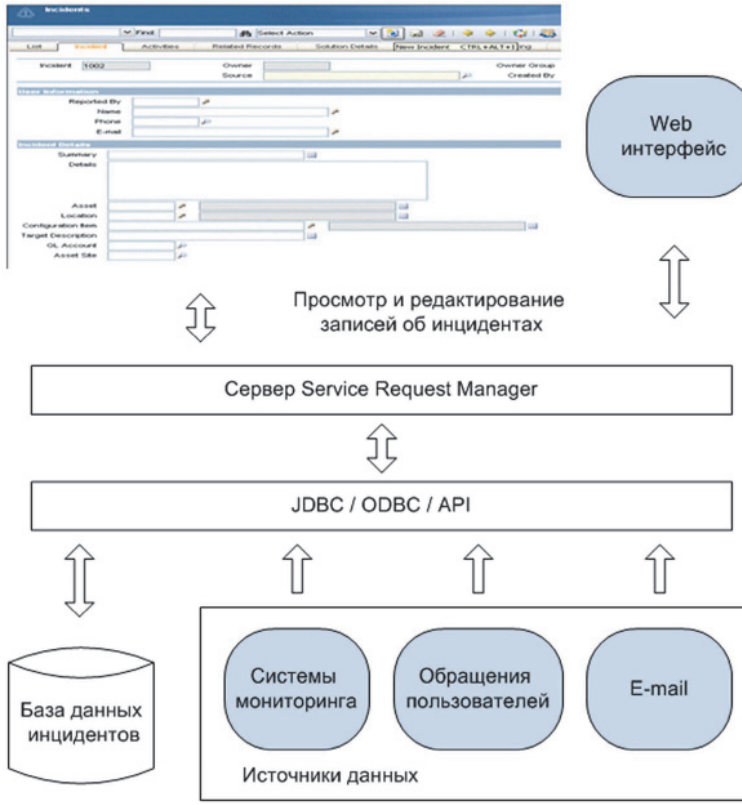

Рис. 2. Архитектура Tivoli Service Request Manager

Источниками данных для наполнения базы инцидентов могут служить обращения пользователей, специализированные приложения сторонних производителей или системы мониторинга. Сервер Tivoli SRM предоставляет собственные API для реализации унифицированного доступа к ним. Просмотр и создание записей об инцидентах осуществляется с использованием административного или Web интерфейсов.

Предустановленные наборы инцидентов или проблем в системе отсутствуют. Вместо них разработчики предоставляют набор шаблонов, которые могут использоваться при создании новых записей для уменьшения объема вводимой вручную информации. Шаблоны со-

держат следующую информацию: приоритет, класс или категорию, ответственного за разрешение и др. Данные, полученные из шаблонов, могут корректироваться до тех пор, пока соответствующие инциденты / заявки не будут разрешены / закрыты.

Связи, существующие между инцидентами, вызвавшими их проблемами и объектами в составе ИС, существуют на уровне связи записей в базе данных (в графическом интерфейсе администратора – на уровне идентификаторов или имен в соответствующих полях) и не имеют наглядного графического отображения.

Tivoli SRM позволяет осуществлять поиск по базе инцидентов, проблем и запросов на обслуживание. Функция поиска является полнотекстовой, т.е. позволяет искать заданную строку как в отдельных полях форм, так и в прикрепленных файлах. Вместе с тем, в случае поиска по базе инцидентов (подразумевается логическое структурирование записей об инцидентах, а не физическое размещение базы данных используемой СУБД), поиск в связанных с инцидентом записях о проблемах и путях ее разрешения осуществляться не будет. Встроенная система поиска позволяет выполнять запросы на русском языке.

ПО Tivoli SRM позволяет получать доступ к базам инцидентов и проблем не только администраторам системы, но и пользователям – при условии, что у них есть соответствующие права в общих настройках безопасности. Использование такого подхода (Self Service Center в терминах Tivoli SRM), с одной стороны, предоставляет заинтересованным пользователям возможность самостоятельно разрешить возникшие инциденты, а с другой – уменьшает нагрузку на службу технической поддержки. Кроме того, пользователи могут совместно участвовать в решении проблем и добавлять в общую базу знаний записи об успешно разрешенных инцидентах.

1.3.3. **Microsoft System Center Service Manager**

Семейство программных инструментальных средств Microsoft System Center [64] построено на основе применения библиотеки ITIL и методологии Microsoft Operations Framework (MOF). Входящий в него компонент Service Manager автоматизирует задачи группы эксплуатации услуг, а также деятельность службы технической поддержки. Использование данного средства подразумевает инте-

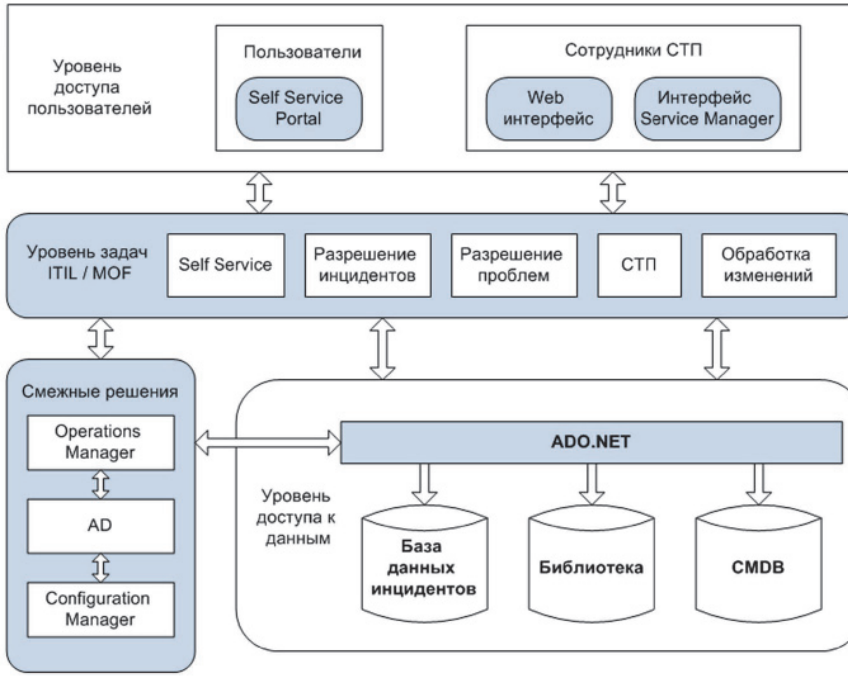

Рис. 3. Архитектура System Center Service Manager

грацию с другими компонентами System Center: Operations Manager, Configuration Manager и со службой каталогов Active Directory. Благодаря интеграции Service Manager получает доступ к информации, содержащейся в AD и конфигурационной базе данных CMDB, и дополняет ее собственной базой данных типовых инцидентов (Рис. 3).

Для хранения информации об инцидентах и способах их разрешения используется СУБД Microsoft SQL. Доступ к данным осуществляется с использованием набора классов ADO.NET в составе платформы .NET Framework. Встроенная форма регистрации инцидента включает в себя такие стандартные поля, как категория, источник возникновения, описание, приоритет и ряд других (Рис. 4). Запись об инциденте также содержит ссылки на конфигурационные единицы базы данных CMDB, с которыми он связан.

Возможность создания и редактирования связей с другими зарегистрированными инцидентами не предусмотрена. Отсутствует воз-

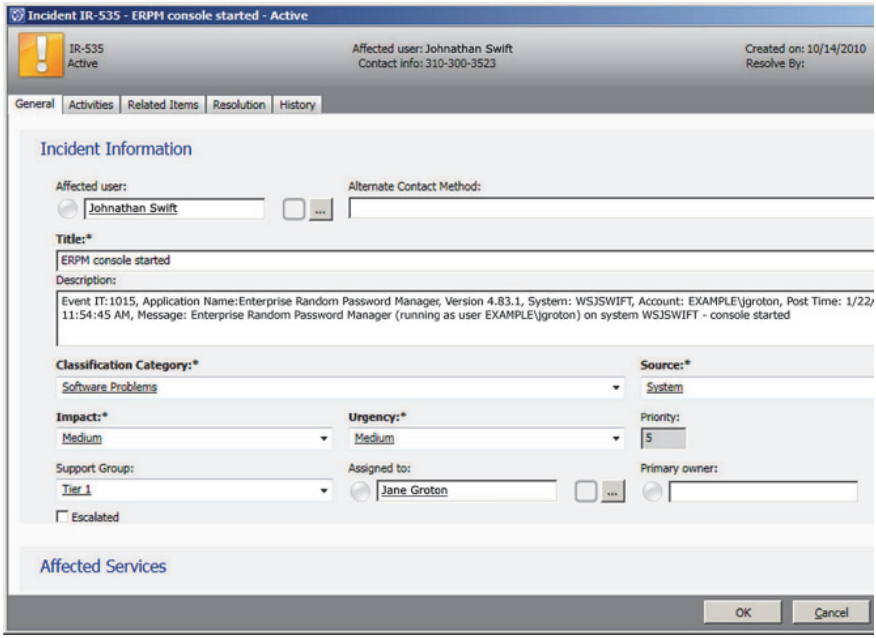

Рис. 4. Форма регистрации инцидента Service Manager

можность графического отображения и редактирования связей инцидентов со связанными объектами.

Новые инциденты в системе регистрируются администратором. Источником информации о возникших инцидентах могут быть как обращения пользователей, так и различные системы мониторинга, в том числе Microsoft System Center Operations Manager. ПО Service Manager содержит набор шаблонов, позволяющих сократить время заполнения большинства типовых полей для различных видов инцидентов.

Глобальная база знаний о зарегистрированных проблемах и путях их разрешения не связана напрямую с базой инцидентов и представляет собой библиотеку информационно-справочных статей, сгруппированных по предметным областям, поддерживающую полнотекстовый поиск, в том числе и с подключением сторонних поисковых служб.

База знаний доступна как сотрудникам СТП (при наличии соответствующего уровня доступа они могут создавать новые и редакти-

ровать существующие записи), так и конечным пользователям (только для поиска необходимой информации). Портал самообслуживания – Self-Service Portal – позволяет пользователям самостоятельно решать возникшие проблемы, используя информацию об известных неисправностях из базы знаний.

Программное средство Microsoft System Center Service Manager позволяет установить поддержку русского языка, однако, она является неполной: записи об инцидентах рекомендуется создавать на английском языке.

1.3.4. **HP Service Manager**

Платформа HP Service Manager (ранее известная как HP Operations Center) разработана в соответствии с методологией ITSM, основанной на рекомендациях библиотеки ITIL. В ее состав входит модуль Knowledge Management. Данный модуль предназначен для сбора, централизованного хранения и многократного использования знаний, накопленных в процессе эксплуатации ИС другими модулями, в том числе, обеспечивающими обработку инцидентов, проблем и изменений.

ПО Knowledge Management [56] позволяет создавать базу знаний с подробным описанием способов разрешения различных инцидентов и вызвавших их проблем. Записи в базе могут содержать не только текстовое описание (поддерживается формат RTF), но и дополнительную информацию в виде ссылок на внешние ресурсы или прикрепленных файлов (Рис. 5). Для хранения данных используется реляционная СУБД. Новые записи могут создаваться как администраторами системы, так и пользователями [5], в зависимости от настроек прав доступа к базе Knowledge Management. Автоматизированное пополнение базы знаний средствами системного мониторинга не предусмотрено.

Доступ к базе Knowledge Management может быть предоставлен внешним пользователям. В соответствии с настройками уровня доступа они могут только просматривать записи, создавать собственные или редактировать уже существующие.

Новые записи могут создаваться на основе пяти стандартных шаблонов, позволяющих сократить время и объем вводимых вручную данных:

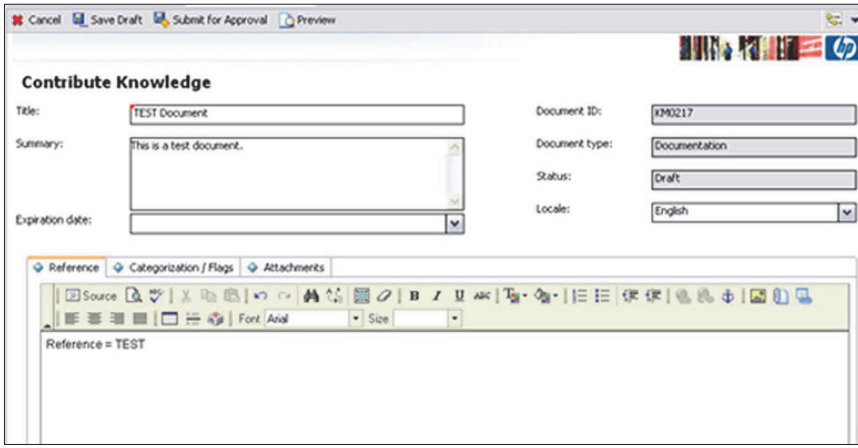

Рис. 5. Создание записи в базе знаний с использованием встроенного редактора

- сообщение об ошибке и ее причине (error message / cause),
- ссылка на внешний документ (external),
- вопрос / ответ (question / answer),
- проблема / решение (problem / solution),
- справочная информация (reference).

База знаний Knowledge Management существует отдельно от базы инцидентов и проблем: описание инцидента и способ его разрешения по умолчанию не заносятся в базу Knowledge Management[5]. Механизм создания и редактирования связей, в том числе и в графическом представлении, между инцидентами, вызвавшими их проблемами и затронутыми ими объектами, не предусмотрен.

При регистрации нового инцидента или проблемы, поиск в базе Knowledge Management может выполняться автоматически по указанным ключевым словам – например, категории инцидента или затронутого объекта. Модуль Knowledge Management поддерживает индексирование записей и осуществление полнотекстового поиска в случае, если добавляемые файлы конвертируются в html-документы. Поиск по прикрепленным документам различных форматов не поддерживается.

[5] Могут быть скопированы в нее по усмотрению пользователя после утверждения администратором системы с соответствующим изменением формы и формата представления.

На данный момент разработчик не предоставляет средств и интерфейсов для осуществления интеграции базы знаний с внешними ресурсами.

ПО HP Service Manager Knowledge Management поддерживает использование русского языка при создании записей в базе знаний и выполнении поисковых запросов.

1.3.5. **BMC Remedy IT Service Management Suite**

Семейство программных инструментальных средств BMC Remedy предназначено для автоматизации обеспечения штатного режима функционирования ИС в соответствии с методологией ITSM. В его состав входит модуль BMC Knowledge Management [58], предоставляющий сотрудникам СТП базу знаний об известных проблемах и способах их решения, а рядовым пользователям – возможность самостоятельно разрешать возникающие инциденты, используя ранее накопленные знания.

Модуль Knowledge Management предназначен для выявления, хранения и дальнейшего использования знаний, собранных в процессе эксплуатации ИС. Экспертные знания организуются в виде структурированных (созданных в соответствии с определенными шаблонами и заданными полями) записей, распределенных по каталогам и категориям. Для хранения могут использоваться как реляционные СУБД (Microsoft SQL Server, MySQL, Oracle), так и файлы в формате XML, размещенные на файловой системе в соответствующей иерархии папок.

Записи в базе Knowledge Management могут создаваться как администраторами системы, так и пользователями, обладающими достаточными для этого правами. Записи создаются на основе существующего XML шаблона, для его заполнения используется встроенный редактор, поддерживающий формат RTF и возможность добавления файлов различных форматов, в том числе аудио- и видео. Модуль Knowledge Management поддерживает пакетный экспорт / импорт данных, например, в XML или CSV файлах.

Записи в базе Knowledge Management (или их часть, помеченная как self-help) доступны сотрудникам организации и внешним пользователям, обладающим необходимыми правами доступа. При этом пользователям может быть доступен лишь ограниченный на-

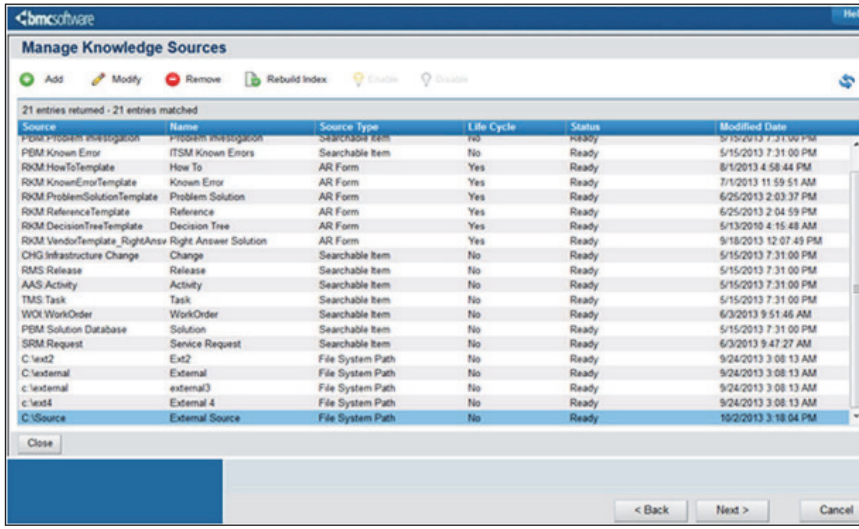

Рис. 6. Управление источниками знаний модуля BMC Remedy Knowledge Management

бор существующих полей, тогда как администраторам – все поля записи.

База Knowledge Management является самостоятельным хранилищем, размещенным отдельно от существующих баз данных инцидентов, проблем и известных решений, и не обязательно связана с ними. Модульная архитектура IT Service Management Suite позволяет устанавливать необходимые модули (Incident Management, Problem Management, Service Desk [50], Knowledge Management) в разных комбинациях независимо друг от друга. Встроенные средства создания и поддержания связей между инцидентами, объектами и записями в базе Knowledge Management отсутствуют.

На рис. 6 представлен интерфейс управления возможными источниками создания записей модуля BMC Remedy Knowledge Management.

Встроенный поисковый механизм (Hummingbird) позволяет осуществлять полнотекстовый поиск среди записей базы Knowledge Management, в прикрепленных файлах в форматах PDF или DOC, базах данных инцидентов, проблем и известных решений Remedy

ITSM Suite. Кроме того, он поддерживает индексирование сторонних источников информации, что позволяет осуществлять в них полноценный поиск.

Интеграция с существующими базами знаний сторонних поставщиков не предусмотрена. ПО BMC Remedy ITSM Suite поддерживает использование русского языка с рядом ограничений (например, не поддерживаются русские названия файлов во вложениях).

1.3.6. **ManageEngine ServiceDesk Plus**

Свободно распространяемая версия Standard Edition коммерческого программного средства ManageEngine ServiceDesk Plus [63] предназначена для автоматизации работы СТП. Коммерческие версии включают также инструментальные средства для автоматизации задач группы эксплуатации услуг.

Записи об инцидентах создаются администратором вручную на основе обращений пользователей (по телефону или e-mail) или уведомлений, генерируемых средствами системного мониторинга. Возможность автоматизированного создания записей об инцидентах на основе получаемых уведомлений не предусмотрена. Записи создаются во встроенном редакторе, содержащем набор предустановленных шаблонов для описания типовых инцидентов. При необходимости администратор может редактировать существующие шаблоны или добавлять собственные. Для хранения записей об инцидентах, проблемах и объектах (в версии Enterprise Edition) используются реляционные СУБД – Microsoft SQL, MySQL или Postgre SQL.

Информация о способах разрешения инцидентов и детальное описание проблем хранится в специально предназначенной базе знаний (knowledge base). База может пополняться как сотрудниками СТП, так и пользователями, имеющими доступ к системе. В случае создания новой записи пользователем обязательным условием ее добавления является подтверждение эффективности и безопасности предложенного метода решения ответственным специалистом (Рис. 7). База знаний доступна сотрудникам СТП, сотрудникам организации и внешним пользователям через Self Service Portal в рамках их уровня доступа: описание технически сложных инфраструктурных решений может быть удалено из области видимости рядовых пользователей.

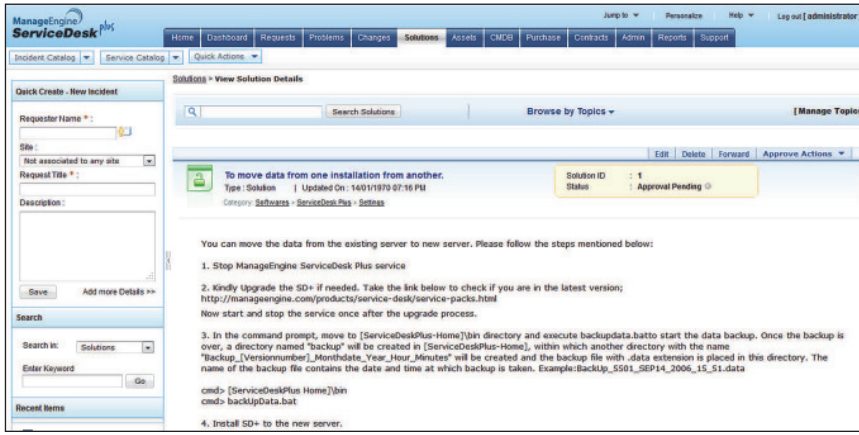

Рис. 7. Запись, ожидающая размещения в базе знаний

Поддерживается создание связей между объектами следующих типов:

- инцидент – инцидент (без указания причинно-следственных отношений);
- проблема – объект в составе ИС;
- проблема – запись в базе знаний.

Возможность редактирования и графического представления существующих связей в визуальном редакторе не предусмотрена.

Поисковый механизм позволяет выполнять поиск по ключевым словам и фразам в записях об инцидентах и проблемах в случае заполнения соответствующих полей при создании записи. Полнотекстовый поиск по вложенным файлам или подключение сторонних поисковых механизмов не поддерживается. Записи баз данных инцидентов и проблем могут экспортироваться и импортироваться (поддерживаются различные форматы, в том числе CSV, XLS и PDF). Однако, открытых интерфейсов для интеграции с существующими базами знаний разработчик на данный момент не предоставляет.

Поддержка русского языка в программном средстве реализована частично: использование Web-интерфейса для доступа к базе знаний позволяет выбрать предпочитаемый язык их создания и редактирования. Записи в базах данных инцидентов, проблем и объектов создаются на английском языке.

1.3.7. **OTRS ITSM**

Свободное программное обеспечение OTRS ITSM с открытыми исходными текстами, построенное на платформе OTRS (Open Ticket Request System), предназначено для автоматизации обработки инцидентов, проблем и изменений в ИС и работы службы технической поддержки.

Записи об инцидентах могут создаваться администраторами системы с помощью встроенного редактора, пользователями портала самообслуживания через Web-форму или средствами системы на основе входящих почтовых уведомлений[6]. Записи создаются с помощью предустановленных шаблонов, форм и типовой структуры категорий; для хранения используются реляционные СУБД.

Для каждого инцидента в базе данных может фиксироваться последовательность действий, выполненная для его разрешения. Первоначально информационное наполнение отсутствует. База знаний (Рис. 8) [65] поддерживает разграничение уровней доступа для разных групп пользователей, включая сотрудников организации и внешних пользователей. Также поддерживается ограничение области видимости записей на основе их статуса – активные и неак-

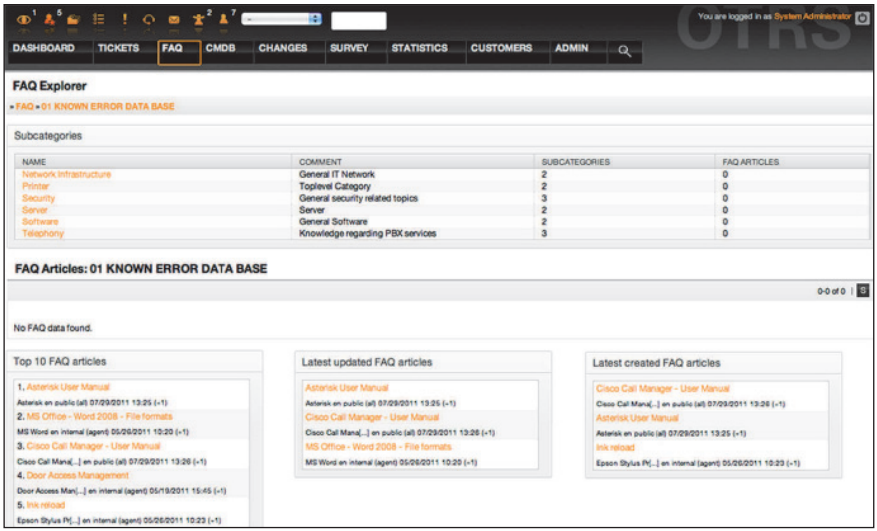

Рис. 8. Поиск статей в базе знаний OTRS ITSM

---

[6] В этом случае создается заготовка записи, требующая дальнейшей корректировки и подтверждения администратором.

тивные. Записи, помеченные как неактивные[7], могут быть доступны только сотрудникам СТП.

В OTRS ITSM поддерживается механизм построения связей между записями об инцидентах. Входящие уведомления и создаваемые администратором записи могут ассоциироваться или объединяться с уже существующими. При этом настройка иерархических связей, позволяющих определить, какой из инцидентов явился причиной возникновения других, не предусмотрена. Записи, содержащие рекомендации по устранению проблем, описания известных ошибок и приемлемых обходных решений, в свою очередь могут быть связаны с записями об инцидентах и объектах из конфигурационной базы данных.

Наличие односторонних связей позволяет из записи об инциденте получить доступ к записи с его подробным описанием, но не дает возможность проанализировать все возможные инциденты, вызванные той или иной проблемой, запись о которой также присутствует в соответствующей базе данных. Средства визуального редактирования и графического отображения связей между различными объектами отсутствуют.

Поиск релевантных записей в базе данных инцидентов по ключевым словам обеспечивается заполнением соответствующих полей в шаблоне. Также поддерживается полнотекстовый поиск по всем индексированным документам, входящим в состав описания, и по различным внешним источникам данных. Механизм OpenSearchDescription используется для поиска в Web-интерфейсе базы знаний (доступ к ней может быть предоставлен внешним пользователям через Self Service портал).

Возможность доступа к исходным текстам системы позволяет разработчикам создать или настроить в соответствии с потребностями пользователей необходимые интерфейсы для связи и интеграции собственных баз данных инцидентов с различными ресурсами сторонних поставщиков.

ПО OTRS ITSM поддерживает использование русского языка при создании записей об инцидентах.

---

[7] Например, связанные с версиями программного обеспечения, больше не использующегося в организации.

1.4. **Результаты сравнительного анализа**

Проведение сравнительного анализа инструментальных средств обеспечения баз экспертных знаний позволило определить основные подходы и методы их реализации, используемые такими производителями, как BMC, HP, IBM, ManageEngine, Microsoft и OTRS. Результаты сравнительного анализа выбранных программных средств по критериям, предложенным в пункте 1.2 монографии, представлены в табл. 2.

В представленных программных инструментах базы данных инцидентов и проблем нередко объединяются в общую постоянно обновляемую базу знаний, содержащую зарегистрированные инциденты, задокументированные способы разрешения и обходные решения, а также установленные корневые причины возникновения, последовательность действий для их устранения и все существующие связи между ними и объектами в составе ИС. Подобные базы знаний могут быть как централизованными, так и распределенными, использоваться в рамках одной организации или предоставлять интерфейсы для совместной работы с внешними пользователями, объединяясь в открытые базы экспертных знаний об инцидентах в ИС.

Для быстрой обработки поступающих данных об инцидентах, их точной первоначальной классификации и привязки к затронутым объектам рядом производителей [58, 63, 65] применяется специализированное программное обеспечение [52, с. 61–63], включающее в себя экспертные системы и базы знаний. Базы знаний содержат предустановленный набор описаний типовых решений для данной предметной области (например, подсистемы ИС или группы предоставляемых ИТ-услуг) и позволяют сотрудникам СТП пополнять их в ходе эксплуатации ИС.

Основным отличием баз знаний от баз данных является наличие информации, позволяющей не только классифицировать инцидент, но и выявить его взаимосвязь (не всегда очевидную) с другими инцидентами и объектами. Таким образом, они позволяют разрешать инциденты на двух уровнях: предоставляют готовое решение для инцидентов, описание которых уже есть в базе, и предлагают рекомендации по разрешению новых инцидентов на основании имеющейся информации о составе компонентов ИС и существующих между ними связях. Использование экспертных систем позволяет прове-

рить наличие связей между объектами в базах данных проблем и инцидентов, чтобы определить, существуют ли незакрытые инциденты, связанные с текущим, и каким образом были разрешены предшествовавшие аналогичные инциденты.

На основе собранных данных можно сформулировать следующие выводы:

1. Все разработчики признают необходимость накопления и дальнейшего использования экспертных знаний, полученных в процессе эксплуатации ИС, и предоставляют необходимые для этого инструментальные средства.
2. В большинстве программных инструментов реализован подход, предусматривающий раздельное хранение информации об инцидентах, проблемах и возможных путях их решения [5]. При этом базы данных инцидентов содержат сведения, используемые сотрудниками первой линии СТП, такие, как класс, категория, дата создания записи, источник, ответственный сотрудник, степень влияния и приоритет, дата разрешения инцидента и закрытия записи. Подробное описание способа разрешения в записи об инциденте в большинстве случаев не приводится; может содержаться идентификатор связанной с ним проблемы или известной ошибки. Базы данных проблем содержат подробное описание корневых причин возникновения инцидентов и способов их разрешения, но могут и не содержать информации обо всех инцидентах, вызванных той или иной проблемой; используется преимущественно второй линией СТП.
3. В состав многих программных инструментов входит модуль организации баз экспертных знаний, собранных в процессе функционирования ИС. При этом под знаниями подразумевается разнородная, слабоструктурированная информация, хранящаяся в виде разрозненных информационно-аналитических статей и руководств с пошаговым описанием выполнения рекомендуемых действий. В качестве приложений текстовые статьи могут содержать файлы различных форматов, в том числе и мультимедиа, что усложняет их обработку (классификацию, поиск, выборку) средствами вычислительной техники. В качестве примера таких статей можно привести руководство по обновлению операционной системы Microsoft Windows Vista до Microsoft Windows 7, уста-

новке и настройке принтера определенной модели или переносу серверного ПО между физическими серверами. Данные статьи, как правило, доступны для просмотра, создания и редактирования сотрудниками и внешними пользователями, и не содержат описания технически сложных инфраструктурных решений или информации о том, с какими зарегистрированными инцидентами и проблемами они связаны.

4. Программные модули, реализующие обработку инцидентов, проблем и запросов на обслуживание, не содержат встроенных механизмов создания и редактирования причинно-следственных связей между записями в базах данных инцидентов и проблем. Также они не предоставляют средств визуального отображения информации об инцидентах, проблемах, конфигурационных единицах и существующих между ними связях.
5. Большинство программных инструментов не содержит интерфейсов, позволяющих использовать информационные ресурсы сторонних поставщиков или формировать открытые базы знаний со свободным доступом.

К недостаткам рассмотренных подходов по накоплению экспертных знаний в процессе эксплуатации ИС можно отнести:

1. Недостаточную формализацию. Базы знаний представляют собой набор разнородных, не связанных друг с другом справочных статей и приложений к ним, содержащих слабоструктурированную информацию. Подобный способ хранения не позволяет эффективно использовать средства вычислительной техники для их обработки.
2. Отсутствие классификации и категоризации. Рассмотренные программные средства не содержат встроенных классификаторов, к узлам которых могли бы привязываться новые записи об инцидентах. В результате усложняется первоначальное определение причин возникновения инцидента и связанных с ним объектов в составе ИС; повышается вероятность появления записей с повторяющимся или пересекающимся содержанием [44].
3. Отсутствие взаимосвязей. Большинство разработчиков не предоставляет средств для создания, редактирования и поддержания в актуальном состоянии связей между инцидентами, проблемами и конфигурационными единицами, что не позволяет администра-

Табл. 2. Результаты сравнительного анализа программных инструментов приобретения экспертных знаний

Название системы	Формат хранения данных	Способы и источники наполнения баз данных и знаний	Возможность доступа внешних пользователей к ресурсам	Наличие предустановленных шаблонов и наборов типовых записей	Возможность создания и редактирования причинно-следственных связей	Поддержка графического отображения данных об инцидентах и связях	Поиск в базах данных и знаний	Возможность интеграции с базами знаний сторонних поставщиков	Поддержка русского языка
IBM Tivoli Monitoring	набор .htm файлов	создание записей вручную администратором на основании пользовательских обращений	не предусмотрена	предустановленный набор типовых ситуаций со стандартным описанием	не предусмотрена	отсутствует	отсутствует	не предусмотрена	элементы интерфейса
IBM Tivoli Service Request Manager	СУБД IBM DB2, MS SQL	создание записей средствами системного мониторинга, приложениями сторонних производителей, вручную администратором	с помощью Self Service Center при наличии необходимых прав	набор предустановленных шаблонов для описания стандартных инцидентов; типовые записи отсутствуют	горизонтальные (неиерархические) связи между инцидентами	отсутствует	полнотекстовый поиск по базам данных инцидентов и проблем	не предусмотрена	элементы интерфейса, создание и редактирование записей, поисковые запросы
Microsoft System Center Service Manager	СУБД MS SQL	создание записей вручную администратором на основании пользовательских обращений и уведомлений систем мониторинга	через портал самообслуживания Self-Service Portal при наличии необходимых прав	набор предустановленных шаблонов для описания стандартных инцидентов; типовые записи отсутствуют	не предусмотрена	отсутствует	полнотекстовый поиск с возможностью подключения сторонних поисковых служб	не предусмотрена	элементы интерфейса; записи рекомендуется создавать на английском языке
HP Service Manager Knowledge Management	СУБД IBM DB2, MySQL	создание записей вручную администратором или пользователями при наличии необходимых прав	с использованием модуля Knowledge Management при наличии необходимых прав	набор из пяти предустановленных шаблонов для описания стандартных инцидентов; типовые записи отсутствуют	не предусмотрена	отсутствует	полнотекстовый поиск по всем документам, преобразованным в HTML	не предусмотрена	элементы интерфейса, создание и редактирование записей, поисковые запросы

Табл. 2. Результаты сравнительного анализа программных инструментов приобретения экспертных знаний *(окончание)*

Название системы	Формат хранения данных	Способы и источники наполнения баз данных и знаний	Возможность доступа внешних пользователей к ресурсам	Наличие предустановленных шаблонов и наборов типовых записей	Возможность создания и редактирования причинно-следственных связей	Поддержка графического отображения данных об инцидентах и связях	Поиск в базах данных и знаний	Возможность интеграции с базами знаний сторонних поставщиков	Поддержка русского языка
BMC Remedy IT Service Management Suite	СУБД MySQL, Sybase, Oracle, MS SQL	создание записей вручную администратором или пользователями при наличии необходимых прав, импорт и экспорт в форматах XML, CSV	через портал самообслуживания self-help Portal при наличии необходимых прав; доступ может быть ограничен на уровне отдельных полей	набор предустановленных шаблонов для описания стандартных инцидентов; типовые записи отсутствуют	не предусмотрена	отсутствует	полнотекстовый поиск с поддержкой индексирования сторонних источников информации	не предусмотрена	элементы интерфейса, создание и редактирование записей
Manage Engine ServiceDesk Plus	СУБД MySQL, Postgre SQL, MS SQL	создание записей вручную администратором на основании пользовательских обращений и уведомлений систем мониторинга	через портал самообслуживания Self Service Portal при наличии необходимых прав	набор предустановленных шаблонов для описания стандартных инцидентов; типовые записи отсутствуют	горизонтальные (неиерархические) связи между записями об инцидентах, проблемах и затронутых объектах	отсутствует	поиск по ключевым словам и фразам	не предусмотрена	Web-интерфейс, создание и редактирование записей
OTRS ITSM	СУБД MySQL, Postgre SQL	создание записей средствами системного мониторинга, вручную администратором или пользователями при наличии необходимых прав	через портал самообслуживания Self Service при наличии необходимых прав	набор предустановленных шаблонов для описания стандартных инцидентов; типовые записи отсутствуют	горизонтальные (неиерархические) связи между записями об инцидентах, проблемах и затронутых объектах	отсутствует	поиск по ключевым словам, полнотекстовый поиск по всем проиндексированным документам	может быть реализована благодаря открытому доступу к исходным текстам	элементы интерфейса, создание и редактирование записей, поисковые запросы

тору видеть причинно-следственные связи между событиями, приводящими к нарушению функционирования ИС. Существующие программные инструментальные средства не позволяют получить ответ на следующие группы вопросов: какие ранее возникшие инциденты и проблемы могли привести к возникновению данного инцидента, сколько зарегистрированных инцидентов связано с отдельно взятым объектом или к возникновению каких инцидентов может привести известная проблема?

4. Ограниченные возможности интеграции. Большинство существующих систем не позволяет интегрировать собственные базы данных инцидентов и проблем с внешними источниками или формировать общие открытые базы знаний.

Проведенный сравнительный анализ позволил выявить основные недостатки рассмотренных подходов к фиксации экспертных знаний в процессе эксплуатации ИС и уточнить состав задач, требующих практического решения.

1.5. **Выводы**

При решении задачи обеспечения штатного режима функционирования современных ИС большое значение имеет использование экспертных знаний об инцидентах, возникающих в процессе их эксплуатации и являющихся уникальными. В случае отсутствия таких знаний или их ограниченного набора, организация сталкивается с рядом таких негативных факторов, как недостатком информации об особенностях взаимодействия различных компонентов ИС, приводящим к принятию необоснованных решений, или необходимостью многократного разрешения однотипных инцидентов и чрезмерной зависимостью от ключевых сотрудников СТП, что в целом приводит к увеличению совокупной стоимости владения ИС [24].

В качестве основных критериев сравнения подходов и программных инструментов ведения баз экспертных знаний выбраны способы и источники наполнения баз знаний, наличие механизма создания и поддержания причинно-следственных связей между записями и средствами их визуализации, а также возможности поиска и интеграции с ресурсами сторонних поставщиков.

В результате проведенного сравнительного анализа программных инструментов по предложенным критериям определены основные

подходы, используемые разработчиками для решения задачи приобретения экспертных знаний, а также существующие ограничения и недостатки.

Основными решениями, принятыми в большинстве рассмотренных систем, является раздельное хранение собранной информации в базах данных инцидентов, проблем и их решений; использование компонента, предназначенного для организации баз знаний отдельно от баз данных. Доступ к нему предоставляется широкому кругу внешних пользователей, тогда как базы данных предназначены, в первую очередь, для сотрудников СТП.

Недостатки рассмотренных подходов заключаются в отсутствии классификаторов, позволяющих осуществить точную привязку инцидента на начальном этапе его разрешения, недостаточной степени формализации собранной информации и отсутствии механизма создания и поддержания причинно-следственных связей между записями в базах данных инцидентов и проблем, конфигурационной базе данных и базе знаний, содержащей способы разрешения известных инцидентов.

ГЛАВА 2

Формализация и классификация экспертных знаний об инцидентах в ИС: модели и методы

2.1. **Онтологический подход в формализации экспертных знаний**

В первой главе монографии определено значение накапливания и последующего использования экспертных знаний об инцидентах в ИС, полученных в процессе ее эксплуатации. Основными недостатками и ограничениями, выявленными в результате проведения сравнительного анализа существующих подходов и программных инструментов организации баз экспертных знаний, являются недостаточная степень формализации собранной информации и отсутствие классификаторов, позволяющих унифицировать привязку и распределение по категориям зарегистрированных инцидентов. Следствием является слабая степень их автоматизации.

Для решения задач формализации и классификации экспертных знаний об инцидентах в ИС в данной книге предложен онтологический подход [20, 25, 45].

Термин «онтология» в информационных технологиях и интеллектуальных информационных системах [1, 12–13, 35] обозначает систему понятий определенной предметной области, представляющую собой набор сущностей и возможных отношений между ними. Онтологии используются для формального определения понятий и отношений, характерных для данной предметной области. Актив-

ное развитие теории и практики построения онтологий, начавшееся в конце 1990-х гг. [27], обусловлено такими факторами[8], как:

- значительный рост объемов информации, представляемой для обработки (анализа, поиска и выборки, изменения) специалистами разных областей деятельности;
- необходимость использования одних и тех же знаний специалистами различных направлений в разных целях;
- необходимость сокращения времени, затрачиваемого на поиск требуемой информации, и повышения релевантности результатов поиска.

Существуют и другие определения онтологии. С позиции информационных систем онтология рассматривается как база знаний специального типа [7], которая может читаться и пониматься, отчуждаться от разработчика и/или физически разделяться пользователями [10, 40].

В интеллектуальных ИС онтология определяется как формальная точная спецификация совместно используемой концептуализации. В данном определении термином «концептуализация» обозначается модель предметной области, для которой определены существенные понятия или концепты. Формальность подразумевает возможность применения средств вычислительной техники для обработки данных онтологии, что ограничивает использование естественных языков при описании входящих в нее понятий. Точность означает, что типы понятий и ограничения на область их использования должны быть определены явным образом. «Совместное использование» в определении указывает на то, что онтология описывает всеобщие знания – знания, принятые в определенной группе или сообществе специалистов [2–3].

Для обеспечения совместного использования вводится понятие онтологического соглашения – объема общепринятого в данной предметной области словаря, достаточного для описания понятий модели и толкования предполагаемых значений слов, входящих в словарь.

Общим для всех определений онтологии являются наличие словаря имен сущностей для представления и обмена знаниями в рас-

[8] Связанными, в первую очередь, с непрерывным повышением сложности решаемых задач и ростом информационных систем организаций

сматриваемой области (тезауруса) и набора связей, которые можно выделить между сущностями. Связи могут быть как универсальными, например, «часть – целое», «причина – следствие», так и специфическими для данной области. Сущности, как и связи, могут обладать различными свойствами, необходимыми для отражения и актуализации свойств предметной области в абстрактной модели.

Таким образом, применение онтологического подхода к организации знаний позволяет решать следующие задачи:

- точно описывать общие знания заданной предметной области за счет формального определения всех используемых понятий и связей между ними;
- совместно использовать знания пользователями, занятыми в различных сферах деятельности и обладающими разными профессиональными навыками;
- интегрировать необходимую информацию и представлять ее в виде, удобном для совместного использования при решении множества прикладных задач [3, с. 370–371].

2.1.1. **Классификация онтологий**

Существующие на сегодняшний день онтологии различаются по целому ряду параметров, что приводит к возникновению большого числа подходов к их классификации: по составу и количеству входящих понятий, типам используемых отношений, способам формирования, решаемым задачам. К основным принципам классификации относятся степень формализации, набор входящих понятий и цель создания [40, с. 12]. В рамках данной книги подробно рассматривается классификация онтологий по цели создания как наиболее подходящая для исследования возможности применения онтологического подхода при решении задач формализации и классификации экспертных знаний об инцидентах в ИС.

По цели создания выделяют онтологии представления, онтологии верхнего уровня, онтологии предметных областей и прикладные онтологии [32, 40].

Онтологии представления определяют абстрактную модель, лежащую в основе формализации представления знаний. Понятия и отношения, вводимые в онтологии последующих, более низких уровней, считаются конкретизацией понятий и отношений онтологии

представления, не зависящих от конкретной предметной области. Благодаря этому они могут использоваться, в том числе, для поддержки нескольких альтернативных теорий в рамках одной и той же или различных областей.

Онтологии верхнего уровня предназначены для приобретения и представления знаний, общих для нескольких предметных областей. Они могут являться как самостоятельным, независимым модулем[9], так и модулем, находящимся в рассматриваемой иерархии над онтологиями предметной области.

Онтологии предметных областей содержат понятия, использующиеся для описания объектов, их качественных характеристик, отличительных особенностей и свойств определенной области знаний. Они состоят из объектов и связей между ними, описанных в терминологии конкретной предметной области. Понятия связаны отношениями с другими основополагающими или определенными в словаре понятиями. Отношения обычно имеют несколько собственных свойств, таких как имя и описание, закрепленных документально [32]. При этом одно и то же множество объектов и связей с одним и тем же набором свойств может использоваться в нескольких областях, в результате чего онтология может быть связана с несколькими областями знаний.

Прикладные онтологии описывают понятия, зависящие как от выбранной области знаний, так и от состава решаемых задач. Совокупность знаний предметной области и модели, описывающие их поведение в рамках определенных в онтологии функций, образуют частную модель, программная реализация которой является онтологией-приложением.

Описание задач или деятельности, для создания которого используется словарь понятий, введенный в онтологиях предметных областей, формирует онтологию задач, не обязательно включаемую в рассматриваемую иерархию. Онтология задач ориентирована на решение конкретных проблем и включает все необходимые для этого понятия словаря онтологии верхнего уровня соответствующей предметной области. Она предназначена для того, чтобы сделать знания доступными для повторного использования посредством их кодирования на основе стандартного словаря. Данная онтология опирается

[9] Описывающим такие общие понятия, как сущность, объект, явление, процесс, материя, пространство, время.

Рис. 9. Классификация онтологий по цели создания

на предположение о том, что определенные виды распространенных задач могут быть решены с использованием одной и той же последовательности действий.

На рис. 9 представлена иерархия и схема отношений между рассмотренными видами онтологий.

2.1.2. **Формальная модель онтологий**

Как было определено ранее, общим для всех онтологий является наличие словаря имен сущностей (тезауруса [15]) для представления и обмена знаниями в рассматриваемой предметной области и набора связей, выделяемых между сущностями. Третьим компонентом онтологии, связанным с указанными, является набор правил их обработки. В этом случае формальная модель произвольной онтологии (O) может быть представлена в виде упорядоченной тройки: O = <X, R, F>, где X – конечное множество понятий (сущностей), R – конечное множество отношений между ними, F – конечное множество аксиом и правил вывода, заданных на сущностях и / или отношениях в составе онтологии.

Очевидными ограничениями, накладываемыми на множества X, R и F, является их конечность. Кроме того, множество сущностей X не может быть пустым. Рассмотрим возможные варианты, связанные с вхождением в состав упорядоченной тройки O пустых и непустых множеств R и F [3].

1. Пусть $R = \varnothing$ и $F = \varnothing$. Тогда модель онтологии представляется в виде

$$O = \langle X, \{\}, \{\}\rangle, \tag{2.1}$$

что соответствует простому словарю V, O = V. Данный вид вырожденной онтологии может использоваться для наполнения и поддержания в актуальном состоянии словарей предметной области. Вместе с тем, применение онтологий-словарей существенно ограничено, так как значения терминов и понятий заданной предметной области заранее хорошо известны в пределах соответствующего сообщества специалистов.

2. Пусть $R = \varnothing$ и $F \neq \varnothing$. Тогда модель онтологии представляется в виде

$$O = \langle X, \{\}, F\rangle. \tag{2.2}$$

Это означает, что каждому элементу множества X ставится в соответствие одно или несколько правил вывода f, $f \in F$. Предположим, $X = X_1 \cup X_2$, причем $X_1 \cap X_2 = \varnothing$, где X_1 – множество интерпретируемых терминов, X_2 – множество не интерпретируемых терминов. При этом $\exists(x \in X_1, y^1, y^2, ..., y^k \in X_2)$ такие, что $x = f(y^1, y^2, ..., y^k)$, $f \in F$.

Если при этом правило вывода задается оператором присваивания сущностям X_1 значений X_2: $X_1 := X_2$, то онтология трансформируется в пассивный словарь V^p,

$$O = V^p = \langle X_1 \cup X_2, \{\}, \{:=\}\rangle, \tag{2.3}$$

так как все определения терминов X_1 берутся из существующего множества X_2. Значение таких терминов вычисляется каждый раз при их интерпретации. Ценность такого словаря выше, чем у предыдущей модели, однако, он не может использоваться для представления

знаний в задачах обработки информации, так как интерпретируемые элементы из множества X_1 никак не связаны друг с другом.

3. Пусть $R \neq \varnothing$ и $F = \varnothing$. Тогда модель онтологии представляется в виде

$$O = \langle X, R, \{\}\rangle, \tag{2.4}$$

т.е. множество аксиом и правил вывода пусто, множество отношений между понятиями не пусто и включает в себя иерархические отношения. Понятия предметной области являются специфическими для каждой прикладной онтологии, тогда как отношения более универсальны. Именно поэтому в качестве базиса обычно выделяют такие отношения модели предметной области, как «элемент класса» IS_A, «вид – род» AKO[10], «часть – целое» POW[11] и др. Данный вариант соответствует специальному подклассу онтологий, называемому простыми таксономиями:

$O = T^o = \langle X, \{IS_A\}, \{\}\rangle$, где IS_A – отношение «быть элементом класса»

$O = T^o = \langle X, \{AKO\}, \{\}\rangle$, где AKO – отношение «вид – род»

$O = T^o = \langle X, \{POW\}, \{\}\rangle$, где POW – отношение «часть – целое».

Под таксономией понимается иерархическая система понятий, связанных друг с другом указанными отношениями с определенной семантикой, что позволяет структурно организовывать сущности в составе онтологии в виде графа. В общем случае такой подход является эффективным и удобным для представления иерархии понятий.

4. Пусть $R \neq \varnothing$ и $F \neq \varnothing$. Тогда модель онтологии представляется в виде

$$O = \langle X, R, F\rangle, \tag{2.5}$$

где непустое множество R может включать в себя как иерархические отношения, так и отношения, специфические для данной предметной области (временные, пространственные, сравнения). В этом случае строится иерархия терминов выбранной области деятельности, в которой каждому из них ставится в соответствие определенная аксиома

[10] A Kind Of
[11] Part Of While

или правило вывода. Такие онтологии являются самыми сложными и представляют собой иерархические структуры, в которых на вводимых сущностях с заданной интерпретацией определяются специфические отношения предметной области.

2.1.3. **Онтологические системы**

Для создания и организации работы онтологий разных уровней применяются онтологические системы, предоставляющие, кроме редактирования и просмотра, возможности экспорта и импорта онтологий в различных форматах, графического редактирования, работы с распределенными библиотеками онтологий [3].

В общем случае онтологическая система включает в себя следующие компоненты:

- метаонтологию, предназначенную для описания таких общих понятий, как объект, свойство, отношение, сущность. Позволяет объявлять понятия в процессе построения произвольных онтологий предметных областей наследниками соответствующих категорий;
- онтологию предметной области. В соответствии с приведенной выше классификацией определяет набор сущностей и описывающих их понятий, используемых при решении конкретных задач. В большинстве случаев строится пользователем онтологической системы;
- онтологию задач. Используется для описания способов преобразования рассматриваемых объектов в ходе решения поставленных задач или их декомпозиции.
- машину вывода. В общем случае определяет исходные и целевые состояния объектов предметной области; используя сетевое представление онтологии, определяет путь (последовательность действий или состояний), необходимый для перехода из исходного в целевое состояние.

На сегодняшний день наиболее широко распространены следующие программные реализации онтологических систем:

- Ontolingua – предоставляет коллективный доступ к библиотеке онтологий, предусматривающей возможности расширения, объединения, а также включения внешних онтологий.
- OilEd – система, предоставляющая функциональные возможности редактора онтологий с использованием технологии OIL (Ontology Inference Layer, уровень логического вывода) и меха-

низма вывода FaCT (Fast Classification of Terminologies, быстрая классификация терминов), позволяющего выполнить проверку полученной онтологии на непротиворечивость.

- Protege – интегрированная среда разработки и редактирования баз знаний, позволяющая создавать онтологии предметных областей и наполнять их знаниями. Также может предоставлять сторонним приложениям интерфейсы для доступа к ранее созданным базам знаний.

2.1.4. **Назначение онтологий**

Исследования различных авторов [8–9, 36, 39, 42], посвященные анализу классов задач, решаемых с помощью онтологий и функциональных возможностей онтологических систем, показывают, что они способны обеспечить:

- стандартизацию знаний, основанную на включении терминов и понятий в состав словаря, объем которого заранее согласован в пределах соответствующего сообщества специалистов и является достаточным для описания конкретной области знаний;
- систематизацию знаний, позволяющую интегрировать различные источники знаний на основе общей таксономии, входящей в совместно используемый словарь;
- разделение знаний между специалистами, занятыми в различных областях деятельности, совместное, а также многократное использование знаний при решении новых задач;
- повышение интеллектуальности систем поддержки принятия решений за счет выявления ранее скрытых или неявных взаимосвязей между сущностями описываемой предметной области.

Перечисленные возможности позволяют использовать онтологические системы в следующих областях [3, 45]:

1. Построение обучающих систем. Онтологии предоставляют точные формальные модели, которые могут применяться при начале работы с новой предметной областью. Также они позволяют легко находить ссылки на различные внешние источники информации.
2. Интеграция данных и знаний. При объединении нескольких самостоятельных баз данных и знаний в одну использование онтологий позволяет уменьшить возможную избыточность за счет установления семантической эквивалентности понятий и сущностей,

определяемых в них разными терминами, а также установить все существенные взаимосвязи.

3. Построение и использование баз общих знаний (common knowledge) для различных интеллектуальных систем. При работе в любой области специалистам приходится использовать знания более высокой степени общности, выходящие за ее рамки. Применение онтологий верхнего уровня позволяют уточнять модели конкретных ситуаций с учетом общих знаний. Прикладные онтологии и онтологии задач могут обращаться к онтологиям верхнего уровня, размещенным в сети Интернет и поддерживающим открытый доступ, для получения требуемой информации.
4. Обеспечение единства используемой терминологии. Значительное количество прикладных задач находится на стыке нескольких предметных областей. Решение таких задач осуществляется в рамках совместной деятельности группы специалистов, имеющих разную подготовку. Применение онтологического подхода позволяет сформировать общее понятийное пространство, обеспечивающее адекватное понимание при обмене информацией между всеми участниками.
5. Построение поисковых систем [30]. Использование онтологий предметной области позволяет точнее интерпретировать смысл терминов в поисковых запросах, а также дополнять или расширять их понятиями, которые связаны с терминами запроса иерархическими или специфическими для данной области знаний отношениями.

Рассмотренные в данном разделе функциональные возможности, области применения и состав решаемых задач, а также общие принципы и особенности формирования онтологий позволяют сделать вывод о целесообразности применения онтологического подхода [20] для решения задач формализации и структурирования экспертных знаний об инцидентах в ИС.

2.2. **Концептуальная модель представления знаний об инцидентах в ИС**

Проектирование онтологии включает в себя разработку ее содержания, состоящую из этапов спецификации [16, 38, 41], концептуализации [28] и формализации, за которыми следует непосредственно реализация с использованием программных средств разработки.

Спецификация определяет общую цель создания и область, знания которой организуются в виде онтологии, набор задач, для решения которых она будет использоваться, а также состав и квалификацию ее пользователей.

На этапе концептуализации определяются значимые объекты (сущности), отношения между ними и их свойства, необходимые для разработки и поддержания в актуальном состоянии модели рассматриваемой предметной области. Информация о них соответствующим образом структурируется [3–4]. Основной задачей при этом является разработка принципов и способов структурирования знаний.

Собранные знания экспертов в заданной области, в большинстве случаев описываемые с помощью естественного языка [14], формализуются с целью дальнейшей автоматизации обработки средствами вычислительной техники. При реализации модель и формализованные знания в ее составе описываются с помощью выбранных технологий и языков программирования.

В монографии онтологический подход применяется для решения задач формализации и классификации экспертных знаний об инцидентах в ИС [25], оказывающих влияние на функционирование аппаратного и программного обеспечения вычислительных комплексов ИС. Пользователями программной реализации онтологии являются специалисты, обеспечивающие функционирование ИС организации, и сотрудники службы технической поддержки.

В соответствии с классификацией, приведенной в разделе 2.1.1 монографии, для решения поставленных задач построена онтология обработки инцидентов в ИС, содержащая понятия, обозначающие сущности, их качественные характеристики, отличительные особенности и свойства.

В соответствии со структурой формальной модели (2.4) $O = \langle X, R, \{\}\rangle$ знания, накопленные в процессе эксплуатации ИС, организуются в виде таксономии, описывающей иерархическую систему понятий, связанных друг с другом отношениями с определенной семантикой, что позволяет структурно организовывать сущности в составе онтологии в виде графа.

2.2.1. **Таксономия объектов**

В общем случае [37] таксономия (классификационная структура) объектов включает в себя следующие элементы:

- *понятия-сущности*, соответствующие значимым объектам рассматриваемой области деятельности; в программной реализации являются экземплярами соответствующих классов или подклассов;
- *понятия-отношения*, соответствующие иерархическим или специфическим для рассматриваемой предметной области бинарным связям между объектами;
- *понятия-свойства*, описывающие значимые параметры и характеристики объектов или их отношений; для большинства свойств в онтологическом соглашении определяется типовая область значений.

В этой книге для структурирования знаний об инцидентах в ИС выделяются следующие *понятия-сущности*:

- информационная система организации (Enterprise) – предназначена для объединения в общую онтологию обработки инцидентов в ИС нескольких прикладных онтологий, разработанных и использующихся различными организациями;
- подсистема ИС (Subsystem) – совокупность аппаратного, программного обеспечения и ресурсов, предназначенных для решения определенной задачи в процессе функционирования ИС; может являться как независимой, обособленной от остальных подсистем, так и тесно с ними связанной;
- аппаратно-программный компонент (Component) – физическое или виртуальное оборудование, а также программное обеспечение в составе подсистем ИС;
- группы объектов (Objgroup) – предназначены для объединения по ряду определенных признаков атомарных объектов, выделенных в составе аппаратно-программных компонентов подсистем ИС;
- объекты (Object) – атомарные с точки зрения установления взаимосвязей с возникающими в ИС инцидентами объекты, в том числе составные части серверного или сетевого оборудования, отдельные модули и компоненты программного обеспечения, процессы, службы или используемые в процессе эксплуатации ИС структуры данных;

- инциденты (Incident) – события, приводящие к нарушению штатного режима или ухудшению заданных параметров функционирования ИС.

Перечисленные *понятия-сущности* на этапе программной реализации рассматриваются как классы. Соответствующие объекты, знания о которых заносятся в базы знаний с помощью онтологического подхода, являются их экземплярами. В табл. 3 приведены значимые параметры и характеристики (понятия-свойства) для перечисленных понятий-сущностей.

Сущность Incident является основной в рассматриваемой предметной области обработки инцидентов в ИС, для нее определены следующие свойства [22, 23]:

- *сообщение об ошибке* (message) – стандартное сообщение, выдаваемое аппаратным или программным обеспечением в соответствии с заданным разработчиком кодом ошибки. В случае возникновения соответствующей ошибки в процессе функционирования ИС поиск нужного инцидента и способа его разрешения может быть выполнен по данному сообщению или его части. В качестве значения свойства message сущности Incident могут использоваться сообщения как на русском, так и на английском языках. Это позволяет фиксировать и классифицировать экспертные знания, полученные при работе с программным и аппаратным обеспечением, не прошедшим русификацию, или с иностранными справочными системами. В качестве примера сообщения об ошибке можно привести: «The DNS server could not create a thread» или «Configuration error for ‘Mem Card Bank’ Major».
- *важность* (severity) – степень воздействия инцидента на штатное функционирование ИС организации. Определяется на основании оценки влияния на объекты и требуемой срочности обработки. Используется для принятия решения о максимально допустимом времени разрешения инцидента и необходимых для этого ресурсах – материальных, финансовых, людских.

  Важность может принимать одно из следующих значений:

Низкая – инцидент незначительно влияет на работу отдельно взятого объекта, аппаратно-программного компонента в составе ИС или пользователя. Разрешение данного инцидента не является приоритетным и не требует специальных навыков. Примером является

Табл. 3. Свойства сущностей предметной области обработки инцидентов в ИС

Свойство	Обозначение	Описание
Сущность Enterprise		
Название организации	enterprisename	Название организации, для ИС которой фиксируются экспертные знания
Описание	comment	Развернутое описание-подсказка, отражающее особенности структуры и состава задач, решаемых в процессе функционирования ИС организации
Сущность Subsystem		
Название подсистемы	subname	Название самостоятельной подсистемы, выделяемой в ИС организации
Описание	comment	Подробное описание, включающее в себя перечень решаемых подсистемой задач и входящих в нее аппаратно-программных компонентов
Сущность Component		
Название компонента	compname	Название серии и модели оборудования или версии ПО
Компания-производитель	vendor	Название компании-производителя аппаратного или программного обеспечения
Описание	comment	Характеристика, позволяющая определить тип и принадлежность данного компонента к определенной подсистеме ИС
Сущность Objgroup		
Название группы объектов	objgname	Название группы объектов в составе аппаратно-программного компонента, объединенных по определенным признакам
Набор общих признаков	belongsto	Описание общих признаков, на основании которых объекты объединены в данную группу
Сущность Object		
Название объекта	objname	Название атомарного с точки зрения привязки к инцидентам объекта
Принадлежность к группе объектов	belongsto	Описание групп объектов, к которым принадлежит данный объект

плановая замена аккумуляторной батареи в ноутбуке или переход на новую версию почтового клиента для отдельно взятого пользователя.

Средняя – инцидент оказывает влияние на работу определенной группы пользователей или не позволяет отдельно взятому объекту выполнять свои функции. При отсутствии инцидентов с более высоким приоритетом должен быть разрешен в течение 24 часов после обнаружения. Примером подобного инцидента является смена пароля учетной записи пользователя в связи с истечением срока действия, определенного политикой безопасности организации, или внесение изменений в настройки DHCP сервера, предназначенного для динамического конфигурирования параметров доступа к ЛВС рабочих станций сотрудников.

Высокая – инцидент не позволяет большому количеству сотрудников выполнять свои задачи или приводит к нарушению штатного режима функционирования целой подсистемы в составе ИС. К разрешению инцидента приступают, как правило, непосредственно после его обнаружения. К инцидентам с высокой степенью воздействия относится отказ программного сервера корпоративной электронной почты или нарушение целостности файловой системы на сервере, выполняющем в организации роль внутреннего DNS сервера.

Критическая – инцидент с указанной степенью воздействия нарушает штатный режим функционирования нескольких отдельных подсистем ИС, в той или иной мере затрагивает всех сотрудников организации. Для его разрешения используются все доступные ресурсы, в случае необходимости привлекаются сторонние специалисты. Примером является аппаратный отказ группы серверов, обеспечивающих функционирование виртуальной инфраструктуры организации или нарушение целостности оптического волокна, соединяющего территориально-распределенные центры хранения и обработки данных (ЦХОД).

- *тип* (type) – определяет, с каким аппаратно-программным компонентом в составе ИС или группой задач, решаемых в процессе ее функционирования, связан данный инцидент. Так, для инцидентов, возникающих в почтовой системе, построенной на основе ПО IBM Lotus, данное свойство принимает значения «Клиент Lotus Notes» или «Сервер Lotus Domino» в зависимости от того,

влияют ли они на клиентское или серверное ПО. Для инцидентов, разрешение которых связано с внесением изменений в настройки операционных систем или системного ПО, данное свойство принимает значение «Администрирование ОС».

- *описание события* (event) – используется для хранения данных, описывающих возникший инцидент. К ним относится информация, дополняющая и поясняющая стандартное сообщение об ошибке (свойство message), а также описывающая возможные предпосылки и условия, приведшие к его возникновению, основные признаки и возможное влияние на другие объекты в составе ИС или возникновение других инцидентов. Примером такого описания является «Ошибка подключения к удаленному рабочему столу при использовании клиентом и сервером несовместимых уровней шифрования» или «Ошибка запуска демона named на 64-битной ОС: не найдена следующая разделяемая библиотека /usr/lib64/x86_64/liblwres.so.80».
- *способ разрешения* (action) – содержит пошаговое руководство по разрешению инцидента, представляющее собой проверенную последовательность действий, выполненных и зафиксированных при его первоначальном разрешении. Примером такого руководства является «Выполните команду tscc.msc и в свойствах подключения RDP-Tcp выберите уровень шифрования «‘Совместимый с клиентским’» или «Установите (например, с использованием утилиты конфигурации yast) пакет bind9-lwresd». Значением данного свойства также могут быть ссылки на сторонние ресурсы, содержащие дополнительную информацию по разрешению соответствующего инцидента, исправления ПО, выпущенные разработчиками, или актуальные версии драйверов.
- *принадлежность к уровню модели ISO/OSI* (osi) – на основании значения данного свойства определяется принадлежность инцидента к определенному уровню модели.
- *краткое описание* (comment) – содержит набор ключевых понятий или фраз, описывающих инцидент в целом. Предназначено для повышения точности и релевантности результата поиска по базе знаний инцидентов поисковому запросу пользователя.

Для описания значимых для предметной области обработки инцидентов в ИС связей между объектами (рассмотренными выше

понятиями-сущностями) в данной работе приняты и используются следующие понятия-отношения:

- подсистемы в составе ИС *(Enterprise – Subsystem*) – связь данного типа предназначена для описания иерархических отношений между ИС организации и подсистемами, выделяемыми в ней. Одна ИС может состоять из произвольного количества подсистем. Одна и та же подсистема может входить в состав нескольких ИС. Это означает, что при реализации для обозначения данного отношения будет использована связь типа «многие ко многим».
- аппаратно-программные компоненты в составе подсистемы ИС (*Subsystem – Component*) – связь обозначает иерархические отношения между подсистемами ИС и формирующими их компонентами. Каждый компонент может входить в состав только одной подсистемы. При этом, нарушение штатного режима функционирования ряда компонентов может повлиять на подсистемы, непосредственно с ними не связанные. Для учета такого влияния, а также возможности выявления и представления неявных взаимосвязей используется соответствующее свойство отношения (isnative, Табл. 4).
- группы объектов в составе аппаратно-программного компонента (*Component – Objgroup*) – данная связь позволяет выделять и группировать множество однотипных объектов, являющихся составными частями рассматриваемого компонента. Группа объектов может входить в состав нескольких компонентов.
- объекты внутри группы (*Objgroup – Object*) – связь данного типа является иерархической и используется для объединения атомарных объектов в группы на основании тех или иных общих признаков.
- отношение инцидент – объект (*Object – Incident*) является основным в рассматриваемой предметной области. Данная связь позволяет не только определить все инциденты, связанные с заданным объектом или все объекты, затронутые определенным инцидентом, но и уточнить, привело ли возникновение инцидента к нарушению функционирования объекта в составе ИС, или, в свою очередь, было им вызвано.
- отношение инцидент – инцидент (*Incident – Incident*) отражает причинно-следственную связь между несколькими инцидентами.

Также данное отношение позволяет выявлять неявные связи между объектами, нарушение функционирования которых приводит к возникновению одного или нескольких инцидентов, и другими инцидентами, вызванными ими в свою очередь.

В табл. 4 приведены значимые параметры и характеристики (понятия-свойства) для перечисленных понятий-отношений.

Таким образом, онтология обработки инцидентов в ИС включает в себя описание шести типов понятий-сущностей, соответствую-

Табл. 4. Свойства отношений предметной области обработки инцидентов в ИС

Свойство	Обозначение	Описание
Отношение Subsystem – Component		
Относится к	isnative	Определяет, является ли аппаратно-программный компонент составной частью подсистемы, или оказывает влияние на ее функционирование
Отношение Object – Incident		
Вызван	issuedby	Определяет, был ли данный инцидент вызван нарушением функционирования связанного с ним объекта
Влияет на	affect	Определяет, на какие объекты повлияло возникновение данного инцидента
Описание	comment	Развернутое описание-подсказка, поясняющее связь инцидента с объектом
Отношение Incident – Incident		
Вызван	issuedby	Определяет, был ли данный инцидент вызван другим инцидентом; используется для выявления неявных причинно-следственных связей между возникшими инцидентами и не связанными с ними непосредственно объектами
Привел к возникновению	leadto	Определяет, привел ли данный инцидент к возникновению других инцидентов в смежных подсистемах ИС
Описание	comment	Развернутое описание-подсказка, поясняющее причинно-следственную связь между инцидентами

щих таким объектам предметной области, как ИС, их подсистемы, аппаратно-программные компоненты в составе подсистем, атомарные объекты и группы, в которые они могут объединяться, а также инциденты, возникающие в процессе эксплуатации ИС и связанные с объектами.

Связи между рассмотренными сущностями описываются с помощью шести типов понятий-отношений. Четыре из них являются иерархическими – подсистемы в составе ИС, аппаратно-программные компоненты подсистем, группы объектов, выделяемые в компоненте, и атомарные объекты, объединяемые в группы. Два оставшихся типа отношений являются специфическими для предметной области обработки инцидентов в ИС и описывают причинно-следственные взаимосвязи между объектами и возникающими инцидентами. Как для понятий-сущностей, так и для понятий-отношений определены свойства, описывающие их значимые параметры и характеристики.

Предложенная онтология позволяет перейти от эмпирического определения, данного во введении, к формальному описанию инцидента, представленному в виде кортежа:

$$I = \langle X, P(X), R, P(R) \rangle, \tag{2.6}$$

где

X – множество объектов действий по разрешению инцидентов,

P(X) – множество значимых свойств объектов,

R – множество отношений между объектами,

P(R) – множество значимых свойств отношений.

На рис. 10 представлена таксономия объектов в соответствии с разработанной концептуальной моделью представления знаний об инцидентах в ИС.

2.2.2. **Классификация и структурирование знаний об инцидентах в ИС**

Онтология включает в себя набор сущностей, отношений между ними и их свойств. Как было установлено ранее, сущности являются специфическими для каждой прикладной онтологии, тогда как отношения в целом более универсальны. Именно поэтому структурирование знаний при построении онтологий чаще всего выполняется с по-

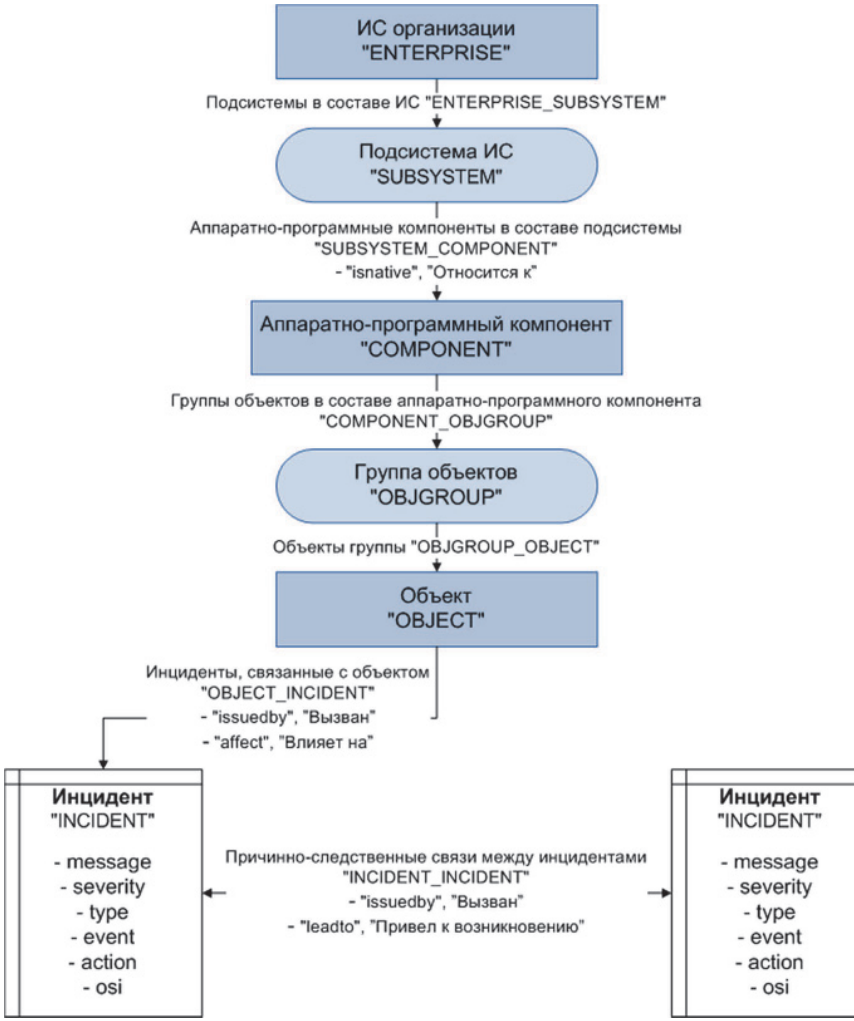

Рис. 10. Таксономия объектов, основанная на концептуальной модели представления знаний об инцидентах в ИС

мощью определения и классификации значимых отношений между сущностями. При этом отношения могут быть как иерархическими, так и специфическими для рассматриваемой предметной области.

В данной книге структурирование экспертных знаний об инцидентах в ИС осуществляется с помощью разработки двух независимых

классификаторов инцидентов. Первый классификатор базируется на структурном подходе, позволяющем рассматривать любую информационную систему как совокупность входящих в нее подсистем (иерархические отношения). В основе второго классификатора лежит эталонная модель взаимодействия открытых систем ISO/OSI (отношения, специфические для рассматриваемой предметной области).

2.2.2.1. Классификация на основе структурного подхода

В основе применения структурного подхода в теории информационных систем лежит утверждение о том, что любая ИС независимо от сферы применения может быть представлена как совокупность своих подсистем. Подсистемы могут быть:

- общими (общесистемными) – формируют основу ИС, без которой она не может существовать; к ним относится активное сетевое оборудование, СКС, серверное оборудование;
- обеспечивающими – используются для организации работы прикладных подсистем, в большинстве случаев не представляют самостоятельной ценности; включают в себя виртуальную инфраструктуру, операционные системы, сети хранения данных;
- функциональными (прикладными) – позволяют автоматизировать решение задач, специфических для заданной области; к ним относятся подсистемы разработки, сборки и тестирования ПО, почтовые системы, системы электронного документооборота.

В зависимости от необходимого уровня детализации и количества возникающих инцидентов в ИС можно выделить произвольное количество подсистем (Рис. 11). Выбор подобной иерархии, допускающей использование отношений типа «многие ко многим», позволит в дальнейшем организовывать открытые базы экспертных знаний на основе онтологического подхода: любая организация сможет использовать существующие или разрабатывать собственные прикладные онтологии для описания подсистем, объединенных общими понятиями и отношениями на уровне онтологии обработки инцидентов в ИС.

На следующем этапе применения структурного подхода в каждой подсистеме выделяется множество аппаратно-программных компонентов, из которых она состоит. Такими компонентами являются физическое и виртуальное оборудование, а также программное обе-

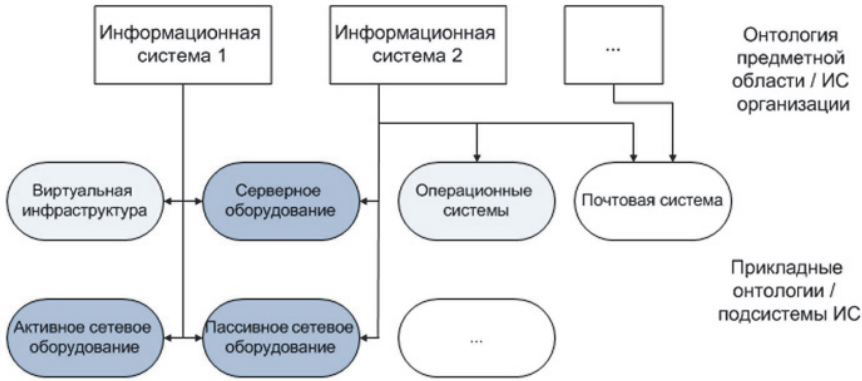

Рис. 11. Пример общих, обеспечивающих и прикладных подсистем

спечение. В качестве примера компонентов можно привести для виртуальной инфраструктуры – программный сервер ESX 4.1, для почтовой системы – почтовый сервер IBM Lotus Domino 8.5.2 и клиент IBM Lotus Notes 8.5.2.

В отличие от подсистем, большинство из которых может входить в произвольное количество рассматриваемых ИС, компоненты входят в состав одной определенной подсистемы. При этом компоненты разных подсистем могут влиять друг на друга – инцидент, связанный с одним из них, может стать причиной возникновения ряда инцидентов в другом. Для учета такого взаимовлияния в концептуальной модели между сущностями типа «подсистема» и «компонент» в данной работе выделяется два типа связей.

Первый является иерархическим – «состоит из» и определяет набор аппаратно-программных компонентов в составе подсистемы. Второй является специфическим для предметной области обработки инцидентов в ИС и используется для выявления скрытых причинно-следственных связей между инцидентами, связанными с различными компонентами. На рис. 12 представлен пример использования обоих типов связей между подсистемами и их компонентами; иерархические связи обозначены сплошными линиями, специфические – пунктирными.

На следующем шаге определяется множество атомарных с точки зрения установления взаимосвязей с инцидентами объектов в соста-

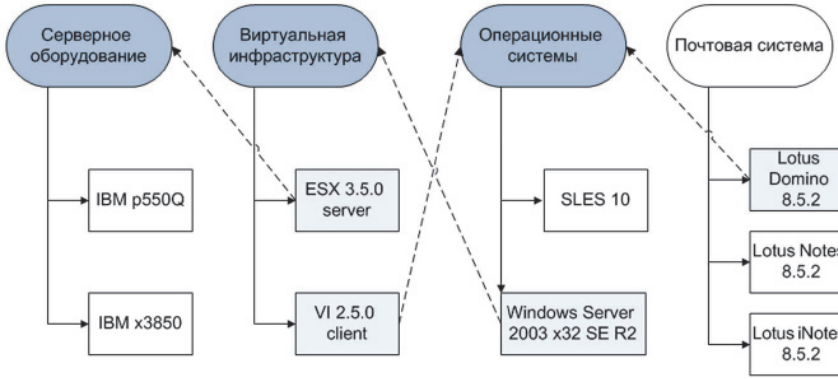

Рис. 12. Пример иерархических и специфических отношений

ве каждого аппаратно-программного компонента. В рассматриваемом контексте такие объекты аналогичны конфигурационным единицам – configuration items – описываемым в базах данных CMDB в терминологии библиотеки ITIL. Объектами могут быть составные части серверного или сетевого оборудования – процессоры, жесткие диски, сетевые интерфейсы, отдельные модули и компоненты программного обеспечения, процессы, службы, а также данные, используемые в процессе эксплуатации ИС.

При необходимости объекты, выделяемые в аппаратно-программном компоненте, могут объединяться в группы на основании тех или иных общих признаков. В качестве примера такой группы можно привести множество сетевых служб в операционной системе Windows Server, каждая из которых рассматривается как самостоятельный объект.

В примере, представленном на рис. 13, в составе ОС Suse Linux Enterprise Server (SLES) выделяются такие группы объектов, как драйверы устройств, системные и сетевые демоны, файловые системы и графические оболочки. Часть групп может быть общей для нескольких компонентов[12], при этом объекты, входящие в них, могут относиться к разным компонентам. К сетевым относятся демоны, обеспечивающие работу таких протоколов, как DHCP (динамиче-

[12] В данном примере – драйверы устройств и файловые системы – являются общими для нескольких операционных систем.

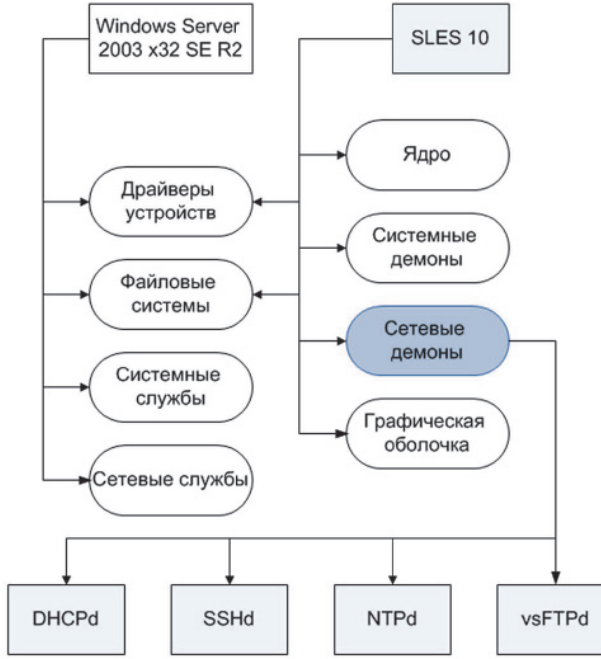

Рис. 13. Группы объектов в составе аппаратно-программного компонента

ская настройка параметров подключения к сети), SSH (удаленный доступ с шифрованием данных), NTP (синхронизация времени), FTP (передача данных). Настройка и разрешение инцидентов, связанных с перечисленными демонами, отличается от настройки сетевых служб, отвечающих за работу этих же протоколов в ОС семейства Microsoft Windows, в связи с чем в данном примере выделяется самостоятельная группа объектов «Сетевые службы».

Выявление атомарных объектов в ИС объясняется необходимостью формальной точной классификации и привязки инцидентов, возникающих в ней. Любой инцидент связан с одним или более объектом: он либо вызван им, либо оказывает влияние, приводящее к ухудшению параметров его функционирования. В большинстве случаев возникновение инцидента связано с нарушением работы нескольких объектов; один и тот же объект может быть затронут рядом не связанных друг с другом непосредственно инцидентов.

Рис. 14. Схема алгоритма привязки инцидента к объектам на основе структурного подхода

Таким образом, в полученном классификаторе используются следующие типы узлов: подсистемы ИС (на практике именно на уровне подсистем реализуются прикладные онтологии предметной области обработки инцидентов в ИС), аппаратно-программные компоненты, группы атомарных объектов, объекты. Понятия инцидента и объек-

та тесно связаны; эта взаимосвязь является основой распределения и привязки инцидентов в рассматриваемом классификаторе.

На рис. 14 представлена схема, иллюстрирующая алгоритм привязки инцидента к объектам в соответствии с рассмотренным классификатором.

2.2.2.2. Классификация на основе эталонной модели взаимодействия открытых систем ISO/OSI

Эталонная модель взаимодействия открытых систем ISO/OSI (Open Systems Interconnection) предназначена для определения и организации совместной работы ИС, являющихся неоднородными по составу аппаратного и программного обеспечения. Модель последовательно описывает правила и процедуры организации взаимодействия на семи уровнях, начиная с оборудования и физических средств связи и заканчивая уровнем приложений. Каждый уровень выполняет определенную задачу, его назначение стандартизируется, определяется набор протоколов, функционирующих в его рамках, описываются интерфейсы, использующиеся для передачи данных от одного уровня к другому [18].

Классификатор, основанный на эталонной модели взаимодействия открытых систем, позволяет логически упорядочивать, группировать и распределять инциденты по соответствующим уровням. Распределение инцидентов может выполняться в зависимости как от того, к какому уровню модели относится объект, связанный с ними[13], так и от того, на каком уровне было нарушено взаимодействие в результате возникновения инцидента.

Таким образом, в отличие от классификатора, основанного на структурном подходе, распределение и привязка инцидентов в соответствии с которым осуществляется только на основании затронутых атомарных объектов в составе ИС, рассматриваемый классификатор не использует связи объектов с инцидентами как основные. Главным принципом классификации для него является определение уровней модели ISO/OSI, между которыми было нарушено взаимодействие в результате возникновения инцидента. Например, инциденты, связанные с коммутатором третьего уровня, формально функ-

[13] Штатное функционирование которого было нарушено инцидентом либо само привело к возникновению инцидента.

ционирующим на канальном и сетевом уровнях модели, также будут распределяться между соответствующими уровнями в зависимости от того, влияют ли они на функции коммутации или IP маршрутизации, выполняемые данным устройством.

Уровни модели, их краткое описание [31] и примеры инцидентов, соответствующих каждому из них, приведены в табл. 5.

Классификатор, основанный на семиуровневой модели взаимодействия ISO/OSI, позволяет распределять по соответствующим уровням накопленные знания о причинах возникновения и способах разрешения, а также существующих взаимосвязях между инцидентами. Подобное распределение дает специалисту, работающему с базой знаний, возможность не только получать подмножество инцидентов, способных привести к нарушению функционирования ИС на заданном уровне модели, но и последовательно просматривать уровни, на-

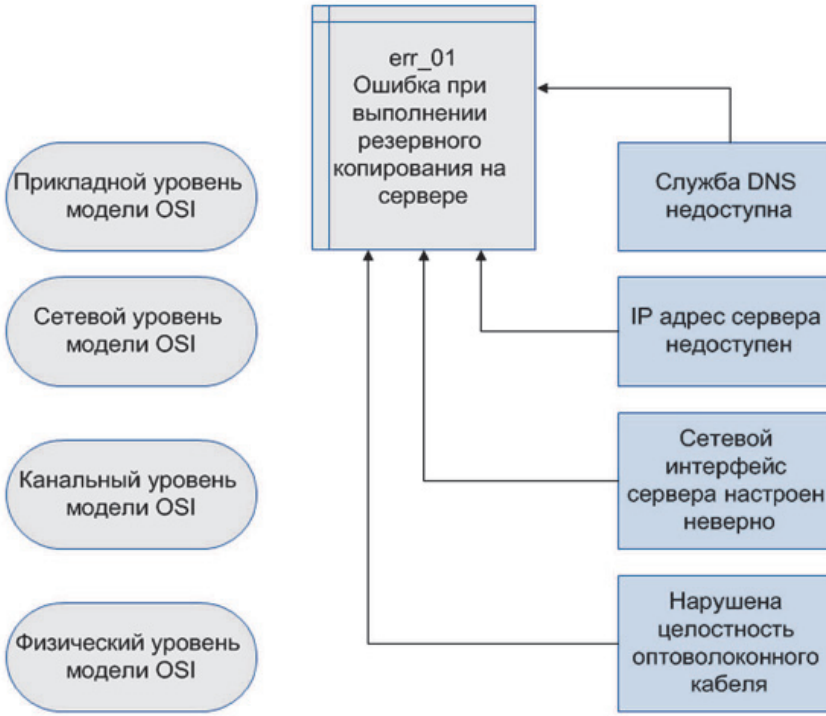

Рис. 15. Распределение инцидентов в соответствии с уровнями модели ISO/OSI

Табл. 5. Примеры распределения инцидентов ИС по уровням модели ISO/OSI

Название уровня	Назначение	Примеры инцидентов
Физический	Описание сетевого и серверного оборудования, а также физической среды передачи данных	• нарушение целостности оптоволоконного кабеля, соединяющего центры обработки данных; • физическое нарушение работоспособности сетевого интерфейса маршрутизатора
Канальный	Управление доступом к среде передачи данных, обеспечение и контроль логической связи	• сбой сетевого подключения, вызванный неверной работой сетевого интерфейса сервера; • отсутствие у виртуального сервера доступа к сети из-за неправильной настройки виртуального сетевого адаптера
Сетевой	Соединение пользовательских и управляющих систем через сети передачи данных, маршрутизация и управление потоками данных	• ошибка передачи данных, вызванная неверной настройкой преобразования сетевых адресов (NAT); • удаленный сервер приложений не доступен по указанному сетевому адресу
Транспортный	Управление передачей данных через коммуникационную сеть	сбой обмена TCP пакетами, вызванный некорректной настройкой параметра MTU[14]
Сеансовый	Установление и поддержание сеанса взаимодействия между удаленными пользователями или прикладными процессами	• ошибка установления соединения с удаленным сервером, вызванная использованием различных версий протокола SSH (Secure Shell); • сбой соединения с удаленной системой, организованного через туннельный протокол L2TP
Представительский	Преобразование данных, передаваемых между прикладными процессами, в требуемый формат, включая шифрование и дешифрование	• ошибка подключения к удаленному рабочему столу при использовании клиентом и сервером несовместимых уровней шифрования; • невозможность доступа к защищенным ресурсам с помощью Web браузера, не поддерживающего требуемую версию протокола, обеспечивающего безопасность связи (SSL, TLS)
Прикладной	Обеспечение набора интерфейсов, предоставляющих прикладным процессам средства доступа к области взаимодействия	• сервер резервного копирования не может получить доступ к параметрам копирования на клиентах из-за недоступности службы DNS; • сервер DHCP не удается запустить из-за повреждения файла Dhcp.mdb, используемого для хранения сведений об активных арендах адресов и резервирований; • почтовое сообщение не было передано по протоколу NRPC на удаленный сервер: в доступе отказано

[14] Maximum Transmission Unit, максимальный размер полезного блока данных

чиная с физического и заканчивая уровнем приложений в поисках корневой причины возникновения инцидента (рис. 15).

В результате применение двух независимых классификаторов для структурирования экспертных знаний об инцидентах в ИС позволяет в зависимости от специфики решаемых задач [24, 26]:

- получать подмножество инцидентов, связанных с заданным объектом, и последовательно, в соответствии с распределением по уровням модели ISO/OSI, просматривать их;
- выбирать нужный уровень модели для указанной подсистемы или компонента ИС, после чего просматривать полученное подмножество инцидентов для нахождения связанных с искомым объектом.

2.3. **Математическая модель представления знаний об инцидентах в ИС**

В разработанной концептуальной модели (2.6), включающей в себя формальное определение инцидента $I = \langle X, P(X), R, P(R) \rangle$, выделяется три уровня [25]. К первому уровню относится совокупность множеств понятий-сущностей $X_{об}$, соответствующих таким объектам предметной области, как ИС, их подсистемы, аппаратно-программные компоненты, атомарные объекты и их группы и множеств понятий-отношений $R_{об}$, описывающих иерархические связи между ними. Таким образом, данный уровень, в дальнейшем называемый объектным и обозначаемый $L_{об}$, описывается следующим образом: $L_{об} = X_{об} \cup R_{об}$, где

$$X_{об} = X_{ис} \cup X_{п} \cup X_{к} \cup X_{г} \cup X_{ао} \quad (2.7)$$

$$R_{об} = R(X_{ис}, X_{п}) \cup R(X_{п}, X_{к}) \cup R(X_{к}, X_{г}) \cup R(X_{г}, X_{ао}), \quad (2.8)$$

В данных выражениях используются следующие обозначения:

$X_{ис}$ – множество рассматриваемых ИС,

$X_{п}$ – множество подсистем, входящих в их состав,

$X_{к}$ – множество аппаратно-программных компонентов, из которых состоят подсистемы ИС,

$X_{г}$ – множество групп объектов, объединяющих объекты на основании тех или иных признаков,

$X_{ао}$ – множество атомарных объектов;

$R(X_{ис}, X_{п})$ – множество иерархических отношений, использующихся для включения подсистем в состав ИС,

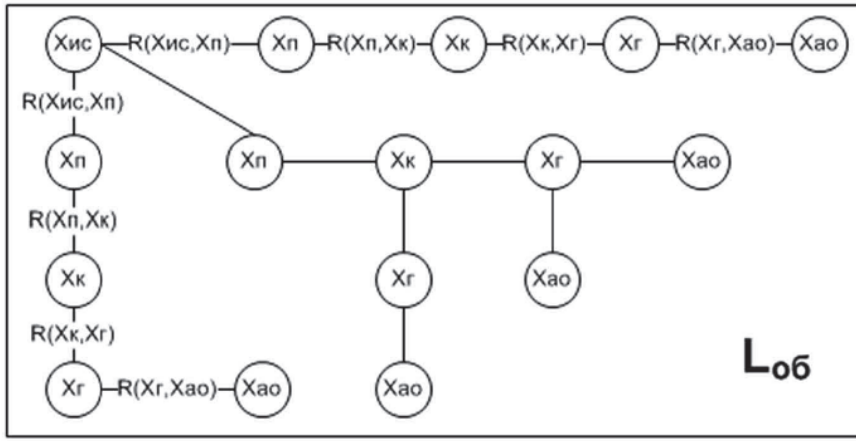

Рис. 16. Фрагмент структуры объектного уровня Lоб модели предметной области

$R(X_п, X_к)$ – множество иерархических отношений, определяющих аппаратно-программные компоненты в составе подсистем,

$R(X_к, X_г)$ – множество иерархических отношений между объектами типа «аппаратно-программный компонент» и «группа объектов»,

$R(X_г, X_{ао})$ множество иерархических отношений, определяющих принадлежность атомарных объектов к соответствующим группам.

Распределение понятий-сущностей и отношений на объектном уровне $L_{об}$ осуществляется с использованием классификатора, основанного на структурном подходе. Фрагмент соответствующей структуры приведен на рис. 16.

На третьем уровне – уровне инцидентов, в дальнейшем обозначаемом $L_и$, – находятся множества объектов типа «инцидент» $X_и$, значения свойств которых позволяют фиксировать признаки и структурную принадлежность в соответствии с предложенными методами классификации разрешенных в процессе эксплуатации ИС инцидентов, и причинно-следственных связей между ними $R_{пс}$:

$$L_и = X_и \cup R_{пс},\ R_{пс} = R(X_и, X_и), \qquad (2.9)$$

Объекты на уровне инцидентов распределяются в соответствии с классификатором, основанном на эталонной модели взаимодействия открытых систем ISO/OSI.

Причинно-следственные связи между объектами $R(X_и, X_и)$ на уровне $L_и$ позволяют определить подмножество инцидентов, для разрешения которых необходимо и достаточно устранить корневую причину возникновения инцидента, приведшего к их возникновению. На рис. 17 данное подмножество обозначено как ε-окрестность, центром которой является исходный инцидент (выделен цветом), входящие дуги – причины, исходящие – следствия. Узлы, или вершины полученного графа, могут быть:

- корневыми – иметь только исходящие дуги и интерпретироваться как причины возникновения других инцидентов;
- терминальными – иметь только входящие дуги и обозначать следствия исходных инцидентов;
- промежуточными – иметь и входящие и исходящие дуги.

В качестве примера причинно-следственных связей между инцидентами можно привести обновление версии драйвера сетевого интерфейса ПК, решающее проблему периодической недоступности ЛВС организации для пользователя. Оно позволяет также решить инциденты, связанные с невозможностью отправки почтовых сообщений, получения доступа к ресурсам сети Интернет и настройки времени на рабочей станции в соответствии с параметрами NTP сервера.

Таким образом, перечисленные инциденты, связанные соответствующими причинно-следственными связями с исходным, в терминах предложенной математической модели располагаются в его ε-окрестности. Классификация и распределение инцидентов при этом выполняется в соответствии с уровнями модели ISO/OSI (на рис. 17 цифрами слева обозначены уровни модели). Выполнение зафиксированной ранее последовательности действий по разрешению исходного инцидента позволяет разрешить все инциденты из данной окрестности при условии отсутствия пересечения таких окрестностей по инцидентам, возникающего в случае, когда один и тот же инцидент вызван несколькими не связанными друг с другом причинами.

При этом возможны следующие варианты:

$$1.\ \bigcap_{i=1}^{n} \varepsilon(X_{и_i}) = \varnothing,\ \overline{i=1,n}\ . \qquad (2.10)$$

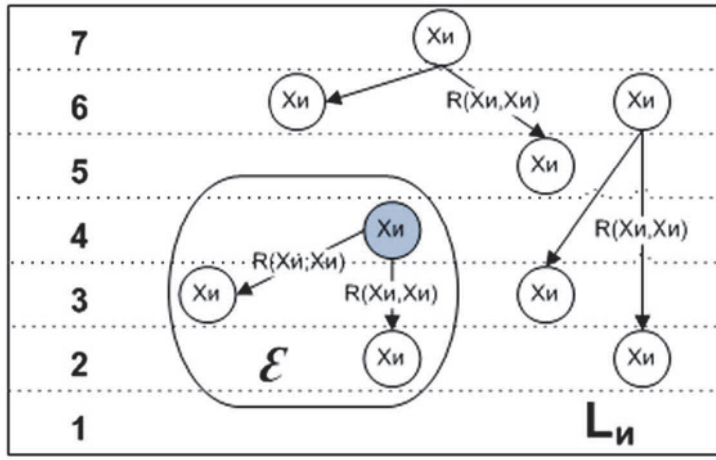

Рис. 17. Фрагмент структуры уровня инцидентов $L_и$ модели предметной области

Данный случай соответствует ситуации, при которой ε-окрестности всех зафиксированных в системе инцидентов $\varepsilon(X_{иi})$ не пересекаются, что позволяет сделать вывод о достаточности разрешения исходного инцидента для разрешения всех инцидентов из его ε-окрестности. Другими словами, в процессе эксплуатации ИС не зафиксированы инциденты, вызванные несколькими независимыми причинами.

$$2.\ \bigcap_{i=1}^{n} \varepsilon(X_{и_i}) = \{X_{и_j}, ..., X_{и_k}\},\ \overline{i,j,k=1,n}\ . \qquad (2.11)$$

В данном случае инциденты $X_{иj}$, $X_{иk}$ вызваны несколькими независимыми причинами (их точное количество определяется выбранным значением переменной n), в результате чего для их разрешения требуется выполнить несколько самостоятельных последовательностей действий.

Особенностью уровня инцидентов $L_и$ в разработанной модели является наличие свойств[15] у объектов множеств инцидентов $X_и$ и причинно-следственных связей между ними $R_{пс}$. Данные множества обозначаются, соответственно, как $P(X_и)$ и $P(R(X_и, X_и))$:

[15] Их подробное описание приведено в разделе 2.2.1 монографии.

$$P(X_и) = \{p_j\,(X_{и_i})\},\ \overline{i=1,n},\ \overline{j=1,m}, \tag{2.12}$$

где p_j – свойства i-го объекта множества $X_и$, (2.13)

$$P(R(X_и, X_и)) = \{p_j\,(R(X_{и_i}, X_{и_k}))\},\ \overline{i,k=1,n},\ \overline{j=1,m},$$

где p_j – свойства отношения между i-м и k-м объектом множества $X_и$.

Данные множества расширяют определение уровня инцидентов $L_и$, описание которого принимает следующий вид:

$$L_и = X_и \cup P(X_и) \cup R(X_и, X_и) \cup P(R(X_и, X_и)). \tag{2.14}$$

Промежуточным (вторым в предложенной модели) между уровнями $L_и$ и $L_{об}$ является уровень действий $L_д$, включающий в себя множество действий, выполняемых над атомарными объектами для разрешения связанных с ними инцидентов.

Действие или последовательность действий, выполненных при первоначальном разрешении инцидента и зафиксированных для дальнейшего использования, определяется благодаря причинно-следственным связям между объектами и инцидентами $R(X_{ао}, X_и)$. Данные связи позволяют не только определить возможные способы разрешения инцидента, но и идентифицировать атомарные объекты, ставшие причиной возникновения инцидента, или инциденты, повлиявшие на их функционирование. Таким образом, связи $R(X_{ао}, X_и)$ используются для представления отношений типа «многие ко многим»: один объект может быть связан с несколькими инцидентами. Для разрешения одного инцидента может потребоваться выполнить последовательность действий, затрагивающую несколько объектов.

У объектов множества $R(X_{ао}, X_и)$ так же, как и у $R(X_и, X_и)$, определяются свойства, формирующие множество $P(R(X_{ао}, X_и))$:

$$P(R(X_{ао}, X_и)) = \{p_j\,(R(X_{ао_i}, X_{и_k}))\},\ \overline{i,k=1,n},\ \overline{j=1,m}, \tag{2.15}$$

где p_j – свойства отношения между i-м объектом множества $X_{ао}$ и k-м объектом множества $X_и$.

На основании приведенных выше обозначений формальное определение инцидента записывается следующим образом:

$$I = <X_{ао}, P(X_и), R(X_{ао}, X_и), P(R(X_{ао}, X_и))> , \quad (2.16)$$

где $X_{ао}$ – объекты действий по разрешению инцидента в составе ИС,

$P(X_и)$ – значимые свойства и признаки инцидента, фиксируемые в процессе его разрешения,

$R(X_{ао}, X_и)$ – причинно-следственные связи между объектами и инцидентом, позволяющие определить последовательность действий по его разрешению,

$P(R(X_{ао}, X_и)$ – свойства связей, позволяющие идентифицировать атомарные объекты, ставшие причиной возникновения инцидента или инциденты, повлиявшие на их функционирование.

Объединение рассмотренных выше уровней математической модели $M_{ис}$ позволяет описать ее в целом следующим образом:

$$M_{ис} = L_{об} \cup L_д \cup L_и,$$

$$M_{ис} = X_{об} \cup R_{об} \cup R(X_{ао}, X_и) \cup P(R(X_{ао}, X_и)) \cup$$
$$\cup X_и \cup P(X_и) \cup R(X_и, X_и) \cup P(R(X_и, X_и)). \quad (2.17)$$

На рис. 18 представлена структура графа, соответствующего данному описанию.

В зависимости от порядка просмотра уровней он позволяет получать как подмножества инцидентов, связанных с заданным объектом (прямой порядок, сверху – вниз), так и подмножества объектов, связанных с выбранными инцидентами (обратный порядок, снизу – вверх).

В соответствии с (2.7) множество $X_{ао}$ входит в состав $X_{об}$ и, таким образом, может рассматриваться как его подмножество: $X_{ао} \subset X_{об}$.

Тогда, исходя из определений инцидента (2.16) и математической модели $M_{ис}$ (2.17), можно утверждать, что для любого набора элементов множеств $X_{ао}$ и $X_и$ существует описание инцидента, заданное в соответствии с формальным определением, для которого единственным образом определяется принадлежность к множеству $M_{ис}$:

$$\forall X_{ао}, X_и \; \exists I \in M_{ис} , \quad (2.18)$$

То есть инцидент I, связанный с любыми элементами множеств атомарных объектов и описаний инцидентов может быть описан

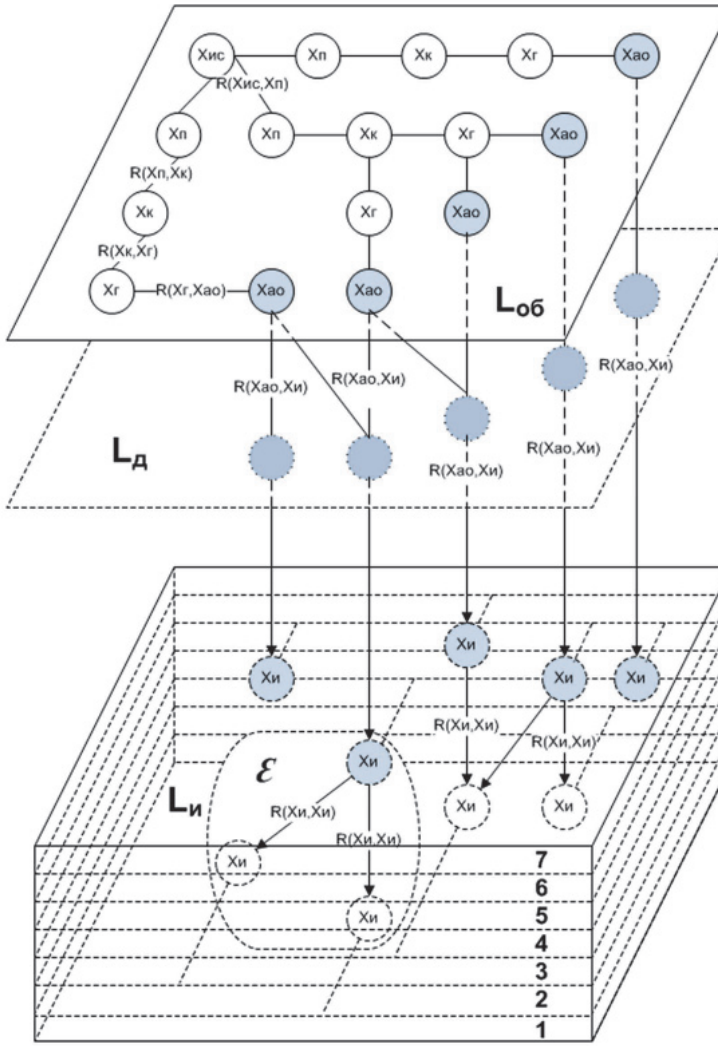

Рис. 18. Модель представления экспертных знаний об инцидентах в ИС

в рамках предложенной математической модели $M_{ис}$. Это позволяет рассматривать предложенную модель $M_{ис}$ как универсум U понятийного пространства предметной области разрешения инцидентов в ИС, определяемого, в первую очередь, описанием инцидентов

и связанных с ними атомарными объектами в соответствии с предложенными методами классификации и структурирования знаний на основе структурного подхода и модели ISO/OSI.

База экспертных знаний, основанная на предложенной модели, позволит описать все доступное множество объектов и инцидентов и содержит все необходимые знания для обработки инцидентов, возникающих в ИС.

2.4. **Выводы**

Рассмотренные функциональные возможности онтологического подхода, состав решаемых задач и области его применения, а также общие принципы и особенности формирования онтологий позволили сделать вывод о целесообразности применения онтологического подхода к решению задач формализации и классификации экспертных знаний, полученных при обработке инцидентов в ИС с целью организации и наполнения базы знаний.

Для структурирования накопленных знаний предложены к реализации два независимых классификатора инцидентов.

Первый классификатор основан на применении структурного подхода, в соответствии с которым осуществляется последовательное выделение подсистем, аппаратно-программных компонентов и атомарных объектов в составе ИС. Понятия инцидента и объекта тесно связаны. Эта взаимосвязь является основой распределения и привязки инцидентов в данном классификаторе.

Второй классификатор, предложенный в монографии, основан на эталонной модели взаимодействия открытых систем ISO/OSI и позволяет логически упорядочивать, группировать и распределять инциденты по ее уровням. Распределение инцидентов может выполняться в зависимости как от того, к какому уровню модели относятся связанные с ними объекты, так и от того, на каком уровне было нарушено взаимодействие в результате возникновения инцидента.

Предложенные классификаторы для структурирования знаний в зависимости от специфики решаемых задач позволили:

- получать подмножество инцидентов, связанных с заданным объектом, и последовательно, в соответствии с распределением по уровням модели ISO/OSI, просматривать их;

- выбирать нужный уровень модели для указанной подсистемы или компонента ИС, после чего просматривать полученное подмножество инцидентов для нахождения связанных с искомым объектом.

Предложенные классификаторы инцидентов и онтологический подход позволили разработать математическую модель представления знаний об инцидентах в ИС, состоящую из трех уровней.

К первому (объектному) уровню модели относится совокупность множеств понятий-сущностей, соответствующих таким объектам, как ИС, их подсистемы, аппаратно-программные компоненты, атомарные объекты и их группы и множеств понятий-отношений, описывающих иерархические связи между ними. Распределение объектов осуществляется в соответствии с классификатором инцидентов, основанном на структурном подходе.

Вторым – промежуточным по отношению к уровням объектов и инцидентов – является уровень действий, включающий в себя множество действий, выполняемых над атомарными объектами для разрешения связанных с ними инцидентов. При этом действие или последовательность действий, выполненные при первоначальном разрешении инцидента и зафиксированные для дальнейшего использования, определяются благодаря причинно-следственным связям между объектами и инцидентами.

На третьем уровне – уровне инцидентов – находятся множества инцидентов, зафиксированных в процессе эксплуатации ИС, и специфических для предметной области обработки инцидентов в ИС причинно-следственных связей между ними. Объекты на уровне инцидентов распределяются в соответствии с классификатором, основанном на эталонной модели взаимодействия открытых систем ISO/OSI.

В зависимости от порядка просмотра уровней модель позволяет получать подмножества инцидентов, связанных с заданным объектом (прямой порядок сверху – вниз), или определять подмножество объектов, связанных с выбранным инцидентом (обратный порядок снизу – вверх). Организованная с использованием предложенной математической модели, содержащей, в том числе, формальное определение инцидента (2.16), база экспертных знаний позволяет описывать все доступное множество объектов и инцидентов. Она содержит все необходимые знания для обработки инцидентов, возникающих в ИС.

ГЛАВА 3

База экспертных знаний об инцидентах в ИС: реализация

3.1. Средства программной реализации системы приобретения и представления экспертных знаний

Основные требования, предъявляемые к программным средствам организации и наполнения базы экспертных знаний в соответствии с предложенной математической моделью (2.3), делятся на две группы.

Первая группа определяет требования, обусловленные спецификой разработанной модели, онтологического подхода и методов классификации и структурирования знаний об инцидентах в ИС и включает в себя:

- возможность создания и редактирования понятий-сущностей и их свойств;
- возможность определения и дальнейшего редактирования понятий-отношений, в том числе и неявных, между заданными понятиями-сущностями;
- поддержку логической организации определенных сущностей и отношений в соответствии с разработанными классификаторами инцидентов.

Вторая группа объединяет общие требования к архитектуре, функциональным возможностям и структурам данных, используемых разработанной системой приобретения и представления экспертных знаний:

- возможность реализации программного инструментального средства как в виде самостоятельного standalone-приложения, так и в виде Java-апплета;

- поддержка внешних источников данных, в том числе реляционных СУБД и XML-файлов, для хранения информации;
- возможность реализации механизма поиска по ключевым словам;
- поддержка визуализации определенных сущностей и отношений между ними, а также их свойств;
- поддержка представления знаний на русском и английском языках, использование русского языка в интерфейсе пользователя.

В качестве возможного средства программной реализации рассматривались как самостоятельные программные инструментальные средства визуализации – такие как Gephi, Graphviz и другие, так и онтологические системы [29], функциональность которых может быть расширена за счет подключения отдельных модулей и плагинов, например, OWLViz. Из всех рассмотренных программных средств выбран комплект средств разработки на основе свободного ПО Thinkmap SDK [66] – как наиболее полно отвечающий сформулированным ранее требованиям.

Данное программное инструментальное средство предназначено для разработки приложений, позволяющих визуализировать и редактировать значительные объемы данных, представленных, в том числе с использованием естественных языков, их графических интерфейсов и интегрированных механизмов поиска. Отказ от использования онтологических систем для решения поставленной задачи в данной работе объясняется ограниченностью их возможностей по созданию самостоятельных приложений[16]. Основным ограничением Thinkmap SDK по сравнению с онтологическими системами является отсутствие подключаемых систем логического вывода.

Вместе с тем, для проверки непротиворечивости онтологии обработки инцидентов в ИС, на основе которой организуется база экспертных знаний, может быть дополнительно использована любая онтологическая система со встроенным или подключаемым механизмом логического вывода, например, Protégé.

---

[16] Под самостоятельными в данном контексте подразумеваются приложения, для работы с которыми пользователь не должен обладать специальными знаниями теории онтологий, в том числе – о правилах построения запросов.

3.1.1. **Функциональные возможности Thinkmap SDK**

Комплект средств разработки Thinkmap SDK предназначен для создания приложений, решающих задачу обработки и визуализации больших объемов данных. К основным областям его применения относятся обработка знаний [14, 34], графическое представление потоков работ, ИТ ресурсов организации и топологий информационных сетей.

Средства разработки Thinkmap позволяют создавать самостоятельные standalone-приложения и Java-апплеты с использованием ряда современных технологий (Java EE, HTML5). В качестве источника данных в обоих случаях могут использоваться реляционные СУБД, XML-файлы или специальным образом подготовленные файлы данных (flatfile). Источники данных могут располагаться как локально на компьютере пользователя, так и на удаленном сервере. Таким образом, существует четыре различных варианта архитектуры приложений, разработанных с помощью Thinkmap:

- Standalone-приложение, использующее локальные источники данных. В этом случае файлы приложения и данных располагаются на компьютере пользователя, необходимость использования сетевого подключения отсутствует. Основным недостатком является невозможность совместного использования данных и их своевременного централизованного обновления.
- Standalone-приложение, использующее в качестве источника данных удаленный сервер. Данный сценарий позволяет централизованно вносить изменения и поддерживать в актуальном состоянии используемую информацию; требует установки приложения на компьютер пользователя и наличия сетевого подключения.
- Java-апплет, использующий локальные источники данных. Позволяет работать с Web-интерфейсом, не требуя установки приложения и наличия доступа к удаленному серверу; обязательным условием является дополнительная настройка параметров, связанных с обеспечением безопасного доступа к данным на диске.
- Java-апплет, использующий в качестве источника данных удаленный сервер. Наиболее распространенная архитектура, позволяющая обеспечить совместную работу и централизованный доступ к используемой информации при наличии сетевого подключения.

Основными структурными единицами, используемыми в Thinkmap для представления информации, являются узлы (Node)

и дуги (Edge). Данные объекты соответствуют понятиям-сущностям и отношениям между ними разработанной ранее концептуальной модели представления знаний об инцидентах в ИС. Также они могут интерпретироваться как вершины и дуги в теории графов. Поддерживается возможность задания свойств узлов и дуг, описывающих значимые параметры объектов и их отношений.

Важной особенностью Thinkmap является возможность создания сложных правил выбора и отображения указанных объектов, основанных на анализе значений их свойств и отношений, а также использования различных условий при обходе полученных графов в зависимости от состава решаемых задач. В данной работе они используются для визуализации неявных причинно-следственных связей между объектами в составе ИС и возникающими инцидентами в базе экспертных знаний. Поддержка настройки критериев отбора объектов в графические представления позволяет логически организовать накопленные в процессе эксплуатации ИС знания в соответствии с разработанными методами их классификации.

Комплект средств разработки Thinkmap SDK включает в себя встроенные механизмы поиска, а также позволяет создавать и интегрировать собственные. Поиск осуществляется по всему объему хранимых в базе знаний, допускается возможность выполнения поисковых запросов на русском языке. Также поддерживается использование русского языка при описании фиксируемых знаний и в графическом интерфейсе приложения.

3.1.2. **Компоненты Thinkmap SDK**

В состав комплекта средств разработки Thinkmap SDK входят следующие элементы:

- ядро (Thinkmap Core);
- сервер приложений (Thinkmap Server);
- интерфейс для подключения источников данных (Thinkmap DataSources).

Ядро объединяет набор компонентов, использующихся для обработки и визуализации данных. Основными задачами, решаемыми ядром Thinkmap, являются получение нужной информации об объектах из источника данных, определение набора объектов, отбираемых в то или иное представление, вычисление расположения выбранных

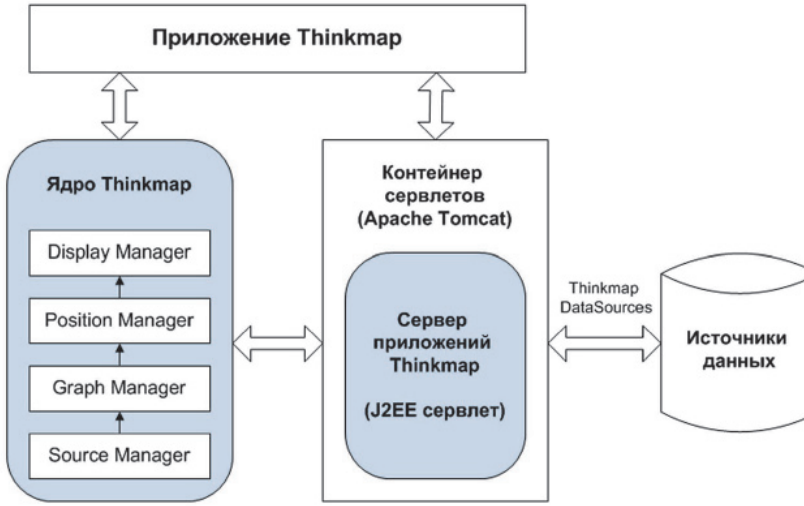

Рис. 19. Взаимодействие компонентов Thinkmap SDK

объектов друг относительно друга на основании связей между ними, их общего числа и других параметров и непосредственно визуализация. Таким образом, ядро состоит из четырех архитектурных уровней, на каждом из которых последовательно решается соответствующая задача. Ее выполнение обеспечивает менеджер каждого уровня: менеджер источников данных (Source Manager), менеджер представлений (Graph Manager), менеджер расположения (Position Manager), менеджер отображения (Display Manager). Перечисленные уровни образуют иерархию – каждый последующий уровень является потомком предыдущего и может обрабатывать только подмножество объектов, полученных от родительского уровня.

Сервер Thinkmap исполняет роль сервера приложений и предназначен для обработки запросов, поступающих от приложений, разработанных с использованием инструментальных средств Thinkmap Core. Запросы создаются на стороне клиента каждый раз при обновлении текущего представления или необходимости получить новый набор данных. В соответствии с содержанием запроса сервер Thinkmap обращается к источнику данных, получает необходимую информацию и возвращает ее клиенту. Обращение осуществляется

сервером с использованием интерфейса для подключения источников данных (Thinkmap Datasources). В качестве источников данных могут использоваться реляционные базы данных. В этом случае интерфейсами являются стандартные JDBC драйверы или JNDI. Также источниками данных могут быть XML-файлы или файлы данных. Вне зависимости от способа хранения накопленной информации взаимодействие сервера приложений и подключенных источников данных обеспечивает актуальность используемых клиентской частью данных за счет их динамического обновления.

На рис. 19 представлена схема взаимодействия компонентов в составе комплекта средств разработки Thinkmap SDK.

3.2. **Объекты и структуры данных онтологии обработки инцидентов в ИС**

Таксономия, основанная на разработанной во второй главе монографии концептуальной модели представления знаний об инцидентах в ИС, включает в себя шесть типов понятий-сущностей и шесть типов понятий-отношений. В процессе ее программной реализации определены структуры данных, описывающие соответствующие объекты в инструментальном средстве Thinkmap SDK – узлы и дуги. Для описания объектов и их свойств используется язык разметки XML[17].

Структура объекта типа «информационная система организации» представлена на рис. 20.

Описание используемых свойств для всех типов узлов (кроме инцидентов) приведено в табл. 3. Для хранения их значений выбран строковый тип данных. Далее приведен пример записи о конкретном объекте в соответствии с рассмотренной структурой.

```
<node type=«ENTERPRISE» id=«1» >
    <property name=«enterprisename» value=«ИС1» />
    <property name=«comment» value=«Информационная система 1» />
</node>                                                    (3.1)
```

[17] Для хранения данных о рассмотренных далее объектах, их свойствах и отношениях между ними также может использоваться реляционная СУБД.

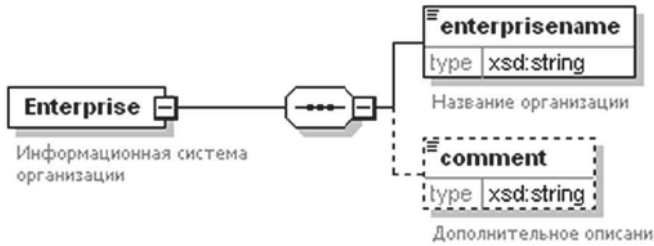

Рис. 20. Структура объекта «информационная система организации»

При создании записей об объектах необходимо гарантировать уникальность каждой из них. Для этого в структуру записи включен служебный параметр «id» – уникальный идентификатор объекта. Наличие иерархических отношений между объектами разных типов, а также использование классификатора, основанного на модели взаимодействия открытых систем ISO/OSI, позволяет задавать значения данного параметра в соответствии с нотацией ASN.1 (Abstract Syntax Notation One [54, 62]), разработанной совместно ISO и ITU-T. В данной нотации идентификатор объекта является иерархическим и состоит из обозначения самого объекта, а также всех объектов верхних уровней, связанных с ним иерархическими отношениями. Использование данного подхода в монографии позволяет определить тип объекта или подсистему, в которой возник описываемый инцидент, на основании его уникального идентификатора.

Объекты типа «информационная система организации» являются корневыми в иерархии концептуальной модели и в программной реализации системы приобретения и представления экспертных знаний имеют уникальный идентификатор от 1 до 9.

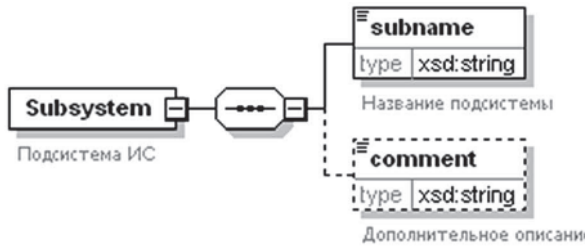

Рис. 21. Структура объекта «подсистема ИС»

Объекты типа «подсистема ИС» описываются с помощью следующей структуры данных, см. рис. 21.

Уникальный идентификатор объектов данного типа состоит из двух цифр: первая указывает на ИС, в которую входит подсистема, вторая обозначает саму подсистему.

<node type=«SUBSYSTEM» id=«19» >
<property name=«subname» value=«Почтовая система» />
<property name="comment" value="Настройки почтового сервера, клиентов, маршрутизация почты" />
</node> (3.2)

Таким образом, значение служебного параметра id, равное 19, уникальным образом идентифицирует объект второго уровня иерархии (уровень подсистем) с порядковым номером 9, входящим в состав первой из рассматриваемых ИС.

В монографии множество подсистем ИС $X_п$ (2.7) состоит из следующих объектов (экземпляров класса «подсистема ИС»):

$X_п$ = {серверное оборудование, активное сетевое оборудование, пассивное сетевое оборудование, рабочие станции, периферийные устройства, сеть хранения данных, операционнные системы, виртуальная инфраструктура, почтовая система, Web инфраструктура}.

Структура данных, описывающая объекты нижележащего уровня иерархии – аппаратно-программные компоненты в составе подсистем, содержит следующие параметры, рис. 22:

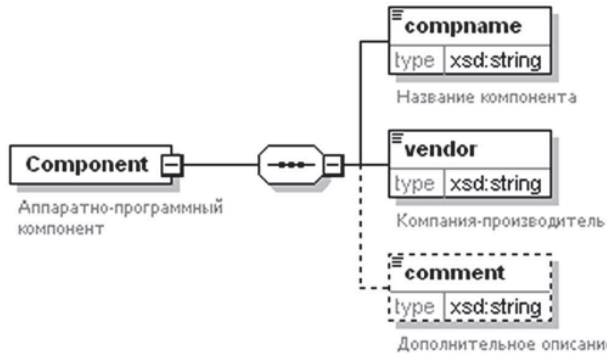

Рис. 22. Структура объекта «аппаратно-программный компонент»

На уровне аппаратно-программных компонентов идентификатор объекта состоит из трех цифр: первая обозначает текущую ИС, вторая – порядковый номер подсистемы в ее составе, третья – сам компонент подсистемы.

```
<node type=«COMPONENT» id=«191» >
    <property name=«compname» value=«Lotus Domino 8.5.2» />
    <property name=«vendor» value=«IBM» />
    <property name="comment" value="Серверное программное обеспечение" />
</node>                                                    (3.3)
```

Идентификатор «191» в данном случае указывает на первый компонент подсистемы «почтовая система» с порядковым номером девять в составе первой из рассматриваемых ИС.

Множество $X_к$ (2.7) для почтовой системы[18] задано следующим образом:

$X_к$ = {Lotus Domino 8.5.2, Lotus Notes 7.0.4, Lotus Notes 8.5.2, Lotus iNotes 8.5.2}.

Группы объектов в составе аппаратно-программных компонентов описываются следующей структурой данных, рис. 23.

Четырехзначные идентификаторы групп объектов состоят из трехзначного префикса, обозначающего аппаратно-программный компонент (3.3), в составе которого они выделяются, и порядкового

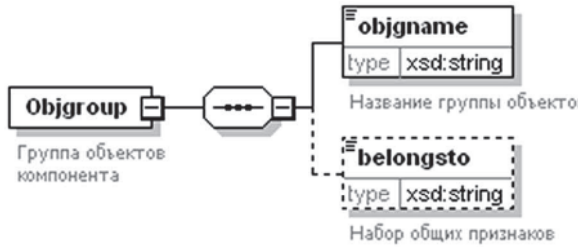

Рис. 23. Структура «группы объектов компонента»

[18] В данном разделе в качестве примера приведена иерархия объектов для почтовой системы. Описание остальных объектов онтологии обработки инцидентов в ИС, их свойств и отношений дано в приложении Б.

номера самой группы. Атрибут «belongsto», содержащий описание общих для всех объектов в группе признаков, является необязательным и может не заполняться.

```
<node type=«OBJGROUP» id=«1910» >
    <property name=«objgname» value=«Почта» />
    <property name=«belongsto» value=« » />
</node>                                                    (3.4)
```

Объекты множества $X_г$ (2.7) для почтовой системы приведены в табл. 6.

Следующим типом в иерархии являются атомарные объекты, с которыми непосредственно связываются возникающие в ИС инциденты. Для их описания используется следующая структура данных, рис. 24.

Пятизначные идентификаторы объектов последовательно формируются из номеров рассматриваемой ИС, ее подсистемы, аппаратно-программного компонента в составе подсистемы, группы объектов внутри компонента и самого объекта.

```
<node type=«OBJECT» id=«19102» >
    <property name=«objname» value=«Серверная задача SMTP» />
    <property name=«belongsto» value=« » />
</node>                                                    (3.5)
```

Объекты множества X_{ao} (2.7) для почтовой системы приведены в табл. 6.

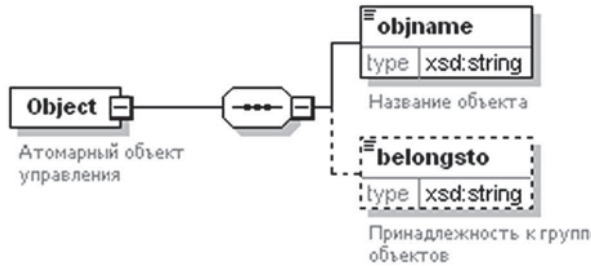

Рис. 24. Структура объекта «атомарный объект»

Табл. 6. Объекты множеств $X_к$, $X_г$, $X_{ао}$ для почтовой системы

№№	$X_к$	$X_г$	$X_{ао}$
1	Lotus Domino 8.5.2	Общие	NServer.exe
2			NService.exe
3			ID сервера
4			ACL
5		Domino Directory	Пароль пользователя
6			Интернет пароль пользователя
7			Срок действия ID пользователя
8		Почта	Серверная задача Router
9			Серверная задача SMTP
10			Серверная задача IMAP
11			Серверная задача POP3
12			mail.box
13		Web	Серверная задача HTTP
14			Серверная задача LDAP
15			Directory Assistance
16	Lotus Notes 7.0.4	Пользовательские настройки	notes.ini
17			Address Book
18		ID пользователя	Корневой сертификатор
19	Lotus Notes 8.5.2	Доступ	Доступ к серверу
20			Доступ к БД на сервере
21			Доступ к документу в БД
22			Доступ к полю документа
23	Lotus iNotes 8.5.2	iNotes Почта	Входящие сообщения
24			Исходящие сообщения
25		iNotes Настройки	Панель управления

Объекты типа «инцидент» являются основными в концептуальной модели представления знаний об инцидентах в ИС. В соответствии с формальным определением (2.16), предложенным во второй главе, для описания значимых свойств и признаков инцидентов (множество $P(X_и)$) в данной работе используется следующая структура данных, рис. 25.

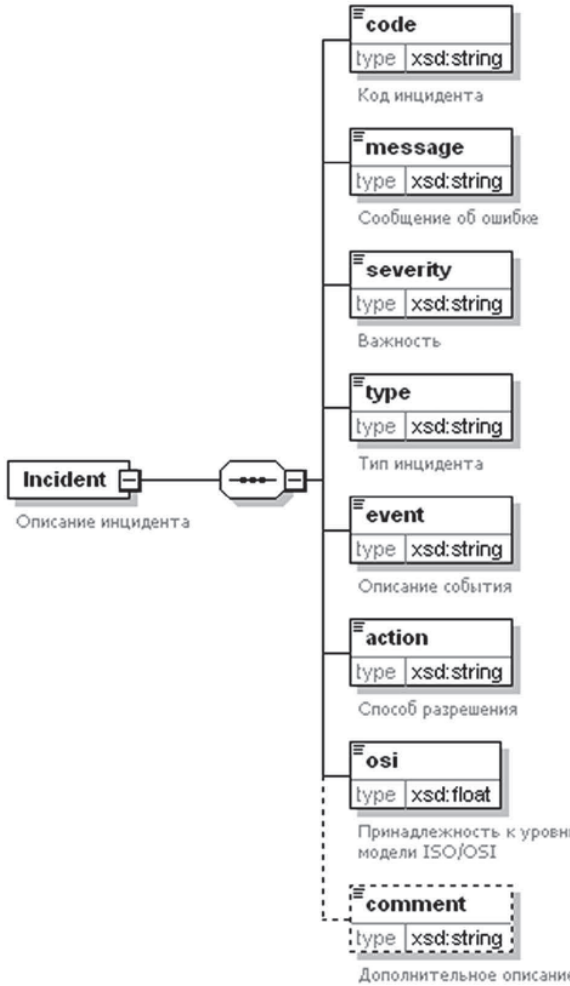

Рис. 25. Структура свойств объекта типа «инцидент»

Описание свойств приведено в разделе 2.2.1 монографии. Дополнительный служебный параметр code, не рассматриваемый ранее, предназначен для хранения символьного идентификатора инцидента в следующем формате: INC_xxxx, где xxxx – порядковый номер инцидента, внесенного в базу знаний, от 0001 до 9999. Использование символьного идентификатора позволяет отображать большое коли-

чество инцидентов, не занимая рабочее пространство графического интерфейса пользователя их детальным описанием. Универсальный идентификатор инцидента представляет собой шестизначное число; первые пять цифр являются идентификатором объекта (3.5), приведшего к его возникновению, шестая – порядковый номер самого инцидента. В случае возникновения инцидента из-за воздействия внешних факторов используется идентификатор атомарного объекта, функционирование которого было в результате нарушено. Объекты множества $X_и$ (2.9) заносятся в базу знаний в процессе функционирования системы приобретения и представления экспертных знаний. Примеры записей об инцидентах, созданных в соответствии с рассмотренной структурой, приведены в приложении Б.

Отношения между объектами в среде разработки Thinkmap также заданы с помощью структур данных в формате XML. Связь типа «подсистема в составе ИС» описывается следующим образом, рис. 26.

Выбор объектов, для которых определяется отношение и его свойства, осуществляется с использованием их уникальных идентификаторов:

<edge type=«ENTERPRISE_SUBSYSTEM» fromid=«1» toid=«19» />
(3.7)

В данном примере задается связь между объектом с идентификатором, равном единице (ИС организации, верхний уровень иерархии) и объектом типа «подсистема ИС», значение параметра id которого равно 19. Разработанная система приобретения и представления экспертных знаний в ходе эксплуатации нескольких ИС позволяет создавать множественные связи для нескольких объектов: одна ИС состоит из множества подсистем, одна подсистема является составной частью нескольких ИС, решающих типовые задачи.

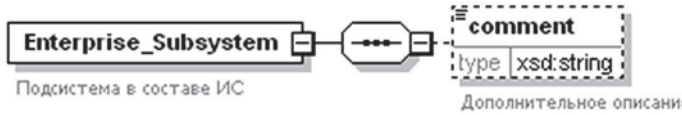

Рис. 26. Структура связи «подсистема в составе ИС»

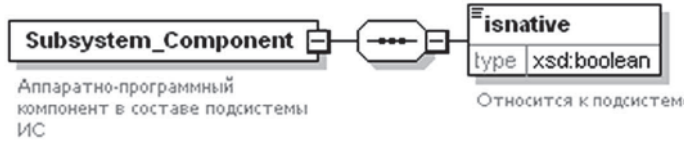

Рис. 27. Структура связи «аппаратно-программный компонент в составе подсистемы ИС»

Структура данных (рис. 27), описывающая связь типа «аппаратно-программный компонент в составе подсистемы ИС», включает в себя параметр isnative (описание и назначение свойств отношений приведено в табл. 4).

Значение данного параметра (тип данных – логический) позволяет определить, является ли связь иерархической или используется для определения отношений, специфических для объектов с заданными идентификаторами.

```
<edge type=«SUBSYSTEM_COMPONENT» fromid=«19»
toid=«191» >
        <property name=«isnative» value=«true» />
</edge>
<edge type=«SUBSYSTEM_COMPONENT» fromid=«19»
toid=«182» >
        <property name=«isnative» value=«false» />
</edge>                                                    (3.8)
```

В первом случае компонент входит в состав подсистемы ИС (идентификатор компонента состоит из значения параметра id подсистемы и номера компонента), во втором – может влиять на ее функционирование (между объектами отсутствуют непосредственные иерархические отношения, компонент относится к другой подсистеме).

Для описания связи типа «группа объектов в составе аппаратно-программного компонента» используется структура данных, представленная на рис. 28.

Как и для остальных типов связей, задание уникальных идентификаторов позволяет определить объекты, отношение между которыми описывается с помощью данной связи.

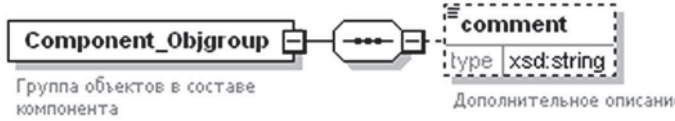

Рис. 28. Структура связи «группа объектов в составе аппаратно-программного компонента»

<edge type=«COMPONENT_OBJGROUP» fromid=«191» toid=«1910» /> (3.9)

В ряде случаев одна и та же группа объектов может входить в состав нескольких аппаратно-программных компонентов. Например, «дисковая подсистема» является частью всех серверов в ИС, для которой фиксируются экспертные знания. При этом все компоненты, включающие общую группу объектов, относятся к одной и той же подсистеме ИС.

Связь между объектами и группой, в которую они объединены, задается следующей структурой данных, рис. 29.

Как и для предыдущего типа связи (3.9), один атомарный объект может входить в состав нескольких групп. Подобная организация связей позволяет, например, зафиксировать возможность как централизованного редактирования учетных записей администратором на сервере, так и локального – пользователем с помощью клиентского ПО. Далее представлен пример связи типа «объекты в составе группы»:

<edge type=”OBJGROUP_OBJECT” fromid=”1910” toid=”19102” /> (3.10)

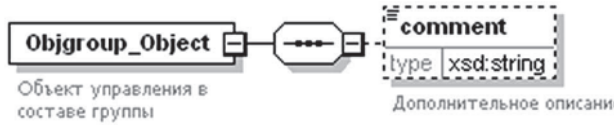

Рис. 29. Структура связи «объект в составе группы»

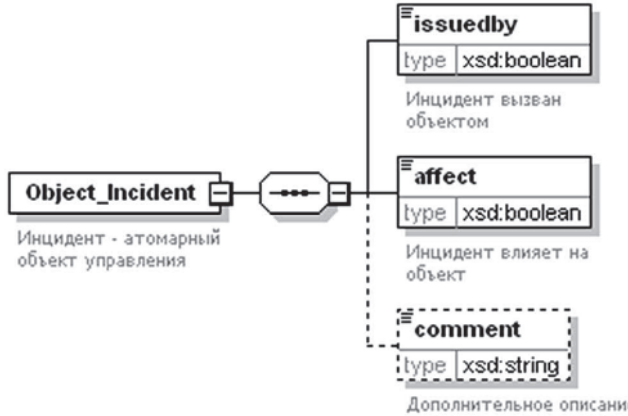

Рис. 30. Структура и свойства связи «инцидент – атомарный объект»

Структура данных, описывающая отношение типа «инцидент – атомарный объект», содержит два параметра, позволяющих определить причинно-следственные связи между указанными объектами, рис. 30.

В отличие от рассмотренных выше (3.7)–(3.10) иерархических связей, данный тип отношений является специфическим для предметной области обработки инцидентов в ИС и позволяет фиксировать связи между объектами и возникающими инцидентами. Количество таких связей для объекта или инцидента не ограничено и зависит как от степени воздействия инцидента (определяемой, в том числе, количеством затронутых объектов), так и от того, сколько различных инцидентов может привести к нарушению штатного режима функционирования отдельно взятого объекта. Множества отношений данного типа $R(X_{ао}, X_{и})$ и их свойств $P(R(X_{ао}, X_{и}))$ входят в состав формального определения инцидента (2.16). Далее представлен пример структуры связи данного типа:

```
<edge type="OBJECT_INCIDENT" fromid="19102" toid="191021" >
    <property name=«issuedby» value=« true» />
    <property name=«affect» value=«false « />
    <property name=«comment» value=«Вызван» />
</edge>                                                    (3.11)
```

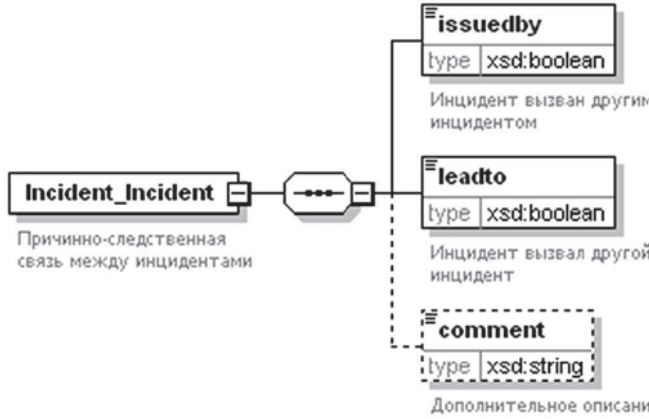

Рис. 31. Структура и свойства связи «инцидент – инцидент»

Вторым специфическим типом отношений в разработанной концептуальной модели представления знаний об инцидентах в ИС является связь «инцидент – инцидент», рис. 31.

Данное отношение позволяет выявлять и фиксировать причинно-следственные связи между несколькими инцидентами в случае, когда источником их возникновения являются не объекты, а другие инциденты, возникающие как в других подсистемах рассматриваемой ИС, так и за ее пределами. Уникальные идентификаторы инцидентов, для которых задается взаимосвязь, не связаны иерархическими отношениями:

```
<edge type=«INCIDENT_INCIDENT» fromid=«190111»
toid=«193111» >
        <property name=«issuedby» value=«false» />
        <property name=«leadto» value=«true» />
        <property name=«comment» value=«Вызвал» />
</edge>                                                    (3.12)
```

Применение двух типов специфических связей позволяет осуществлять поиск инцидента как по затронутым им объектам, так и по известным связям с другими инцидентами. Объекты множеств $R(X_{ao}, X_и)$ и $R(X_и, X_и)$ заносятся в базу знаний в процессе функционирования системы приобретения и представления экспертных знаний.

Табл. 7. Количественные показатели для объектов в онтологии обработки инцидентов в ИС

Класс объекта	Количество экземпляров класса
Понятия-сущности	
ИС организации	1
Подсистема ИС	10
Аппаратно-программный компонент	31
Группа объектов	60
Атомарный объект	109
Понятия-отношения	
Подсистемы в составе ИС	10
Аппаратно-программные компоненты в составе подсистемы ИС	42
Группы объектов в составе аппаратно-программного компонента	75
Объекты в составе группы	115

Рассмотренные структуры данных (3.1)–(3.12) позволяют описать значимые объекты и отношения между ними в соответствии с концептуальной моделью представления знаний об инцидентах в ИС в формате XML. При использовании в качестве источника данных реляционной СУБД заданные параметры сохраняются; меняется только формат их описания. Так, уникальные идентификаторы объектов становятся первичными ключами для соответствующих записей.

Общее количество объектов разных классов в онтологии обработки инцидентов в ИС на этапе ее программной реализации приведено в табл. 7.

Описание объектов в соответствии с приведенными структурами данных дано в приложении Б.

3.3. Принципы построения представлений

Все объекты, обрабатываемые и визуализируемые с помощью средств Thinkmap SDK, представляются в виде основных структурных элементов – узлов и дуг. Это позволяет организовывать данные, получаемые из внешних источников, в виде графа. При большом количестве объектов в ИС и связанных с ними инциден-

тов хранение в оперативной памяти и визуализация всего графа может потребовать использования значительных системных ресурсов. Более эффективной с этой точки зрения является работа с определенным подмножеством объектов в заданный момент времени с возможностью перехода к новому подмножеству при необходимости. В данной книге такое подмножество, выбранное в соответствии со значениями указанных параметров, рассматривается как представление.

С позиции онтологического подхода данная операция аналогична поиску термина, представляющего интерес для решения определенной задачи, и формирования соответствующего контекста. Под контекстом здесь подразумевается часть (срез) онтологии, значимая для рассматриваемой задачи.

В соответствии с концептуальной моделью и методами классификации знаний об инцидентах разработанная система приобретения и представления экспертных знаний позволяет работать с *шестью* типами представлений, формируемых в зависимости от центрального для данного представления понятия-сущности.

Первый тип, используюемый в качестве точки входа в графическом интерфейсе пользователя, содержит название организации, для которой фиксируются экспертные знания, а также все подсистемы ее ИС. При переходе к произвольной подсистеме – выборе ее в качестве нового центрального элемента – становится доступно описание и перечень всех аппаратно-программных компонентов, входящих в ее состав. Для построения данного представления (отбора всех входящих в него объектов и связей между ними) используются следующие условия обхода графа, в виде которого организованы данные о сущностях и отношениях предметной области.

Для подсистемы ИС, выбранной в качестве нового центрального узла, последовательно просматриваются все связанные с ней дуги, соответствующие связям типа «аппаратно-программные компоненты в составе подсистемы ИС», а для каждой из них – все узлы типа «аппаратно-программный компонент», с которыми они, в свою очередь, связаны. В случае если узлы и дуги удовлетворяют заданным условиям, они отбираются в соответствующее представление.

В используемом комплекте средств разработки Thinkmap SDK данный алгоритм записывается с помощью т.н. терминальных усло-

вий (terminal condition) и может быть представлен следующим образом в формате XML:

```
<sequence match=«all» >
    <condition>is('SUBSYSTEM')</condition>
    <condition>is('SUBSYSTEM_COMPONENT') &amp;amp;&amp;amp;
isnative</condition>
    <condition>is('COMPONENT')</condition>
</sequence>                                                    (3.13)
```

Данная синтаксическая конструкция означает, что в представление, центральным элементом которого является объект типа «подсистема ИС» (Subsystem), отбираются только объекты типа «аппаратно-программный компонент» (Component), связанные с ним непосредственно дугами, соответствующими связям типа «аппаратно-программные компоненты в составе подсистемы ИС», рис. 32.

На рис. 32 пунктиром выделены объекты, входящие в представление, формируемое для подсистем в составе ИС. Дополнительным условием отбора является истинность логического параметра isnative, определяющего принадлежность компонента к подсистеме. Это позволяет отбирать объекты, связанные иерархическими отношениями с центральным элементом и игнорировать специфические отноше-

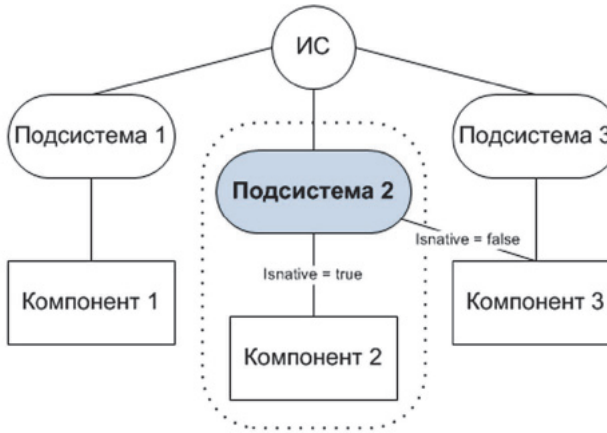

Рис. 32. Представление, формируемое для объекта типа «Подсистема ИС»

ния, обозначаемые с помощью того же типа связей, в данном представлении (на рис. 32 – компонент 3).

Представление следующего уровня строится для объектов типа «аппаратно-программный компонент». В соответствии с концептуальной моделью представления знаний об инцидентах в ИС любой компонент может входить в состав только одной подсистемы; при этом нарушение штатного режима его функционирования может оказывать влияние на подсистемы, не связанные с ним иерархическими отношениями. Таким образом, в данное представление отбираются не только «родительская» подсистема, но и все подсистемы, на функционирование которых компонент может влиять, что подразумевает использование дополнительных условий при обходе графа. Кроме того, представление данного уровня содержит группы объектов, выделенные в составе аппаратно-программного компонента.

Условия построения представлений двух следующих уровней (строящихся, соответственно, для центральных элементов типа «группы объектов в составе аппаратно-программного компонента» (3.14) и «объекты в составе группы» (3.15)) описываются с помощью следующих конструкций в формате XML:

```
<sequence match=«all» >
    <condition>is(‘OBJGROUP’)</condition>
    <condition>is(‘EDGE’)</condition>
    <condition>is(‘NODE’)</condition>
</sequence>                                        (3.14)

<sequence match=«all» >
    <condition>is(‘OBJECT’)</condition>
    <condition>is(‘EDGE’)</condition>
    <condition>is(‘NODE’)</condition>
</sequence>                                        (3.15)
```

Их отличительной особенностью является использование универсальных для Thinkmap SDK типов объектов Node – узлы и Edge – дуги. В соответствии с разработанной концептуальной моделью это позволяет отбирать в представления все объекты, связанные с цен-

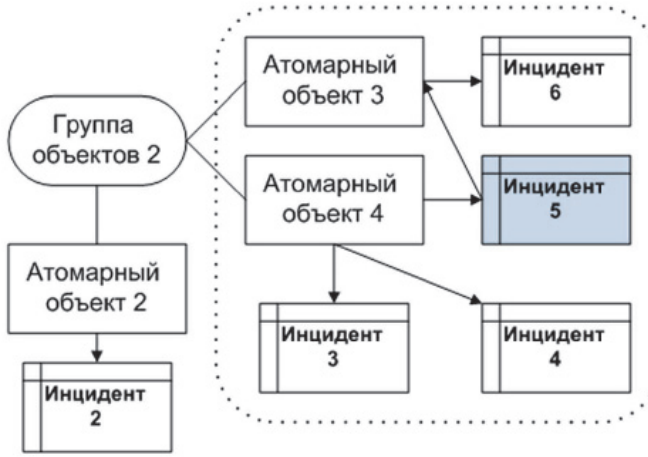

Рис. 33. Представление, содержащее в качестве центрального элемент типа «инцидент»

тральными элементами иерархическими отношениями, включая возможные связи типа «многие ко многим»[19].

Шестой тип представлений содержит в качестве центрального элемент типа «инцидент», рис. 33.

При отборе объектов в данное представление учитываются следующие положения концептуальной модели:

- инцидент может приводить к нарушению функционирования объектов в составе ИС, и, таким образом, являться корневой причиной возникновения других инцидентов;
- инцидент может быть вызван как нарушением штатного режима функционирования произвольного атомарного объекта, так и другим инцидентом;
- один инцидент может оказывать влияние на большое число объектов в составе различных подсистем ИС; неисправность различных объектов в составе ИС может приводить к возникновению одного и того же инцидента.

В соответствии с данными утверждениями условие обхода графа записывается следующим образом:

[19] Например, одна группа объектов, выделенная в разных аппаратно-программных компонентах или атомарный объект, входящий в состав нескольких групп.

```
<sequence match=«any» >
    <condition>is('INCIDENT')</condition>
    <condition>is('EDGE')</condition>
    <condition>is('NODE')</condition>
    <condition>is('OBJECT_INCIDENT')</condition>
    <condition>is('OBJECT')</condition>
</sequence>                                          (3.16)
```

Условие (3.16) позволяет отбирать в представление все атомарные объекты, связанные с рассматриваемым инцидентом, все инциденты, которые могли привести к его возникновению или, в свою очередь, быть вызваны им (таким образом формируется ε-окрестность инцидента, понятие которой рассмотрено в разделе 2.3 монографии), а также все элементы типа «объект», с ними связанные. Ключевое слово «any», использующееся в отличие от «all» в рассмотренных выше условиях, позволяет выбирать объекты, частично соответствующие указанной последовательности.

На рис. 33 подмножество объектов, отобранных в представление с центральным элементом «Инцидент 5», строится следующим образом: выбираются элементы типа «объект», связанные с ним – «Атомарный объект 4», приведший к его возникновению, и «Атомарный объект 3», чье функционирование было нарушено данным инцидентом. Далее для каждого объекта выбираются все элементы типа «инцидент», связанные с ним причинно-следственными отношениями типа «инцидент – атомарный объект».

На рис. 34 представлена общая схема формирования представлений для подмножеств объектов разных типов.

Цветом выделены объекты, являющиеся центральными элементами, пунктиром обведены все объекты и связи между ними, отбираемые в соответствующие представления.

3.4. **Компоненты графического интерфейса**

Комплект средств разработки Thinkmap SDK позволяет создавать графические интерфейсы разрабатываемых приложений с использованием типовых средств. Основным из них является библиотека элементов (окон, меню, кнопок, текстовых полей) пользовательских интерфейсов – GEL. Данная библиотека ориентирована на приме-

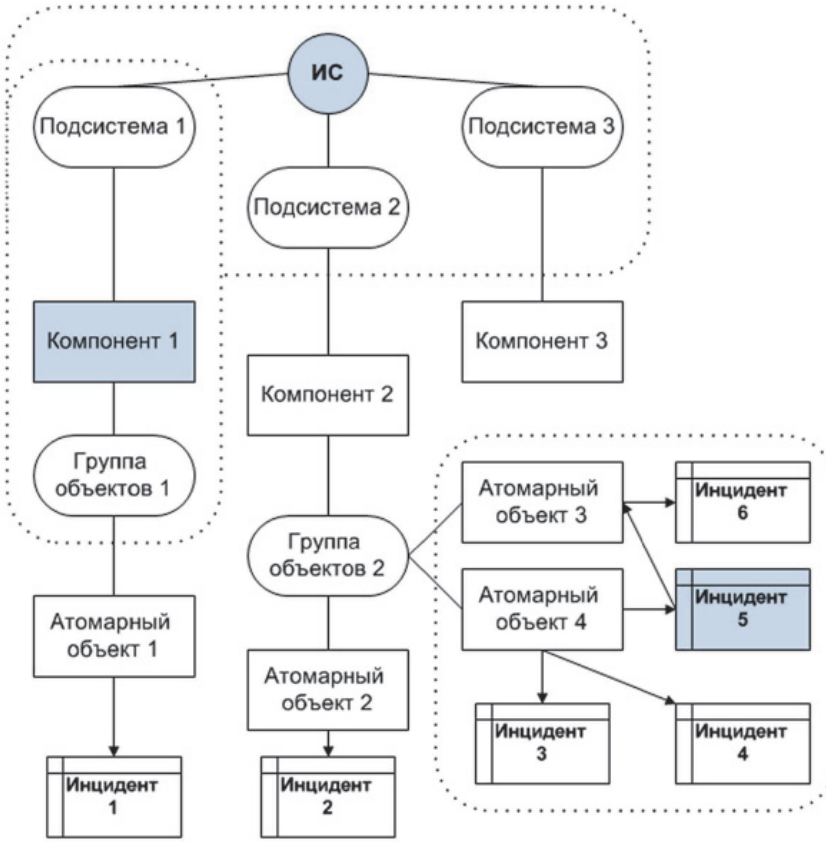

Рис. 34. Подмножества объектов разных типов, отбираемых в представления

нение в Java-приложениях и может использоваться совместно с Java AWT или Swing. При этом она содержит все необходимые компоненты для обеспечения взаимодействия с ядром Thinkmap и предоставляет возможности конфигурирования и настройки отображения как с использованием языка программирования Java, так и с помощью языка разметки XML и каскадных таблиц стилей CSS.

На рис. 35 представлен интерфейс системы приобретения и представления знаний, использующейся для организации и наполнения базы экспертных знаний об инцидентах в ИС.

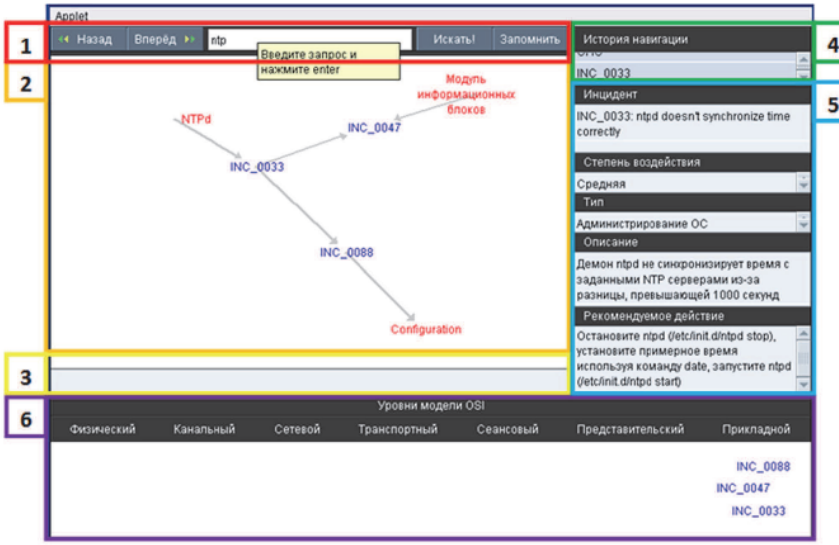

Рис. 35. Интерфейс системы приобретения и представления экспертных знаний

Ее основными компонентами являются:

Панель инструментов (1). На ней расположены инструменты навигации – кнопки «Назад» – для перехода к предыдущему представлению в рабочей области, «Вперёд» – для возвращения к исходному представлению и строка поиска. Поиск осуществляется по всем записям об объектах в системе и является полнотекстовым. Для каждой записи просматриваются значения всех полей, что позволяет находить объекты не только по соответствию ключевых слов, но и по вхождению поискового запроса или его части, например, в описание способа разрешения инцидента.

Для выполнения поиска достаточно нажать клавишу «Enter» после ввода поискового запроса или кнопку «Искать», расположенную после строки поиска; данная информация выводится также во всплывающем сообщении-подсказке, появляющемся при помещении курсора в поле ввода. В случае наличия единственного объекта, удовлетворяющего поисковому запросу, автоматически осуществляется переход к новому представлению, в котором он является центральным элементом. Множественные результаты поиска содержатся в дина-

мически изменяемой области под панелью инструментов. Данная область занимает часть рабочего пространства и может отображать до пяти ссылок на объекты одновременно (для доступа к остальным объектам используется полоса прокрутки).

Рабочая область (2). Является основным компонентом графического интерфейса; предназначена для отображения представлений рассмотренных выше типов в соответствии с разработанным классификатором инцидентов, основанном на структурном подходе.

Узлы, соответствующие всем типам понятий-сущностей, кроме инцидентов, обозначаются с помощью своих названий – значений соответствующих полей структур данных. Названия могут записываться как на латинице (сетевой демон NTPd на рис. 35), так и на кириллице (например, Модуль информационных блоков, там же). Для обозначения объектов, соответствующих инцидентам, в рабочей области используется значение служебного параметра code описания инцидента.

Связи между узлами – дуги – отображаются в виде ненаправленных линий для обозначения иерархических отношений между объектами в представлении, например, подсистем в составе ИС или атомарных объектов, объединенных в группы, и направленных – для обозначения причинно-следственных связей между инцидентами или объектами и инцидентами.

Возможны следующие варианты направлений, рис. 35:

- от инцидента к инциденту – определяет, был ли данный инцидент (INC_0047) вызван другими (INC_0033), или, в свою очередь, привел к их возникновению;
- от инцидента к объекту – исходный инцидент (INC_0088) нарушает штатный режим функционирования данного объекта;
- от объекта к инциденту – неисправность или неверная конфигурация объекта (NTPd) приводит к возникновению одного или нескольких инцидентов (INC_0033).

Для перемещения между представлениями и поиска объектов может использоваться как строка поиска панели инструментов, так и рабочая область: наведение указателя мыши на произвольный узел и щелчок левой кнопкой позволяет перейти к представлению, для которого выбранный объект является центральным. Таким образом, рабочая область позволяет пользователю последовательно просматривать граф, в виде которого организованы данные о сущностях

и отношениях предметной области обработки инцидентов в ИС, а выбор подмножества объектов на каждом шаге осуществляется в соответствии с рассмотренными выше условиями.

Строка состояния (3). Расположена непосредственно под рабочей областью и предназначена для вывода контекстных сообщений. При наличии дополнительной информации в поле comment структуры данных, описывающих объект, достаточно навести указатель мыши на соответствующий узел в рабочей области для ее отображения в строке состояния. Данная информация, в зависимости от типов объектов (приведены в табл. 3), может включать в себя развернутое описание-подсказку, отражающее особенности структуры объекта и состава решаемых задач, характеристику аппаратно-программного компонента или набор ключевых понятий и фраз, описывающих инцидент в целом.

История навигации (4). Дополнительным способом перехода между представлениями является история навигации (расположена справа от панели инструментов и рабочей области). В отличие от кнопок «Назад» и «Вперёд», позволяющих переходить к предыдущему и возвращаться к текущему представлениям, история навигации может использоваться для перехода к любому представлению, которое было открыто в течение сеанса работы с системой.

Панель свойств инцидента (5). Для представлений, содержащих в качестве центрального элемента объект типа «инцидент», панель свойств заполняется значениями полей соответствующей структуры данных. При этом выполняются следующие сопоставления:

- в область «Инцидент» записываются значения полей code и message;
- в область «Степень воздействия» – записывается значение поля severity;
- в область «Тип» – записывается значение поля type;
- в область «Описание» – записывается значение поля event;
- в область «Рекомендуемое действие» – записывается значение поля action.

Назначением панели свойств является предоставление пользователю системы всей доступной информации об инциденте, собранной при его первоначальном разрешении: стандартное сообщение об ошибке, степень воздействия на функционирование ИС, возмож-

ные предпосылки и условия, приведшие к его возникновению, основные признаки и возможное влияние на другие объекты, а также руководство по разрешению и ссылки на сторонние ресурсы.

Панель инцидентов (6). Предназначена для распределения инцидентов в соответствии с разработанным классификатором, основанном на модели взаимодействия открытых систем ISO/OSI; состоит из семи секций, соответствующих уровням модели.

Распределение инцидентов по уровням модели ISO/OSI доступно в представлениях, центральными элементами которых являются атомарные объекты или инциденты. Все объекты типа «Инцидент» из таких представлений дополнительно отображаются в панели инцидентов, озаглавленной «Уровни модели OSI» (расположена под строкой состояния и панелью свойств). Эта панель обладает всеми функциональными возможностями рабочей области и позволяет, в том числе, выбирать нужный объект в качестве центрального в новом представлении и просматривать его свойства.

Распределение осуществляется в соответствии со значением служебного поля «osi» структуры данных объекта типа «Инцидент».

Таким образом, графический интерфейс позволяет одновременно работать с представлениями, отражающими как иерархические (структурные) отношения, так и отношения, специфические для области обработки инцидентов, используя оба классификатора.

3.5. **Выводы**

Для реализации действующего прототипа системы приобретения и представления экспертных знаний выбран комплект средств разработки Thinkmap SDK, как наиболее полно отвечающий сформулированным архитектурным, функциональным и специфическим для области обработки инцидентов требованиям. Данный инструментарий предназначен для создания приложений, решающих задачу обработки и визуализации больших объемов данных. К основным областям его применения относятся обработка знаний, графическое представление потоков работ, ИТ ресурсов организации и топологий информационных сетей.

В соответствии с разработанными моделями представления знаний об инцидентах в ИС определены структуры данных, описывающие соответствующие объекты в среде разработки Thinkmap SDK –

узлы и дуги. Разработанные структуры данных используются для описания шести типов понятий-сущностей, четырех типов иерархических и двух типов специфических понятий-отношений, включенных в ранее предложенную таксономию объектов.

Для описания структур данных на этапе программной реализации использован язык разметки XML. Уникальность записей об объектах, созданных на их основе, гарантируется дополнительным служебным параметром – идентификатором объекта. Наличие иерархических отношений между объектами разных типов, а также использование классификатора, основанного на модели взаимодействия открытых систем ISO/OSI, позволяет задавать значения данного параметра в соответствии с нотацией ASN.1, разработанной совместно ISO и ITU-T. Использование данного подхода в монографии позволило определять тип объекта или подсистему, в которой возник описываемый инцидент, на основании его уникального идентификатора.

Все обрабатываемые и визуализируемые объекты представляются в виде элементов графовой модели. В каждый момент времени пользователю доступно определенное подмножество объектов предметной области (фрагмент графа) с возможностью перехода к новому подмножеству при необходимости. Такое подмножество, выбранное на основе значений указанных параметров, рассматривается в данной книге как представление.

В соответствии с концептуальной и математической моделями и классификаторами инцидентов система приобретения и представления экспертных знаний позволяет работать с шестью типами представлений, формируемых в зависимости от выбранного понятия-сущности, являющегося для данного представления центральным. Для построения представлений (отбора всех входящих в них объектов и связей между ними) применяются специальные условия обхода графа, реализованные с использованием программных инструментальных средств Thinkmap SDK.

Рассмотрены свойства и функциональные возможности основных компонентов графического интерфейса пользователя, используемых для работы с представлениями, навигацией и просмотром свойств объектов, такими как панели инструментов, рабочей области, строки состояния, истории навигации, панели свойств инцидентов и панели инцидентов.

ГЛАВА 4

Апробация системы приобретения и представления экспертных знаний

4.1. **Информационное наполнение базы знаний**

Апробация разработанной системы приобретения и представления экспертных знаний осуществлялась в течение нескольких лет в процессе эксплуатации аппаратно-программного комплекса Федерального центра информационно-образовательных ресурсов [53] (ФЦИОР).

Проект ФЦИОР направлен на организацию и поддержку открытого доступа к централизованному хранилищу интерактивных электронных образовательных ресурсов для общего образования и НПО/СПО и предоставление сервисов для образовательных учреждений всех уровней образования.

ИС ФЦИОР представляет собой сложный гетерогенный аппаратно-программный комплекс, включающий уровни аппаратного обеспечения, виртуализации, операционных и прикладных систем. На каждом уровне используется оборудование и программное обеспечение различных производителей, сконфигурированное и настроенное в соответствии со спецификой решаемых задач.

Существенным требованием, предъявляемым к организации процесса эксплуатации данного комплекса, является необходимость обеспечения высокой доступности предоставляемых ресурсов – общее время их недоступности для конечных пользователей, связанное с возникновением инцидентов, не должно превышать 50 минут в год; с проведением регламентных работ – не более 2 часов в год.

Разработанный и представленный в книге программный инструмент использовался для организации и наполнения базы экспертных

знаний, полученных при обработке инцидентов, возникающих в процессе эксплуатации ИС ФЦИОР.

За время опытной эксплуатации средствами системы зафиксированы и внесены в базу знаний описания 98 инцидентов, 128 связей типа «инцидент – атомарный объект», 38 связей типа «инцидент – инцидент» в соответствии со структурами данных, приведенными в третьей главе монографии. Далее представлен пример записи об инциденте в формате XML.

```
<node type="INCIDENT" id="111511" >
    <property name=«code» value=«INC_0089» />
    <property name=«message» value=«Unable to start the operating system through HMC» />
    <property name=«severity» value=«Высокая» />
    <property name=«type» value=«Серверное оборудование» />
    <property name=«event» value=«После смены IP адреса сервер под управлением ОС AIX недоступен с консоли управления (HMC)» />
    <property name="action" value="Необходимо обновить системный микрокод до SF240_299; микрокоды доступны по адресу http://www14.software.ibm.com/webapp/set2/firmware/gjsn" />
    <property name=«osi» value=«3» />
    <property name=«comment» value=«Отсутствует доступ к серверу с консоли HMC» />
</node>
```

Записи в формате XML, созданные средствами данного программного инструмента представления экспертных знаний в ходе опытной эксплуатации и представляющие собой информационное наполнение базы знаний, приведены в приложении Б.

4.2. **Построение OWL-онтологии**

Проверка целостности и непротиворечивости базы экспертных знаний после ее информационного наполнения системы приобретения и представления экспертных знаний средствами разработанного инструментария осуществлялась с использованием онтологической системы Protégé, содержащей встроенные механизмы логического вы-

вода (FaCT++ и HermIT). Для этого XML-описание разработанной ранее таксономии объектов предметной области обработки инцидентов в ИС было преобразовано в соответствии со стандартами языка описания онтологий OWL [67] (Web Ontology Language).

Стандартизированный язык OWL, относящийся к Web-ориентированным языкам спецификации онтологий, основан на предшествующих проектах OIL (Ontology Interchange Level, 2000 г.) и DAML+OIL (совместная реализация OIL с DARPA Agent Markup Language) и в настоящее время является рекомендуемым стандартом консорциума W3C. Как и RDF (Resource Description Framework), основанный на синтаксисе XML, язык OWL позволяет создавать и описывать свойства классов, отдельных экземпляров и отношений между ними, что соответствует рассмотренной во второй главе структуре онтологий.

Вместе с тем, OWL расширяет функциональность RDF, сохраняя при этом совместимость с предшествующими ему языками описания онтологий. К его дополнительным возможностям относятся создание локальных ограничений области распространения свойств и отношений; поддержка выполнения основных операций теории множеств – объединения, пересечения, дополнения; введение понятия мощности для свойств объектов [27].

Основные структурные элементы полученной средствами Protégé OWL-онтологии и их соответствие элементам таксономии объектов, рассмотренной во второй главе, представлены в табл. 8.

В процессе построения OWL-онтологии можно выделить три основных этапа. На первом определяется множество используемых классов и задается их описание, после чего в систему вносятся все объекты этих классов. Далее приведены примеры структур данных (4.1), задающих классы «Аппаратно-программный компонент» и «Объект».

```
<Declaration>
    <Class IRI="#Аппаратно-программный_компонент"/>
</Declaration>

<Declaration>
    <Class IRI=«#Объект»/>
</Declaration>                                          (4.1)
```

Табл. 8. Соответствие элементов OWL-онтологии и таксономии объектов

OWL-онтология	Таксономия объектов	Пояснение
Class	Понятие-сущность	Позволяет выделять и группировать в соответствии с определенными признаками объекты рассматриваемой предметной области
DataProperty	Понятие-свойство	Позволяет задавать значимые параметры и характеристики объектов и связей между ними
ObjectProperty	Понятие-отношение	Используется для определения иерархических или специфических для рассматриваемой предметной области бинарных связей между объектами
NamedIndividual	Объект	Экземпляр класса, соответствующий конкретному объекту предметной области
Annotation	Описание (свойство)	Элемент, позволяющий в OWL-онтологии дать необходимое пояснение к объекту или отношению; при этом, в отличие от таксономии объектов, не является его свойством

После определения всех используемых классов необходимо создать их экземпляры:

```
<Declaration>
    <NamedIndividual IRI=«#146_GB_15K_RPM_SAS»/>
</Declaration>                                          (4.2)
```

Для явного указания принадлежности объекта к классу используется тег *<ClassAssertion>* (4.3), внутри которого указываются ранее созданные сущности в последовательности «класс – экземпляр класса».

```
<ClassAssertion>
    <Class IRI=«#Объект»/>
    <NamedIndividual IRI=«#146_GB_15K_RPM_SAS»/>
</ClassAssertion>                                       (4.3)
```

На втором этапе определяется все множество свойств, которыми будут обладать объекты и связи в OWL-онтологии.

```
<Declaration>
    <DataProperty IRI=«#Производитель»/>
</Declaration>

<Declaration>
    <DataProperty IRI=«#Способ_разрешения»/>
</Declaration>                                          (4.4)
```

После определения возможных свойств осуществляется задание их конкретных значений для экземпляров классов. Как и в случае указания принадлежности объекта к классу, внутри тега *<DataPropertyAssertion>* (4.5) указываются ранее созданные сущности в последовательности «свойство – экземпляр класса – значение свойства».

```
<DataPropertyAssertion>
    <DataProperty IRI=«#Производитель»/>
    <NamedIndividual IRI=«#Catalyst_3750G»/>
    <Literal datatypeIRI=«&rdf;PlainLiteral»>Cisco</Literal>
</DataPropertyAssertion>                                (4.5)
```

Третьим этапом задается множество отношений, значимых для рассматриваемой предметной области.

```
<Declaration>
    <ObjectProperty IRI=«#влияет_на»/>
</Declaration>

<Declaration>
    <ObjectProperty IRI=«#состоит_из»/>
</Declaration>                                          (4.6)
```

После определения возможных отношений с помощью тега *<ObjectPropertyAssertion>* (4.7) выстраиваются бинарные связи между соответствующими объектами.

```
<ObjectPropertyAssertion>
    <ObjectProperty IRI=«#состоит_из»/>
```

```
        <NamedIndividual IRI=«#Flexible_Service_Processor»/>
        <NamedIndividual IRI=«#Системный_микрокод»/>
    </ObjectPropertyAssertion>                                    (4.7)
```

Тег <*AnnotationAssertion*> (4.8) используется для добавления комментария к произвольному объекту:

```
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI=«rdfs:comment»/>
        <IRI>#ASDM</IRI>
        <Literal datatypeIRI=«&rdf;PlainLiteral»>Adaptive Security
Device Manager</Literal>
    </AnnotationAssertion>                                        (4.8)
```

Графическое представление фрагмента полученной OWL-онтологии, выполненное средствами Protégé (для его построения используется подключаемый модуль OntoGraf), приведено на рис. 36. Здесь показаны экземпляры класса «подсистема ИС», отобранные для конкретной ИС, что аналогично рассмотренному в третьей главе представлению, центральным элементом которого является объект типа «информационная система организации».

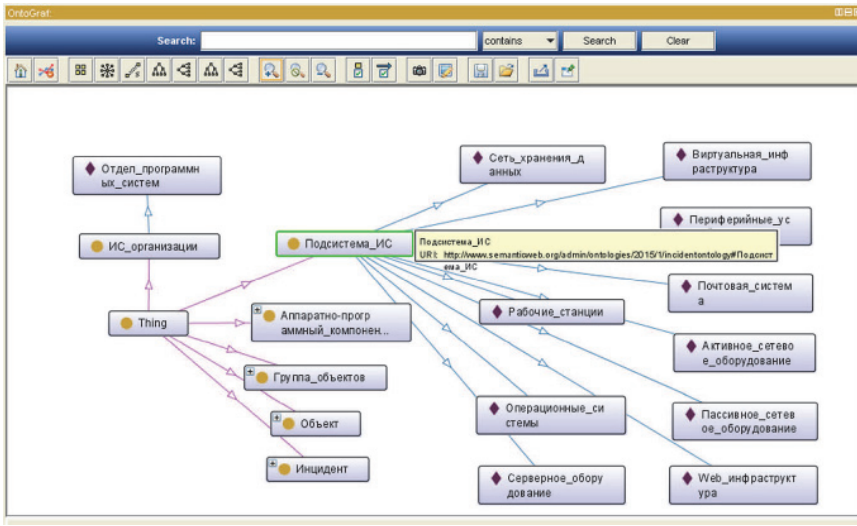

Рис. 36. Объекты класса «подсистема ИС»

4.3. **Запросы к OWL-онтологии**

Онтологическая система Protégé не предоставляет функциональных возможностей для создания самостоятельных приложений; вместе с тем, она содержит механизмы логического вывода, позволяющие выполнять проверку целостности и непротиворечивости для уже построенных онтологий.

В данной работе система Protégé применялась для проверки целостности полученной OWL-онтологии, информационное наполнение которой осуществлялось средствами разработанной системы приобретения и представления экспертных знаний. На рис. 37 представлен фрагмент полученных результатов.

Для проверки целостности построенной онтологии после ее информационного наполнения были использованы встроенные машины логического вывода FaCT++ и HermiT. Иерархия классов, автоматически вычисленная (inferred) машиной вывода, позволяет определить наличие либо отсутствие противоречий, а также наличие классов, не вошедших в иерархию, что является признаком нарушения целостности онтологии.

Как видно на рис. 37, вычисленная иерархия классов (вкладка Class hierarchy (inferred), слева) совпадает с иерархией, созданной на первом этапе построения OWL-онтологии (вкладка Class hierarchy, в центре), и содержит супер-класс Thing, включающий в себя все шесть класов онтологии. Класс Nothing, предназначенный для классов, не вошедших в иерархию, является пустым.

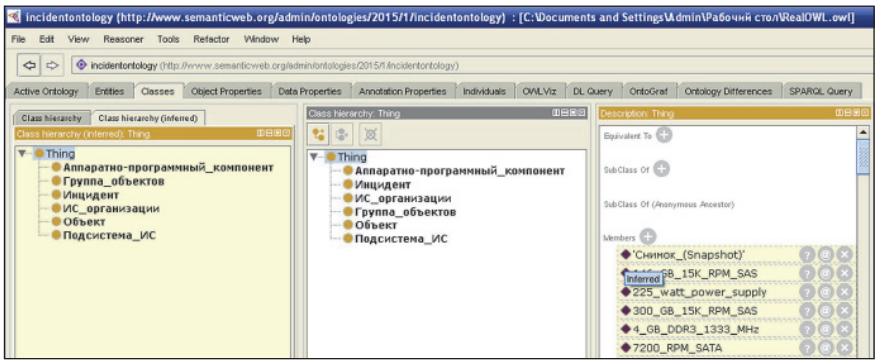

Рис. 37. Иерархия классов, вычисленная машиной вывода HermiT

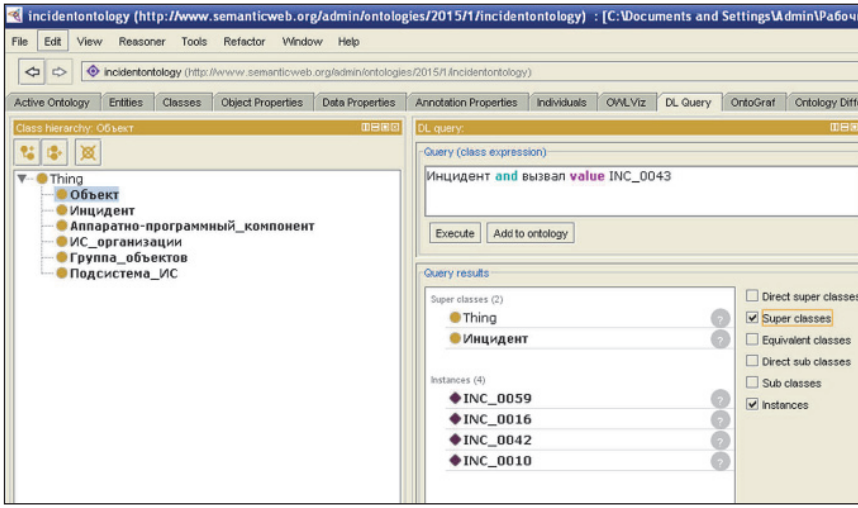

Рис. 38. Результат выполнения поискового запроса (1)

Таким образом, в онтологии отсутствуют противоречия и классы, не вошедшие в общую иерархию, что подтверждает ее целостность и непротиворечивость.

Другой функциональной возможностью среды Protégé является механизм выполнения специальных запросов, построенных в соответствии с синтаксисом OWL, к разработанной онтологии. Она позволяет не только проверять правильность структуры таксономии объектов и иерархии классов в онтологии, но и использовать ее для получения знаний о выбранной предметной области непосредственно при решении прикладных задач.

В качестве примеров запросов к разработанной онтологии можно привести следующие:

1. Какие инциденты могли привести к возникновению инцидента с символьным идентификатором INC_0043? (Рис. 38)

OWL-запрос: Инцидент and вызвал value INC_0043

Данный запрос предназначен для выбора всех экземпляров класса «Инцидент», связанных бинарным отношением типа «вызвал» с инцидентом, символьный идентификатор которого имеет значение «INC_0043» (ср. с рис. 50, приложение В).

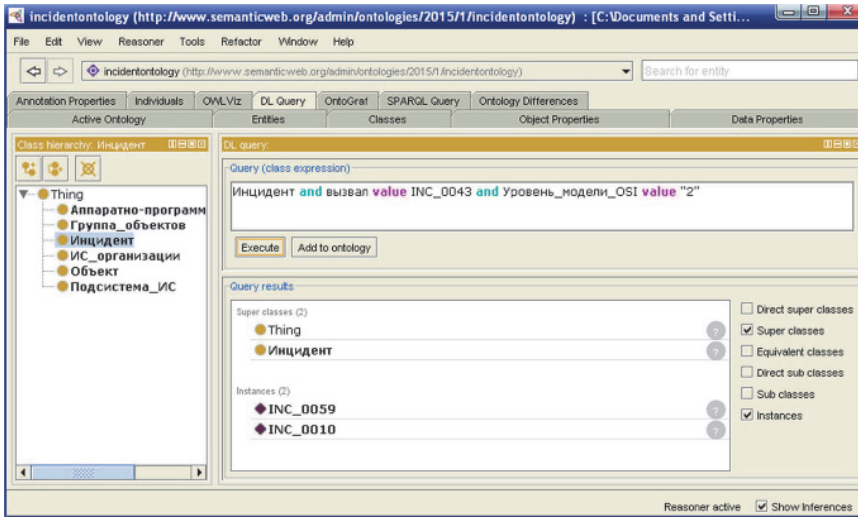

Рис. 39. Результат выполнения поискового запроса (2)

2. Какие из найденных инцидентов (см. предыдущий запрос) относятся к канальному уровню в соответствии с предложенной классификацией инцидентов, основанной на эталонной модели взаимодействия открытых систем ISO/OSI? (Рис. 39)

OWL-запрос: Инцидент and вызвал value INC_0043 and Уровень_модели_OSI value «2»

К предыдущему запросу добавляется дополнительный параметр – Уровень_модели_OSI, по указанному значению которого выполняется фильтрация среди результатов предшествующего поискового запроса (ср. с рис. 50, приложение В).

3. Какие аппаратно-программные компоненты включают в себя группу объектов «Файловая система», из каких объектов она состоит? (Рис. 40)

OWL-запрос: (Объект and inverse (состоит_из) value Файловая_система) or (Аппаратно-программный_компонент and состоит_из value Файловая_система)

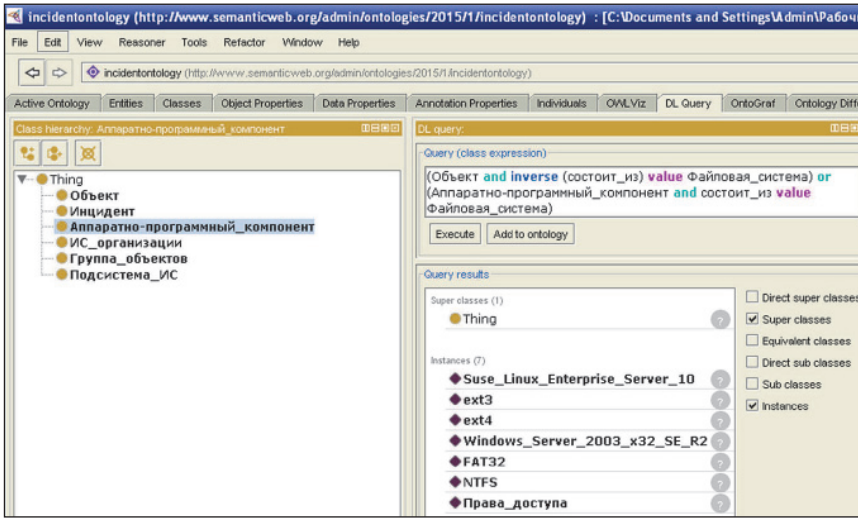

Рис. 40. Результат выполнения поискового запроса (3)

В результате выполнения данного запроса отбираются все экземпляры классов «Объект» и «Аппаратно-программный компонент», связанные бинарным отношением типа «состоит_из» с экземпляром «Файловая система» класса «Группа объектов». При этом для отбора экземпляров класса «Объект» необходимо использовать оператор inverse, так как отношение «состоит_из» не является симметричным и направлено от группы объектов к объекту (ср. с рис. 49, приложение В).

4. Какие инциденты могут оказать влияние на функционирование демона DHCP в Linux? (Рис. 41)

OWL-запрос: Инцидент and влияет_на value DHCPd

Данный запрос позволяет обнаруживать все инциденты, связанные с указанным атомарным объектом. Запросы с аналогичным синтаксисом, таким образом, позволяют в зависимости от специфики решаемых задач получать подмножество инцидентов, связанных с заданным объектом. При указании дополнительных параметров (см. пример 2) полученное подмножество может быть дополнительно ограничено для таких свойств, как тип инцидента, его важность или соответствующий уровень модели OSI.

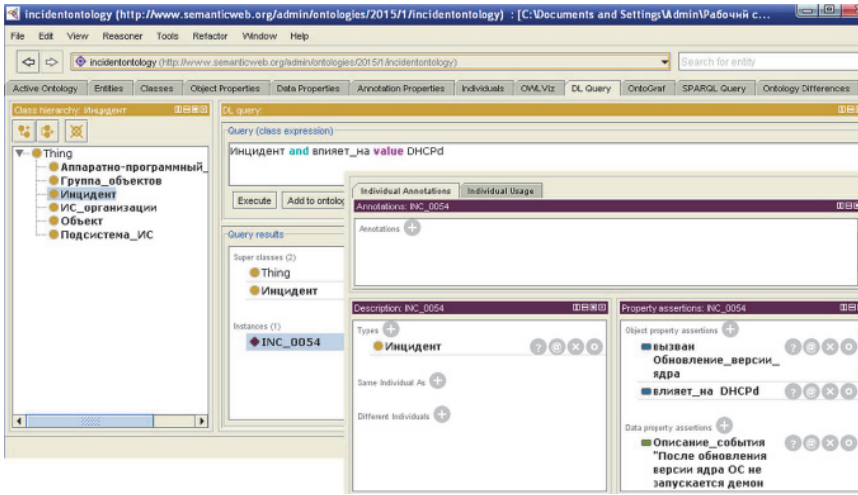

Рис. 41. Результат выполнения поискового запроса (4)

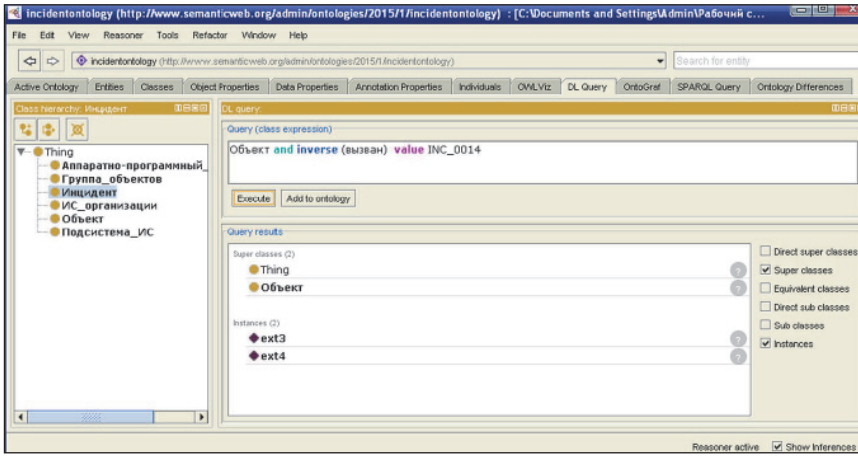

Рис. 42. Результат выполнения поискового запроса (5)

5. Какие атомарные объекты были затронуты указанным инцидентом? (Рис. 42)

OWL-запрос: Объект and inverse (вызван) value INC_0014

Подобные запросы используются для решения обратной задачи: определения всех атомарных объектов, функционирование которых

было нарушено указанным инцидентом. Это позволяет точно установить подмножество всех объектов, функционирование которых необходимо восстановить для каждого ранее зафиксированного инцидента.

Как и в случае третьего примера, для получения корректного результата необходимо использовать оператор inverse для правильной обработки несимметричного бинарного отношения «вызван».

6. Какие атомарные объекты могли стать причиной возникновения инцидента, на функционирование каких объектов это, в свою очередь, повлияло? (Рис. 43)

OWL-запрос: Объект and inverse (вызван) value INC_0020 or inverse (влияет_на) value INC_0020

Комбинация двух предыдущих запросов позволяет специалисту СТП получить полную картину возникновения инцидента и связанных с ним атомарных объектов за счет выявленных и зафиксированных ранее причинно-следственнных связей (ср. с рис. 54, приложение В).

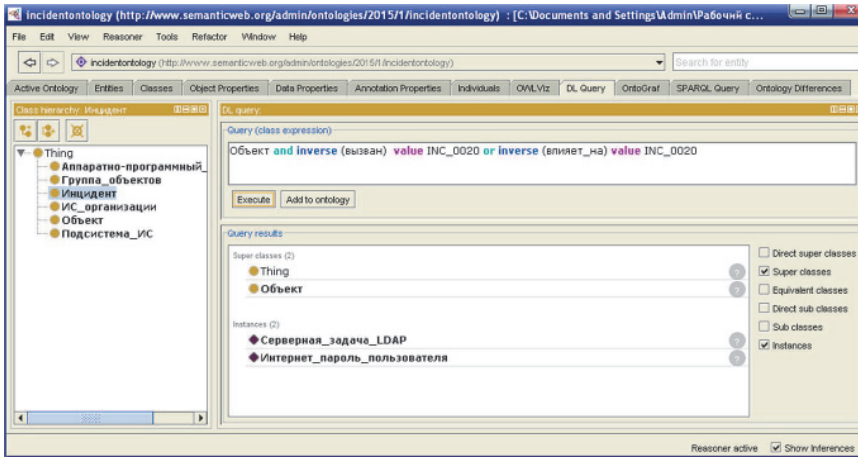

Рис. 43. Результат выполнения поискового запроса (6)

4.4. **Программа испытаний**

В монографии сформулированы следующие функциональные требования, предъявляемые к разработанному программному инструменту приобретения и представления экспертных знаний об инцидентах в ИС:

- возможность создания и редактирования записей об объектах рассматриваемой области и их свойствах, а также связях между ними, в том числе и неявных в базе экспертных знаний;
- поддержка логической организации объектов и связей между ними в соответствии с классификаторами инцидентов;
- поддержка графического представления объектов, их свойств и связей между ними;
- возможность получать подмножество инцидентов, связанных причинно-следственными связями с выбранным объектом;
- возможность определять все объекты, связанные причинно-следственными связями с заданным инцидентом;
- реализация механизма поиска по ключевым словам, названиям объектов, описаниям свойств и ключевых признаков, а также рекомендуемым действиям;
- поддержка использования русского и английского языков в интерфейсе приложения, создаваемых записях об объектах и поисковых запросах.

Для определения соответствия разработанной системы приведенным функциональным требованиям определена следующая программа испытаний:

1. **Создание записей об объектах и связях между ними в базе экспертных знаний.** Записи должны обеспечивать возможность представления экспертных знаний об объектах рассматриваемой области и их свойствах, в формате, позволяющем осуществлять их редактирование, повторное использование, обработку и визуализацию средствами системы.
2. **Просмотр созданных записей об объектах в графическом интерфейсе пользователя.** Информация об объектах и их взаимосвязях должна быть представлена в рабочей области, о свойствах объектов – дополнительно – в строке состояния и панели свойств инцидента.
3. **Просмотр представлений, формируемых в соответствии с классификаторами инцидентов.** Состав и структура подмножества

объектов, отображаемого в рабочей области в определенный момент времени, должны определяться в соответствии с классификаторами инцидентов, разработанными на основе использования структурного подхода и модели ISO/OSI.

4. **Поиск и выбор представления, содержащего подмножество инцидентов, связанных с объектом.** Представление должно содержать все объекты типа «Инцидент», зафиксированные средствами системы, связанные причинно-следственными отношениями с выбранным объектом.
5. **Поиск и выбор представления, содержащего подмножество объектов, связанных с инцидентом.** Представление должно содержать все объекты типа «Атомарный объект», зафиксированные средствами системы, связанные причинно-следственными отношениями с выбранным инцидентом.
6. **Выполнение поисковых запросов.** Система должна обеспечивать поиск объектов в базе экспертных знаний по ключевым словам, названиям, свойствам, а также по описанию действий, зафиксированных при первоначальном разрешении связанных с ними инцидентов, или их фрагментам. Инструменты навигации и поиска должны предоставлять возможность выполнения поисковых запросов с использованием русского и английского языков при наличии соответствующих описаний объектов.

Состав, порядок и методика проведения испытаний приведены в приложении В.

4.5. Результаты апробации

Основным требованием, предъявляемым к оценке эффективности наполнения базы экспертных знаний, является использование количественных показателей, позволяющих определять значения отдельных параметров и сравнивать текущие значения с показателями, зафиксированными ранее, для проведения анализа и выполнения необходимых корректирующих действий.

Для решения этой задачи вводится понятие метрик – количественных мер измерения значимых параметров функционирования ИС [5, 46].

4.5.1. **Количественная оценка показателей с использованием метрик**

К задачам, решаемым за счет использования метрик, относятся:

- согласование деятельности ИТ подразделения, включающего в себя, в том числе и СТП, со спецификой задач, решаемых информационной системой;
- количественная оценка параметров функционирования ИС, проведение на ее основе сравнительного анализа и выполнение последовательности корректирующих действий в случае необходимости.

При этом метрики являются не только средством мониторинга значений выбранных параметров для ведения статистики и составления отчетной документации, но и инструментом воздействия на ход выполнения задач – на рис. 44 показано, что метрики могут выполнять в них функцию обратной связи.

На схеме показано, что каждая задача состоит из ряда подзадач, способных возвращать значения заданных параметров при изменении состояния; эти значения могут использоваться для формирования метрик. Результатом анализа полученных таким образом значений может стать внесение изменений в состав и алгоритм выполнения подзадач.

В табл. 9 приведены примеры метрик и их описаний для задач, связанных с приобретением и последующим использованием экспертных знаний в процессе эксплуатации ИС [5, 46].

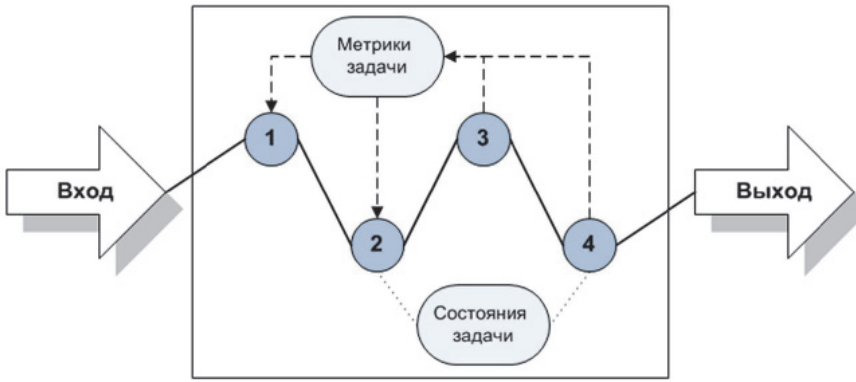

Рис. 44. Использование метрик для организации обратной связи

Табл. 9. Примеры метрик задач, связанных с использованием экспертных знаний

Метрика	**Описание**	**Возможное значение**
Задача разрешения инцидентов		
Процент инцидентов, разрешенных в течение заданного времени согласно приоритету	Определяет процент инцидентов, разрешаемых в заданный срок в соответствии с приоритетом, определенным на этапе регистрации заявки первой линией СТП	0 – 100
Процент заявок, закрытых с первого раза (на одного специалиста)	Процент заявок, для которых найдено известное решение в базе экспертных знаний после первого обращения и не требующих передачи на следующую линию СТП	0 – 100
Задача разрешения проблем		
Число решенных проблем	Число записей о закрытых проблемах, информация о которых добавлена в базу экспертных знаний	0 – 9999
Число проблем, не решенных в течение заданного времени	Проблемы, информация о которых отсутствует в базе экспертных знаний	0 – 9999

Количество метрик для оценки значимых параметров является неограниченным. Для общей оценки используется интегральный показатель, рассчитываемый с применением различных методов. Одним из них является метод расчета, использующий средневзвешенные значения метрик:

$$Z = \frac{\sum_{i=1}^{n} R_i * P_i}{n}, \qquad (4.9)$$

где Z – результирующий интегральный показатель, R_i – значение i-й метрики, P_i – ее приоритет.

Значение одних метрик может выражаться в процентах, другие могут принимать практически любые числовые значения, в результате чего использовать их непосредственно в приведенной формуле нельзя – необходимо первоначально выполнить операцию масштабирования. Под масштабированием подразумевается отображение про-

извольного значения метрики на общую для всех задач шкалу, принятую в данной организации. Так, при использовании шкалы от 1 до 10 с шагом, равным 1, значениям метрик, равным 72% и 48 при возможном значении от 1 до 70, будут соответствовать значения, равные 7.

Полученные значения R_i в приведенной формуле необходимо умножить на приоритеты соответствующих метрик, определяющие степень их значимости (вес) при расчете результирующего интегрального показателя. Значение приоритетов может динамически изменяться в ходе эксплуатации ИС в соответствии с изменением состава решаемых задач.

4.5.2. **Оценка эффективности**

В этой книге для оценки эффективности организации и наполнения базы экспертных знаний об инцидентах в ИС выбраны следующие метрики:

- процент инцидентов, разрешенных на первой линии поддержки – определяет долю инцидентов, разрешенных за счет использования экспертных знаний, зафиксированных ранее, или обходных решений;
- процент инцидентов, переданных на второй уровень поддержки – позволяет определить долю инцидентов, для разрешения которых требуется дополнительная информация;
- среднее время разрешения – позволяет оценить среднее за рассматриваемый период время от открытия заявки до разрешения инцидента;
- процент инцидентов, разрешенных с первого раза – используется для определения доли инцидентов, не требующих после своего разрешения повторного открытия или открытия новых инцидентов, связанных с ними причинно-следственными связями, описанными в базе экспертных знаний;
- процент неправильно классифицированных инцидентов – определяет часть инцидентов, неверно описанных или классифицированных при первоначальной регистрации;
- процент инцидентов, правильно связанных с объектами с первого раза – используется для оценки части инцидентов, корректно связанных с затронутыми объектами в ИС на этапе первоначальной классификации;

- процент инцидентов, разрешенных проактивно – позволяет оценить часть инцидентов, разрешенных за период времени от появления первых признаков до нарушения штатного режима функционирования ИС за счет использования экспертных знаний, зафиксированных ранее;
- число инцидентов, разрешенных с использованием известных решений – определяет абсолютное количество инцидентов, для разрешения которых использованы ранее зафиксированные экспертные знания;
- процент инцидентов, не связанных с известными проблемами – используется для определения доли инцидентов, для которых отсутствует описание в базе экспертных знаний.

В табл. 10 приведены выбранные приоритеты и абсолютные значения данных метрик, измеренные до осуществления опытной эксплуатации системы приобретения и представления экспертных знаний и через 1, 2 и 3 месяца после ее внедрения. В скобках приведены эти же значения относительно используемой для масштабирования шкалы от 0 до 10 с шагом, равным 1.

Приведенные значения метрик (по столбцам) измерены для временного интервала, равного одному календарному месяцу. Общее количество инцидентов, разрешенных за месяц до внедрения, составило 50, за последующие месяцы, соответственно, 52, 46, 48.

На рис. 45 представлена диаграмма, позволяющая сравнить значения метрик, выраженные в процентах, для соответствующих периодов наблюдения.

Значения метрик, определяющих такие положительные факторы при обработке инцидентов в ИС, как точность первоначальной классификации, правильность и скорость их разрешения с использованием базы экспертных знаний, значительно увеличиваются в первый месяц после внедрения системы. Дальнейший непрерывный рост оцениваемых показателей (при относительно постоянном количестве возникающих в ИС инцидентов) обусловлен постепенным наполнением базы знаний, позволяющей решать большее их количество, как проактивно, так и на первой линии поддержки.

Значения метрик, оценивающих негативные факторы (в табл. 10 их приоритеты заданы с использованием отрицательных значе-

Табл. 10. Приоритеты и значения используемых метрик

№	Название	Приоритет	Значение до внедрения системы	Значение после внедрения		
				через 1 месяц	через 2 месяца	через 3 месяца
1	Процент инцидентов, разрешенных на первой линии поддержки	2	44 (4)	63 (6)	67 (7)	71 (7)
2	Процент инцидентов, переданных на вторую линию поддержки	-2	56 (6)	37 (4)	33 (3)	29 (3)
3	Среднее время разрешения инцидента (мин.)	-1	155 (4)[20]	97 (3)	90 (3)	92 (3)
4	Процент инцидентов, разрешенных с первого раза	2	62 (6)	77 (8)	80 (8)	79 (8)
5	Процент неправильно классифицированных инцидентов	-3	42 (4)	31 (3)	26 (3)	23 (2)
6	Процент инцидентов, правильно связанных с объектами с первого раза	1	56 (6)	71 (7)	76 (8)	79 (8)
7	Процент инцидентов, разрешенных проактивно	3	18 (2)	29 (3)	26 (3)	31 (3)
8	Число инцидентов, разрешенных с использованием накопленных экспертных знаний	2	22 (4)[21]	36 (7)	32 (7)	33 (7)
9	Процент инцидентов, не связанных с известными проблемами	-1	42 (4)	29 (3)	28 (3)	23 (2)

ний) – неверную классификацию, невозможность определить корневую причину возникновения инцидентов – уменьшаются; диаграмма в целом при этом смещается влево.

Для выполнения расчета интегрального показателя на основе средневзвешенного значения метрик (4.9) необходимо дать следующие пояснения:

- значения приоритетов (весов) используемых метрик выбраны в диапазоне от –3 до 3 в зависимости от их значимости для обе-

[20] Для расчета относительного значения за 0 принималось минимальное время разрешения инцидента, зафиксированное в данном месяце, за 10 – максимальное.

[21] Для расчета относительного значения определялась доля соответствующих инцидентов от их общего количества за рассматриваемый месяц.

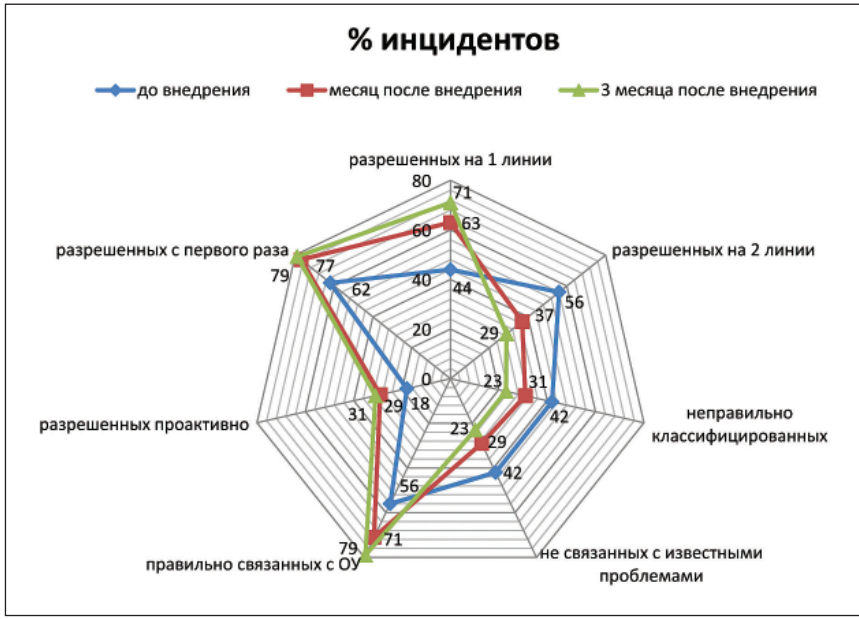

Рис. 45. Значения метрик до и после внедрения системы приобретения и представления экспертных знаний

спечения штатного режима функционирования ИС в данный момент времени; в дальнейшем, при изменении важности отдельных показателей, они также могут быть изменены;

- предельные значения для метрик, значения которых измеряются в процентах, принимаются равными 100% при положительных приоритетах и 0 при отрицательных; для времени разрешения инцидента – минимальному значению за данный месяц; для числа инцидентов, разрешенных с использованием известных решений – общему числу разрешенных за месяц инцидентов;
- в соответствии с предыдущим пунктом, максимально возможное значение интегрального показателя достигается при относительном значении всех метрик с положительными приоритетами, равном 10, и всех метрик с отрицательным приоритетом, равном нулю. Минимальное – при относительном значении всех метрик с положительным приоритетом, равном нулю, и всех метрик с отрицательным приоритетом, равном 10.

Значение до внедрения системы приобретения и представления экспертных знаний:

$$Z_{до} = \frac{2*4+(-2)*6+(-1)*4+2*6+(-3)*4+6+3*2+2*4+(-1)*4}{9} = 0{,}9.$$

Значение через месяц после внедрения:

$$Z_1 = \frac{2*6+(-2)*4+(-1)*3+2*8+(-3)*3+7+3*3+2*7+(-1)*3}{9} = 3{,}9.$$

Значение через 2 месяца после внедрения:

$$Z_2 = \frac{2*7+(-2)*3+(-1)*3+2*8+(-3)*3+8+3*3+2*7+(-1)*3}{9} = 4{,}4.$$

Значение через 3 месяца после внедрения:

$$Z_3 = \frac{2*7+(-2)*3+(-1)*3+2*8+(-3)*2+8+3*3+2*7+(-1)*2}{9} = 4{,}9.$$

На рис. 46 представлен график изменения интегрального показателя по месяцам; в качестве горизонтальных асимптот используются его минимально и максимально возможные значения.

Увеличение интегрального показателя в результате применения разработанного программного инструмента приобретения и представления экспертных знаний в ходе эксплуатации аппаратно-программного комплекса ФЦИОР позволяет сделать вывод о целесообразности использования полученных классификаторов и математической модели представления знаний об инцидентах в ИС для решения поставленных задач.

4.6. **Выводы**

Апробация разработанного программного инструмента приобретения и представления экспертных знаний осуществлена в процессе эксплуатации и технической поддержки аппаратно-программного комплекса ФЦИОР. Система использовалась для организации и наполнения базы экспертных знаний, полученных в ходе разрешения инцидентов, возникавших при эксплуатации ИС ФЦИОР, требую-

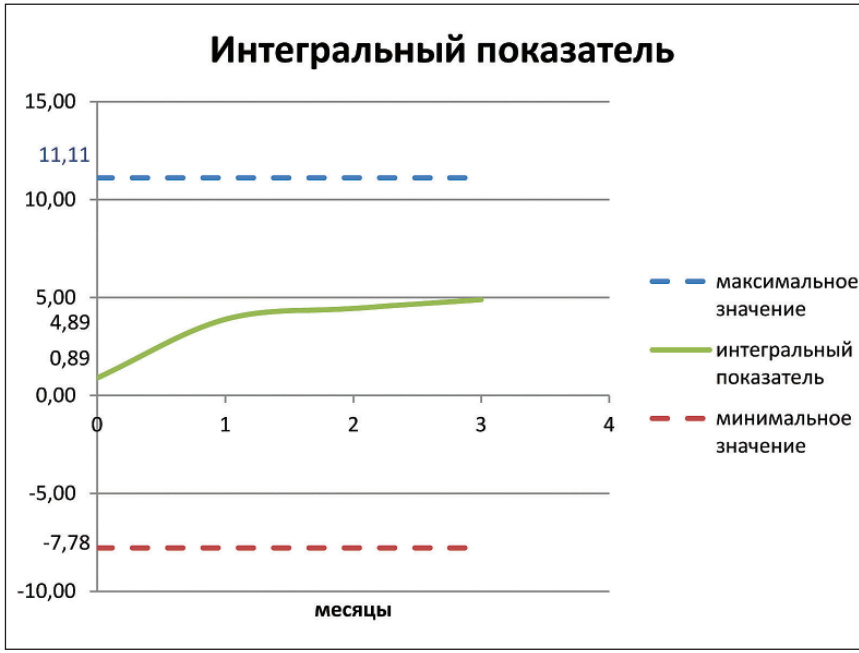

Рис. 46. График изменения интегрального показателя эффективности организации и наполнения базы экспертных знаний

щей обеспечения высокой доступности предоставляемых ресурсов[22]. За время апробации с помощью разработанных инструментальных средств зафиксированы описания 98 инцидентов; фрагменты соответствующего информационного наполнения базы знаний приведены в приложении Б.

Выполнена проверка целостности и непротиворечивости полученной базы экспертных знаний средствами онтологической системы Protégé, для чего XML-описание таксономии объектов было преобразовано в соответствии со стандартами языка описания онтологий OWL. Дано описание основных структурных элементов полученной OWL-онтологии и их соответствие элементам таксономии объектов; представлены примеры выполнения специальных запросов, построенных в соответствии с синтаксисом OWL, к разработанной онтологии.

[22] Недоступность, связанная с нарушением штатного режима функционирования, не более 50 минут, с проведением регламентных работ – не более 2 часов в год.

Для определения соответствия разработанной системы сформулированным в монографии функциональным требованиям составлена программа испытаний, включающая в себя различные сценарии создания записей об объектах рассматриваемой области и связях между ними, просмотра созданных записей и навигации в представлениях разных типов, выполнения в базе знаний поисковых запросов разной степени сложности. Состав, порядок и методика проведения испытаний приведены в приложении В.

Оценка повышения эффективности организации и наполнения базы экспертных знаний об инцидентах в ИС за счет применения разработанной системы, позволяющая определять количественные значения отдельных параметров и сравнивать их текущие значения с показателями, зафиксированными ранее, проводилась с использованием метрик – количественных мер измерения значимых параметров функционирования ИС.

Оценка эффективности проводилась на основании таких метрик как процент инцидентов, решенных на первой линии поддержки или проактивно, среднее время разрешения, число инцидентов, разрешенных с использованием базы экспертных знаний.

Для общей оценки эффективности использован интегральный показатель, рассчитываемый на основе значений перечисленных метрик и приоритетов, определяющих степень их значимости при расчете.

Увеличение интегрального показателя в результате использования базы экспертных знаний, организованной средствами разработанной системы приобретения и представления экспертных знаний в ходе эксплуатации аппаратно-программного комплекса ФЦИОР, позволяет сделать вывод о целесообразности использования предложенного подхода для решения задач данного класса.

Заключение

Целью данной книги является описание процесса разработки и использования интегрированного программного инструментария, позволяющего повысить эффективность, достоверность и скорость принятия решений по локализации и диагностике, при проявлении дефектов ИС в ходе её эксплуатации, на основе использования знаний о нарушении функциональности ИС, представленных в виде инцидентов. Программный инструментарий разработан на основании функциональной интеграции разработанной системы классификации инцидентов для формирования *баз знаний инцидентов,* информации о причинно-следственных связях для всех инцидентов, вызванных какой-либо проблемой и механизма создания и поддержания в актуальном состоянии связей между записями в базах данных инцидентов, проблем и информационно-справочных баз знаний, для чего:

1. Разработаны два независимых метода классификации и структурирования знаний об инцидентах в ИС. Первый базируется на структурном подходе, позволяющем рассматривать ИС как совокупность входящих в нее подсистем и компонентов, второй основан на эталонной модели взаимодействия открытых систем ISO/OSI. Их совместное использование позволило решить задачу формирования подмножеств инцидентов и объектов в соответствии с заданными параметрами.
2. На основе предложенных методов классификации и математической модели представления знаний об инцидентах в ИС разработана онтология обработки инцидентов в ИС. Данная онтология, являющаяся основой для организации базы экспертных знаний, позволила описать все доступное множество объектов предметной области, значимое для обработки инцидентов в ИС.

3. Разработан программный инструментарий приобретения и представления экспертных знаний об инцидентах в ИС; в качестве инструментального средства разработки выбрано ПО Thinkmap SDK.
4. В процессе эксплуатации сложного аппаратно-программного комплекса (ФЦИОР) выполнена апробация разработанной системы приобретения и представления экспертных знаний.
5. По результатам опытной эксплуатации выполнена практическая оценка повышения эффективности наполнения базы экспертных знаний об инцидентах в ИС ФЦИОР. Оценка проводилась с использованием метрик – количественных мер измерения значимых параметров функционирования ИС.

Дальнейшее развитие работ в развитие данного подхода планируется вести в *двух направлениях*.

Первое связано с формализацией выявления новых знаний из разных источников (преимущественно инструментальных средств мониторинга) на основе технологий Big Data, позволяющих автоматизировать наполнение баз экспертных знаний в виде открытых репозиториев.

Второе направление связано с определением общих принципов и практических аспектов развития онтологического инжиниринга информационных систем. Это позволит использовать существующие или разрабатывать собственные прикладные онтологии, объединенные общими понятиями и отношениями на уровне онтологии обработки инцидентов в ИС.

Список абревиатур

AD	active directory (активный каталог)
ASN.1	abstract syntax notation one (абстрактная синтаксическая нотация)
CI	configuration item (конфигурационная единица)
CMDB	configuration management database (конфигурационная база данных)
CSS	cascading style sheets (каскадные таблицы стилей)
DHCP	dynamic host configuration protocol (протокол динамической настройки узла)
DNS	domain name system (система доменных имен)
HTML	hypertext markup language (язык гипертекстовой разметки)
IMDB	incident management database (база данных управления инцидентами)
IP	internet protocol (межсетевой протокол)
ISO	international organization for standardization (международная организация по стандартизации)
ITIL	information technology infrastructure library (библиотека инфраструктуры информационных технологий)
ITSM	information technology service management (управление ИТ услугами)
ITU-T	international telecommunication union telecommunication standardization sector (сектор стандартизации электросвязи международного союза электросвязи)
JDBC	java database connectivity (интерфейс связи с базами данных для платформы Java)
JNDI	java naming and directory interface (интерфейс для служб каталогов и именования в Java)
L2TP	layer 2 tunneling protocol (протокол туннелирования второго уровня)
MOF	Microsoft operations framework (операционная среда Microsoft)
MTU	maximum transmission unit (максимально допустимый размер пакета)
NAT	network address translation (преобразование сетевых адресов)
NRPC	notes remote procedure call (вызов удаленных процедур Notes)
OSI	open systems interconnection (взаимодействие открытых систем)
OWL	web ontology language (язык описания онтологий для семантической сети)
RDF	resource description framework (среда описания ресурсов)
RDP	remote desktop protocol (протокол удаленного рабочего стола)

SDK	software development kit (комплект средств разработки ПО)
SSH	secure shell (безопасная оболочка)
SSL	secure sockets layer (уровень безопасных сокетов)
TCP	transmission control protocol (протокол управления передачей данных)
TLS	transport layer security (безопасность транспортного уровня)
XML	extensible markup language (расширяемый язык разметки гипертекста)
БД	база данных
БЗ	база знаний
ИС	информационная система
ИТ	информационная технология
ЛВС	локальная вычислительная сеть
ОС	операционная система
ПО	программное обеспечение
ППО	прикладное программное обеспечение
СКС	структурированная кабельная система
СТП	служба технической поддержки
ЦХОД	центр хранения и обработки данных

Глоссарий

интеллектуальная информационная система: Совокупность программного, аппаратного, математического и лингвистического обеспечения, предназначенного для поддержки принятия решений пользователем в определенной предметной области, информация о которой хранится в базе знаний.

информационная система: Совокупность программно-аппаратного комплекса, использующего информационные технологии для достижения поставленных целей, и необходимых для этого ресурсов – людских, временных, финансовых.

информационная технология: Модели, методы и средства сбора и обработки информации, использующиеся для получения информации нового качества.

инцидент: Любое событие, оказывающее отрицательное воздействие на штатное функционирование информационной системы, ее компонентов или предоставляемых услуг, приводящее к нарушению их функционирования.

контекст: Часть (срез) онтологии, значимый для решения рассматриваемой задачи.

конфигурационная база данных: База данных, используемая для хранения записей об объектах в составе ИС, их свойствах и взаимозависимостях на всем протяжении жизненного цикла.

конфигурационная единица: Объект, запись о котором внесена в конфигурационную базу данных.

концептуальная модель представления знаний: Описание предметной области за счет выявления и определения значимых для нее объектов (концептов), их свойств и отношений между ними.

масштабирование метрик: Отображение произвольного значения метрики на общую для всех задач шкалу, принятую в данной организации.

метрика: Количественная мера измерения заданных параметров в методологии ITSM.

нормативные знания: Знания, содержащиеся в справочной литературе и эксплуатационной документации, описывающие функционирование аппаратно-программных комплексов и компьютерных сетей в нормальных условиях без учета внешних воздействий.

обеспечивающая подсистема: Составная часть ИС, предназначенная для организации работы ее прикладных подсистем.

обработка инцидентов: Совокупность действий по выявлению возникающих инцидентов, их классификации и разрешению, в том числе с использованием ранее приобретенных экспертных знаний.

онтологическое соглашение: Объем общепринятого в данной предметной области словаря, достаточного для описания значимых объектов и отношений между ними, а также толкования предполагаемых значений слов, входящих в сам словарь.

онтология: Формальная точная система понятий заданной предметной области, описывающая набор объектов, возможных отношений между ними и их значимых свойств.

предметная область: Определенная сфера человеческой деятельности, рассматриваемая в рамках данного контекста.

представление: Подмножество объектов предметной области, отобранное системой приобретения экспертных знаний в заданный момент времени в соответствии со значениями указанных параметров.

прикладная онтология: Онтология, описывающая подмножество объектов предметной области, определяемое в соответствии со спецификой решаемой задачи.

прикладная подсистема: Составная часть ИС, предназначенная для автоматизации решения прикладных задач заданной предметной области.

проблема: Корневая причина возникновения одного или нескольких инцидентов, в большинстве случаев не описанная к моменту их возникновения.

таксономия: Иерархическая система понятий предметной области, связанных друг с другом указанными отношениями с определенной семантикой.

тезаурус: Словарь имен объектов для представления и обмена знаниями в заданной предметной области.

уникальный идентификатор объекта: Идентификатор, присваиваемый объекту и позволяющий отличать его от всех прочих объектов данного типа.

фиксация (накопление) экспертных знаний: Сбор и классификация экспертных знаний, полученных специалистами, с целью их дальнейшей организации в базы знаний.

экспертная система: Интеллектуальная информационная система, предназначенная для решения задач с помощью накопленных знаний и получения логических выводов.

экспертные знания: Знания, накопленные специалистами в определенной предметной области в процессе работы, отчужденные от них и хранимые в форме, обеспечивающей автоматизированную обработку (в т.ч. в базе знаний).

Литература

1. Башмаков, А.И. Интеллектуальные информационные технологии / А.И. Башмаков, И.А. Башмаков. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 302 с.
2. Башмаков, А.И. Информационные ресурсы для сферы образования: каталогизация, классификация, онтология / А.И. Башмаков, В.Н. Белозеров, В.А. Старых. – Информационные системы и технологии. – 2013. – №6 (80). – Орел: ФГБОУ ВПО «Государственный университет – учебно-научно-производственный комплекс», 2013. – с. 88–102.
3. Болотова, Л.С. Системы искусственного интеллекта: модели и технологии, основанные на знаниях / Л.С. Болотова. – М.: Финансы и статистика, 2012. – 664 с.
4. Болотова, Л.С. Организация многонаправленности иерархического подъема (спуска) по структуре неоднородных знаний / Л.С. Болотова, А.Н. Данчул, А.П. Новиков, А.А. Никишина. – Прикладная информатика. – 2014. – №1. – М.: ООО «Синергия Пресс», 2014. – с. 107–113.
5. Брукс, П. Метрики для управления ИТ-услугами / Питер Брукс [Peter Brooks]. – Пер. с англ. В. Первушиной. – М.: Альпина Бизнес Букс, 2008. – 283 с.
6. Ван Бон, Я. ИТ Сервис-менеджмент, введение. / Ян Ван Бон [Jan van Bon]. – Пер. с англ. М. Потоцкого. – М.: IT Expert, 2003. – 225 с.
7. Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2000. – 382 с.
8. Гаврилова, Т.А. Формирование прикладных онтологий / Т.А. Гаврилова. – Труды X национальной конференции по искусственному интеллекту, Обнинск, 25–28 сентября 2006 г. – М.: Физматлит, 2006. – Т. 2.

9. Гаврилова, Т.А. Использование онтологий в системах управления знаниями [Электронный ресурс] / Т.А. Гаврилова. – Режим доступа: http://kmsoft.ru/publications/km/big/use_ontology_in_suz.html#1.
10. Гладун, А.Я. Онтологии в корпоративных системах, часть I [Электронный ресурс] / А.Я. Гладун, Ю.В. Рогушина // Корпоративные системы. – 2006. – №1. – Режим доступа: http://www.management.com.ua/ims/ims116.html.
11. Глоссарий терминов и определений ITIL V3/ ITIL V3 Glossary / Пер. с англ. ITIL V3 Translation Project. – М.: itSMF Russia, 2009. – 146 с.
12. Глухих, И.Н. Интеллектуальные информационные системы / И.Н. Глухих. – М.: Академия, 2010. – 112 с.
13. Грибова, В.В. Управление программными средствами в интеллектуальных системах / В.В. Грибова, А.С. Клещев, Е.А. Шалфеева. – Известия Российской академии наук. Теория и системы управления. – 2010. – №6. – М.: Наука, 2010. – с. 122–137.
14. Данчул, А.Н Представление знаний о предметной области на основе гносеологической модели / А.Н. Данчул, А.П. Новиков. – Информационные технологии. – 2009. – №10. – М.: Новые технологии, 2009. – с. 44–50.
15. Джарратано, Д. Экспертные системы. Принципы разработки и программирование / Джозеф Джарратано [Joseph Giarratano]. – Пер. с англ. К. Птицына. – М.: Вильямс, 2007. – 1152 с.
16. Добров Б.В. Онтологии и тезаурусы. Модели, инструменты, приложения / Б.В. Добров, В.В. Иванов, Н.В. Лукашевич, В.Д. Соловьев. – М.: Бином. Лаборатория знаний, 2009. – 176 с.
17. Загорулько, Ю.А. Онтологии и их практическое применение в системах, основанных на знаниях / Ю.А. Загорулько, Г.Б. Загорулько // Материалы всероссийской конференции с международным участием «Знания – Онтологии – Теории» под ред. Д.Е. Пальчунова. – Новосибирск: Институт математики им. С.Л. Соболева СО РАН, 2011 г. – с. 132–141.
18. Капустин, Д.А. Информационно-вычислительные сети: учебное пособие / Д.А. Капустин, В.Е. Дементьев. – Ульяновск: УлГТУ, 2011. – 141 с.

19. Карасев, А.А. Использование свободного программного обеспечения для управления корпоративными информационными системами / А.А. Карасев. – Материалы Седьмой научно-технической конференции студентов, аспирантов и молодых специалистов Российского государственного университета инновационных технологий и предпринимательства. – М.: РГУИТП, 2010. – с. 86–88.
20. Карасев, А.А. Использование онтологического подхода в процессе управления инцидентами в современной корпоративной информационной системе / А.А. Карасев. – Труды XIX Всероссийской научно-методической конференции «Телематика 2012». – Т. 2. – СПб.: «Университетские телекоммуникации», 2012. – с. 276–277.
21. Карасев, А.А. Семиотическая система управления инцидентами в ИТ-инфраструктуре вуза / Л.С. Болотова, А.А. Карасев, С.С. Смирнов, В.А. Смольянинова. – Научные ведомости Белгородского государственного университета. Серия История, экономика, политология, информатика. – 2012. – №19 (138). – Белгород: Издательский дом «Белгород», 2012. – с. 190–196.
22. Карасев, А.А. Определение структуры инцидента в системе управления инцидентами на основе онтологического подхода и семиотического моделирования / А.А. Карасев, С.С. Смирнов, В.А. Старых. – Труды XX Всероссийской научно-методической конференции «Телематика 2013». – Т. 1. – СПб.: «Университетские телекоммуникации», 2013. – с. 93–94.
23. Карасев, А.А. Формирование инцидентов в системе управления сложной информационной системой на основе онтологического подхода и семиотического моделирования / А.А. Карасев, В.А. Старых. – Материалы X международной научной конференции «Новые информационные технологии и менеджмент качества» (NIT&QM'2013) под общ. ред. А.Н. Тихонова. – М.: ООО «Арт-Флэш», 2013. – с. 90–93.
24. Карасев, А.А. Онтологическое обеспечение процессов администрирования информационных систем / Л.С. Болотова, А.А. Карасёв, С.С. Смирнов, В.А. Старых. – Качество. Инновации. Образование. – 2013. – №12. – М.: Европейский центр по качеству, 2013. – с. 88–94.

25. Карасев, А.А. Методика построения модели предметной области управления информационными системами с использованием онтологического подхода / А.А. Карасев. – Материалы научно-технической конференции студентов, аспирантов и молодых специалистов НИУ ВШЭ. – М.: МИЭМ НИУ ВШЭ, 2014. – с. 115–116.
26. Карасев, А.А. Формализация экспертных знаний для управления инцидентами информационных систем на основе онтологического подхода / Л.С. Болотова, А.А. Карасев, В.А. Старых. – Информационные технологии. – 2014. – №6. – М.: Новые технологии, 2014. – с. 3–10.
27. Константинова, Н.С. Онтологии как системы хранения знаний / Н.С. Константинова, О.А. Митрофанова. – СПб.: Санкт-Петербургский государственный университет, 2009. – 54 с.
28. Кучкаров, З.А. Концептуальный анализ и проектирование / З.А. Кучкаров, С.П. Никаноров, Н.К. Никитина. – М.: Метасинтез, 2002. – 20 с.
29. Лапшин, В.А. Онтология в компьютерных системах / В.А. Лапшин. – М.: Научный мир, 2011. – 224 с.
30. Лукашевич, Н.В. Тезаурусы в задачах информационного поиска / Н.В. Лукашевич. – М.: Издательство МГУ, 2011. – 512 с.
31. Модель OSI [Электронный ресурс] / Режим доступа: http://citforum.ru/nets/switche/osi.shtml.
32. Найханова, Л.В. Технология создания методов автоматического построения онтологий с применением генетического и автоматного программирования: монография / Л.В. Найханова. – Улан-Удэ: Изд-во БНЦ СО РАН, 2008. – 244 с.
33. Поддержка услуг: библиотека ITIL. – М.: Компания «Ай-Теко», 2006. – 395 с.
34. Попов, Э.В. Статические и динамические экспертные системы: учебное пособие / Э.В. Попов, И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. – М.: Финансы и статистика, 1996. – 320 с.
35. Путькина, Л.В. Интеллектуальные информационные системы / Л.В. Путькина, Т.Г. Пискунова. – СПб.: Издательство СПбГУП, 2008. – 228 с.
36. Рубашкин, В.Ш. Онтологии – проблемы и решения. Точка зрения разработчика [Электронный ресурс] / В.Ш. Рубашкин. – Режим

доступа: http://www.dialog–21.ru/digests/dialog2007/materials/html/74.htm

37. Рыбина, Г.В. Некоторые аспекты разработки онтологий в обучающих интегрированных экспертных системах / Г.В. Рыбина, М.В. Юсова. – Инженерия знаний и технологии семантического веба. – 2011. – №2. – СПб.: Издательство Санкт-Петербургского НИУ информационных технологий, механики и оптики, 2011. – с. 118–126.
38. Рыбина, Г.В. Онтологии в обучающих интегрированных экспертных системах / Г.В. Рыбина, М.В. Юсова, Е.В. Чурдалев. – Информационно-измерительные и управляющие системы. – 2012. – №8. – М.: Радиотехника, 2012. – с. 13–21.
39. Смирнов, А.В. Онтологии в системах искусственного интеллекта: способы построения и организации / А.В. Смирнов, М.П. Пашкин, Н.Г. Шилов, Т.В. Левашова // Новости искусственного интеллекта. – 2002. – №1.
40. Соловьев, В.Д. Онтологии и тезаурусы: учебно-методическое пособие / В.Д. Соловьев, Б.В. Добров, В.В. Иванов, Н.В. Лукашевич. – Казань: Издательство Казанского государственного университета, 2006. – 157 с.
41. Сосинская, С.С. Представление знаний в информационной системе. Методы искусственного интеллекта и представления знаний / С.С. Сосинская. – Старый Оскол: ООО «Тонкие наукоемкие технологии», 2011. – 216 с.
42. Хакалин, Г.К. Предметная онтология для понимания текстов геометрических задач [Электронный ресурс] / Г.К. Хакалин. – Режим доступа: http://www.dialog–21.ru/digests/dialog2008/materials/html/Khakhalin.htm.
43. Харламов, А.А. Семантические сети как формальная основа решения проблемы интеграции интеллектуальных систем. Формализм автоматического формирования семантической сети с помощью преобразования в многомерное пространство / А.А. Харламов, Т.В. Ермоленко. – Труды Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS’2011). – Минск, 2011.
44. Харламов, А.А. Семантический анализ как основа для выявления дублирующих фрагментов текстов / Харламов А.А., Сергиевский Н.А. – Программная инженерия. – № 11, 2013

45. Щеглов, С.Н. Онтологический подход и его использование в системах представления знаний / С.Н. Щеглов. – Известия Южного федерального университета. Технические науки. – 2009. – №4. – Таганрог: Издательство Технологического института Южного федерального университета, 2009. – с. 146–153
46. Brooks, P. Metrics for Service Management: Designing for ITIL. / Peter Brooks. – Zaltbommel.: Van Haren Publishing, 2012. – 169 p.
47. ITIL Service Operation: ITIL Library. – London.: TSO, 2011. – 384 p.
48. ITIL Service Transition: ITIL Library. – London.: TSO, 2011. – 360 p.
49. ITIL Continual Service Improvement: ITIL Library. – London.: TSO, 2011. – 260 p.
50. IT Service Management (ITSM) Implementation Workshop Student Guide. – Houston: BMC Software Inc., 2008. – 508 p.
51. Karasev, A.A. Classification of incidents for management of information system. Innovative Information Technologies / A.A. Karasev, V.A. Starykh. – Materials of international scientific practical conference (Prague). Part 3. / Ed. Uvaysov S.U. – M.: HSE, 2014. – p. 219–225
52. Sharon, T. ITIL Small-scale Implementation. / Taylor Sharon, Macfarlane Ivor. – London.: TSO, 2007. – 102 p.
53. URL: http://fcior.edu.ru/
54. URL: http://tools.ietf.org/html/rfc3641
55. URL: http://www–03.ibm.com/software/products/ru/ru/servicerequestmanager
56. URL: http://www8.hp.com/ru/ru/software-solutions/software.html?compURI=1173779
57. URL: http://www.atkearney.com/ru/strategic-it/featured-article/-/asset_publisher/BqWAk3NLsZIU/content/it–2020-preparing-for-the-future/10192
58. URL: http://www.bmc.com/products/offering/Knowledge-Management.html
59. URL: http://www.gartner.com/newsroom/id/2603215?fnl=search
60. URL: http://www.idc.com/getdoc.jsp?containerId=244914
61. URL: http://www.itil-officialsite.com/nmsruntime/saveasdialog.aspx?lID=923
62. URL: http://www.itu.int/ITU-T/asn1/uses/index.htm
63. URL: http://www.manageengine.com/products/service-desk

64. URL: http://www.microsoft.com/ru-ru/server-cloud/system-center/service-manager.aspx
65. URL: http://www.otrs.com/ru/po/otrs-itsm/vozmozhnosti/upravlenieznanijami
66. URL: http://www.thinkmap.com
67. URL: http://www.w3.org/2004/OWL/

ПРИЛОЖЕНИЕ А

Фрагменты исходных текстов системы приобретения и представления экспертных знаний

1. **Представление знаний**

Модуль представления знаний взаимодействует с одной или несколькими базами данных, XML или flat файлами в качестве источника данных с одной стороны, и с модулем, реализующим графический интерфейс пользователя, с другой. Для выполнения модуля требуется серверная часть среды Thinkmap. Модуль использует соответствующее пространство имен (http://www.thinkmap.com/2.8/thinkmap.xml) и ряд специфических атрибутов, оформленных в виде XML-тегов.

```
<sourcemanager id=«sm» xmlns=«http://www.thinkmap.com/2.8/thinkmap.xml»
        xmlns:config=«http://www.thinkmap.com/2.8/config.xml»>

        <datasource>
                <remote />
        </datasource>

        <entitytype name=«NODE» >
                <!-- parent type for all nodes -->
                <property name=«PositionEntityFactory» >
                        <value>
                                <positionnode isFixed=«IsCenter || IsMouseDown» />
```

```
                    </value>
            </property>
    </entitytype>

    <entitytype name=«EDGE» >
            <!-- parent type for all edges -->
            <property name=«PositionEntityFactory» >
                    <value>
                            <positionedge />
                    </value>
            </property>
            <property name=«DisplayEntityFactory» >
                    <value>
                            <displayentity>
                                    <lineaspect
                                            thickness=«1»
                                            drawSinkArrow=«true»
                                            color=«blend
( #C8C8C8, #FFFFFF, LifeSpan)»
                                            isinteractive=«false» />
                            </displayentity>
                    </value>
            </property>
    </entitytype>

    <!-- define ENTERPRISE, SUBSYSTEM, COMPONENT,
OBJGROUP, OBJECT, INCIDENT -->
    <entitytype name=«ENTERPRISE» supertype=«NODE» >
            <property name=«DisplayEntityFactory» >
                    <value>
                            <displayentity>
                                    <imageaspect layer=«1»
image=«'images/ent.gif'» halign=«1» xoffset=«–10»/>
                                    <labelaspect layer=«1»
color=«blend ( #000000, #FFFFFF, LifeSpan )»
                                    label=«enterprisename»
valign=«.5» halign=«0» xoffset=«–8» />
```

```
                        </displayentity>
                    </value>
                </property>
            </entitytype>

            <entitytype name=«SUBSYSTEM» supertype=«NODE» >
                <property name=«DisplayEntityFactory» >
                    <value>
                        <displayentity>
                            <imageaspect layer=«2»
image=«'images/sub.gif'» xoffset=«5» yoffset=«–22»/>
                            <labelaspect layer=«2»
color=«blend( #0000FF, #FFFFFF, LifeSpan )»
                            label=«subname»
wrapWIDTH=«110» textAlign=«1»/>
                        </displayentity>
                    </value>
                </property>
            </entitytype>

            <entitytype name=«COMPONENT» supertype=«NODE» >
                <property name=«DisplayEntityFactory» >
                    <value>
                        <displayentity>
                            <imageaspect layer=«3»
image=«'images/com.gif'» xoffset=«10» yoffset=«–25»/>
                            <labelaspect layer=«3»
label=«vendor + ' ' + compname»
                            color=«blend(#FF0000,
#FFFFFF, LifeSpan)» wrapWIDTH=«100» textAlign=«1»/>
                        </displayentity>
                    </value>
                </property>
            </entitytype>

            <entitytype name=«OBJGROUP» supertype=«NODE» >
                <property name=«DisplayEntityFactory» >
```

```
                        <value>
                                <displayentity>
                                        <imageaspect layer=«4»
image=«’images/obj.gif’» yoffset=«–20»/>
                                        <labelaspect layer=«4»
color=«blend( #0000FF, #FFFFFF, LifeSpan )»
                                        label=«objgname»
wrapWIDTH=«125» HAlign=«.5»/>
                                </displayentity>
                        </value>
                </property>
        </entitytype>

        <entitytype name=«OBJECT» supertype=«NODE» >
                <property name=«DisplayEntityFactory» >
                        <value>
                                <displayentity>
                                        <labelaspect layer=«5»
label=«objname»
                                        color=«blend( #FF0000,
#FFFFFF, LifeSpan )» wrapWIDTH=«80» textAlign=«1»/>
                                </displayentity>
                        </value>
                </property>
        </entitytype>

        <entitytype name=«INCIDENT» supertype=«NODE» >
                <property name=«DisplayEntityFactory» >
                        <value>
                                <displayentity>
                                        <labelaspect layer=«6»
color=«blend( #0000FF, #FFFFFF, LifeSpan )»
                                        label=«code»
wrapWIDTH=«60» textAlign=«1»/>
                                </displayentity>
                        </value>
                </property>
```

```
        </entitytype>

        <!-- define Edges -->
        <entitytype name=«ENTERPRISE_SUBSYSTEM»
supertype=«EDGE» >
        </entitytype>

        <entitytype name=«SUBSYSTEM_COMPONENT»
supertype=«EDGE» >
                <property name=«DisplayEntityFactory» >
                        <value>
                                <displayentity>
                                        <lineaspect
                                        thickness=«2»
                                        drawSinkArrow=«isnative»
                                        color=«blend( #C8C8C8,
#FFFFFF, LifeSpan )» />
                                        <!-- isinteractive=«false» -->
                                </displayentity>
                        </value>
                </property>
        </entitytype>

        <entitytype name=«COMPONENT_OBJGROUP»
supertype=«EDGE» >
        </entitytype>

        <entitytype name=«OBJGROUP_OBJECT»
supertype=«EDGE» >
                <property name=«solength» value=«75» type=«Float»
/>
        </entitytype>

        <entitytype name=«OBJECT_INCIDENT»
supertype=«EDGE» >
                <property name=«solength» value=«50» type=«Float» />
                <property name=«DisplayEntityFactory» >
```

```
                <value>
                        <displayentity>
                                <lineaspect
                                drawSinkArrow=«issuedby»
                                drawSourceArrow=«affect»
                                thickness=«2»
                                color=«blend( #C8C8C8,
#FFFFFF, LifeSpan )» isinteractive=«true» />
                                <labelaspect
visible=«IsMouseOver» layer=«1» isinteractive=«false»
label=«comment» halign=«0»/>
                        </displayentity>
                </value>
        </property>
    </entitytype>

    <entitytype name=«INCIDENT_INCIDENT»
supertype=«EDGE» >
        <property name=«solength» value=«50» type=«Float» />
        <property name=«DisplayEntityFactory» >
                <value>
                        <displayentity>
                                <lineaspect
                                drawSinkArrow=«leadto»
                                drawSourceArrow=«issuedby»
                                thickness=«2»
                                color=«blend( #C8C8C8,
#FFFFFF, LifeSpan )» isinteractive=«true» />
                                <labelaspect
visible=«IsMouseOver» layer=«1» isinteractive=«false»
label=«comment» halign=«0»/>
                        </displayentity>
                </value>
        </property>
    </entitytype>

    <graphmanager id=«graphManager»>
```

```
<!-- what to display -->
<searchresults id=«searchResults»/>
<historymanager id=«historyManager» />
<terminationcondition >
        <sequence match=«all» >
                ...
        </sequence>

        <sequence match=«all» >
                ...
        </sequence>

        <sequence match=«all» >
                <!--when component centered -->
                <condition>is('COMPONENT')
</condition>
                <condition>is('EDGE')</condition>
                <condition>is('NODE')</condition>
        </sequence>

        <sequence match=«all» >
                <!-- when object group centered -->
                <condition>is('OBJGROUP')
</condition>
                <condition>is('EDGE')</condition>
                <condition>is('NODE')</condition>
        </sequence>

        <sequence match=«all» >
                <!--when object centered -->
                <condition>is('OBJECT')
</condition>
                <condition>is('EDGE')</condition>
                <condition>is('NODE')</condition>
        </sequence>
```

```
                    <sequence match=«any» >
                        ...
                    </sequence>
            </terminationcondition>
            <positionmanager id=«positionManager»
isolated=«true»>
                    <!-- where and how to display for working
space-->
                    <layouts>
                        <shell />
                    </layouts>

                    <informotions>
                        <!-- item's movement physics -->
                        <magnet />
                        <elasticity edgeLength=«if(solength,sol
ength,110)» />
                        <sphere radius=«Depth * 80» />
                        <textprotect buffer=«25» />
                        <viscousity />
                        <clamp />
                        <limit />
                    </informotions>

                    <displaymanager id=«displayManager»>
                        <!-- now displaying -->

                        <paintorder>if( is('EDGE'),0, 1 )
</paintorder>
                            <!-- nodes after edges -->
                        <dragentity />
                            <!-- to drag the whole view
alltogether -->
                        <scroll mousebutton=«LEFT_
BUTTON» />
                        <trackcenter />
                            <!-- two plugins for saying -->
```

```
                                <autoorigin triggeredOnResize=«false» />
                                        <!-- center entity is center -->
                                <centerentity id=«centerEntity» >
                                        <!-- who can be in the centre -->
                                        <isCenterable>

<![CDATA[is(‘ENTERPRISE’) || is(‘SUBSYSTEM’) ||
is(‘COMPONENT’)
                                                || is (‘OBJGROUP’) || is
(‘OBJECT’) || is (‘INCIDENT’) ]]>
                                        </isCenterable>
                                </centerentity>
                        </displaymanager>
                </positionmanager>
                <!-- next follows OSI layers positionmanagers -->

                <positionmanager id=«ISO1Position»
isolated=«true»>
                        <condition><![CDATA[ is(‘INCIDENT’) &&
osi == 1 ]]></condition>
                        <layouts>
                                <absolutelayout
Xoffset=«random(20,90)» Yoffset=«random(10,90)» />
                        </layouts>
                        <informotions>
                                <textprotect buffer=«–2»/>
                                <viscosity />
                        </informotions>
                        <displaymanager id=«ISO1Display»>
                                <dragentity />
                                <centerentity />
                        </displaymanager>
                </positionmanager>

                <positionmanager id=«ISO2Position» isolated=«true»>
                        <condition><![CDATA[ is(‘INCIDENT’) &&
osi == 2 ]]></condition>
```

```
                    <layouts>
                        <absolutelayout
Xoffset=«random(20,90)» Yoffset=«random(10,90)» />
                    </layouts>
                    <informotions>
                        <textprotect buffer=«–2»/>
                        <viscosity />
                    </informotions>
                    <displaymanager id=«ISO2Display»>
                        <dragentity />
                        <centerentity />
                    </displaymanager>
                </positionmanager>

                <positionmanager id=«ISO3Position»
isolated=«true»>
                    <condition><![CDATA[ is(‘INCIDENT’) &&
osi == 3 ]]></condition>
                    <layouts>
                        <absolutelayout
Xoffset=«random(20,90)» Yoffset=«random(10,90)» />
                    </layouts>
                    <informotions>
                        <textprotect buffer=«–2»/>
                        <viscosity />
                    </informotions>
                    <displaymanager id=«ISO3Display»>
                        <dragentity />
                        <centerentity />
                    </displaymanager>
                </positionmanager>

                <positionmanager id=«ISO4Position» isolated=«true»>
                    <condition><![CDATA[ is(‘INCIDENT’) &&
osi == 4 ]]></condition>
                    <layouts>
                        <absolutelayout
```

```
Xoffset=«random(20,90)» Yoffset=«random(10,90)» />
                        </layouts>
                        <informotions>
                                <textprotect buffer=«–2»/>
                                <viscosity />
                        </informotions>
                        <displaymanager id=«ISO4Display»>
                                <dragentity />
                                <centerentity />
                        </displaymanager>
                </positionmanager>

                <positionmanager id=«ISO5Position» isolated=«true»>
                        <condition><![CDATA[ is('INCIDENT') &&
osi == 5 ]]></condition>
                        <layouts>
                                <absolutelayout
Xoffset=«random(20,90)» Yoffset=«random(10,90)» />
                        </layouts>
                        <informotions>
                                <textprotect buffer=«–2»/>
                                <viscosity />
                        </informotions>
                        <displaymanager id=«ISO5Display»>
                                <dragentity />
                                <centerentity />
                        </displaymanager>
                </positionmanager>

                <positionmanager id=«ISO6Position» isolated=«true»>
                        <condition><![CDATA[ is('INCIDENT') &&
osi == 6 ]]></condition>
                        <layouts>
                                <absolutelayout
Xoffset=«random(20,90)» Yoffset=«random(10,90)» />
                        </layouts>
                        <informotions>
```

```
                        <textprotect buffer=«–2»/>
                        <viscosity />
                    </informotions>
                    <displaymanager id=«ISO6Display»>
                        <dragentity />
                        <centerentity />
                    </displaymanager>
                </positionmanager>

                <positionmanager id=«ISO7Position» isolated=«true»>
                    <condition><![CDATA[ is('INCIDENT') &&
osi == 7 ]]></condition>
                    <layouts>
                        <absolutelayout
Xoffset=«random(20,90)» Yoffset=«random(10,90)» />
                    </layouts>
                    <informotions>
                        <textprotect buffer=«–2»/>
                        <viscosity />
                    </informotions>
                    <displaymanager id=«ISO7Display»>
                        <dragentity />
                        <centerentity />
                    </displaymanager>
                </positionmanager>
            </graphmanager>
        </sourcemanager>
```

ПРИЛОЖЕНИЕ Б

Структуры данных базы экспертных знаний об инцидентах в ИС и фрагменты их информационного наполнения

1. **Структура данных записи об ИС организации:**

```
<nodetype name=«ENTERPRISE» >
    <property name=«enterprisename»  type=«String»   />
    <property name=«comment»  type=«String»  />
</nodetype>
```

Записи в базе знаний:

```
<node type=«ENTERPRISE» id=«1» >
    <property name=«enterprisename»  value=«ОПС» />
    <property name="comment"  value="Онтология инцидентов
для Отдела Программных Систем" />
</node>
```

2. **Структура данных записи о подсистеме ИС:**

```
<nodetype name=«SUBSYSTEM» >
    <property name=«subname»  type=«String»   />
    <property name=«comment»  type=«String»   />
</nodetype>
```

Записи в базе знаний:

<node type=«SUBSYSTEM» id=«11» >
<property name=«subname» value=«Серверное оборудование» />
<property name="comment" value="Конфигурирование и управление серверным аппаратным обеспечением" />
</node>
<node type=«SUBSYSTEM» id=«12» >
<property name="subname" value="Активное сетевое оборудование" />
<property name="comment" value="Коммутаторы, маршрутизаторы, аппаратные средства обеспечения информационной безопасности" />
</node>
<node type=«SUBSYSTEM» id=«13» >
<property name=«subname» value=«Пассивное сетевое оборудование» />
<property name=«comment» value=«Структурированная кабельная система» />
</node>
<node type=«SUBSYSTEM» id=«14» >
<property name=«subname» value=«Рабочие станции» />
<property name="comment" value="Персональные компьютеры и ноутбуки пользователей" />
</node>
<node type=«SUBSYSTEM» id=«15» >
<property name=«subname» value=«Периферийные устройства» />
<property name="comment" value="Принтеры, МФУ, мониторы, серверные консоли" />
</node>
<node type=«SUBSYSTEM» id=«16» >
<property name=«subname» value=«Сеть хранения данных» />
<property name="comment" value="Дисковые массивы, ленточные библиотеки, ПО резервного копирования" />
</node>
<node type=«SUBSYSTEM» id=«17» >

```
        <property name=«subname»  value=«Операционные системы» />
        <property name=”comment”  value=”Конфигурирование и настройка используемых ОС” />
    </node>
    <node type=«SUBSYSTEM» id=«18» >
        <property name=«subname»  value=«Виртуальная инфраструктура» />
        <property name=«comment»  value=«ПО виртуализации, гипервизоры, виртуальные машины» />
    </node>
    <node type=«SUBSYSTEM» id=«19» >
        <property name=«subname»  value=«Почтовая система» />
        <property name=”comment”  value=”Настройки почтового сервера, клиентов, маршрутизация почты” />
    </node>
    <node type=«SUBSYSTEM» id=«20» >
        <property name=«subname»  value=«Web инфраструктура» />
        <property name=«comment»  value=«Web серверы, серверы приложений, CMS» />
    </node>
```


3. **Структура связи записей об ИС и подсистемах в ее составе:**


```
    <edgetype name=«ENTERPRISE_SUBSYSTEM» from=«ENTERPRISE» to=«SUBSYSTEM» />
```


Записи в базе знаний:


```
    <edge type=”ENTERPRISE_SUBSYSTEM” fromid=”1”  toid=”11” />
    <edge type=”ENTERPRISE_SUBSYSTEM” fromid=”1”  toid=”12” />
    <edge type=”ENTERPRISE_SUBSYSTEM” fromid=”1”  toid=”13” />
    <edge type=”ENTERPRISE_SUBSYSTEM” fromid=”1”  toid=”14” />
    <edge type=”ENTERPRISE_SUBSYSTEM” fromid=”1”  toid=”15” />
    <edge type=”ENTERPRISE_SUBSYSTEM” fromid=”1”  toid=”16” />
    <edge type=”ENTERPRISE_SUBSYSTEM” fromid=”1”  toid=”17” />
    <edge type=”ENTERPRISE_SUBSYSTEM” fromid=”1”  toid=”18” />
```

```
<edge type="ENTERPRISE_SUBSYSTEM" fromid="1" toid="19" />
<edge type="ENTERPRISE_SUBSYSTEM" fromid="1" toid="20" />
```

4. **Структура данных записи об аппаратно-программном компоненте в составе подсистемы ИС:**

```
<nodetype name=«COMPONENT» >
    <property name=«compname» type=«String» />
    <property name=«vendor» type=«String» />
    <property name=«comment» type=«String» />
</nodetype>
```

Записи в базе знаний:

```
<node type=«COMPONENT» id=«111» >
    <property name=«compname» value=«p520Q» />
    <property name=«vendor» value=«IBM» />
    <property name=«comment» value=«Сервер» />
</node>
<node type=«COMPONENT» id=«112» >
    <property name=«compname» value=«HMC» />
    <property name=«vendor» value=«IBM» />
    <property name=«comment» value=«Консоль» />
</node>
<node type=«COMPONENT» id=«113» >
    <property name=«compname» value=«x3650» />
    <property name=«vendor» value=«IBM» />
    <property name=«comment» value=«Сервер» />
</node>
<node type=«COMPONENT» id=«114» >
    <property name=«compname» value=«x3850» />
    <property name=«vendor» value=«IBM» />
    <property name=«comment» value=«Сервер» />
</node>
<node type=«COMPONENT» id=«121» >
    <property name=«compname» value=«Catalyst 3750G» />
    <property name=«vendor» value=«Cisco» />
    <property name=«comment» value=«Коммутатор» />
```

```
</node>
<node type=«COMPONENT» id=«122» >
    <property name=«compname» value=«Catalyst 6509E» />
    <property name=«vendor» value=«Cisco» />
    <property name=«comment» value=«Коммутатор» />
</node>
<node type=«COMPONENT» id=«123» >
    <property name=«compname» value=«7206VXR» />
    <property name=«vendor» value=«Cisco» />
    <property name=«comment» value=«Маршрутизатор» />
</node>
<node type=«COMPONENT» id=«124» >
    <property name=«compname» value=«PIX Firewall 535» />
    <property name=«vendor» value=«Cisco» />
    <property name=«comment» value=«Межсетевой экран» />
</node>
<node type=«COMPONENT» id=«131» >
    <property name=«compname» value=«Smart Giga 24» />
    <property name=«vendor» value=«RiT» />
    <property name=«comment» value=«Кросс-панель» />
</node>
<node type=«COMPONENT» id=«132» >
    <property name=«compname» value=«Smart Giga UTP 5e» />
    <property name=«vendor» value=«RiT» />
    <property name=«comment» value=«Витая пара» />
</node>
<node type=«COMPONENT» id=«141» >
    <property name=«compname» value=«ThinkCentre A52» />
    <property name=«vendor» value=«Lenovo» />
    <property name=«comment» value=«Персональный компью-
тер» />
</node>
<node type=«COMPONENT» id=«142» >
    <property name=«compname» value=«ThinkPad T60» />
    <property name=«vendor» value=«Lenovo» />
    <property name=«comment» value=«Ноутбук» />
</node>
```

```
<node type=«COMPONENT» id=«151» >
        <property name=«compname» value=«LaserJet 4350n» />
        <property name=«vendor» value=«HP» />
        <property name=«comment» value=«Принтер» />
</node>
<node type=«COMPONENT» id=«152» >
        <property name=«compname» value=«LaserJet 3052» />
        <property name=«vendor» value=«HP» />
        <property name=«comment» value=«Многофункциональное
устройство» />
</node>
<node type=«COMPONENT» id=«161» >
        <property name=«compname» value=«TS SAN–32B–2» />
        <property name=«vendor» value=«IBM» />
        <property name=«comment» value=«Оптический коммута-
тор» />
</node>
<node type=«COMPONENT» id=«162» >
        <property name=«compname» value=«TS3500» />
        <property name=«vendor» value=«IBM» />
        <property name=”comment” value=”Ленточная библиотека” />
</node>
<node type=”COMPONENT” id=”163” >
        <property name=”compname” value=”DS6800” />
        <property name=”vendor” value=”IBM” />
        <property name=”comment” value=”Дисковый массив” />
</node>
<node type=”COMPONENT” id=”164” >
        <property name=”compname” value=”DS4700” />
        <property name=”vendor” value=”IBM” />
        <property name=”comment” value=”Дисковый массив” />
</node>
<node type=”COMPONENT” id=”171” >
        <property name=”compname” value=”Windows Server 2003
x32 SE R2” />
        <property name=”vendor” value=”Microsoft” />
        <property name=”comment” value=”Операционная система” />
```

```
    </node>
    <node type="COMPONENT" id="172" >
        <property name="compname"  value="Suse Linux Enterprise
Server 10" />
        <property name="vendor"  value="Novell" />
        <property name="comment"  value="Операционная система" />
    </node>
    <node type="COMPONENT" id="173" >
        <property name="compname"  value="IOS 15" />
        <property name="vendor"  value="Cisco" />
        <property name="comment"  value="Операционная система" />
    </node>
    <node type="COMPONENT" id="181" >
        <property name="compname"  value="ESX 3.5.0 server" />
        <property name="vendor"  value="VMware" />
        <property name="comment"  value="Серверное программное
обеспечение" />
    </node>
    <node type=«COMPONENT» id=«182» >
        <property name=«compname»  value=«VI 2.5.0 client» />
        <property name=«vendor»  value=«VMware» />
        <property name="comment"  value="Клиентское программ-
ное обеспечение" />
    </node>
    <node type=«COMPONENT» id=«191» >
        <property name=«compname»  value=«Lotus Domino 8.5.2» />
        <property name=«vendor»  value=«IBM» />
        <property name="comment"  value="Серверное программное
обеспечение" />
    </node>
    <node type=«COMPONENT» id=«192» >
        <property name=«compname»  value=«Lotus Notes 7.0.4» />
        <property name=«vendor»  value=«IBM» />
        <property name="comment"  value="Клиентское программ-
ное обеспечение" />
    </node>
    <node type=«COMPONENT» id=«193» >
```

```
        <property name=«compname» value=«Lotus Notes 8.5.2» />
        <property name=«vendor» value=«IBM» />
        <property name="comment" value="Клиентское программ-
ное обеспечение" />
    </node>
    <node type=«COMPONENT» id=«194» >
        <property name=«compname» value=«Lotus iNotes 8.5.2» />
        <property name=«vendor» value=«IBM» />
        <property name=«comment» value=«Web-клиент» />
    </node>
    <node type=«COMPONENT» id=«201» >
        <property name=«compname» value=«Apache v2.2» />
        <property name=«vendor» value=«ASF» />
        <property name=«comment» value=«Программное обеспече-
ние» />
    </node>
```


5. **Структура связи записей о подсистемах ИС с аппаратно-программными компонентами в их составе:**


```
    <edgetype name=«SUBSYSTEM_COMPONENT»
from=«SUBSYSTEM» to=«COMPONENT» >
        <property name=«isnative» type=«Boolean» />
    </edgetype>
```


Фрагменты записей в базе знаний:


```
    <edge type=«SUBSYSTEM_COMPONENT» fromid=«11»
toid=«111» >
        <property name=«isnative» value=«true» />
    </edge>
    <edge type=«SUBSYSTEM_COMPONENT» fromid=«11»
toid=«112» >
        <property name=«isnative» value=«true» />
    </edge>
    <edge type=«SUBSYSTEM_COMPONENT» fromid=«11»
toid=«113» >
        <property name=«isnative» value=«true» />
```

```
    </edge>
    <edge type=«SUBSYSTEM_COMPONENT» fromid=«11»
toid=«181» >
        <property name=«isnative»  value=«false» />
    </edge>
    <edge type=«SUBSYSTEM_COMPONENT» fromid=«12»
toid=«121» >
        <property name=«isnative»  value=«true» />
    </edge>
    <edge type=«SUBSYSTEM_COMPONENT» fromid=«12»
toid=«122» >
        <property name=«isnative»  value=«true» />
    </edge>
    <edge type=«SUBSYSTEM_COMPONENT» fromid=«12»
toid=«123» >
        <property name=«isnative»  value=«true» />
    </edge>
    <edge type=«SUBSYSTEM_COMPONENT» fromid=«16»
toid=«164» >
        <property name=«isnative»  value=«true» />
    </edge>
    <edge type=«SUBSYSTEM_COMPONENT» fromid=«16»
toid=«165» >
        <property name=«isnative»  value=«true» />
    </edge>
    <edge type=«SUBSYSTEM_COMPONENT» fromid=«17»
toid=«142» >
        <property name=«isnative»  value=«false» />
    </edge>
    <edge type=«SUBSYSTEM_COMPONENT» fromid=«17»
toid=«171» >
        <property name=«isnative»  value=«true» />
    </edge>
    <edge type=«SUBSYSTEM_COMPONENT» fromid=«17»
toid=«172» >
        <property name=«isnative»  value=«true» />
    </edge>
```

```
<edge type=«SUBSYSTEM_COMPONENT» fromid=«17»
toid=«182» >
        <property name=«isnative»  value=«false» />
</edge>
<edge type=«SUBSYSTEM_COMPONENT» fromid=«17»
toid=«191» >
        <property name=«isnative»  value=«false» />
</edge>
<edge type=«SUBSYSTEM_COMPONENT» fromid=«18»
toid=«181» >
        <property name=«isnative»  value=«true» />
</edge>
```

6. **Структура данных записи о группах объектов в составе аппаратно-программных компонентов:**

```
<nodetype name=«OBJGROUP» >
        <property name=«objgname»   type=«String»    />
        <property name=«belongsto»   type=«String»    />
</nodetype>
```

Записи в базе знаний:

```
<node type=«OBJGROUP» id=«1110» >
        <property name=«objgname»  value=«Блок питания» />
        <property name=«belongsto»  value=« » />
</node>
<node type=«OBJGROUP» id=«1111» >
        <property name=«objgname»  value=«Процессорный модуль» />
        <property name=«belongsto»  value=« » />
</node>
<node type=«OBJGROUP» id=«1112» >
        <property name=«objgname»  value=«Оперативная память» />
        <property name=«belongsto»  value=« » />
</node>
<node type=«OBJGROUP» id=«1113» >
        <property name=«objgname»  value=«Дисковая подсистема» />
        <property name=«belongsto»  value=« » />
```

```
</node>
<node type=«OBJGROUP» id=«1114» >
    <property name=«objgname» value=«Сетевой адаптер» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1115» >
    <property name=«objgname» value=«Flexible Service
Processor» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1121» >
    <property name=«objgname» value=«Упраление LPAR» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1122» >
    <property name=«objgname» value=«Управление LIC» />
    <property name=«belongsto» value=«Licensed Internal Code» />
</node>
<node type=«OBJGROUP» id=«1231» >
    <property name=«objgname» value=«Конфигурация марш-
рутизатора» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1241» >
    <property name=«objgname» value=«Конфигурация межсе-
тевого экрана» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1321» >
    <property name=«objgname» value=«Целостность» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1410» >
    <property name=«objgname» value=«Системная плата» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1411» >
```

```
        <property name=«objgname» value=«Блок питания» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJGROUP» id=«1412» >
        <property name=«objgname» value=«Оперативная память» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJGROUP» id=«1413» >
        <property name=«objgname» value=«Жесткий диск» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJGROUP» id=«1414» >
        <property name=«objgname» value=«Сетевой адаптер» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJGROUP» id=«1421» >
        <property name=«objgname» value=«Аккумуляторная бата-
рея» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJGROUP» id=«1422» >
        <property name=«objgname» value=«Беспроводной модуль» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJGROUP» id=«1521» >
        <property name=«objgname» value=«Конфигурация МФУ» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJGROUP» id=«1621» >
        <property name=«objgname» value=«Конфигурация ленточ-
ной библиотеки» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJGROUP» id=«1631» >
        <property name=«objgname» value=«Конфигурация диско-
вого массива» />
        <property name=«belongsto» value=« » />
```

```
</node>
<node type=«OBJGROUP» id=«1651» >
    <property name=«objgname» value=«NetWorker server» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1652» >
    <property name=«objgname» value=«NetWorker client» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1700» >
    <property name=«objgname» value=«Файловая система» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1702» >
    <property name=«objgname» value=«Загрузчик ОС» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1710» >
    <property name=«objgname» value=«Сетевые службы» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1711» >
    <property name=«objgname» value=«Системные службы» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1712» >
    <property name=«objgname» value=«Системные настройки» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1713» >
    <property name=«objgname» value=«Реестр» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1720» >
    <property name=«objgname» value=«Сетевые демоны» />
    <property name=«belongsto» value=« » />
</node>
```

```
<node type=«OBJGROUP» id=«1721» >
    <property name=«objgname» value=«Системные демоны» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1722» >
    <property name=«objgname» value=«Ядро» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1723» >
    <property name=«objgname» value=«Среда рабочего стола» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1812» >
    <property name=«objgname» value=«Настройки виртуаль-
ной машины» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1813» >
    <property name=«objgname» value=«Конфигурация сервера» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1821» >
    <property name=«objgname» value=«Клиентские службы» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1822» >
    <property name=«objgname» value=«Интерфейс пользователя» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1901» >
    <property name=«objgname» value=«Общие» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1902» >
    <property name=«objgname» value=«Domino Directory» />
    <property name=«belongsto» value=« » />
</node>
```

```
<node type=«OBJGROUP» id=«1910» >
    <property name=«objgname» value=«Почта» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1911» >
    <property name=«objgname» value=«Web» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1920» >
    <property name=«objgname» value=«Пользовательские на-
стройки» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1921» >
    <property name=«objgname» value=«ID пользователя» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1941» >
    <property name=«objgname» value=«iNotes Почта» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«1942» >
    <property name=«objgname» value=«iNotes Настройки» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«2021» >
    <property name=«objgname» value=«Каталог сайтов» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«2022» >
    <property name=«objgname» value=«Контроллер сайтов» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJGROUP» id=«2023» >
    <property name=«objgname» value=«Структура сайта» />
    <property name=«belongsto» value=« » />
</node>
```

7. **Структура связи записей об аппаратно-программных компонентах с группами объектов в их составе:**

<edgetype name=«COMPONENT_OBJGROUP» from=«COMPONENT» to=«OBJGROUP» />

Фрагменты записей в базе знаний:

<edge type=«COMPONENT_OBJGROUP» fromid=«111» toid=«1110» />
<edge type=«COMPONENT_OBJGROUP» fromid=«111» toid=«1111» />
<edge type=«COMPONENT_OBJGROUP» fromid=«111» toid=«1112» />
<edge type=«COMPONENT_OBJGROUP» fromid=«112» toid=«1121» />
<edge type=«COMPONENT_OBJGROUP» fromid=«112» toid=«1122» />
<edge type=«COMPONENT_OBJGROUP» fromid=«114» toid=«1110» />
<edge type=«COMPONENT_OBJGROUP» fromid=«114» toid=«1111» />
<edge type=«COMPONENT_OBJGROUP» fromid=«114» toid=«1112» />
<edge type=«COMPONENT_OBJGROUP» fromid=«114» toid=«1113» />
<edge type=«COMPONENT_OBJGROUP» fromid=«123» toid=«1231» />
<edge type=«COMPONENT_OBJGROUP» fromid=«123» toid=«1232» />
<edge type=«COMPONENT_OBJGROUP» fromid=«124» toid=«1241» />
<edge type=«COMPONENT_OBJGROUP» fromid=«124» toid=«1242» />
<edge type=«COMPONENT_OBJGROUP» fromid=«132» toid=«1321» />
<edge type=«COMPONENT_OBJGROUP» fromid=«132» toid=«1322» />

```
<edge type=«COMPONENT_OBJGROUP» fromid=«133»
toid=«1331» />
<edge type=«COMPONENT_OBJGROUP» fromid=«141»
toid=«1410» />
<edge type=«COMPONENT_OBJGROUP» fromid=«141»
toid=«1411» />
<edge type=«COMPONENT_OBJGROUP» fromid=«141»
toid=«1412» />
```

8. **Структура данных записи об атомарных объектах:**

```
<nodetype name=«OBJECT» >
    <property name=«objname»  type=«String»  />
    <property name=«belongsto»  type=«String»  />
</nodetype>
```

Записи в базе знаний:

```
<node type=«OBJECT» id=«11101» >
    <property name=«objname»  value=«MAX5 HE 675W» />
    <property name=«belongsto»  value=« » />
</node>
<node type=«OBJECT» id=«11121» >
    <property name=«objname»  value=«4 GB DDR3 1333 MHz» />
    <property name=«belongsto»  value=« » />
</node>
<node type=«OBJECT» id=«11131» >
    <property name=«objname»  value=«146 GB 15K RPM SAS» />
    <property name=«belongsto»  value=« » />
</node>
<node type=«OBJECT» id=«11132» >
    <property name=«objname»  value=«300 GB 15K RPM SAS» />
    <property name=«belongsto»  value=« » />
</node>
<node type=«OBJECT» id=«11151» >
    <property name=«objname»  value=«Системный микрокод» />
    <property name=«belongsto»  value=« » />
</node>
```

```
<node type=«OBJECT» id=«11161» >
        <property name=«objname»  value=«IMM» />
        <property name=«belongsto»  value=«Integrated Management
Module» />
</node>
<node type=«OBJECT» id=«11211» >
        <property name=«objname»  value=«Vterm» />
        <property name=«belongsto»  value=« » />
</node>
<node type=«OBJECT» id=«12321» >
        <property name=«objname»  value=«NAT» />
        <property name=«belongsto»  value=«Трансляция сетевых
адресов» />
</node>
<node type=«OBJECT» id=«12411» >
        <property name=«objname»  value=«ASDM» />
        <property name=«belongsto»  value=«Adaptive Security
Device Manager» />
</node>
<node type=«OBJECT» id=«13211» >
        <property name=«objname»  value=«Внешний экран» />
        <property name=«belongsto»  value=« » />
</node>
<node type=«OBJECT» id=«13221» >
        <property name=«objname»  value=«RJ11» />
        <property name=«belongsto»  value=« » />
</node>
<node type=«OBJECT» id=«13222» >
        <property name=«objname»  value=«RJ45» />
        <property name=«belongsto»  value=« » />
</node>
<node type=«OBJECT» id=«13311» >
        <property name=«objname»  value=«Rx» />
        <property name=«belongsto»  value=« » />
</node>
<node type=«OBJECT» id=«13312» >
        <property name=«objname»  value=«Tx» />
```

```
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«14101» >
        <property name=«objname» value=«CMOS» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«14121» >
        <property name=«objname» value=«DDR2 PC2–5300» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«14131» >
        <property name=«objname» value=«7200 RPM SATA» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«14141» >
        <property name=«objname» value=«Intel PRO/1000 PL» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«15121» >
        <property name=«objname» value=«Модуль RAM HP LJ
4350» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«15122» >
        <property name=«objname» value=«Сетевой адаптер HP LJ
4350» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«15211» >
        <property name=«objname» value=«Сканер» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«16211» >
        <property name=«objname» value=«Привод» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«16311» >
```

```
        <property name=«objname» value=«Системный контроллер» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«16312» >
        <property name=«objname» value=«Логические партиции» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«16321» >
        <property name=«objname» value=«Storage manager» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«16521» >
        <property name=«objname» value=«VSS System Boot save
set» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«17000» >
        <property name=«objname» value=«FAT32» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«17001» >
        <property name=«objname» value=«NTFS» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«17002» >
        <property name=«objname» value=«ext3» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«17003» >
        <property name=«objname» value=«ext4» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«17011» >
        <property name=«objname» value=«Intel PRO/1000 LAN
Adapter» />
        <property name=«belongsto» value=« » />
    </node>
```

```
<node type=«OBJECT» id=«17021» >
    <property name=«objname» value=«NTLDR» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«17022» >
    <property name=«objname» value=«GRUB» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«17100» >
    <property name=«objname» value=«DNS» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«17101» >
    <property name=«objname» value=«DHCP» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«17102» >
    <property name=«objname» value=«Active Directory» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«17103» >
    <property name=«objname» value=«RDP» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«17104» >
    <property name=«objname» value=«W32Time» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«17111» >
    <property name=«objname» value=«Shadow Copy Provider» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«17122» >
    <property name=«objname» value=«Маска подсети» />
    <property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«17123» >
```

```
        <property name=«objname» value=«Журнал безопасности» />
        <property name=«belongsto» value=« » />
    </node><node type=«OBJECT» id=«17200» >
        <property name=«objname» value=«Named» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«17201» >
        <property name=«objname» value=«DHCPd» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«17202» >
        <property name=«objname» value=«SSHd» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«17203» >
        <property name=«objname» value=«NTPd» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«17204» >
        <property name=«objname» value=«vsFTPd» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«17211» >
        <property name=«objname» value=«Udevd» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«18111» >
        <property name=«objname» value=«Виртуальный сетевой
адаптер» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«18112» >
        <property name=«objname» value=«Виртуальная оператив-
ная память» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«18113» >
```

```
        <property name=«objname»  value=«Виртуальный жёсткий
диск» />
        <property name=«belongsto»  value=« » />
    </node>
    <node type=«OBJECT» id=«18114» >
        <property name=«objname»  value=«Виртуальный SCSI кон-
троллер» />
        <property name=«belongsto»  value=« » />
    </node>
    <node type=«OBJECT» id=«18121» >
        <property name=«objname»  value=«VMware Tools» />
        <property name=«belongsto»  value=« » />
    </node>
    <node type=”OBJECT” id=”18123” >
        <property name=”objname”  value=”Шаблон” />
        <property name=”belongsto”  value=” “ />
    </node>
    <node type=”OBJECT” id=”18124” >
        <property name=”objname”  value=”Параметры загрузки” />
        <property name=”belongsto”  value=” “ />
    </node>
    <node type=”OBJECT” id=”18211” >
        <property name=”objname”  value=”VMware VirtualCenter
Server” />
        <property name=”belongsto”  value=” “ />
    </node>
    <node type=”OBJECT” id=”18212” >
        <property name=”objname”  value=”VMware License Server”
/>
        <property name=”belongsto”  value=” “ />
    </node>
    <node type=”OBJECT” id=”19011” >
        <property name=”objname”  value=”NServer.exe” />
        <property name=”belongsto”  value=” “ />
    </node>
    <node type=”OBJECT” id=”19012” >
        <property name=”objname”  value=”NService.exe” />
```

```
        <property name="belongsto"  value=" " />
    </node>
    <node type="OBJECT" id="19013" >
        <property name="objname"  value="ID сервера" />
        <property name="belongsto"  value=" " />
    </node>
    <node type="OBJECT" id="19015" >
        <property name="objname"  value="ACL" />
        <property name="belongsto"  value=" " />
    </node>
    <node type="OBJECT" id="19020" >
        <property name="objname"  value="Пароль пользователя" />
        <property name="belongsto"  value=" " />
    </node>
    <node type="OBJECT" id="19021" >
        <property name="objname"  value="Интернет пароль пользо-
вателя» />
        <property name=«belongsto»  value=« » />
    </node>
    <node type=«OBJECT» id=«19022» >
        <property name=«objname»  value=«Срок действия ID поль-
зователя» />
        <property name=«belongsto»  value=« » />
    </node>
    <node type=«OBJECT» id=«19101» >
        <property name=«objname»  value=«Серверная задача
Router» />
        <property name=«belongsto»  value=« » />
    </node>
    <node type=«OBJECT» id=«19102» >
        <property name=«objname»  value=«Серверная задача
SMTP» />
        <property name=«belongsto»  value=« » />
    </node>
    <node type=«OBJECT» id=«19103» >
        <property name=«objname»  value=«Серверная задача
IMAP» />
```

```
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«19104» >
        <property name=«objname» value=«Серверная задача POP3» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«19110» >
        <property name=«objname» value=«Серверная задача
HTTP» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«19111» >
        <property name=«objname» value=«Серверная задача
LDAP» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«19112» >
        <property name=«objname» value=«Directory Assistance» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«19201» >
        <property name=«objname» value=«notes.ini» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«19202» >
        <property name=«objname» value=«Address Book» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«19211» >
        <property name=«objname» value=«Корневой сертифика-
тор» />
        <property name=«belongsto» value=« » />
    </node>
    <node type=«OBJECT» id=«19312» >
        <property name=«objname» value=«Доступ к БД на сервере» />
        <property name=«belongsto» value=« » />
    </node>
```

<node type=«OBJECT» id=«19411» >
<property name=«objname» value=«Входящие сообщения» />
<property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«19412» >
<property name=«objname» value=«Исходящие сообщения» />
<property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«20233» >
<property name=«objname» value=«Модуль информационных блоков» />
<property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«20234» >
<property name=«objname» value=«Модуль AD/LDAP интеграции» />
<property name=«belongsto» value=« » />
</node>
<node type=«OBJECT» id=«20235» >
<property name=«objname» value=«Модуль опросов и голосований» />
<property name=«belongsto» value=« » />
</node>

9. **Структура связи записей о группах объектов с входящими в них атомарными объектами:**

<edgetype name=«OBJGROUP_OBJECT» from=«OBJGROUP» to=«OBJECT» />

Фрагменты записей в базе знаний:

<edge type=«OBJGROUP_OBJECT» fromid=«1110» toid=«11101» />
<edge type=«OBJGROUP_OBJECT» fromid=«1112» toid=«11121» />
<edge type=«OBJGROUP_OBJECT» fromid=«1113» toid=«11131» />

<edge type=«OBJGROUP_OBJECT» fromid=«1113» toid=«11132» />
<edge type=«OBJGROUP_OBJECT» fromid=«1115» toid=«11151» />
<edge type=«OBJGROUP_OBJECT» fromid=«1121» toid=«11211» />
<edge type=«OBJGROUP_OBJECT» fromid=«1122» toid=«11151» />
<edge type=«OBJGROUP_OBJECT» fromid=«1232» toid=«12321» />
<edge type=«OBJGROUP_OBJECT» fromid=«1241» toid=«12411» />
<edge type=«OBJGROUP_OBJECT» fromid=«1242» toid=«12421» />
<edge type=«OBJGROUP_OBJECT» fromid=«1242» toid=«12422» />
<edge type=«OBJGROUP_OBJECT» fromid=«1321» toid=«13211» />
<edge type=«OBJGROUP_OBJECT» fromid=«1321» toid=«13212» />
<edge type=«OBJGROUP_OBJECT» fromid=«1710» toid=«17104» />
<edge type=«OBJGROUP_OBJECT» fromid=«1711» toid=«17111» />
<edge type=«OBJGROUP_OBJECT» fromid=«1712» toid=«17121» />
<edge type=«OBJGROUP_OBJECT» fromid=«1712» toid=«17122» />
<edge type=«OBJGROUP_OBJECT» fromid=«1720» toid=«17200» />
<edge type=«OBJGROUP_OBJECT» fromid=«1720» toid=«17201» />
<edge type=«OBJGROUP_OBJECT» fromid=«1721» toid=«17211» />
<edge type=«OBJGROUP_OBJECT» fromid=«1722» toid=«17221» />
<edge type=«OBJGROUP_OBJECT» fromid=«1722»

```
toid=«17222» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1811»
toid=«18111» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1811»
toid=«18112» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1812»
toid=«18122» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1813»
toid=«16312» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1821»
toid=«18211» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1821»
toid=«18212» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1822»
toid=«18221» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1901»
toid=«19011» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1901»
toid=«19012» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1901»
toid=«19013» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1902»
toid=«19020» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1902»
toid=«19021» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1910»
toid=«19101» />
    <edge type=«OBJGROUP_OBJECT» fromid=«1910»
toid=«19102» />
```

10. **Структура данных записи об инциденте:**

```
<nodetype name=«INCIDENT» >
    <property name=«code»   type=«String»      />
    <property name=«message»   type=«String»   />
    <property name=«severity»   type=«String»  />
    <property name=«type»   type=«String»      />
```

```
        <property name=«event»   type=«String»        />
        <property name=«action»   type=«String»        />
        <property name=«osi» type=«Float»     />
        <property name=«comment» type=«String»     />
    </nodetype>
```


Записи в базе знаний:


```
    <node type=«INCIDENT» id=«111011» >
        <property name=«code» value=«INC_0094» />
        <property name=«message» value=«Internal power supply
fault» />
        <property name=«severity» value=«Высокая» />
        <property name=”type” value=”Серверное оборудование” />
        <property name=”event” value=”При включенном питании
на задней панели сервера горит желтый индикатор, свидетельствую-
щий о неисправности соответствующего блока питания” />
        <property name=”action” value=”При использовании резерв-
ного блока питания (или подключения по схеме N+N) замените не-
исправный блок питания не останавливая работу сервера” />
        <property name=«osi» value=«1» />
        <property name=«comment» value=«Внутренняя ошибка пи-
тания» />
    </node>
    <node type=«INCIDENT» id=«111211» >
        <property name=«code» value=«INC_0093» />
        <property name=«message» value=«Configuration error for
‘Mem Card Bank’ Major» />
        <property name=«severity» value=«Средняя» />
        <property name=”type” value=”Серверное оборудование” />
        <property name=”event” value=”После обновления IMM
до версии 1.28 в журнал событий записываются ошибки конфигу-
рации памяти. При этом ОС использует всю доступную память без
потери функциональности” />
        <property name=”action” value=”Ошибка не связана с конфи-
гурацией установленной оперативной памяти; исправлена в IMM
(Integrated Management Module) версии 1.29” />
        <property name=«osi» value=«1» />
```

```
        <property name=«comment» value=«Ошибка установки оперативной памяти» />
    </node>
    <node type=«INCIDENT» id=«111311» >
        <property name=«code» value=«INC_0090» />
        <property name=«message» value=«A new grown defect has been detected on a disk» />
        <property name=«severity» value=«Высокая» />
        <property name=”type” value=”Серверное оборудование” />
        <property name=”event” value=”Система мониторинга генерирует уведомления о постоянно увеличивающемся количестве зафиксированных на жестком диске сервера дефектов” />
        <property name=”action” value=”Размонтируйте файловые системы на диске и выполните команду fsck. В случае необходимости замените диск на аналогичный и восстановите данные из резервной копии” />
        <property name=”osi” value=”1” />
        <property name=”comment” value=”Увеличивается количество дефектов на жестком диске” />
    </node>
    <node type=«INCIDENT» id=«111321» >
        <property name=«code» value=«INC_0092» />
        <property name=«message» value=«SCSI disk error. I/O error» />
        <property name=«severity» value=«Высокая» />
        <property name=”type” value=”Серверное оборудование” />
        <property name=”event” value=”Отсутствует доступ к консоли виртуальных машин, файлы которых расположены на физическом сервере” />
        <property name=”action” value=”Перезапустите физический сервер вручную. В случае необходимости восстановите нужные файлы из резервных копий и замените жесткий диск” />
        <property name=«osi» value=«1» />
        <property name=«comment» value=«Ошибка ввода/вывода дисковой подсистемы» />
    </node>
    <node type=«INCIDENT» id=«111511» >
        <property name=«code» value=«INC_0089» />
```

```
        <property name=«message» value=«Unable to start the
operating system through HMC» />
        <property name=«severity» value=«Высокая» />
        <property name=«type» value=«Серверное оборудование» />
        <property name=«event» value=«После смены IP адреса
сервер под управлением ОС AIX недоступен с консоли управления
(HMC)» />
        <property name=”action” value=”Необходимо обновить си-
стемный микрокод до SF240_299; микрокоды доступны по адресу
http://www14.software.ibm.com/webapp/set2/firmware/gjsn” />
        <property name=«osi» value=«2» />
        <property name=«comment» value=«Отсутствует доступ
к серверу с консоли HMC» />
    </node>
    <node type=«INCIDENT» id=«112111» >
        <property name=«code» value=«INC_0091» />
        <property name=«message» value=«Virtual terminal (vterm)
inaccessible on HMC» />
        <property name=«severity» value=«Средняя» />
        <property name=”type” value=”Серверное оборудование” />
        <property name=”event” value=”Виртуальный терминал кон-
соли HMC отображает черный экран и не реагирует на нажатие кла-
виш. При этом LPAR функционирует корректно” />
        <property name=”action” value=”В консоли HMC выполните
команду mkvterm -m [machine] -p [partition] для открытия новой
сессии, а затем закройте ее (~.). После этого заново запустите вирту-
альный терминал” />
        <property name=”osi” value=”7” />
        <property name=”comment” value=”Виртуальный терминал
консоли недоступен” />
    </node>
    <node type=«INCIDENT» id=«124211» >
        <property name=«code» value=«INC_0095» />
        <property name=«message» value=«Deny inbound UDP due to
DNS Response» />
        <property name=«severity» value=«Высокая» />
        <property name=«type» value=«Сетевое оборудование» />
```

```
        <property name=«event» value=«Входящие запросы к DNS
серверу блокируются межсетевым экраном; опция DNS inspection
включена в глобальной политике» />
        <property name=”action” value=”При настройках по умол-
чанию DNS пакеты более 512 байт отбрасываются (см. RFC 1035 п.
2.3.4). Добавьте параметр inspect dns maximum-length 1024» />
        <property name=«osi» value=«3» />
        <property name=«comment» value=«DNS пакеты отбрасыва-
ются межсетевым экраном» />
    </node>
    <node type=«INCIDENT» id=«132131» >
        <property name=«code» value=«INC_0009» />
        <property name=«message» value=«Continuity test failed» />
        <property name=«severity» value=«Критическая» />
        <property name=«type» value=«Кабельная система» />
        <property name=”event” value=”Не пройден тест на целост-
ность витой пары. Возможно наличие механических повреждений
или обрывов пар проводников” />
        <property name=”action” value=”Локализуйте повреждение,
и, в случае необходимости, замените поврежденный участок” />
        <property name=«osi» value=«1» />
        <property name=«comment» value=«Нарушена целостность
витой пары» />
    </node>
    <node type=«INCIDENT» id=«141111» >
        <property name=«code» value=«INC_0073» />
        <property name=«message» value=«Power supply error» />
        <property name=«severity» value=«Средняя» />
        <property name=”type” value=”Персональный компьютер” />
        <property name=”event” value=”При нажатии кнопки вклю-
чения питания компьютер не включается. На задней панели горит
желтый светодиод” />
        <property name=”action” value=”Желтый светодиод указыва-
ет на неисправность блока питания. В случае необходимости заме-
ните блок питания на аналогичный” />
        <property name=”osi” value=”1” />
```

```
        <property name=«comment» value=«Аппаратный сбой блока питания» />
    </node>
    <node type=«INCIDENT» id=«141211» >
        <property name=«code» value=«INC_0071» />
        <property name=«message» value=«CMOS settings does not match memory on system board» />
        <property name=«severity» value=«Средняя» />
        <property name=”type” value=”Персональный компьютер” />
        <property name=”event” value=”Автотест при включении питания (POST) завершается ошибкой с кодом 164 ‘Размер системной памяти меньше ожидаемого’. ОС не загружается” />
        <property name=”action” value=”Установите дополнительный модуль оперативной памяти или примите новое значение при помощи программы Setup Utility” />
        <property name=«osi» value=«1» />
        <property name=«comment» value=«Размер системной памяти меньше ожидаемого» />
    </node>
    <node type=«INCIDENT» id=«163111» >
        <property name=«code» value=«INC_0081» />
        <property name=«message» value=«I/O is running in Single Cluster mode» />
        <property name=«severity» value=«Высокая» />
        <property name=«type» value=«Дисковые массивы» />
        <property name=”event” value=”На устройстве горит лампа ‘alarm’, ячейки массива недоступны для большинства серверов, мониторинг системы в реальном времени недоступен” />
        <property name=”action” value=”Убедитесь, что системный контроллер дискового массива функционирует корректно. В случае необходимости замените его” />
        <property name=”osi” value=”1” />
        <property name=”comment” value=”Дисковый массив недоступен на большинстве серверов” />
    </node>
    <node type=«INCIDENT» id=«162111» >
        <property name=«code» value=«INC_0085» />
```

```
        <property name=«message» value=«The drive is temporarily in service mode» />
        <property name=«severity» value=«Средняя» />
        <property name=«type» value=«Ленточные библиотеки» />
        <property name="event" value="Привод автоматически переходит в режим service mode. При этом текущие задачи копирования / восстановления продолжают выполняться, новые обрабатываются вторым приводом" />
        <property name="action" value="Убедитесь, что картридж в приводе исправен. Увеличьте значение параметра Max Consecutive Errors (число ошибок, после которого привод автоматически переходит в service mode)" />
        <property name=«osi» value=«1» />
        <property name=«comment» value=«Привод переходит в режим service mode» />
    </node>
    <node type=«INCIDENT» id=«163121» >
        <property name=«code» value=«INC_0083» />
        <property name=«message» value=«VMware ESX 3.5 do not support 2 terabyte LUNs» />
        <property name=«severity» value=«Средняя» />
        <property name=«type» value=«Дисковые массивы» />
        <property name="event" value="Сервер VMware ESX 3.5 не определяет логические партиции дискового массива объемом более 2 ТБ" />
        <property name="action" value="Серверы VMware ESX/ESXi 3.x/4.x не поддерживают партиции объемом более 2 ТБ. Измените размер используемой партиции на меньший, например – 1.99 ТБ" />
        <property name="osi" value="1" />
        <property name="comment" value="VMware ESX 3.5 не поддерживает партиции более 2 ТБ" />
    </node>
    <node type=«INCIDENT» id=«163211» >
        <property name=«code» value=«INC_0082» />
        <property name=«message» value=«Storage manager administrative account locked out» />
```

```
        <property name=«severity» value=«Средняя» />
        <property name=«type» value=«Дисковые массивы» />
        <property name="event" value="После заданного числа неудачных попыток ввести пароль, учетная запись администратора блокируется, доступ к интерфейсу управления невозможен" />
        <property name="action" value="Для ОС Windows из директории C:\Program Files\IBM\dsniserver\bin запустите утилиту сброса административного пароля securityRecoveryUtility.bat -r" />
        <property name=«osi» value=«1» />
        <property name=«comment» value=«Учетная запись администратора storage manager заблокирована» />
    </node>
    <node type=«INCIDENT» id=«165111» >
        <property name=«code» value=«INC_0087» />
        <property name=«message» value=«Probe job had unrecoverable failure(s)» />
        <property name=«severity» value=«Высокая» />
        <property name="type" value="Администрирование ОС" />
        <property name="event" value="Ошибка при выполнении резервного копирования: сервер NetWorker не может получить доступ к параметрам копирования на клиентах" />
        <property name="action" value="Убедитесь, что доменные имена клиентов успешно разрешаются на сервере в IP адреса с использованием DNS или локального файла hosts" />
        <property name="osi" value="7" />
        <property name="comment" value="Ошибка резервного копирования из-за недоступности DNS" />
    </node>
    <node type=«INCIDENT» id=«165112» >
        <property name=«code» value=«INC_0088» />
        <property name=«message» value=«Unsynchronized client clock detected» />
        <property name=«severity» value=«Высокая» />
        <property name="type" value="Администрирование ОС" />
        <property name="event" value="При наличии разницы в системном времени между сервером и клиентами более двух минут резервное копирование по расписанию не выполняется" />
```

```
        <property name="action" value="Убедитесь, что служба синхронизации системного времени на сервере и клиентах запущена и корректно функционирует" />
        <property name="osi" value="7" />
        <property name="comment" value="Системное время на сервере и клиенте не синхронизировано" />
    </node>
    <node type=«INCIDENT» id=«170011» >
        <property name=«code» value=«INC_0098» />
        <property name=«message» value=«Cannot save file» />
        <property name=«severity» value=«Средняя» />
        <property name="type" value="Администрирование ОС" />
        <property name="event" value="Невозможно сохранить файл: часть имени и расширение не сохраняется, что приводит к невозможности открыть файл в дальнейшем" />
        <property name="action" value="Убедитесь, что длина имени файла с учетом полного пути не превышает 256 символов. При необходимости используйте более короткое имя" />
        <property name="osi" value="7" />
        <property name="comment" value="Ошибка сохранения файла с заданным именем" />
    </node>
    <node type=«INCIDENT» id=«170021» >
        <property name=«code» value=«INC_0014» />
        <property name=«message» value=«Cannot touch file» />
        <property name=«severity» value=«Высокая» />
        <property name="type" value="Администрирование ОС" />
        <property name="event" value="Невозможно создать файл: файловая система смонтирована в режиме 'только чтение'" />
        <property name="action" value="Смонтируйте файловую систему в режиме 'чтение-запись'. При необходимости выполните команду 'fsck'" />
        <property name="osi" value="7" />
        <property name="comment" value="Невозможно создать файл" />
    </node>
    <node type=«INCIDENT» id=«170022» >
```

```
        <property name=«code» value=«INC_0015» />
        <property name=«message» value=«Cannot touch file» />
        <property name=«severity» value=«Высокая» />
        <property name="type" value="Администрирование ОС" />
        <property name="event" value="Невозможно создать файл:
закончилось свободное дисковое пространство" />
        <property name="action" value="Удалите неиспользуемые
файлы (журналы событий, дистрибутивы) или увеличьте размер
файловой системы" />
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Невозможно создать
файл» />
    </node>
    <node type=«INCIDENT» id=«171001» >
        <property name=«code» value=«INC_0032» />
        <property name=«message» value=«The DNS server could not
create a thread» />
        <property name=«severity» value=«Критическая» />
        <property name="type" value="Администрирование ОС" />
        <property name="event" value="Сервер DNS не отвечает
на запросы из-за утечки памяти: недостаточно системных ресурсов" />
        <property name="action" value="В качестве временного
решения необходимо перезагрузить сервер DNS. Для решения про-
блемы нужно установить обновление (http://support.microsoft.com/
kb/946565)" />
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Утечка памяти препят-
ствует работе DNS сервера» />
    </node>
    <node type=«INCIDENT» id=«171011» >
        <property name=«code» value=«INC_0036» />
        <property name=«message» value=«DHCP Database
Integrity» />
        <property name=«severity» value=«Высокая» />
        <property name="type" value="Администрирование ОС" />
        <property name="event" value="Сервер DHCP не удается
запустить из-за повреждения файла Dhcp.mdb, используемого для
```

хранения сведений об активных арендах адресов и резервирований” />
<property name=”action” value=”Восстановите базу данных \system32\dhcp\Dhcp.mdb из резервной копии \dhcp\backup и согласуйте ее с данными реестра: Scope – Active Leases – Reconcile” />
<property name=«osi» value=«7» />
<property name=«comment» value=«Поврежден файл базы данных DHCP» />
</node>
<node type=«INCIDENT» id=«171012» >
<property name=«code» value=«INC_0037» />
<property name=«message» value=«Network connection limited or no connectivity» />
<property name=«severity» value=«Средняя» />
<property name=”type” value=”Администрирование ОС” />
<property name=”event” value=”Сервер настроен на автоматическое конфигурирование стека протоколов TCP/IP, но назначенный DHCP сервер не отвечает” />
<property name=”action” value=”Восстановите связь с DHCP сервером; в случае необходимости сконфигурируйте параметры TCP/IP сервера вручную” />
<property name=”osi” value=”7” />
<property name=”comment” value=”Сетевое подключение ограничено или отсутствует” />
</node>
<node type=«INCIDENT» id=«171021» >
<property name=«code» value=«INC_0038» />
<property name=«message» value=«The remote procedure call timed out and was canceled» />
<property name=«severity» value=«Высокая» />
<property name=”type” value=”Администрирование ОС” />
<property name=”event” value=”При попытке выполнения удаленного вызова процедуры Active Directory (репликация изменений на контроллерах домена) время ожидания истекло, вызов отменен” />
<property name=«action» value=«Установите значение параметра реестра RPC Replication Timeout (HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters), равное 45 минутам» />

```
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Ошибка репликации изменений службы каталогов Active Directory» />
    </node>
    <node type=«INCIDENT» id=«171031» >
        <property name=«code» value=«INC_0034» />
        <property name=«message» value=«The terminal server has exceeded the maximum number of allowed connections» />
        <property name=«severity» value=«Средняя» />
        <property name="type" value="Администрирование ОС" />
        <property name="event" value="Ошибка подключения к удаленному рабочему столу при наличии текущих RDP сессий, как активных, так и отключенных" />
        <property name="action" value="Завершите открытые сессии на сервере назначения командой logoff ID /server:address (ID – идентификатор сессии) в консоли, в т.ч. удаленно" />
        <property name="osi" value="7" />
        <property name="comment" value="Для сервера терминалов достигнуто максимально возможное число разрешенных подключений" />
    </node>
    <node type=«INCIDENT» id=«171032» >
        <property name=«code» value=«INC_0044» />
        <property name=«message» value=«The RDP protocol component 'Data Encryption' detected an error» />
        <property name=«severity» value=«Средняя» />
        <property name="type" value="Администрирование ОС" />
        <property name="event" value="Ошибка подключения к удаленному рабочему столу при использовании клиентом и сервером несовместимых уровней шифрования" />
        <property name="action" value="Выполните команду tscc.msc и в свойствах подключения RDP-Tcp выберите уровень шифрования 'Совместимый с клиентским'" />
        <property name="osi" value="6" />
        <property name="comment" value="Компонент RDP обнаружил ошибку в потоке протокола и отключил этого клиента" />
    </node>
```

```
<node type=«INCIDENT» id=«171041» >
    <property name=«code» value=«INC_0046» />
    <property name=«message» value=«The time service has
detected that system time needs to be changed by +seconds» />
    <property name=«severity» value=«Средняя» />
    <property name=”type” value=”Администрирование ОС” />
    <property name=”event” value=”Служба времени не может
изменить системное время: разница с источником времени превы-
шает допустимую” />
    <property name=”action” value=”Проверьте правильность
установленного времени и часового пояса, в случае необходимости
установите время вручную” />
    <property name=”osi” value=”7” />
    <property name=”comment” value=”Служба времени обнару-
жила, что системное время нужно изменить на +сек.” />
</node>
<node type=«INCIDENT» id=«172011» >
    <property name=«code» value=«INC_0054» />
    <property name=«message» value=«Error occured while
restarting the DHCP daemon» />
    <property name=«severity» value=«Высокая» />
    <property name=”type” value=”Администрирование ОС” />
    <property name=”event” value=”После обновления версии
ядра ОС не запускается демон DHCP: Семейство адресов не под-
держивается протоколом” />
    <property name=”action” value=”При обновлении ядра ком-
поненты старой версии удаляются из файловой системы; для кор-
ректной работы новой версии выполните перезапуск ОС” />
    <property name=«osi» value=«7» />
    <property name=«comment» value=«Ошибка при запуске де-
мона DHCP» />
</node>
<node type=«INCIDENT» id=«172021» >
    <property name=«code» value=«INC_0035» />
    <property name=«message» value=«SSH Connection Failure» />
    <property name=«severity» value=«Средняя» />
    <property name=”type” value=”Администрирование ОС” />
```

```
        <property name="event" value="После перезапуска невозможно установить соединение с удаленным сервером по протоколу SSH" />
        <property name="action" value="Демон sshd не стартовал при запуске операционной системы. Для запуска вручную необходимо выполнить команду /etc/init.d/sshd start" />
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Удаленный сервер не доступен по SSH» />
    </node>
    <node type=«INCIDENT» id=«172031» >
        <property name=«code» value=«INC_0033» />
        <property name=«message» value=«ntpd doesn't synchronize time correctly» />
        <property name=«severity» value=«Средняя» />
        <property name="type" value="Администрирование ОС" />
        <property name="event" value="Демон ntpd не синхронизирует время с заданными NTP серверами из-за разницы, превышающей 1000 секунд" />
        <property name="action" value="Остановите ntpd (/etc/init.d/ntpd stop), установите примерное время используя команду date, запустите ntpd (/etc/init.d/ntpd start)" />
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Некорректная работа демона ntpd» />
    </node>
    <node type=«INCIDENT» id=«172221» >
        <property name=«code» value=«INC_0055» />
        <property name=«message» value=«Requested filesystem type is neither well-known nor in /proc/filesystems nor in /etc/filesystems» />
        <property name=«severity» value=«Средняя» />
        <property name=«type» value=«Администрирование ОС» />
        <property name=«event» value=«После обновления компонентов ядра ОС не монтируются внешние ntfs накопители: ошибка – неизвестная файловая система 'ntfs'» />
        <property name=«action» value=«Добавьте 'ntfs' в файл /etc/filesystems (строка могла быть удалена при обновлении ядра). Также возможно монтирование с использованием команды mount" />
```

```
        <property name=”osi” value=”7” />
        <property name=”comment” value=”Ошибка монтирования
съемных устройств” />
    </node>
    <node type=«INCIDENT» id=«181111» >
        <property name=«code» value=«INC_0010» />
        <property name=«message» value=«Virtual NIC is
misconfigured» />
        <property name=«severity» value=«Высокая» />
        <property name=”type” value=”Виртуальное оборудование” />
        <property name=”event” value=”Доступ виртуальной маши-
ны к сети отсутствует из-за неверной конфигурации сетевого адап-
тера” />
        <property name=”action” value=”Убедитесь, что в настройках
подключения виртуального сетевого адаптера указана правильная
сеть” />
        <property name=«osi» value=«2» />
        <property name=«comment» value=«Неверная конфигурация
сетевого адаптера ВМ» />
    </node>
    <node type=«INCIDENT» id=«181211» >
        <property name=«code» value=«INC_0030» />
        <property name=«message» value=«VMware Tools not
installed» />
        <property name=«severity» value=«Средняя» />
        <property name=«type» value=«Виртуальное оборудование» />
        <property name=«event» value=«Информация о гостевой
операционной системе недоступна внешним системам мониторинга
(например, IBM Tivoli Monitoring)» />
        <property name=«action» value=«Установите VMware Tools
выбрав из контекстного меню ВМ пункт Install/Upgrade VMware
Tools» />
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Не установлен набор
утилит VMware Tools» />
    </node>
    <node type=«INCIDENT» id=«182111» >
```

```
        <property name=«code» value=«INC_0042» />
        <property name=«message» value=«Service ‘VMware
VirtualCenter Server’ terminated» />
        <property name=«severity» value=«Высокая» />
        <property name=”type” value=”Администрирование ОС” />
        <property name=”event” value=”При попытке автоматическо-
го запуска службы, обеспечивающей функционирование клиентов
VMware, произошла ошибка” />
        <property name=”action” value=”Запустите службу вруч-
ную, выполнив в консоли сервера команду net start ‘VMware
VirtualCenter Server’” />
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Служба ‘VMware
VirtualCenter Server’ завершена из-за внутренней ошибки» />
    </node>
    <node type=«INCIDENT» id=«182211» >
        <property name=«code» value=«INC_0043» />
        <property name=«message» value=«VMware Infrastructure
Client connection failed» />
        <property name=«severity» value=«Средняя» />
        <property name=«type» value=«Администрирование ОС» />
        <property name=«event» value=«Клиент VMware
Infrastructure не смог установить соединение с сервером» />
        <property name=«action» value=«Убедитесь, что сетевое
соединение исправно, указан правильный сетевой адрес, сервер
VMware VirtualCenter запущен» />
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Ошибка соединения
клиента VMware Infrastructure с сервером» />
    </node>
    <node type=«INCIDENT» id=«190111» >
        <property name=«code» value=«INC_0008» />
        <property name=«message» value=«Lotus Domino server shut
down» />
        <property name=«severity» value=«Высокая» />
        <property name=«type» value=«Сервер Lotus Domino» />
        <property name=«event» value=«Процесс nserver.exe
```

(nservice.exe) не запущен» />
<property name=«action» value=«Запустите сервер Domino как приложение (nserver.exe) или службу (nservice.exe)» />
<property name=«osi» value=«7» />
<property name=«comment» value=«Не запущен сервер Lotus Domino» />
</node>
<node type=«INCIDENT» id=«190152» >
<property name=«code» value=«INC_0011» />
<property name=«message» value=«Unable to open mailbox file on remote server. No messages transferred via Notes» />
<property name=«severity» value=«Высокая» />
<property name=«type» value=«Сервер Lotus Domino» />
<property name="event" value="Почтовое сообщение не было передано по протоколу NRPC на удаленный сервер: в доступе отказано" />
<property name="action" value="Убедитесь, что на вкладке Security серверного документа запрашиваемого сервера пользователю (серверу) разрешен доступ» />
<property name=«osi» value=«7» />
<property name=«comment» value=«Пользователю (серверу) отказано в доступе к удаленному серверу" />
</node>
<node type=«INCIDENT» id=«190221» >
<property name=«code» value=«INC_0003» />
<property name=«message» value=«Your certificate has expired» />
<property name=«severity» value=«Средняя» />
<property name=«type» value=«Клиент Lotus Notes» />
<property name="event" value="Срок действия ID файла закончился. Пользователь не был аутентифицирован" />
<property name=«action» value=«Перевыпустите ID файл пользователя (Configuration – Certification – Certify) с новым сроком действия» />
<property name=«osi» value=«7» />
<property name=«comment» value=«Истек срок действия сертификата пользователя» />

```
</node>
<node type=«INCIDENT» id=«191011» >
    <property name=«code» value=«INC_0006» />
    <property name=«message» value=«Message delivery has been
delayed» />
    <property name=«severity» value=«Средняя» />
    <property name=«type» value=«Сервер Lotus Domino» />
    <property name="event" value="Одно или несколько по-
чтовых сообщений не были доставлены, так как сервер получателя
в данный момент недоступен" />
    <property name="action" value="Убедитесь, что адрес по-
лучателя указан верно, а сервер получателя запущен и доступен
по сети" />
    <property name=«osi» value=«7» />
    <property name=«comment» value=«Почтовый сервер полу-
чателя недоступен» />
</node>
<node type=«INCIDENT» id=«191012» >
    <property name=«code» value=«INC_0001» />
    <property name=«message» value=«No messages transferred to
SMTP host» />
    <property name=«severity» value=«Высокая» />
    <property name=«type» value=«Сервер Lotus Domino» />
    <property name="event" value="Невозможно соединиться
с SMTP сервером: при проверке связи не удалось обнаружить ука-
занный узел" />
    <property name="action" value="Убедитесь, что использу-
емый DNS сервер позволяет корректно разрешать указанное имя
SMTP сервера" />
    <property name=«osi» value=«7» />
    <property name=«comment» value=«Почтовый сервер полу-
чателя не найден» />
</node>
<node type=«INCIDENT» id=«191013» >
    <property name=«code» value=«INC_0026» />
    <property name=«message» value=«No route found to domain
from server» />
```

```
        <property name=«severity» value=«Высокая» />
        <property name=«type» value=«Сервер Lotus Domino» />
        <property name="event" value="Не найден маршрут для
передачи почтового сообщения в указанный домен по протоколу
NRPC" />
        <property name="action" value="Проверьте настройки доку-
ментов Server и Domain; убедитесь в наличии правильно сконфигу-
рированного документа типа Connection" />
        <property name="osi" value="7" />
        <property name="comment" value="Не найден маршрут
к указанному почтовому домену" />
    </node>
    <node type=«INCIDENT» id=«191021» >
        <property name=«code» value=«INC_0012» />
        <property name=«message» value=«Connection from SMTP
server rejected for policy reasons» />
        <property name=«severity» value=«Средняя» />
        <property name=«type» value=«Сервер Lotus Domino» />
        <property name="event" value="Запрос на соединение SMTP
сервера отклонен, передано 0 сообщений. Данное подключение за-
прещено в текущей конфигурации" />
        <property name="action" value="Измените настройки
в секции Inbound Connection Controls документа Configurations –
Router/SMTP – Restrictions and Controls" />
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Соединение с почтовым
сервером разорвано» />
    </node>
    <node type=«INCIDENT» id=«191101» >
        <property name=«code» value=«INC_0021» />
        <property name=«message» value=«The connection has timed
out» />
        <property name=«severity» value=«Высокая» />
        <property name=«type» value=«Сервер Lotus Domino» />
        <property name="event" value="Невозможно открыть ука-
занную Web страницу. Web сервер Domino может быть остановлен
или неверно сконфигурирован" />
```

```
        <property name=”action” value=”Проверьте настройки Web сервера Domino (Configuration – Web) В случае необходимости запустите задачу HTTP выполнив в консоли сервера команду load http” />
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Web сервер Domino не отвечает» />
    </node>
    <node type=«INCIDENT» id=«191121» >
        <property name=«code» value=«INC_0022» />
        <property name=«message» value=«Directory Assistance failed opening Primary Domino Directory» />
        <property name=«severity» value=«Высокая» />
        <property name=«type» value=«Сервер Lotus Domino» />
        <property name=”event” value=”База names.nsf повреждена или используется другим пользователем и не может быть открыта Directory Assistance” />
        <property name=”action” value=”Выполните команды nfixup.exe -F names.nsf ncompact.exe -c -d -i names.nsf, предварительно остановив сервер Domino” />
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Domino Directory повреждена или недоступна» />
    </node>
    <node type=«INCIDENT» id=«191111» >
        <property name=«code» value=«INC_0005» />
        <property name=«message» value=«Invalid credentials specified» />
        <property name=«severity» value=«Средняя» />
        <property name=«type» value=«Клиент Lotus Notes» />
        <property name=”event” value=”Указан пароль, не соответствующий используемому ID файлу. Пользователь не был аутентифицирован” />
        <property name=”action” value=”Убедитесь, что пароль вводится правильно, с учетом языка и регистра; соответствует указанному ID файлу” />
        <property name=«osi» value=«7» />
```

```
        <property name=«comment» value=«Указан неверный пароль» />
    </node>
    <node type=«INCIDENT» id=«191112» >
        <property name=«code» value=«INC_0020» />
        <property name=«message» value=«LDAP Server: Invalid credentials specified» />
        <property name=«severity» value=«Средняя» />
        <property name=«type» value=«Сервер Lotus Domino» />
        <property name="event" value="Указан неверный логин / пароль. Пользователь не был аутентифицирован" />
        <property name="action" value="Убедитесь, что пароль вводится правильно, с учетом языка и регистра; соответствует указанному логину" />
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Указан неверный интернет пароль» />
    </node>
    <node type=«INCIDENT» id=«192111» >
        <property name=«code» value=«INC_0023» />
        <property name=«message» value=«Your Name and Address Book does not contain a cross certificate for this organization» />
        <property name=«severity» value=«Средняя» />
        <property name=«type» value=«Клиент Lotus Notes» />
        <property name="event" value="Указанный ID файл заверен корневым сертификатором, отличным от сертификатора запрашиваемого сервера" />
        <property name="action" value="Укажите другой ID файл пользователя либо создайте кросс-сертификат в локальной адресной книге" />
        <property name="osi" value="7" />
        <property name="comment" value="Выбран ID файл, заверенный сторонним корневым сертификатором" />
    </node>
    <node type=«INCIDENT» id=«193111» >
        <property name=«code» value=«INC_0004» />
        <property name=«message» value=«Unable to find path to
```

server (Lotus Domino)» />
<property name=«severity» value=«Высокая» />
<property name=«type» value=«Сервер Lotus Domino» />
<property name="event" value="Удаленный сервер Lotus Domino больше не отвечает" />
<property name="action" value="Убедитесь, что сетевое соединение исправно, указан правильный сетевой адрес, сервер Lotus Domino запущен" />
<property name=«osi» value=«3» />
<property name=«comment» value=«Путь к серверу не найден» />
</node>
<node type=«INCIDENT» id=«194111» >
<property name=«code» value=«INC_0075» />
<property name=«message» value=«Error opening user's Inbox in Firefox» />
<property name=«severity» value=«Средняя» />
<property name=«type» value=«Клиент Lotus iNotes» />
<property name="event" value="При открытии почтовой базы пользователя в клиенте iNotes 8.5.2 (используется браузер Firefox 4 и выше) не отображаются входящие сообщения" />
<property name="action" value="Для обеспечения поддержки клиентом iNotes Web браузера Firefox 4 и выше установите обновление IBM Interim Fix 401.8HF6 для iNotes 8.5.2 Fix Pack 2" />
<property name="osi" value="7" />
<property name="comment" value="В Firefox не отображаются входящие почтовые сообщения" />
</node>
<node type=«INCIDENT» id=«194121» >
<property name=«code» value=«INC_0076» />
<property name=«message» value=«Attachment area does not load» />
<property name=«severity» value=«Средняя» />
<property name=«type» value=«Клиент Lotus iNotes» />
<property name="event" value="При создании письма (используется браузер Firefox 4 и выше) нажатие на кнопку 'Вложить файл' не позволяет переместить файлы в область 'Вложения'" />

```
        <property name="action" value="Для обеспечения клиентом iNotes корректной работы с вложениями установите обновление IBM Interim Fix 401.8HF6 для iNotes 8.5.2 Fix Pack 2" />
        <property name="osi" value="7" />
        <property name="comment" value="В Firefox не отображается поле вложений почтовых сообщений" />
    </node>
    <node type=«INCIDENT» id=«202312» >
        <property name=«code» value=«INC_0048» />
        <property name=«message» value=«Disk quota is exceeded» />
        <property name=«severity» value=«Средняя» />
        <property name="type" value="Система управления содержимым" />
        <property name="event" value="При попытке загрузить файл(ы) или сохранить отредактированную страницу появляется сообщение о превышении дисковой квоты: 'Исчерпан выделенный дисковый ресурс'" />
        <property name="action" value="Для изменения размера квоты найдите требуемый сайт в контроллере – Редактировать – Группа и выберите группу с подходящей дисковой квотой" />
        <property name=«osi» value=«7» />
        <property name=«comment» value=«Исчерпан выделенный дисковый ресурс» />
    </node>
    <node type=«INCIDENT» id=«202341» >
        <property name=«code» value=«INC_0049» />
        <property name=«message» value=«Fatal error: Class 'CLdapServer' not found» />
        <property name=«severity» value=«Высокая» />
        <property name="type" value="Система управления содержимым" />
        <property name="event" value="После ввода логина и пароля на странице авторизации панели управления появляется сообщение об ошибке авторизации" />
        <property name="action" value="Переустановите модуль: Настройки продукта – Модули – AD/LDAP интеграция – Удалить; после успешного удаления – Установить" />
```

<property name=«osi» value=«7» />
<property name=«comment» value=«Ошибка авторизации: класс ‘CLdapServer’ не найден» />
</node>

11. **Структура связи записей об инцидентах со связанными с ними объектами:**

<edgetype name=«OBJECT_INCIDENT» from=«OBJECT» to=«INCIDENT» >
<property name=«issuedby» type=«Boolean» />
<property name=«affect» type=«Boolean» />
<property name=«comment» type=«String» />
</edgetype>

Фрагменты записей в базе знаний:

<edge type=«OBJECT_INCIDENT» fromid=«11101» toid=«111011» >
<property name=«issuedby» value=«true» />
<property name=«affect» value=«false» />
<property name=«comment» value=«Вызван» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«11121» toid=«111211» >
<property name=«issuedby» value=«false» />
<property name=«affect» value=«true» />
<property name=«comment» value=«Влияет на» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«11131» toid=«111311» >
<property name=«issuedby» value=«true» />
<property name=«affect» value=«false» />
<property name=«comment» value=«Вызван» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«14141» toid=«141411» >
<property name=«issuedby» value=«false» />

```
        <property name=«affect»  value=«true» />
        <property name=«comment» value=«Влияет на» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«14211»
toid=«142111» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«15112»
toid=«151121» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«15112»
toid=«151122» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«15113»
toid=«151131» >
        <property name=«issuedby»  value=«false» />
        <property name=«affect»  value=«true» />
        <property name=«comment» value=«Влияет на» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«16521»
toid=«165211» >
        <property name=«issuedby»  value=«false» />
        <property name=«affect»  value=«true» />
        <property name=«comment» value=«Влияет на» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«17004»
toid=«172041» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
```

```
        <property name=«comment» value=«Вызван» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«17004»
toid=«202321» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«17011»
toid=«141411» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«17101»
toid=«171012» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«17102»
toid=«171021» >
        <property name=«issuedby»  value=«false» />
        <property name=«affect»  value=«true» />
        <property name=«comment» value=«Влияет на» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«17102»
toid=«171022» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«17203»
toid=«172031» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
```

```
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«18111»
toid=«172111» >
        <property name=«issuedby»  value=«false» />
        <property name=«affect»  value=«true» />
        <property name=«comment» value=«Влияет на» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«18114»
toid=«181141» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«19015»
toid=«190151» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«19015»
toid=«190152» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«19015»
toid=«193141» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
    </edge>
    <edge type=«OBJECT_INCIDENT» fromid=«19020»
toid=«191111» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
    </edge>
```

```
<edge type=«OBJECT_INCIDENT» fromid=«19021»
toid=«191112» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«19104»
toid=«191041» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«19105»
toid=«190152» >
        <property name=«issuedby»  value=«false» />
        <property name=«affect»  value=«true» />
        <property name=«comment» value=«Влияет на» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«19110»
toid=«191101» >
        <property name=«issuedby»  value=«true» />
        <property name=«affect»  value=«false» />
        <property name=«comment» value=«Вызван» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«19111»
toid=«191112» >
        <property name=«issuedby»  value=«false» />
        <property name=«affect»  value=«true» />
        <property name=«comment» value=«Влияет на» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«19312»
toid=«190152» >
        <property name=«issuedby»  value=«false» />
        <property name=«affect»  value=«true» />
        <property name=«comment» value=«Влияет на» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«20231»
```

toid=«202312» >
<property name=«issuedby» value=«false» />
<property name=«affect» value=«true» />
<property name=«comment» value=«Влияет на» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«20231» toid=«202313» >
<property name=«issuedby» value=«false» />
<property name=«affect» value=«true» />
<property name=«comment» value=«Влияет на» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«20232» toid=«202312» >
<property name=«issuedby» value=«false» />
<property name=«affect» value=«true» />
<property name=«comment» value=«Влияет на» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«20232» toid=«202313» >
<property name=«issuedby» value=«false» />
<property name=«affect» value=«true» />
<property name=«comment» value=«Влияет на» />
</edge>
<edge type=«OBJECT_INCIDENT» fromid=«20232» toid=«202321» >
<property name=«issuedby» value=«false» />
<property name=«affect» value=«true» />
<property name=«comment» value=«Влияет на» />
</edge>

12. **Структура причинно-следственной связи записей об инцидентах:**

<edgetype name=«INCIDENT_INCIDENT» from=«INCIDENT» to=«INCIDENT» >
<property name=«issuedby» type=«Boolean» />
<property name=«leadto» type=«Boolean» />

```
        <property name=«comment» type=«String» />
    </edgetype>
```


Фрагменты записей в базе знаний:


```
    <edge type=«INCIDENT_INCIDENT» fromid=«124211»
toid=«165111» >
        <property name=«issuedby»  value=«false» />
        <property name=«leadto»  value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«124211»
toid=«191012» >
        <property name=«issuedby»  value=«false» />
        <property name=«leadto»  value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«132131»
toid=«191011» >
        <property name=«issuedby»  value=«false» />
        <property name=«leadto»  value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«132131»
toid=«193111» >
        <property name=«issuedby»  value=«false» />
        <property name=«leadto»  value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«132131»
toid=«151221» >
        <property name=«issuedby»  value=«false» />
        <property name=«leadto»  value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«133111»
toid=«171021» >
        <property name=«issuedby»  value=«false» />
```

```
        <property name=«leadto» value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«133111»
toid=«191011» >
        <property name=«issuedby» value=«false» />
        <property name=«leadto» value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«163121»
toid=«181141» >
        <property name=«issuedby» value=«false» />
        <property name=«leadto» value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«170021»
toid=«202311» >
        <property name=«issuedby» value=«false» />
        <property name=«leadto» value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«170022»
toid=«202311» >
        <property name=«issuedby» value=«false» />
        <property name=«leadto» value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«171021»
toid=«171022» >
        <property name=«issuedby» value=«false» />
        <property name=«leadto» value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«172031»
toid=«202331» >
        <property name=«issuedby» value=«false» />
        <property name=«leadto» value=«true» />
```

```
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«181111»
toid=«151221» >
        <property name=«issuedby»  value=«false» />
        <property name=«leadto»  value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«181111»
toid=«182211» >
        <property name=«issuedby»  value=«false» />
        <property name=«leadto»  value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
    <edge type=«INCIDENT_INCIDENT» fromid=«190111»
toid=«191011» >
        <property name=«issuedby»  value=«false» />
        <property name=«leadto»  value=«true» />
        <property name=«comment» value=«Вызвал» />
    </edge>
```

ПРИЛОЖЕНИЕ В

Испытания системы приобретения и представления экспертных знаний: методика

Объект испытаний: система приобретения и представления экспертных знаний и организованная с ее использованием база экспертных знаний об инцидентах в ИС.

Цель испытаний: проверка соответствия разработанной системы, функционирующей в эксплуатационной среде, и организованной с ее использованием базы экспертных знаний перечисленным требованиям.

Тест 1. **Создание записей об объектах и связях между ними в базе экспертных знаний**

Порядок тестирования:

Сценарий 1. Использование в качестве источника данных XML-файлов.

Сценарий 2. Использование в качестве источника данных реляционной СУБД.

Сценарий 1

Исходные данные:

XML-файл incident.xml в директории данных системы, предназначенный для хранения записей об объектах типа «Инцидент»;

XML-файл object-incident.xml в директории данных системы, предназначенный для хранения записей о причинно-следственных связях типа «Инцидент – атомарный объект».

Последовательность выполняемых действий и ожидаемый результат:

№№ действия	Порядок выполнения действий и их содержание	Ожидаемые результаты выполнения действий
1.	В открытый на редактирование файл incident.xml внести запись об инциденте в соответствии с разработанной структурой данных; сохранить изменения	В файл incident.xml добавлена новая запись об инциденте
2.	В открытый на редактирование файл object-incident.xml внести запись о связи инцидента, описанного на предыдущем шаге, с объектом в соответствии с разработанной структурой данных; сохранить изменения	В файл object-incident.xml добавлена новая запись о связи инцидента с объектом

Сценарий 2

Исходные данные:

Табл. incident в реляционной базе данных, содержащая поля, соответствующие разработанной структуре данных;

Табл. object-incident в реляционной базе данных, содержащая поля, соответствующие разработанной структуре данных.

Последовательность выполняемых действий и ожидаемый результат:

№№ действия	Порядок выполнения действий и их содержание	Ожидаемые результаты выполнения действий
1.	В таблицу incident внести запись об инциденте в соответствии с заданными полями; сохранить изменения	В таблицу incident добавлена новая запись об инциденте
2.	В таблицу object-incident.xml внести запись о связи инцидента, описанного на предыдущем шаге, с объектом в соответствии с заданными полями; сохранить изменения	В таблицу object-incident добавлена новая запись о связи инцидента с объектом

Тест 2. **Просмотр созданных записей об объектах в графическом интерфейсе пользователя.**

Порядок тестирования:

Сценарий 1. Запуск системы и просмотр созданных записей в рабочей области графического интерфейса.

Сценарий 1

Исходные данные:

- Файлы системы;
- Источники данных в виде XML-файлов или реляционной БД.

Последовательность выполняемых действий и ожидаемый результат:

№№ действия	Порядок выполнения действий и их содержание	Ожидаемые результаты выполнения действий
1.	Запустить систему приобретения и представления экспертных знаний	Система запущена
2.	В строке поиска (1) графического интерфейса (Рис. 47) ввести поисковый запрос к базе экспертных знаний, содержащий часть информации из записи об инциденте, созданной в процессе выполнения теста 1; нажать клавишу Enter или кнопку «Искать!» панели инструментов	В динамически изменяемой области под панелью инструментов показан список объектов, удовлетворяющих введенному запросу
3.	Выбрать из списка запись об инциденте, выполнив на ней щелчок левой кнопкой мыши	В рабочей области (2) открыто представление, центральным элементом которого является выбранный инцидент, и его связь с объектом. В панели свойств (3) инцидента отображены значения полей выбранной записи
4.	Навести указатель мыши на обозначение инцидента в рабочей области	В строке состояния (4) отображена дополнительная информация об инциденте (при условии ее наличия)

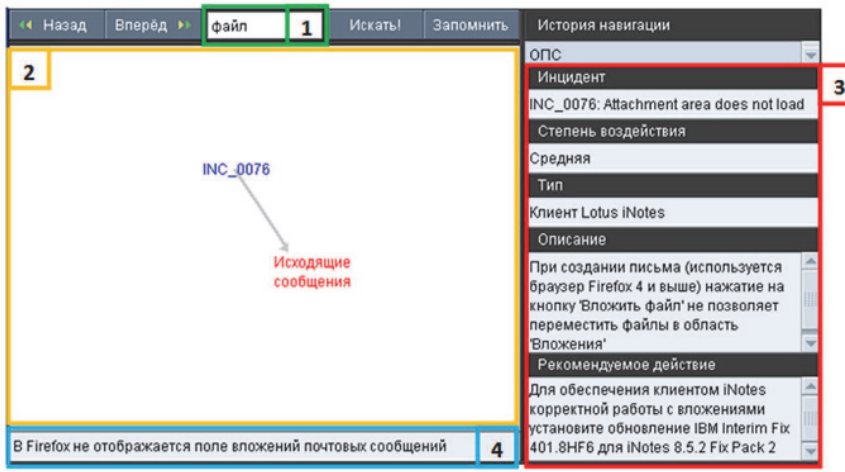

Рис. 47. Представление инцидента в рабочей области

Тест 3. **Просмотр представлений, формируемых в соответствии с классификаторами инцидентов**

Порядок тестирования:

Сценарий 1. Последовательный переход между представлениями от уровня ИС к уровню объектов в рабочей области.

Сценарий 2. Переход между объектами типа «Инцидент» в соответствующем представлении с использованием панели инцидентов.

Сценарий 1

Исходные данные:

- Файлы системы;
- Источники данных в виде XML-файлов или реляционной БД.

Последовательность выполняемых действий и ожидаемый результат:

№№ действия	Порядок выполнения действий и их содержание	Ожидаемые результаты выполнения действий
1.	В исходном представлении выбрать (навести указатель мыши на обозначение и нажать левую кнопку) подсистему «Операционные системы»	В рабочей области открыто представление, центральным объектом которого является выбранная подсистема ИС, и все ее связи с объектами типа «аппаратно-программный компонент»
2.	В открывшемся представлении выбрать аппаратно-программный компонент «Novell Suse Linux Enterprise Server 10»	В рабочей области (1) открыто представление, центральным объектом которого является выбранный компонент, и все его связи с объектами типа «подсистема ИС» и «группа объектов» (Рис. 48)
3.	В открывшемся представлении выбрать группу объектов «Файловая система»	В рабочей области (1) открыто представление, центральным объектом которого является выбранная группа объектов, и все ее связи с объектами типа «аппаратно-программный компонент» и «атомарный объект» (Рис. 49)
4.	В открывшемся представлении выбрать атомарный объект «ext3»	В рабочей области открыто представление, центральным объектом которого является выбранный объект, и все его связи с объектами типа «группа объектов» и «инцидент»

Рис. 48. Представление, центральным элементом которого является аппаратно-программный компонент

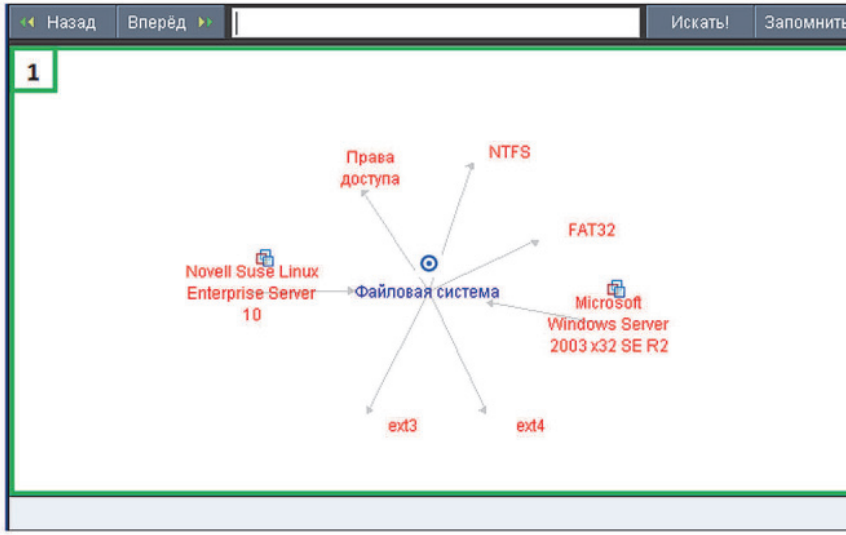

Рис. 49. Представление, центральным элементом которого является группа объектов

Сценарий 2

Исходные данные:

- Файлы системы;
- Источники данных в виде XML-файлов или реляционной БД.

Последовательность выполняемых действий и ожидаемый результат:

№№ действия	Порядок выполнения действий и их содержание	Ожидаемые результаты выполнения действий
1.	В строке поиска ввести код инцидента «INC_0043»; нажать клавишу Enter или кнопку «Искать!» панели инструментов	В рабочей области (1) открыто представление, центральным объектом которого является указанный инцидент. Все инциденты, связанные с ним причинно-следственными связями, распределены по уровням модели ISO/OSI в панели инцидентов (2) (Рис. 50)
2.	В панели инцидентов (2) выбрать инцидент «INC_0059»	В новом представлении все инциденты, связанные с выбранным причинно-следственными связями, также распределены по уровням модели ISO/OSI в панели инцидентов

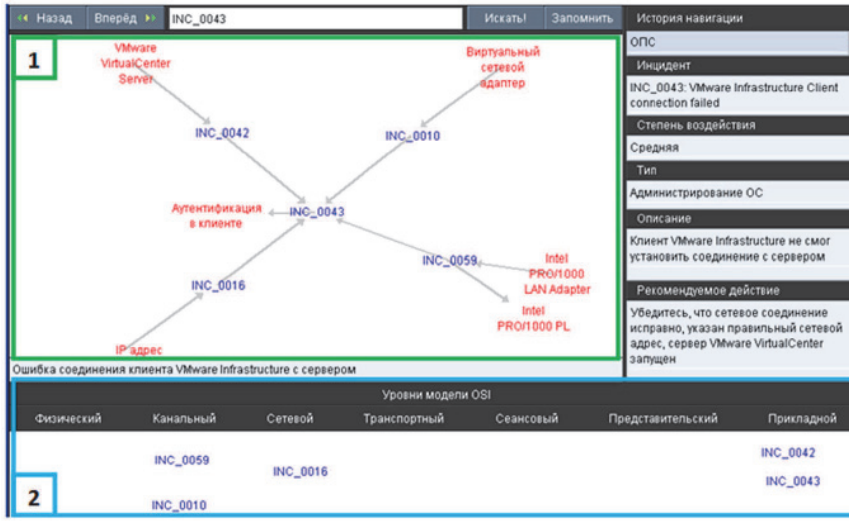

Рис. 50. Распределение инцидентов выбранного представления в соответствии с моделью ISO/OSI

Тест 4. **Поиск и выбор представления, содержащего подмножество инцидентов, связанных с объектом**

Порядок тестирования:

Сценарий 1. Последовательный переход к уровню объектов в рабочей области.

Сценарий 2. Переход к требуемому представлению с использованием строки поиска.

Сценарий 1

Последовательность выполняемых действий и ожидаемый результат аналогичны сценарию 1 теста 3.

Сценарий 2

Исходные данные:

- Файлы системы;
- Источники данных в виде XML-файлов или реляционной БД.

Последовательность выполняемых действий и ожидаемый результат:

№№ действия	Порядок выполнения действий и их содержание	Ожидаемые результаты выполнения действий
1.	В строке поиска ввести название объекта «ядро»; нажать клавишу Enter или кнопку «Искать!» панели инструментов	В рабочей области открыто представление, центральным объектом которого является группа объектов «Ядро». В состав группы входят объекты «Сборка ядра» и «Обновление версии ядра», также отобранные в представление
2.	В открывшемся представлении выбрать объект «Обновление версии ядра»	В рабочей области (1) открыто представление, содержащее объект «Обновление версии ядра» и подмножество инцидентов, связанных с ним: «Ошибка при запуске демона DHCP» и «Ошибка монтирования съемных устройств» (Рис. 51)

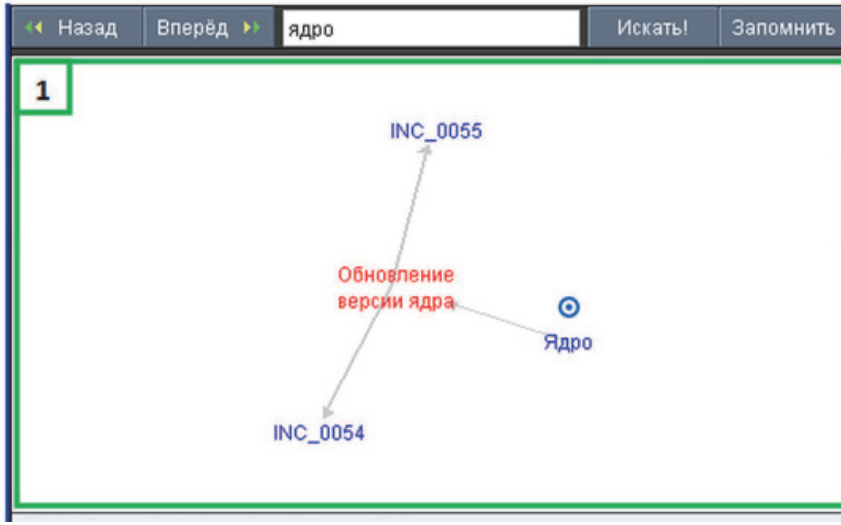

Рис. 51. Подмножество инцидентов, связанных с объектом в составе ИС

Тест 5. **Поиск и выбор представления, содержащего подмножество объектов, связанных с инцидентом**

Порядок тестирования:

Сценарий 1. Последовательный переход к уровню инцидентов в рабочей области.

Сценарий 2. Переход к требуемому представлению с использованием строки поиска.

Сценарий 1

Исходные данные:

- Файлы системы;
- Источники данных в виде XML-файлов или реляционной БД.

Последовательность выполняемых действий и ожидаемый результат:

№№ действия	Порядок выполнения действий и их содержание	Ожидаемые результаты выполнения действий
1-4.	Аналогичны соответствующим действиям сценария 1 теста 3	Аналогичны соответствующим результатам сценария 1 теста 3
5.	В открывшемся представлении выбрать инцидент с кодом «INC_0015»	В рабочей области (2) открыто представление, содержащее инцидент «Невозможно создать файл» и подмножество объектов, связанных с ним: «ext3» и «ext4» (Рис. 52)

Сценарий 2

Исходные данные:

- Файлы системы;
- Источники данных в виде XML-файлов или реляционной БД;

Последовательность выполняемых действий и ожидаемый результат:

№№ действия	Порядок выполнения действий и их содержание	Ожидаемые результаты выполнения действий
1.	В строке поиска (1) ввести код инцидента «INC_0015»; нажать клавишу Enter или кнопку «Искать!» панели инструментов	В рабочей области (2) открыто представление, содержащее инцидент «Невозможно создать файл» и подмножество объектов, связанных с ним: «ext3» и «ext4» (Рис. 52)

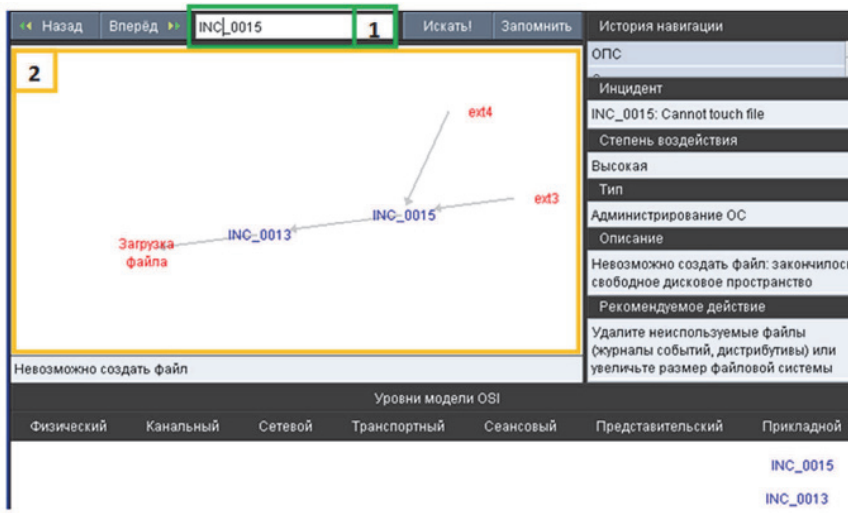

Рис. 52. Подмножество объектов, связанных с инцидентом

Тест 6. **Выполнение поисковых запросов**

Порядок тестирования:

Сценарий 1. Выполнение поиска по нескольким ключевым словам на русском языке.

Сценарий 2. Выполнение поиска по названиям объектов на английском языке.

Сценарий 3. Выполнение поиска по фрагментам описания действий на русском языке.

Сценарий 1

Исходные данные:

- Файлы системы;
- Источники данных в виде XML-файлов или реляционной БД.

Последовательность выполняемых действий и ожидаемый результат:

№№ действия	Порядок выполнения действий и их содержание	Ожидаемые результаты выполнения действий
1.	В строке поиска ввести ключевые слова запроса к базе знаний «создать файл»; нажать клавишу Enter или кнопку «Искать!» панели инструментов	В динамически изменяемой области (1) под панелью инструментов показан список объектов, удовлетворяющих введенному запросу (Рис. 53). К результатам поиска относятся все объекты, в описании которых присутствует хотя бы одно из указанных слов
2.	В строке поиска ввести ключевые слова запроса к базе знаний, заключенные в двойные кавычки: «"создать файл"»; нажать клавишу Enter или кнопку «Искать!» панели инструментов	В динамически изменяемой области под панелью инструментов показан список объектов, удовлетворяющих введенному запросу. К результатам поиска относятся все объекты, в описании которых присутствует указанное словосочетание целиком

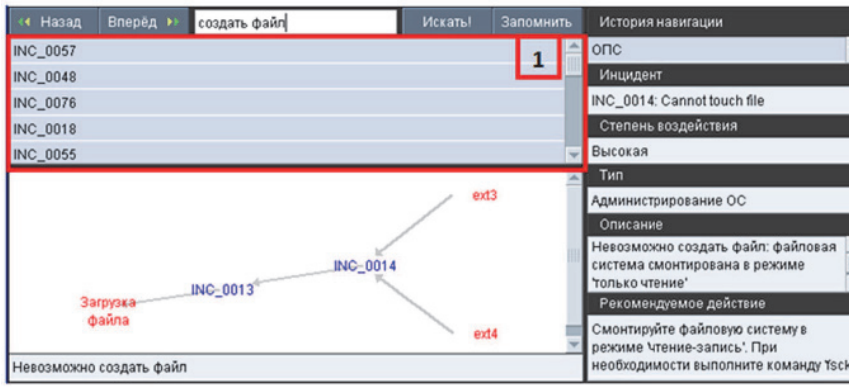

Рис. 53. Список объектов, удовлетворяющих введенному поисковому запросу

Сценарий 2

Исходные данные:

- Файлы системы;
- Источники данных в виде XML-файлов или реляционной БД.

Последовательность выполняемых действий и ожидаемый результат:

№№ действия	Порядок выполнения действий и их содержание	Ожидаемые результаты выполнения действий
1.	В строке поиска ввести название объекта на английском языке «ldap»; нажать клавишу Enter или кнопку «Искать!» панели инструментов	В динамически изменяемой области (1) под панелью инструментов показан список объектов, удовлетворяющих введенному запросу (Рис. 54)
2.	Выбрать из списка запись об объекте «Серверная задача LDAP», выполнив на ней щелчок левой кнопкой мыши	В рабочей области открыто представление, содержащее выбранный объект, группу объектов, в которую он входит, и связанные с ним инциденты
3.	Выбрать из списка запись об инциденте с кодом «INC_0020», выполнив на ней щелчок левой кнопкой мыши	В рабочей области открыто представление, содержащее инцидент «Указан неверный интернет пароль» и подмножество объектов, связанных с ним: «Интернет пароль пользователя» и «Серверная задача LDAP»

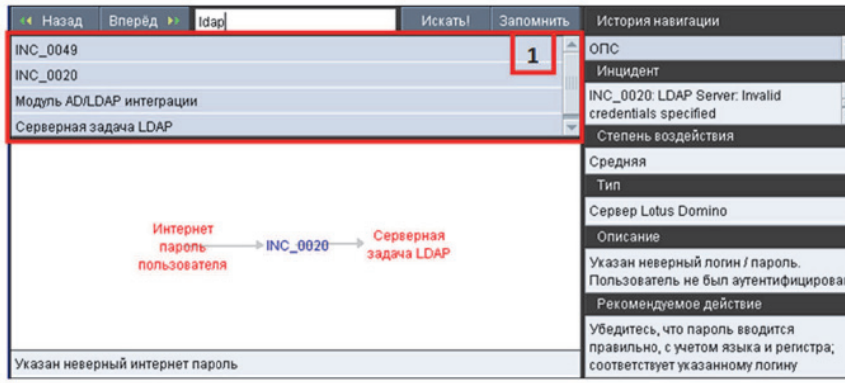

Рис. 54. Результат выполнения поискового запроса на английском языке

Сценарий 3

Исходные данные:

- Файлы системы;
- Источники данных в виде XML-файлов или реляционной БД.

Последовательность выполняемых действий и ожидаемый результат:

№№ действия	Порядок выполнения действий и их содержание	Ожидаемые результаты выполнения действий
1.	В строке поиска (1) ввести фрагмент описания действия «установить обновление»; нажать клавишу Enter или кнопку «Искать!» панели инструментов	В динамически изменяемой области под панелью инструментов (2) показан список объектов, удовлетворяющих введенному запросу (Рис. 55). В рабочей области открыто представление, содержащее объект «Обновление версии ядра», группу объектов, в которую он входит, и связанные с ним инциденты
2.	В строке поиска (1) ввести фрагмент описания действия, заключенный в двойные кавычки «"установить обновление"»; нажать клавишу Enter или кнопку «Искать!» панели инструментов	В рабочей области открыто представление, содержащее инцидент «Утечка памяти препятствует работе DNS сервера», в описании которого присутствует ссылка (2) на требуемое обновление (Рис. 56)

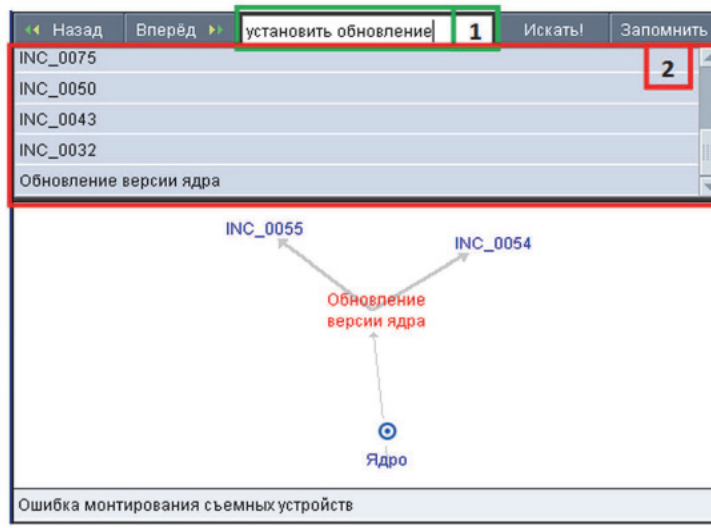

Рис. 55. Результат выполнения поискового запроса, содержащего фрагмент описания действия

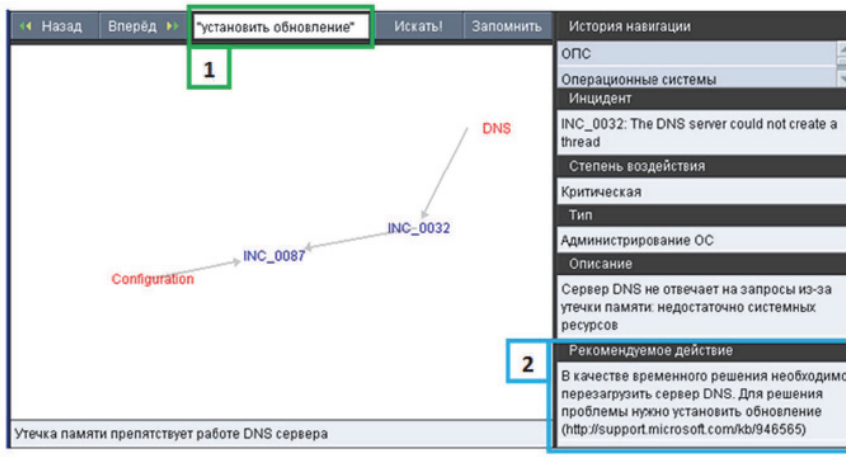

Рис. 56. Результат выполнения уточненного поискового запроса

Научное издание

Андрей Анатольевич Карасёв
Владимир Александрович Старых

ИНТЕЛЛЕКТУАЛЬНАЯ ОБРАБОТКА И РАЗРЕШЕНИЕ ИНЦИДЕНТОВ В ИНФОРМАЦИОННЫХ СИСТЕМАХ: ОНТОЛОГИЧЕСКИЙ ПОДХОД И КЛАССИФИКАЦИЯ

Научный редактор *Болотова Л.С.*
Корректоры *Клышинская С.П.,*
Конюхова Ю.Э.
Оформление и верстка *Логинов К.В.*

Подписано в печать
Формат 145 × 210 мм. Бумага офсетная.
Печать офсетная. Усл. печ. л. 14,0. Тираж 1000 экз.
Зак. 596.

Отпечатано в ООО «Строгановские мастерские».
Москва, Волоколамское шоссе, д. 9.