



УДК 007:681.512.2

А. И. Водяхо, С. С. Голяк, С. А. Гордеев, Н. А. Жукова

АРХИТЕКТУРНЫЙ ФРЕЙМВОРК, ОРИЕНТИРОВАННЫЙ НА ПОДДЕРЖКУ ПРОЦЕССА РАЗРАБОТКИ СИСТЕМ ОБРАБОТКИ МНОГОМЕРНЫХ ИЗМЕРЕНИЙ ПАРАМЕТРОВ ПРОСТРАНСТВЕННО РАСПРЕДЕЛЕННЫХ ОБЪЕКТОВ

Предлагается архитектурный фреймворк, ориентированный на поддержку процесса разработки систем обработки многомерных измерений параметров пространственно распределенных объектов, которые могут быть отнесены к классу data fusion-систем. Фреймворк базируется на нашедших широкое применение фреймворках DoDAF, TOGAF, фреймворке Захмана и JDL-модели и может быть отнесен к классу интеграционных архитектурных фреймворков.

Информационно-управляющие системы, доменный фреймворк, JDL-модель

Системы обработки многомерных измерений параметров пространственно распределенных объектов (СОМ ИП ПРО) являются одним из важнейших подклассов класса информационно-управляющих систем (ИУС). К СОМ ИП ПРО можно отнести ИУС, работающие с данными об объектах реального мира, поступающими от большого числа пространственно разнесенных датчиков. Полученные данные проходят многоуровневую обработку, из них извлекается информация, а из информации – знания, которые используются (обычно человеком) для принятия решений. К СОМ ИП ПРО не могут быть отнесены: простые ИУС с небольшим числом источников данных, для которых не требуется реализовывать высокоуровневые механизмы обработки данных, а также системы с критическими функциями (жесткое реальное время, сверхвысокая производительность, сверхвысокая надежность, очень низкая стоимость), а также ИУС, изначально ориентированные на конкретную платформу (например, мультиагентные системы), и ИУС, ориентированные на решение задач, связанных с конфликтом интересов (например, системы управления боевыми роботами).

Концептуальная (Computational Independent Model, CIM), описывающая процесс функционирования СОМ ИП ПРО описана в [1]. СОМ ИП ПРО можно рассматривать как подкласс data fusion-систем (DF-систем). Для построения DF-систем было предложено несколько различных фреймворков, среди которых наиболее широкое применение получила JDL-модель [2]. Первоначально она использовалась как функциональный фреймворк, однако существующее многообразие создаваемых DF-систем и ограниченность самой модели привели к тому, что сегодня JDL – это скорее не модель, а парадигма, ориентированная на координацию работ в области DF-систем [2], поскольку ее использование не позволяет решить задачу уменьшения трудоемкости разработки ИС, относящихся к данному классу, в частности, за счет повторного использования кода.

Принимая во внимание достаточно широкое распространение систем СОМ ИП ПРО, их сложность и высокую стоимость разработки, представляется целесообразным разработать специальный архитектурный фреймворк доменного уровня для СОМ ИП ПРО. Международный стандарт ISO/IEC 42010 определяет архитектурный фреймворк как «совокупность соглашений, принципов и практик, используемых для описаний архитектур и принятых применительно к некоторому предметному домену и (или) в сообществе специалистов (заинтересованных лиц)». Это касается, прежде всего, возможности использовать лучшие практики (фреймворки) и, в частности, входящие в их состав инструментальные средства.

Предлагаемый фреймворк (рис. 1) базируется на нашедших широкое применение фреймворках DoDAF [3], TOGAF[4], фреймворке Захмана [4] и JDL-модели [2] и может быть отнесен к классу интеграционных архитектурных фреймворков. Предлагаемый подход можно позиционировать как полицентрический, поскольку в зависимости от точки зрения его можно рассматривать как data-centric, process-centric или net-centric. Основной задачей является интеграция перечисленных фреймворков и их адаптация к конкретной предметной области – системам обработки многомерных измерений параметров пространственно распределенных объектов.

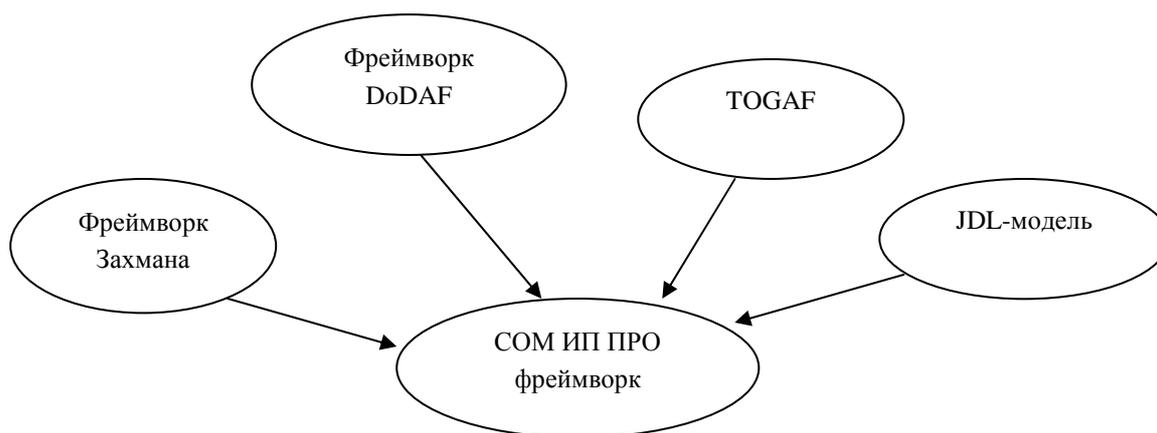


Рис. 1

Под архитектурой понимается совокупность основных элементов, входящих в состав систем рассматриваемого класса, и связи между ними, заинтересованные лица (Stakeholders), точки зрения (Viewpoints), а также процессы, определяющие процедуру создания и управления жизненным циклом образовательного ресурса.

Структура архитектурного описания показана на рис. 2. Основными элементами архитектурного описания являются модели (models), виды (view) и точки зрения (viewpoints). В качестве пользователей выступают заинтересованные лица (Stakeholders). В рамках предлагаемого фреймворка определяются 6 аспектов рассмотрения с точки зрения разных заинтересованных лиц (stakeholders), в число которых входят: аналитики, менеджеры, архитекторы, разработчики, системные администраторы, пользователи, что соответствует подходу, принятому в DoDAF [3].

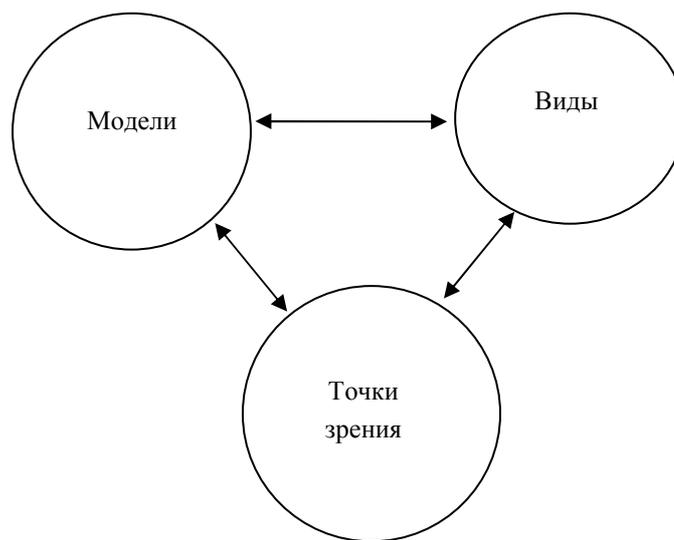


Рис. 2

Пользователь – специалист, использующий систему в процессе выполнения своих функций, в частности, при принятии решений.

Аналитик – специалист, оценивающий качество функционирования системы на всех этапах ее жизненного цикла.

Архитектор – специалист, отвечающий за формирование, разработку и сопровождение системы на всех этапах ее жизненного цикла.

Разработчик – специалист, обычно программист, занятый разработкой и модернизацией системы.

Менеджер – специалист, отвечающий за принятие финансовых, организационных и кадровых решений.

Системный администратор (Сисадмин) – специалист, отвечающий за поддержание аппаратных и программных средств в работоспособном состоянии, а также за работу с пользователями и безопасность системы.

Модель определяется как шаблон (template) для сбора данных разного рода, включая требования, при этом выделяются следующие типы моделей: таблицы, графические изображения, описывающие поведенческие аспекты архитектурного решения, онтологии, разного рода временные диаграммы и картинки в свободном формате. Состав используемых моделей в целом аналогичен моделям, используемым во фреймворке Захмана [4].

Вид (View) определяется как способ представления связанного набора данных в форме, понятной пользователю. Это могут быть документы, таблицы, графики и т. п.

Точка зрения (Viewpoints) описывает данные, поступающие от одного или нескольких источников, которые организованы таким образом, чтобы быть полезными при принятии решений, и может включать несколько видов. Точка зрения представляет собой упорядоченное множество видов.

В рамках предлагаемого фреймворка определяются 9 точек зрения: обобщенная; алгоритмическая; данные, информация и знания; системная; сервисная; операционная; проектная; учитывающая стандарты.

Обобщенная точка зрения (All Viewpoint, AV) интегрирует все точки зрения и образует архитектурный контекст для других точек зрения.

Точка зрения, определяющая потенциальные возможности (Capability Viewpoint, CV), использует такие понятия, как, например, сроки поставки оборудования, обучения персонала и т. д.

Алгоритмическая точка зрения (Algorithmic Viewpoint, AV) описывает систему в терминах используемых алгоритмов, при этом за основу берется общепринятая шестилеveled JDL-модель [2].

Системная точка зрения (Systems Viewpoint, SV) рассматривает систему как совокупность взаимодействующих подсистем, рассматривает способы взаимодействия подсистем и используется в качестве структурного описания системы.

Точка зрения «данные, информация и знания» (Data, Information and Knowledge Viewpoint, DIV) рассматривает структуры данных и способы их представления для разных целей. Это ключевая для данного фреймворка точка зрения, поскольку унификация форматов представления данных и знаний создает предпосылки для внедрения типовых решений. В табл. 1 приведены способы представления данных (знаний) применительно к уровням 0–3 JDL-модели. Способ визуализации представления зависит от специфики используемого человекомашинного интерфейса. Это может быть, например, ГИС-интерфейс или форматы представления информации, используемые в системах Business Intelligence.

Таблица 1

Номер уровня	Действия	Типовые алгоритмы	Способ представления
0	Обработка информации об отдельных элементах объектов (Processing subobject data refinement)	Алгоритмы предварительной обработки данных	XML/XML Schema
1	Обработка информации об объектах (Object refinement)	Алгоритмы сегментации	XML/XML Schema
2	Формирование информации о ситуации. Формирование модели ситуации (Situation refinement)	Алгоритмы кластерного анализа	RDF/ RDF Schema
3	Прогноз развития событий. Оценка угроз (Impact assessment)	Логический вывод	OWL
4	Формирование реакции (Process refinement)	Логический вывод	OWL

Сервисная точка зрения (*Services Viewpoint, SvcV*) рассматривает систему как совокупность взаимодействующих сервисов, при этом каждый сервис может являться точкой входа в бизнес-процесс.

Операционная точка зрения (*Operational Viewpoint, OV*) рассматривает систему с точки зрения сценариев работы, активностей, бизнес-процессов.

Проектная точка зрения (*Project Viewpoint, PV*) рассматривает систему с точки зрения требуемых характеристик и возможностей, а также с точки зрения процесса проектирования. В основу этой точки зрения представляется целесообразным положить модельно-ориентированный подход (*Model Driven Architecture, MDA*) [5].

Точка зрения, учитывающая стандарты (*Standards Viewpoint, StdV*), учитывает, в частности, действующие технические стандарты, методики, руководства, ограничения и т. п.

Каждое заинтересованное лицо рассматривает систему с нескольких точек зрения (табл. 2).

Таблица 2

Точка зрения	Заинтересованное лицо					
	Аналитик	Менеджер	Архитектор	Разработчик	Сисадмин	Пользователь
Обобщенная точка зрения	+	+	+	+	+	+
Точка зрения, определяющая потенциальные возможности	+	+	+	+		
Алгоритмическая точка зрения	+		+	+		
Системная точка зрения	+		+	+		
Точка зрения «данные, информация и знания»	+		+	+		
Сервисная точка зрения	+	+	+	+	+	
Операционная точка зрения	+	+	+	+	+	+
Проектная точка зрения			+	+		
Точка зрения, учитывающая стандарты			+	+	+	

Предлагаемый подход является обобщением опыта проектирования ряда реальных систем обработки многомерных измерений параметров пространственно распределенных объектов для различных предметных доменов. Его использование позволило существенно сократить сроки и уменьшить стоимость разработки реальных систем за счет повторного использования решений разного уровня.

СПИСОК ЛИТЕРАТУРЫ

1. Жукова Н. А., Водяхо А. И. Архитектурный подход к построению систем обработки многомерных измерений параметров пространственно распределенных объектов//Изв. СПбГЭТУ «ЛЭТИ». 2013. Вып. 2. С. 21–27.
2. Handbook of multisensor data fusion : theory and practice / ed. M. Liggins, D. Hall, J. Llinas. 2nd ed. Boca Raton, FL: Taylor & Francis Group, 2009.
3. Dam S. DoD Architecture Framework: A Guide to Applying System Engineering to Develop Integrated Executable Architectures. Marshall, VA: SPEC, 2006.
4. Finkelstein C. Enterprise Architecture for Integration. Rapid Delivery Methods and Technologies. Norwood, MA: Artech House Publishers, 2006.
5. Duffy D. Domain Architectures: Models and Architectures for UML Applications. John Wiley & Sons, 2004.

ARCHITECTURAL FRAMEWORK FOR MULTIDIMENSIONAL PARAMETER MEASUREMENTS OF SPATIALLY DISTRIBUTED OBJECTS OF IT SYSTEMS DEVELOPMENT

The paper discusses an architectural approach to development of IT systems for multidimensional parameter measurements of spatially distributed objects. A platform independent model of IT systems for multidimensional parameter measurements of spatially distributed objects is suggested. Suggested Framework is to be used to support process of development of IT data fusion systems for processing information from multidimensional parameter measurements of spatially distributed objects. The framework is based on well known DoDAF, TOGAF, Zachman frameworks and JDL-model and can be positioned as architectural integration framework.

Information control systems, domain framework , JDL-model

УДК 621.3.049.77.001.2

С. Э. Миронов, А. О. Менько

СЖАТИЕ ТОПОЛОГИИ С ИЗМЕНЕНИЕМ ФОРМЫ ЭЛЕМЕНТОВ

Рассматриваются вопросы повышения эффективности систем сжатия топологии СБИС путем автоматизации изменения формы элементов.

Технологически инвариантное проектирование топологии, изменение формы элементов, автоматизация

Технологически инвариантное проектирование топологии СБИС. Одной из основных отличительных особенностей современных микроэлектронных проектов является технологическая инвариантность – возможность настройки проекта на любые проектные нормы. Она достигается с помощью систем сжатия топологии, минимизирующих расстояние между элементами в соответствии с проектными нормами и задаваемыми разработчиком ограничениями на расположение отдельных частей топологии. В развитии методов технологически инвариантного проектирования топологии основными направлениями являются повышение плотности упаковки топологии, и упрощение алгоритмов ее сжатия. Совершенно очевидно, что и плотность упаковки топологии, и алгоритмы сжатия неразрывно связаны с конструктивом систем символического проектирования.

Этапы развития конструктива в системах символического проектирования топологии. В первых системах технологически инвариантного проектирования топологии элементы конструктива располагались в узлах и на линиях некоторой координатной сетки с постоянным шагом (рис. 1, а). Шаг определялся как максимальное из минимально допустимых проектными нормами расстояний между осями соседних элементов топологии и обеспечивал зазоры между различными элементами, не меньшие допустимых проектными нормами конкретной технологии. Сжатие как таковое отсутствовало, но подобный метод позволял предельно просто и быстро осуществить перевод проекта на новые нормы. Очевидный недостаток данного метода – низкая плотность упаковки – привел к созданию систем проектирования топологии на основе виртуальной сетки (ВС) – сетки с переменным шагом, систем, предполагающих сжатие топологии с постоянным шагом сетки.