

Шнепс-Шнеппе М.А.¹, Сухомлин В.А.^{2,3}, Намиот Д.Е.²¹ ЦКБ-АбаваНет, г. Москва, Россия² Московский государственный университет имени М.В. Ломоносова, г. Москва, Россия³ Федеральный исследовательский центр «Информатика и управление» РАН, г. Москва, Россия**ОБ ИНФОРМАЦИОННЫХ МОДЕЛЯХ ЦИФРОВОЙ ЭКОНОМИКИ*****Аннотация**

Рассматривается принципиальный вопрос об информационных моделях цифровой экономики. Наличие единой модели позволит согласовывать усилия разработчиков разных областей и обеспечит разработку единых программных средств. Кратко рассмотрен советский опыт внедрения вычислительной техники в народное хозяйство. Рассмотрен американский опыт разработки единой архитектуры информационных систем, анализируются трудности внедрения единой архитектуры и ее недостатки. В этой связи рассмотрены работы Дж. Захмана по архитектурам информационных систем. Архитектура предприятия представлялась как "набор описательных представлений (моделей), которые применимы для описания Предприятия в соответствии с требованиями управленческого персонала (качество) и которые могут развиваться в течение определенного периода (динамичность)". Основная идея модели заключается в том, чтобы обеспечить возможность последовательного описания каждого отдельного аспекта системы в координации со всеми остальными. Для любой достаточно сложной системы общее число связей, условий и правил обычно превосходит возможности для одновременного рассмотрения. Далее рассматривается модель электронного правительства, созданная в NIST, архитектура DoDAF и язык SysML. В качестве прототипа российской информационной модели цифровой экономики предлагается новейший язык Lifecycle Modeling Language.

Ключевые слова

Цифровая экономика; информационная модель; ОГАС; ЕССКТ; модель Захмана; электронное правительство; DODAF; LML.

Sneps-Sneppe M.A.¹, Sukhomlin V.A.^{2,3}, Namiot D.E.²¹ SKB-AbavaNet, Moscow, Russia² Lomonosov Moscow State University, Moscow, Russia³ Federal Research Center Computer Science and Control of the Russian Academy of Sciences, Moscow, Russia**ON INFORMATION MODELS OF THE DIGITAL ECONOMY****Abstract**

This paper discusses the principal issue of information models of the digital economy. The presence of a unified model will allow coordinating the efforts of developers from different areas and will ensure the development of unified software. The Soviet experience of introducing computer technology into the national economy was briefly reviewed. The American experience of developing a unified architecture of information systems is considered, the difficulties of introducing a single architecture and its shortcomings are analyzed. In this connection, the work of J. Zachman on the architectures of information systems is considered. The enterprise architecture was presented as "a set of descriptive ideas (models) that are applicable to the description of an Enterprise in accordance with the requirements of management personnel (quality) and which can develop over a certain period (dynamism)." The main idea of the model is to provide the possibility of a sequential description of each individual aspect of the system in coordination with all the others. For any sufficiently complex system, the total number of links, conditions, and rules usually exceeds the possibilities for simultaneous consideration. Next, the e-government model created in NIST, the DoDAF architecture,

* Труды II Международной научной конференции «Конвергентные когнитивно-информационные технологии» (Convergent'2017), Москва, 24-26 ноября, 2017

Proceedings of the II International scientific conference "Convergent cognitive information technologies" (Convergent'2017), Moscow, Russia, November 24-26, 2017

and the SysML language are discussed. As the prototype of the Russian information model of the digital economy, we propose the newest language Lifecycle Modeling Language.

Keywords

Digital economy; information model; OGAS; ESKTT; Zachman model; e-government; DODAF; LML.

1. Введение

Общегосударственная программа «Цифровая экономика Российской Федерации» [1] определяет цели и задачи в рамках восьми направлений развития цифровой экономики на период до 2025 года:

- 1) Государственное регулирование;
- 2) Информационная инфраструктура;
- 3) Исследования и разработки;
- 4) Кадры и образование;
- 5) Информационная безопасность;
- 6) Государственное управление;
- 7) Умный город;
- 8) Цифровое здравоохранение.

Крупнейшей работой в области цифровой экономики, которая, по нашему мнению, касается всех ее направлений, является разработка единых программных средств: единой информационной модели, единой архитектуры, единого языка описания. В настоящей статье мы рассматриваем создание информационной модели цифровой экономики. Наличие единой модели позволит согласовывать усилия разработчиков разных областей и обеспечит разработку единых программных средств.

Настоящая статья является продолжением нашей статьи [2]. Далее, в разделе 2 кратко рассмотрен советский опыт внедрения вычислительной техники в народное хозяйство. В разделах 3–7 рассмотрен американский опыт разработки единой архитектуры информационных систем, анализируются трудности внедрения единой архитектуры и ее недостатки. Раздел 8 посвящен новейшему языку LML (Lifecycle Modeling Language), что, на наш взгляд, может послужить прототипом для российской информационной модели цифровой экономики.

2. Что можем взять из российского опыта

О наследии маршала Огаркова. Некоторое время назад отечественные СМИ выдали сенсацию: «Американцы украли Доктрину маршала Огаркова» (Комсомольская правда, 9 июля 2010 г.). Оказывается, что, позаимствовав идеи у начальника Генерального штаба Николая Огаркова (в 1977–1984 годы), американцы совершили революцию в военном деле. Именно после этого в Пентагоне была переоценена роль систем управления и автоматизации и родилась концепция сете-центрической войны (Network Centric Warfare): в современном бою победу одержит не тот, у кого больше «платформ», а тот, кто сможет быстрее и эффективнее использовать их в нужном месте в нужное время и нужным образом.

Под руководством маршала Н.В. Огаркова в сентябре 1981 г. прошли крупнейшие оперативно-стратегические учения «Запад-81», по своим масштабам сравнимые с операциями времён Великой Отечественной войны. Впервые была опробована автоматизированная система управления и некоторые виды высокоточного оружия. Единая система автоматизированного боевого управления АСУВ фронта «Маневр» объединяла более 2800 целевых каналов (от пилота истребителя-перехватчика и стрелка-зенитчика с ПЗРК до систем дальнего действия С-300В и С-200), действующих в едином информационном пространстве. По расчетам, система «Маневр» повышала эффективность вооружения в 3...5 раз. После распада Варшавского договора один такой комплекс, оставшийся в ГДР, достался американцам. Они и провели с его использованием штабную игру, результаты которой повергли их в шок. Благодаря автоматизации управления условная армия Варшавского договора уничтожила НАТО за три дня без всякого ядерного оружия. Затем алгоритмы «Маневра» были положены в основу аналогичной системы армии США.

О началах АСУ. Следует вспомнить имя инженер-полковника Анатолия Ивановича Китова (1920–2010). В 1959 г. А.И. Китову пришла в голову идея о целесообразности создания единой автоматизированной системы управления для Вооружённых Сил и народного хозяйства страны на базе общей сети вычислительных центров, создаваемых и обслуживаемых Министерством обороны. Он предложил создать общегосударственную автоматизированную систему управления (ОГАС).

7 января 1959 г. А.И. Китов послал письмо в ЦК КПСС главе СССР Н.С. Хрущёву. В этом письме Китов предложил создать общенациональную компьютерную сеть многоцелевого назначения, предназначенную для планирования и управления экономикой в масштабе всей страны. Осенью 1959 г. Китов послал Хрущёву второе письмо. Оно содержало разработанный им 200-страничный проект

«Красная книга» — проект создания Общесоюзной сети ВЦ двойного назначения — военного и гражданского, для управления экономикой страны в мирное время и Вооружёнными силами СССР в военное. К сожалению, проект «Красная книга» был отвергнут, а сам Китов был подвергнут гонениям: его исключили из КПСС, сняли с должности начальника, созданного им ВЦ-1 МО — без права занимать руководящие должности.



Рис. 1. АСУВ «Маневр» [3]: а) командно-штабная машина МП-22Р начальника дивизии ПВО, б) пульт управления АСУВ «Маневр»

Проекты В.М. Глушкова. Идеи А.И. Китова далее развил академик В.М. Глушков. Уже в 1964 г. под его руководством был разработан первый эскизный проект Государственной сети вычислительных центров (ГСВЦ), предназначенной для перестройки на основе безбумажной технологии организационно-экономического управления на всех уровнях (от отдельных предприятий и учреждений до Госплана СССР).

Замысел ученого получил одобрение А.Н. Косыгина, председателя Совета Министров СССР, и В.М. Глушков со свойственной ему энергией приступил к делу, которое впоследствии назвал главным в своей жизни. Он подсчитал, что при умелой организации работ уже через пять лет затраты на ОГАС станут окупаться, а после ее реализации возможности экономики и благосостояние населения по меньшей мере удвоятся. Было еще одно обязательное условие, которое он поставил, а именно: создание авторитетного, наделенного всеми полномочиями государственного органа управления ходом выполнения программы ОГАС – Государственного комитета по управлению программой (Госкомупр), наподобие тех комитетов, с помощью которых осуществлялись космическая и ядерная программы. Это в то время было расценено политической крамолы: научное управление обществом – о ужас! – не на основе марксизма-ленинизма, а на базе точных наук – математики, автоматического управления, статистики и пр. Работы по ОГАС свернули.

Заметим, что завершение работ по ОГАС Глушков относил на 90-е годы, т.е. при политической поддержке ОГАС могла бы действительно стать реальностью. Не надо думать, что свершившийся сейчас переход от планового хозяйства к рыночной экономике сделал бы ОГАС ненужной и неэффективной. Как раз наоборот, ее техническая база, накопленное программно-алгоритмическое обеспечение, банки данных, накопившие опыт кадры сослужили бы очень полезную службу народному хозяйству новой России.

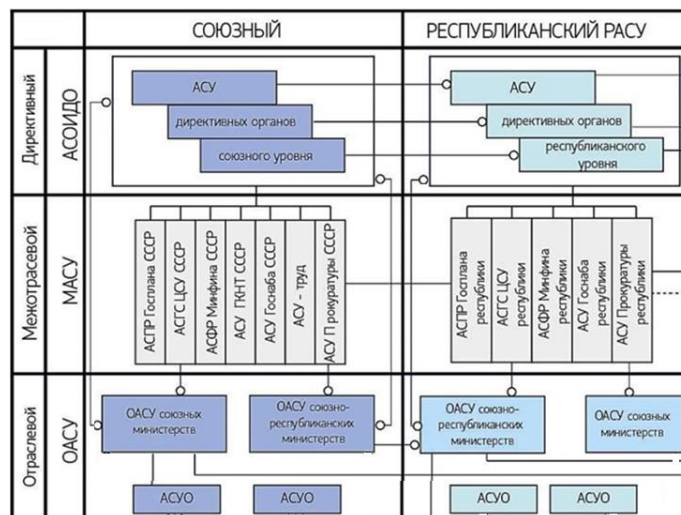


Рис. 2. Функциональная структура ОГАС (фрагмент) [4]

Научный задел по ОГАС нашел некоторое продолжение в постсоветское время. Еще в начале 70-х годов в Институте кибернетики в Киеве (сейчас имени В.М. Глушкова) начались работы по переосмысливанию

технологии программирования. Была разработана новая технология графического программирования. В 1989г. был получен первый и до сих пор единственный в странах-членах СЭВ стандарт ISO/IEC 8631 по Р-технологии графического программирования [5]. Р-технология графического программирования имеет сходство с языком UML (Unified Modeling Language) [6].

Программные средства для телефонных станций. В 1970-ые годы Международный Союз Электросвязи разработал стандарты трех языков программирования телефонных станций [7]: язык спецификаций и описаний SDL (Specification and Description Language), язык программирования высокого уровня CHILL и язык общения персонала с телефонной станцией MML (man-machine language).

Используя эти языковые средства, в странах членах СЭВ разрабатывалось поколение электронных станций ЕССКТ (Единая Система Средств Коммутационной Техники). Головным предприятием по проекту ЕССКТ (сравнимому с проектом ЕСЭВМ) являлся институт НИИ ВЭФ (Рига, Латвия). После развала СССР проект ЕССКТ прекратил свое существование. В качестве примера последовательного применения программных средств МСЭ можно указать цифровую электронную коммутационную систему EWSD, разработанную немецким концерном Siemens.

Цифровое предприятие. Укажем два новейших российских проекта в области цифрового предприятия. 13–16 декабря 2016 г. в Москве было проведено заседание Межведомственной экспертной комиссии по приемке результатов проектов по направлению «Цифровое предприятие».

Объектами испытаний были следующие программные комплексы (всего 11 комплексов программ в импорто-независимой реализации):

- система управления производственным предприятием;
- система управления материальными ресурсами;
- система управления трудовыми ресурсами;
- система управления средствами производства;
- система управления производственными процессами;
- система экономического управления;
- система информационного анализа;
- комплекс средств интеграции;
- система управления основными данными;
- система управления производственными документами;
- комплекс порталных сервисов.

20-22 июня 2017 г. в Ижевске состоялся шестой международный форум «Информационные технологии на службе оборонно-промышленного комплекса России» (ИТОПК-2017). На форуме, среди прочего, обсуждали информационно-технологическую систему ТИС ЯОК. ИТ-специалисты ФГУП «РФЯЦ-ВНИИЭФ» разработали и внедрили на предприятии импорто-независимую информационно-технологическую систему ядерно-оружейного комплекса ТИС ЯОК, которая обеспечивает автоматизацию предприятия в контурах «для служебного пользования» и «государственная тайна». Система начала успешно тиражироваться на предприятиях ядерно-оружейного комплекса. Поскольку в СССР все оборонные предприятия создавались по единой производственной методологии, то и ТИС ЯОК может успешно использоваться во всей отрасли ОПК.

3. Модель Захмана

В 1987 г. появилась статья Дж.А. Захмана «Структура архитектуры информационных систем» и впервые было введено понятие «архитектура предприятия» [8]. Джон Захман много лет занимался внедрением информационных систем IBM и переосмыслил их архитектуру. Он предложил идею, которая для ИТ-отрасли сравнима с Периодической таблицей Менделеева. Он придумал новое описание таких сложных систем, как транснациональная корпорация. Тем самым он стал «отцом» архитектуры предприятия (Enterprise Architecture).

"Модель Захмана" со временем претерпела значительные изменения, появились различные ее модификации (в работах 1992-96 гг.). Например, в статье 1992 года [9] были добавлены формализованные графические средства. Модель Захмана была использована такими крупнейшими корпорациями, как General Motors, Bank of America и др. Модель Захмана послужила основой Институту NIST для создания Федеральной архитектуры США (FEAF – Federal Enterprise Architecture Framework), методикой создания архитектуры TOGAF (The Open Group Architecture Framework) сообществом Open Group и, что особенно важно, методикой описания архитектуры Министерства обороны США DoDAF (Department of Defence Architecture Framework).

Дж. Захман определил архитектуру предприятия как "набор моделей (представлений), которые применимы для описания любого предприятия в соответствии с требованиями управленческого персонала и которые могут развиваться в течение длительного периода". Основная идея архитектуры предприятия заключается в том, чтобы обеспечить возможность последовательного описания каждого

отдельного аспекта системы в координации со всеми остальными [10]. Для любой достаточно сложной системы общее число связей, условий и правил обычно превосходит возможности для их одновременного рассмотрения. В то же время отдельное, в отрыве от других, рассмотрение каждого аспекта системы чаще всего приводит к неоптимальным решениям как в плане производительности, так и стоимости реализации.

Собственно, исходная модель Захмана представляется в виде таблицы, имеющей пять строк и шесть столбцов, которая приведена на рис. 3. Заметим, что в исходной модели было именно пять строк. Шестая строка (на рис. 3) появилась позже, она выходит за пределы описания архитектуры, а описывает работающую систему или предприятие в целом.

	Данные ЧТО	Функции КАК	Дислокация ГДЕ	Люди КТО	Время КОГДА	Мотивация ПОЧЕМУ	
Бизнес-руководители	Список важных понятий и объектов	Список основных бизнес-процессов	Территориальное расположение	Ключевые организации	Важнейшие события	Бизнес-цели и стратегии	Сфера действия (контекст)
	Концептуальная модель данных	Модель бизнес-процессов	Схема логистики	Модель потока работ (workflow)	Мастер-план реализации	Бизнес-план	Модель предприятия
ИТ-менеджеры и разработчики	Логические модели данных	Архитектура приложений	Модель распределенной архитектуры	Архитектура интерфейса пользователя	Структура процессов	Роли и модели бизнес-правил	Модель системы
	Физическая модель данных	Системный проект	Технологич. архитектура	Архитектура презентации	Структуры управления	Описания бизнес-правил	Технологическая (физическая) модель
	Описание структуры данных	Программный код	Сетевая архитектура	Архитектура безопасности	Определение временных привязок	Реализация бизнес-логики	Детали реализации
	Данные	Работающие программы	Сеть	Реальные люди, организации	Бизнес-события	Работающие бизнес-стратегии	Работающее предприятие

Рис. 3. Расширенная модель Захмана

Итак, модель Захмана содержит шесть уровней. На каждом из этих уровней участники рассматривают одни и те же вопросы, соответствующие столбцам в таблице, но с различным уровнем абстракции и детализации. В содержание этих колонок входят ответы на шесть вопросов:

- используемые данные (что?);
- процессы и функции (как?);
- места выполнения этих процессов (где?);
- организации и участники (кто?);
- управляющие события (когда?);
- цели и ограничения, определяющие работу системы (почему?).

Первые три уровня формируют бизнес-руководители.

Первая строка соответствует уровню планирования бизнеса в целом (бизнес-модель). На этом уровне вводятся достаточно общие основные понятия, определяющие саму суть бизнеса (продукты, услуги, клиенты), а также формулируется бизнес-стратегия. Фактически, данная строка определяет контекст всех последующих строк.

Вторая строка (концептуальная модель) предназначена для определения (в терминах менеджеров) структуры организации, ключевых и вспомогательных бизнес-процессов.

Третий уровень (логическая модель) соответствует рассмотрению с точки зрения системного архитектора. Здесь бизнес-процессы описываются уже в терминах информационных систем, включая различные типы данных, правила их преобразования и обработки для выполнения определенных на уровне бизнес-функций.

Уровни с четвертого и далее описывают детали, которые представляют интерес для ИТ-менеджеров и проектировщиков, но ведущую роль играют разработчики.

На четвертом уровне — технологической (физической) модели — осуществляется привязка данных и операций над ними к выбранным технологиям реализации. Например, здесь может быть определен выбор реляционной СУБД или средств работы с неструктурированными данными, или объектно-ориентированной среды.

Пятый уровень соответствует детальной реализации системы, включая конкретные модели оборудования, топологию сети, производителя и версию СУБД, средства разработки и собственно готовый программный код. Многие из работ на данном уровне часто выполняются субподрядчиками.

Шестой уровень описывает работающую систему. В исходной работе Захмана содержание этого уровня не детализируется. На этом уровне могут быть введены такие объекты, как инструкции для работы с системой, фактические базы данных, работа службы HelpDesk и т.д.

4. Архитектура электронного правительства

На основе модели Захмана институт NIST разработал модель электронного правительства FEAF (Federal Enterprise Architecture Framework) для федерального правительства США [11]. Из модели Захмана были взяты лишь первые три столбца, и разработчики сосредоточились на первых трех строчках модели.

Архитектура федеральной организации — это попытка привести бесчисленное множество агентств (министерств) федерального правительства США к единой и повсеместно используемой архитектуре.

На рис. 4 приведена схема сегментов федерального правительства из «Практического руководства по FEA». Из рисунка видно, что многие сегменты (вертикальные столбцы) используются во многих агентствах и все или почти все эти сегменты можно использовать повторно.

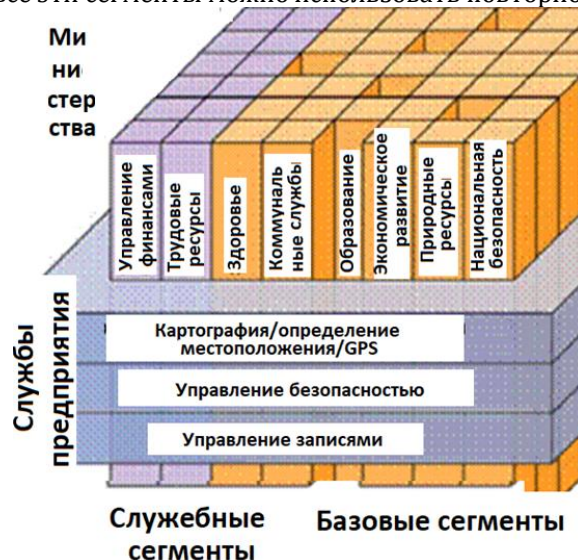


Рис. 4. Схема сегментов федерального правительства

С точки зрения FEA, архитектура предприятия состоит из отдельных сегментов. Сегмент представляет собой один из основных аспектов бизнеса, например, трудовые ресурсы. Сегменты подразделяются на два типа: базовые и служебные.

Базовый сегмент представляет собой ключевой аспект деятельности предприятия в границах политико-административного деления. Например, для Министерства здравоохранения и социальных служб США базовым сегментом является здоровье.

Служебный сегмент — это сегмент, который является фундаментальным если не для всех, то для большинства политических организаций. Например, управление финансами является служебным сегментом, обязательным для всех федеральных агентств.

Другим типом активов в архитектуре предприятия являются службы предприятия. Служба предприятия — это четко определенная функция в границах политико-административного деления. В качестве примера службы предприятия можно привести управление безопасностью. Это служба, единообразно реализованная по всему предприятию.

Различие между службами предприятия и сегментами, особенно служебными сегментами, неочевидно. И службы, и сегменты охватывают все предприятие. Различие заключается в том, что область действия служебных сегментов распространяется только на одну политическую организацию. Область же действия служб предприятия распространяется на все предприятие.

Например, и в Министерстве здравоохранения и социальных служб, и в Агентстве по охране окружающей среды федерального правительства США используется служебный сегмент трудовые ресурсы. Однако трудовые ресурсы для Министерства здравоохранения и социальных служб отличаются

от трудовых ресурсов для Агентства по охране окружающей среды. То же самое относится к службе управления безопасностью: она, по сути, одна и та же и в Министерстве здравоохранения и социальных служб, и в Агентстве по охране окружающей среды используется такая. Эффективное управление учетными записями для безопасного доступа обеспечивается только в том случае, если оно осуществляется на уровне предприятия.

К сожалению, результаты разработки программы FEA оказались не радужными. В официальном отчете Федеральной счетной палаты для Конгресса США о состоянии программы FEA в 2002 году был сделан вывод о том, что «в целом система FEA недостаточно развита для обоснованного принятия инвестиционных решений в области ИТ» [12]. К тому же программа FEA оказалась чрезвычайно дорогой. Например, «к концу 2010 года федеральное правительство потратило более миллиарда долларов на корпоративную архитектуру, и многое, если не большая часть из них была потрачена впустую» [13]

5. Архитектура DODAF

DoDAF (Department of Defense Architecture Framework) — наиболее известный проект в области цифровой экономики. DoDAF — это архитектура, позволяющая Министерству обороны США облегчить управление на всех уровнях разработки (рис. 5).



Рис. 5. Структура описания единой информационной среды МО США

Единая метамодель DoDAF (рис. 6) разрабатывается с конца 1980х гг. на базе модели Захмана в полном объеме (не в урезанном виде, как для FEA).

Рисунок 6 иллюстрирует взаимосвязи между основными понятиями метамодели DoDAF. Модель содержит шесть описаний, которые объединены ключевым понятием Действие:

- 1) описание данных (Data Description) — отвечает на вопрос ЧТО (включает и описание Ресурсов, кроме самих Данных);
- 2) описание функции (Function Description) — отвечает на вопрос КАК (содержит также описание Исполнителя, который выполняет Функции и учитывает связанные с ними Действия, Правила и Условия);
- 3) описание сети (Network Description) — ГДЕ;
- 4) описание участников (People Description) — КТО (включает и сами Организации);
- 5) описание времени (Time Description) — ГДЕ;
- 6) описание мотивации (Motivation Description) — ПОЧЕМУ (с расширением, что включает описание требований к Функциям).

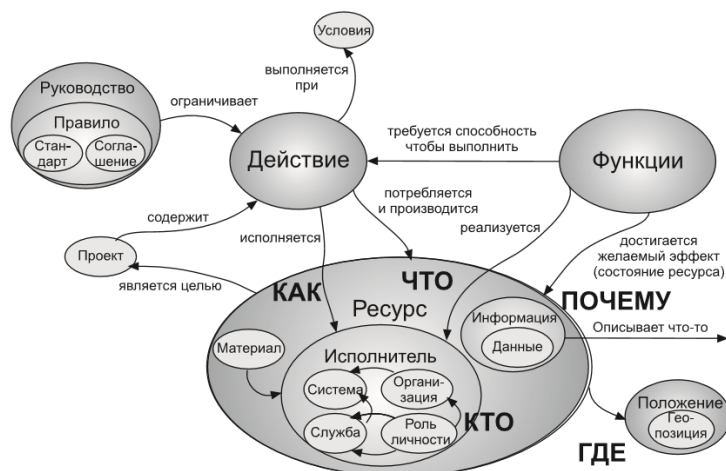


Рис. 6. Иллюстрация метамодели DoDAF

Документация по единой информационной среде GIG (Global Information Grid) представлена с восьми

точек зрения – представлений (Viewpoint) и содержит колоссальный объем материалов – целых 52 тома. (Неудивительно, что программистам, как и другим лицам, причастным к созданию DoDAF, сложно, да и нет желания изучать все эти тома.) Тома описывают все аспекты архитектурного контекста, которые относятся к DoDAF (рис. 7):

- общее описание (All Viewpoint) — 2 тома;
- описание сервисных компонентов (Capability Viewpoint) — 7 томов: описывает требования к возможностям системы; время, необходимое для развертывания системы; возможности развертки;
- описание данных и информации (Data and Information Viewpoint) — 3 тома: описывает взаимодействия между данными системы и согласование архитектуры данных;
- описание операций (Operational Viewpoint) — 9 томов: включает операционный сценарий, активности и требования по поддержке сервисных компонентов;
- описание проекта (Project Viewpoint) — 3 тома: описывает отношения между операционными требованиями и требованиями к возможностям системы;
- описание сервисов (Services Viewpoint) — 13 томов: описывает идентификацию сервисов, сервисных элементов и их взаимодействий;
- описание системы (System Viewpoint) — 13 томов: описывает системы и соединения, поддерживающие работу Департамента Защиты;
- описание стандартов (Standard Viewpoint) — 2 тома: описывает некоторые стандарты, которые используются в разработке решений.

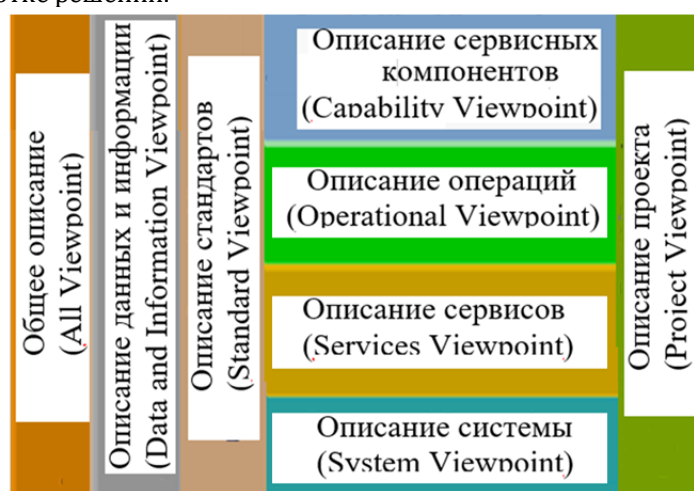


Рис. 7. Структура описания архитектуры DoDAF (версия v.2) с восьми точек зрения

Разработка DoDAF началась в конце 1980х и называлась информационной сетью C4ISR, что по-английски означает концепцию «Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance». В русском варианте – это «Управление, контроль, связь, сбор и компьютерная обработка информации, наблюдение и разведка». Предполагается, что все эти функции являются автоматизированными. C4ISR – это оперативно-стратегический уровень, или, другими словами, уровень больших начальников и штабов.

Система C4ISR включает различные подсистемы, необходимые для технического обеспечения процесса боевого управления вооруженными силами. Комплекс C4ISR, по замыслу, позволяет:

- автоматически определять положение своих подразделений;
- автоматически определять положение противника и его перемещения, что отображается на электронных картах;
- автоматически выбирать маршруты движения войск;
- автоматически задавать цели средствам огневого поражения;
- автоматически постоянно осведомлять свои подразделения о действиях и местоположении их соседей и противника;
- полностью автоматизировать сбор информации и ее обработку;
- автоматизировать процесс предложения командиру решений на основании автоматически полученных разведанных, а также информации о силах и средствах, имеющихся в распоряжении;
- моделировать бой и его возможные результаты;
- предлагать частные решения для командира в ходе боя на основе текущей ситуации.

Таким образом, система C4ISR помогает созданию единого информационного пространства, обеспечению возможности адаптивного управления, комплексного использования возможностей боевых сил.

Опять же, как и в случае электронного правительства США, результаты оказались не радужными. Как отмечали критики уже в 1992 году, Министерство обороны США представляет собой наиболее впечатляющий пример непрерывно растущих денежных вложений, получая все те же неудовлетворительные результаты: «Хотя DOD потратил более 10 лет и по меньшей мере 379 миллионов долларов на архитектуру своего бизнес-предприятия, его способность использовать Архитектуру для руководства и ограничения инвестиций осталась ограниченной» [14]. Подобный же пессимистический вывод эксперты делают более чем через двадцать лет работы – в отчете Федеральной счетной палаты 2015 года [15].

Итак, разработка DoDAF идет уже более 25 лет, но завершить ее никак не удастся. Напрашивается вывод, что сам замысел создания единой информационной системы для столь сложного предприятия, как Министерство обороны США, является сегодня непосильной задачей или же сама методика разработки по модели Захмана является ошибочной.

6. Язык SysML

Для упрощения работы с документацией единой информационной среды для МО США требуются графические средства. За прошедшие годы апробированы различные средства. В итоге выбран графический язык SysML (System Modeling Language). Напомним его предысторию. Язык UML давно уже стал стандартом общения между участниками разработки программного обеспечения крупных проектов. Его богатые выразительные средства и широкий спектр поддерживаемых продуктов способствовали тому, что UML начал проникать в другие области деятельности, связанные с моделированием бизнес-процессов. Исходных средств языка UML оказалось недостаточно для моделирования аппаратуры, поэтому понадобилось добавить ряд новых графических элементов и диаграмм, которые позволяют описывать нюансы каждого элемента модели и взаимосвязи между элементами, а также строго задавать границы модели. С другой стороны, в рамках поставленной задачи UML характеризуется некоторой избыточностью, поэтому не все его элементы вошли в новый клон. Изменения были специфицированы в виде профиля UML 2.0 и названы новым именем — SysML. В спецификацию этого языка вошли новые диаграммы — требований, внешних и внутренних блоков, времени, параметрическая. Кстати, на основе языка SysML был спроектирован орбитальный телескоп «Хаббл».

7. Критика DoDAF

Разработка архитектуры DODAF длится уже более 25 лет, в ней участвуют тысячи разработчиков. В ходе разработки Министерство обороны многократно подвергалось критике за «пустую» трату денег. Новейший обзор состояния (на июль 2017 года) работ в области архитектуры предприятий [16] имеет любопытное название «Структуры архитектуры предприятия: поветрие века». В обзоре утверждается: исторический анализ показывает, что использование DoDAF в американском Министерстве обороны обеспечивает "самый захватывающий пример непрерывно растущих затрат времени и денежных инвестиций в [архитектуру предприятия], но получение все тех же самых неудовлетворительных результатов".

Одним из ярких критиков является Стивен Дем. Сам он – многолетний разработчик архитектур C4ISR и DoDAF. Разработкой информационных систем для оборонной отрасли занимается более 40 лет. Приводим рисунок из его презентации под названием «Нужна ли еще архитектура DODAF?» [17]. На рис. 8 использованы обозначения: DODAF – архитектура МО США, MODAF – архитектура Британского МО, NAF – архитектура НАТО (развернутая в Афганистане), DNDAF – архитектура МО Канады.

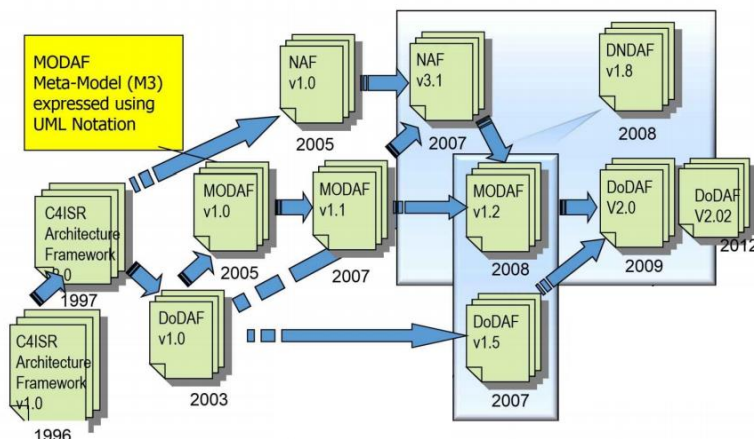


Рис. 8. Эволюция архитектурных структур оборонных ведомств

В отличие от множества критиков информационных систем сложных предприятий, в том числе оборонных ведомств, критика Стивена Дема является положительной: он предлагает новый язык описания сложных систем LML (Lifecycle Modeling Language)

8. Язык LML

Язык Lifecycle Modeling Language (LML) является языком моделирования открытого стандарта, разработанным для системных инженеров. Он поддерживает полный жизненный цикл системы: от разработки концепции до ее замены новым продуктом. Спецификация была опубликована в 2013 году [18], руководство пользователю – в 2014 г. [19].

Целью языка является замена языков-предшественников, таких как UML и SysML, которые, по мнению разработчиков LML, излишне усложняют процесс разработки системы. SysML – это в основном конструкции и имеет ограниченную онтологию, в то время как DoDAF MetaModel 2.0 (DM2) имеет только онтологию. Вместо этого LML упрощает как конструкции, так и онтологию, чтобы сделать их более полными, но и более удобными в использовании.

LML объединяет логические конструкции с онтологией для сбора информации. Существует только 12 первичных классов сущностей (primary entity classes). Устраняется один из недостатков SysML – использование объектно-ориентированного подхода. SysML был разработан по методологии системного мышления программистов, однако, следует заметить, никакая другая дисциплина за все время жизненного цикла системы не пользуется объектно-ориентированным подходом.

LML – это новый подход к анализу, планированию, определению, проектированию, созданию и обслуживанию современных систем (рис. 9). LML фокусируется на шести целях:

1. Чтобы было легко понять
2. Легко расширить
3. Обеспечить поддержку как функционального, так и объектно-ориентированного подхода в рамках одного проекта
4. Быть языком, который может быть понят большинством заинтересованных сторон системы, а не только системными инженерами
5. Поддерживать систему в течение всего цикла жизни
6. Поддерживать как эволюционные, так и революционные изменения системных планов и проектов в течение всего срока службы системы

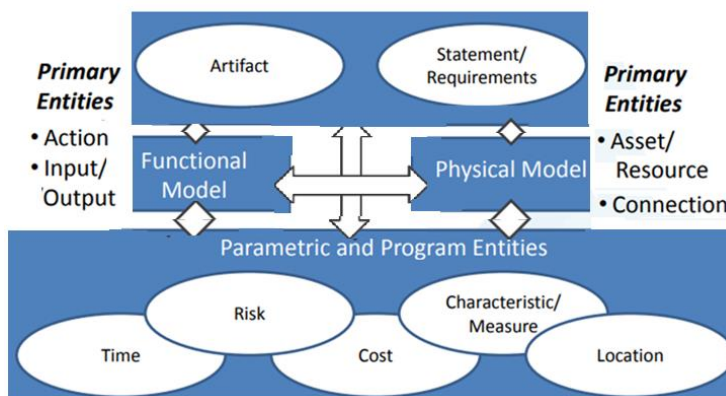


Рис. 9. Модели LML

Онтология. Онтологии предоставляют набор определенных терминов и отношений между терминами для сбора информации, описывающей физические, функциональные, рабочие и программные аспекты системы. Обычными способами описания таких онтологий являются три понятия: «Entity», «Relationship» и «Attribute» (ERA). Комплекс ERA часто используется для определения схем базы данных. LML расширяет схему ERA добавлением понятия "Attributes on Relationship" (атрибуты отношений), тем самым сокращая количество требуемых «сущностей» в ERA. В соответствии с основной мыслью языка LML эти четыре понятия: «Entity», «Relationship», «Attribute» и «Attribute on Relationship» представляют собой эквивалентные элементы английского языка: существительное, глагол, прилагательное и наречие (noun, verb, adjective и adverb).

Сущность: Entity (noun). Сущность (entity) определяется как нечто уникально идентифицируемое и может существовать сама по себе. В LML имеется всего 12 родительских сущностей (parent entities):

- Action
- Artifact
- Asset

Characteristic	Connection
Cost	
Decision	
Input/Output	Location
Risk	
Statement	
Time	

Определено несколько дочерних сущностей для учета информации, необходимой пользователям. У дочерних сущностей в свою очередь имеются свои атрибуты и отношения к родителям плюс дополнительные атрибуты и отношения, которые делают их уникальными. Среди дочерних сущностей от родительских сущностей назовем:

- Conduit (от Connection)
- Logical (от Connection)
- Measure (от Characteristic)
- Orbital (от Location) Physical (от Location)
- Requirement (от Statement)
- Resource (от Asset)
- Virtual (от Location)

Прилагательное: Attribute (adjective). Атрибуты используются наподобие прилагательных в обычном языке. Сущности (существительные) могут иметь имена, числа и атрибуты описания. Собственная характеристика или качество сущности является атрибутом. Каждый атрибут имеет имя, которое однозначно идентифицирует его внутри сущности. Имена атрибутов уникальны внутри сущности, но могут использоваться и в других сущностях. Имя дает обзор информации об атрибуте. Тип данных атрибута указывает на данные, связанные с этим атрибутом.

Отношение (глагол): Relationship (verb). Отношение работает так же, как глагол – соединяет существительные или, другими словами, сущности. Отношения дают простой способ увидеть, как сущности соединяются. Имена отношений уникальны по всей схеме LML.

Атрибуты отношений (наречие): Attributes on Relationships (adverb). Классическая модель ERA не имеет понятия «атрибуты отношений». Это есть существенное дополнение языковых средств. Аналогично тому, как атрибуты относятся к сущностям, «атрибут отношения» имеет имя, уникальное для его отношения, но не должно быть уникальным для других отношений.

С языком LML можно ознакомиться по документу [20]. Разработчики заверяют, что LML можно транслировать в UML, SysML, DoDAF (DM2) и другие языки. На рис. 10 показано соответствие между концептуальной моделью DM2 и LML Schema.

DM2 Element	LML Equivalent
Activity	Action
Capability	Action with "Capability" type
Condition	Characteristic with "Condition" type
Information/Data	Input/Output
Desired Effect	Statement with "Desired Effect" type
Guidance	Statement with "Guidance" type
Measure	Measure
Measure Type	Measure types
Location	Location
Project	Action with "Project" type
Resource	Asset
Skill	Characteristic with "Skill" type
Vision	Statement with "Vision" type

Рис. 10. Соответствие между концептуальной моделью DM2 и LML Schema

Уже открыто доступны первые средства разработки на языке LML – Innoslate [21]. Innoslate – это первый инструмент для внедрения нового языка моделирования Lifecycle Modeling Language (LML).

Innoslate объединяет программное обеспечение системного проектирования с требованиями управления требованиями, анализом требований и инструментами совместной работы. Все это содержится в едином решении. Innoslate, по замыслу, представляет собой будущий стандарт моделирования сложных систем по подходу Model-Based Systems Engineering (MBSE). Innoslate – это облачное веб-приложение, разработанное новой компанией SPEC Innovations

Выводы

Завершая обзор состояния информационных моделей цифровой экономики, отметим:

- За всей этой громадной работой стоит множество серьезных организаций. Прежде всего укажем на ассоциацию Open Group, членами которой в 1995 г. была опубликована первая версия языка проектирования TOGAF (The Open Group Architecture Framework), что и легло в основу DODAF, MODAF, NAF и DNDAF. В 2013 г. появилась модернизированная версия языка TOGAF 9.1.

- На основе ядра OMG UML был разработан Lifecycle Modeling Language (LML), который, как утверждают его создатели, понятен не только системным инженерам, но даже обычным акционерам, не имеющим математического образования. Итак, сделано немало, тем не менее, проблемы еще остались.

- Важнейшим направлением развития языковых средств является разработка языка онтологических описаний ODM (Ontology Definition MetaModel). Все термины в нем даются через описание конкретного применения, чтобы избежать синонимического смешивания понятий. На основе ODM разработана методика UPDM (на ее базе создаются новейшие версии DoDAF, MODAF и NAF).

В качестве заключения укажем, что 13 апреля 2017 г. Приказом ректора МГУ создан «Национальный центр компетенции в области цифровой экономики» с целью координации и проведения научно-исследовательских и методических работ. На наш взгляд, Факультет ВМК МГУ мог бы возглавить координацию работ по созданию единой информационной модели цифровой экономики.

Литература

1. Программа Цифровой Экономики <http://d-russia.ru/wp-content/uploads/2017/05/programmaCE.pdf> Retrieved: Aug, 2017
2. Шнепс-Шнеппе М.А., Сухомлин В.А., Намиот Д. Е. О глобальных информационных системах // International Journal of Open Information Technologies. — 2017. — Т. 5, № 4. — С. 55–62.
3. ТАКТИЧЕСКОЕ ЗВЕНО АСУВ «МАНЕВР»: 30 ЛЕТ СПУСТЯ http://otvaga2004.ru/kaleydoskop/kaleydoskop-c4/takticheskoe-zveno-asuv-manevr-30-let-spustya/attachment/otvaga2004_manevr_13/ Retrieved: Aug, 2017.
4. Эскизный проект Общегосударственной автоматизированной системы сбора и обработки информации для учета, планирования и управления народным хозяйством (ОГАС), 1980.
5. ISO/IEC 8631:1989 Information technology -- Program constructs and conventions for their representation
6. Andrey Terekhov, Timofey Bryksin, and Yuriy Litvinov. How to make visual modeling more attractive to software developers http://laser.inf.ethz.ch/2014/material/terekhov/terekhov_report.pdf Retrieved: Aug, 2017
7. Carrelli, D.J. Roche. CCITT Languages for SCP Switching Systems// IEEE Trans. COMM, Vol. 30, No 6, June 1982, pp. 1304-1309
8. J.A. Zachman (1987). "A Framework for Information Systems Architecture". In: IBM Systems Journal, vol 26, no 3.
9. J.F. Sowa and J. Zachman (1992). "Extending and Formalizing the Framework for Information Systems Architecture" In: IBM Systems Journal, Vol 31, no.3, 1992. p. 590-616.
10. С. Карпенко. Применение модели Захмана для проектирования ИТ-архитектуры предприятия. <http://www.management.com.ua/ims/ims177.html> Retrieved: Aug, 2017
11. The Chief Information Officers Council (1999). Federal Enterprise Architecture Framework, Version 1.1. September 1999
12. GAO. 2002. "Information Technology: Enterprise Architecture Use Across the Federal Government Can Be Improved," GAO-02-6, Government Accountability Office, Washington, DC.
13. S.B. Gaver (2010) "Why Doesn't the Federal Enterprise Architecture Work?," Technology Matters, McLean, VA.
14. A.L. Lederer, and V Gardiner (1992). "Strategic Information Systems Planning: The Method/1 Approach," Information Systems Management (9:3), pp. 13-20.
15. GAO. 2015. "DOD Business Systems Modernization: Additional Action Needed to Achieve Intended Outcomes," GAO-15-627, Government Accountability Office, Washington, DC.
16. S. Kotusev. Enterprise Architecture Frameworks: The Fad of the Century", British Computer Society (BCS), July 2016 <http://www.bcs.org/content/conWebDoc/56347> Retrieved: Aug, 2017
17. Steven H. Dam. Do We Still Need a DoD Architecture Framework? <http://mpcog.com/lib/download/asin=1502757621&type=stream> Retrieved: Aug, 2017
18. Lifecycle Modeling Language (LML) Specification 1.0. October 2013.
19. Steven H. Dam. DoD Architecture Framework 2.0 – A Guide to Applying System Engineering to Develop Integrated, Executable Architectures. Published by SPEC Innovations, 2014
20. Steven H. Dam, Warren K. Vaneman. A New Open Standard: Lifecycle Modeling Language (LML) a Language for Simple, Rapid Development, Operations and Support, January 25, 2014 http://cdn2.hubspot.net/hub/316256/file-493267217-pdf/LML_Overview_for_Lifecycle_Management_WG-Dam_and_Vaneman.pdf?t=1391103350000 Retrieved: Aug, 2017
21. Straightforward, powerful lifecycle management with Innoslate the complete product lifecycle management tool <http://www.systemsengineeringtool.com/innoslate/> Retrieved: Aug, 2017 Retrieved: Aug, 2017

References

1. Программа Цифровой Экономики <http://d-russia.ru/wp-content/uploads/2017/05/programmaCE.pdf> Retrieved: Aug, 2017
2. Sneps-Sneppе M., Sukhomlin V., Namiot D. On global information systems //International Journal of Open Information Technologies. – 2017. – Т. 5. – №. 4. – С. 55-62.
3. ТАКТИЧЕСКОЕ ЗВЕНО АСУВ «МАНЕВР»: 30 ЛЕТ СПУСТЯ <http://otvaga2004.ru/kaleydoskop/kaleydoskop-c4/takticheskoe-zveno-asuv-manevr-30-let-spustya/>

- zveno-asuv-manevr-30-let-spustya/attachment/otvaga2004_manevr_13/ Retrieved: Aug, 2017.
4. Jeskiznyj proekt Obshhegosudarstvennoj avtomatizirovannoj sistemy sbora i obrabotki informacii dlja ucheta, planirovanija i upravljenija narodnym hozjajstvom (OGAS), 1980.
 5. ISO/IEC 8631:1989 Information technology -- Program constructs and conventions for their representation
 6. Andrey Terekhov, Timofey Bryksin, and Yurii Litvinov. How to make visual modeling more attractive to software developers http://laser.inf.ethz.ch/2014/material/terekhov/terekhov_report.pdf Retrieved: Aug, 2017
 7. Carrelli, D.J. Roche. CCITT Languages for SCP Switching Systems// IEEE Trans. COMM, Vol. 30, No 6, June 1982, pp. 1304-1309
 8. J.A. Zachman (1987). "A Framework for Information Systems Architecture". In: IBM Systems Journal, vol 26, no 3.
 9. J.F. Sowa and J. Zachman (1992). "Extending and Formalizing the Framework for Information Systems Architecture" In: IBM Systems Journal, Vol 31, no.3, 1992. p. 590-616.
 10. S. Karpenko. Primenenie modeli Zahmana dlja proektirovanija IT-arhitektury predpriyatija. <http://www.management.com.ua/ims/ims177.html> Retrieved: Aug, 2017
 11. The Chief Information Officers Council (1999). Federal Enterprise Architecture Framework, Version 1.1. September 1999
 12. GAO. 2002. "Information Technology: Enterprise Architecture Use Across the Federal Government Can Be Improved," GAO-02-6, Government Accountability Office, Washington, DC.
 13. S.B. Gaver (2010) "Why Doesn't the Federal Enterprise Architecture Work?," Technology Matters, McLean, VA.
 14. A.L. Lederer, and V Gardiner (1992). "Strategic Information Systems Planning: The Method/1 Approach," Information Systems Management (9:3), pp. 13-20.
 15. GAO. 2015. "DOD Business Systems Modernization: Additional Action Needed to Achieve Intended Outcomes," GAO-15-627, Government Accountability Office, Washington, DC.
 16. S. Kotusev. Enterprise Architecture Frameworks: The Fad of the Century", British Computer Society (BCS), July 2016 <http://www.bcs.org/content/conWebDoc/56347> Retrieved: Aug, 2017
 17. Steven H. Dam. Do We Still Need a DoD Architecture Framework? <http://mpcog.com/lib/download/asin=1502757621&type=stream> Retrieved: Aug, 2017
 18. Lifecycle Modeling Language (LML) Specification 1.0. October 2013.
 19. Steven H. Dam. DoD Architecture Framework 2.0 – A Guide to Applying System Engineering to Develop Integrated, Executable Architectures. Published by SPEC Innovations, 2014
 20. Steven H. Dam, Warren K. Vaneman. A New Open Standard: Lifecycle Modeling Language (LML) a Language for Simple, Rapid Development, Operations and Support, January 25, 2014 http://cdn2.hubspot.net/hub/316256/file-493267217-pdf/LML_Overview_for_Lifecycle_Management_WG-Dam_and_Vaneman.pdf?t=1391103350000 Retrieved: Aug, 2017
 21. Straightforward, powerful lifecycle management with Innoslate the complete product lifecycle management tool <http://www.systemsengineeringtool.com/innoslate/> Retrieved: Aug, 2017 Retrieved: Aug, 2017.

Об авторах:

Шнепс-Шнеппе Манфред Александрович, доктор технических наук, профессор, генеральный директор компании ЦКБ-АбаваНет, sneps@mail.ru

Сухомлин Владимир Александрович, доктор технических наук, профессор, заведующий лабораторией открытых информационных технологий факультета вычислительной математики и кибернетики, Московский государственный университет имени М.В. Ломоносова; ведущий научный сотрудник, Федеральный исследовательский центр «Информатика и управление» Российской академии наук, Президент Фонда «Лига интернет-медиа», sukhomlin@mail.ru

Намиот Дмитрий Евгеньевич, кандидат физико-математических наук, старший научный сотрудник факультета вычислительной математики и кибернетики, Московский государственный университет имени М.В. Ломоносова, dnamiot@gmail.com

Note on the authors:

Sneps-Sneppe Manfred A., Doctor of Engineering Sciences, Full Professor, CEO of CKB-AbavaNet, sneps@mail.ru

Sukhomlin Vladimir A., doctor of technical sciences, professor, head of the Laboratory of Open Information Technologies, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University; leading researcher, Federal Research Center Computer Science and Control of the Russian Academy of Sciences; President of the Fund «League Internet-Media», sukhomlin@mail.ru

Namiot Dmitry E., PhD, Senior Researcher faculty of computational mathematics and Cybernetics, Lomonosov Moscow State University; dnamiot@gmail.com