

**Характеристический анализ современных подходов
к спецификации семантики программ**

**Бабенко Людмила Петровна
Институт программных систем
Национальной Академии Наук Украины**

Полезность обмена информационными ресурсами определяется способностью пользователя понять их смысл и назначение.

- **Анализируются подходы ряда проектов к спецификациям информационных ресурсов.**
- **Проводится анализ их общих и специфических концептуальных и проектных решений, наборов характеристик, применяемых ими для достижения понимаемости спецификаций.**

В ПОЛЕ ЗРЕНИЯ:

Object Management Group (OMG), курирующая такие успешные направления работ, как компонентное программирование (проект CORBA), развитие UML, ориентированные на сервисы архитектуры (SOA);

W3C - концерн, декларирующий своей целью интеграцию усилий по максимальному использованию потенциала веб-сообщества;

The Organization for the Advancement of Structured Information Standards (OASIS) - организация по продвижению структурированных информационных стандартов

НАПРАВЛЕНИЯ ИССЛЕДОВАНИЯ:

1. **Инженерия требований.** Цель - добиться взаимопонимания между заказчиком и исполнителем разработки, четко и однозначно зафиксировать **для разработчика**

- **КТО** его пользователи,
- **ЧТО** им надо по их мнению,
- **ЧТО ОНИ ХОТЯТ на самом деле,**

2. Описание бизнес-процессов. Цель спецификации – сделать понятными и регламентировать процессы, позволяющие вести определенный бизнес и успешно взаимодействовать с другими деловыми кругами, понимая взаимные возможности и обязательства. Современные бизнес-услуги, как правило, поддерживаются программно, более того, интенсивно развиваются идеи электронного бизнеса (eBusiness), электронного правительства, электронной администрации, в которых минимизирована роль человека.

Речь идет уже не только о понимании программ человеком, но и о взаимопонимании между программами.

3. Описание повторно-используемых ресурсов программной инженерии. Цель спецификации – понимание разработчиком тех готовых решений и ресурсов, которыми он может воспользоваться в своей разработке, и тем самым сократить затраты и время разработки

4. Описание ресурсов типа веб-сервисы. Цель спецификации – понимание тех услуг, которые обеспечивает веб-сервис, и правил доступа к ним.

Возникло специальное научное направление – Semantic Web –декларирующее цель всемирной системы распределенных веб-вычислений, что дало новый толчок к разработкам средств понимания программ, хотя и реализованных как веб-сервисы.

5. Описание знаний в системах управления знаниями.
Информационное общество как концепция новой ступени развития цивилизации Информационное общество как концепция новой ступени развития цивилизации, главным продуктом производства которой есть информация, приобретает все более широкое признание. Концепция предполагает создание глобального информационного пространства, которое обеспечивает эффективное информационное взаимодействие людей, их доступ к мировым информационным ресурсам для удовлетворения их информационных потребностей.

*При выборе объектов анализа мы включили в рассмотрение не только те проекты, что ставили своей прямой целью разработку средств описания (спецификации) программ, обеспечивающих понимание, но и те, реализация собственных целей которых в большей или меньшей мере способствовала **появлению новых идей и продвижению** в интересующем нас направлении..*

В качестве объектов анализа нами рассмотрены следующие проекты:

UML –стандарт де факто моделирования в программировании, широко используемый и непосредственно, и как **прародитель множества идей**, получивших развитие в других проектах.

Reusing Assets Specifications (RAS) – стандартный каркас спецификации готовых ресурсов разработки программного обеспечения, пригодных для повторного использования (assets). Разработан под эгидой OMG

Данный проект следует рассматривать как представительный портфель возможных аспектов спецификации и их характеристик. Отметим, что это единственный из попавших в поле зрения проектов, в котором предлагается рассматривать промежуточные продукты этапов жизненного цикла разработки программ как готовые ресурсы, заслуживающие самостоятельной аттестации для дальнейшего использования отдельно с продуктами других этапов разработки того же продукта. Такое решение представляется перспективным при смене среды исполнения

7. Business Processes Modeling Notation (BPMN) – проект языка моделирования процессов бизнеса, разработанный под эгидой OMG. Одна из целей проекта – предоставить средства для формулирования требований к программам в привычной для бизнесменов нотации. Нотация нацелена на понимание внутренних процедур бизнеса (своего или партнерского), обеспечение их взаимодействия, представление требований на разработку программ, обеспечивающих функционирование бизнес-процессов. При этом она имеет визуальное представление, ориентированное на удобства человека, хотя степень ее формализации позволяет относительно простое отображение на языки, непосредственно транслируемые в исполнительский код (executable language), такие, как BPEL4.

Web Services Description Language (WSDL) – проект, имеющий статус рекомендаций для использования концерна W3C . Хотя название свидетельствует о намерениях его создателей описывать веб-сервисы, удалось им лишь создать язык для описания их интерфейсов.

В ряде рассматриваемых далее проектов WSDL используется для описания элементов интерфейсов (BPMN, UDDI, ebXML, WSMO). В то же время в ряде проектов, предлагаемых для рассмотрения W3C, содержатся механизмы для дополнительного аннотирования конструкций WSDL семантическими комментариями, позволяющими понять смысл и функциональное назначение сервиса

Universal Description, Discovery and Integration (UDDI) – проект стандарта для спецификации сервисов, ориентированный на устройство хранилища (называемого регистром), в котором каждый может зарегистрировать свой бизнес и услуги (сервисы), которые предлагаются, описывая их настолько понятно, чтобы будущие пользователи могли их обнаружить и интегрировать в свои проекты. Заметим, что под термином «бизнес» в контексте UDDI можно понимать любую предметную область. Имеются сведения об успешных реализациях регистров сервисов для ряда предметных областей.

XML для электронного бизнеса (ebXML) - совместный проект OASIS и Центра ООН по торговле и электронному бизнесу, назначение которого - определение среды для обмена коммерческой информацией для глобального электронного бизнеса, в котором любые два бизнес-партнера могут интегрировать свои бизнес-процессы, используя сведения о предоставляемых каждым из них услугах. Эти сведения находятся в хранилище, называемом репозиторием, а описание того, как устроены эти сведения – в так называемом регистре – хранилище спецификаций. использующем стандартизованные и расширяемые метаданные. Используется в ряде специализированных регистров, как, например, фирмой IBM при создании регистра медицинских записей, фирмой SUN Microsystems при создании регистра сервисно-ориентированных архитектур и ряда других.

WEB Services Modeling Ontology (WSMO) - предложения для использования W3C в проекте Semantic Web Services, нацеленном на поддержку всемирной системы распределенных веб-вычислений как эффективной инфраструктуры для электронного бизнеса, где смысл и назначение сервисов и правила их использования понятны не только человеку, но и взаимодействующим программам, способным находить и использовать нужные веб-сервисы без участия человека.

Semantic Annotations for WSDL and XML Schema (SAWSDL) рекомендации W3C по дополнительному аннотированию конструкций WSDL семантическими комментариями, позволяющими понять смысл и функциональное назначение сервиса).

Web Service Semantics (WSDL-S) – предложения для обсуждения W3C по расширению описаний операций веб-сервиса в WSDL семантическими комментариями, связывающими функциональное назначение операций сервиса с соответствующей ему семантической моделью.

**Сходные концептуальные и проектные решения,
присущие большинству из рассматриваемых
проектов.**

- **1) Наличие определенного хранилища информационных ресурсов, при этом, предлагается раздельное хранение описания объекта спецификации (так называемых метаданных) и собственно его представления (т.н. контента), подобно тому, как в традиционных библиотеках раздельно хранятся каталоги и книги, причем в электронных хранилищах, как правило, обеспечивается прямой доступ из единицы каталога спецификаций к объекту спецификации.**

2) **Аспектный подход к описанию.** Целесообразность структурирования описаний программ отдельными категориями или аспектами осознана еще в ранних языках программирования, где отдельными средствами описывались **аспект данных** и **аспект их преобразования**, то есть алгоритмический или поведенческий.

В UML разделение описания на аспекты получило дальнейшее развитие – к аспектам **данных** (диаграмма *классов*) и **алгоритмов их обработки** (диаграммы *состояний и действий*) добавлены аспекты **взаимодействия объектов программы друг с другом** и с **внешними по отношению к программе объектами, называемыми актерами** (диаграммы *последовательности и взаимодействия* на основе посылки сообщений), Кроме того, диаграммы *компонент, пакетирования и размещения* отражали **аспекты среды реализации** программы.

В проекте Reusing Assets Specifications представлен наиболее широкий набор возможных аспектов спецификации и их характеристик. . Выделяется **базовое ядро спецификации**, общее для всех типов ресурсов на всех стадиях разработки. В него входят такие аспекты: **идентификация ресурса, контекст** (стадия моделирования: модель бизнеса, модель требований или проектирования, компонента кода, модель теста, документация), **использование** (характеризует виды работ (пользователя или инструментального средства), необходимые для использования готового ресурса, в том числе для настраивания его вариантных свойств), **решение** (артефакты, соответствующие готовому ресурсу, среди которых различают *главный тип*, например, *компонента кода*, и несколько вторичных типов, которые его объясняют, например соответствующие ему *требования*, *отдельные диаграммы UML*, *разделы инструкций для пользователя* и т.п.), так называемые *точки вариантности* (концептуальные позиции артефакта, в которые пользователь имеет возможность вносить изменения после передачи ему артефакта для использования), *зависимости* (определяет ссылки на артефакты, связанные с данным определенными отношениями, например, ассоциация, агрегация, наследование, зависимость по изменениям и другие.).

Для ресурсов конкретного типа (**продуктов конкретной деятельности на конкретном отрезке пути разработки**) предлагается строить так называемые **профили** спецификации. Профиль есть расширение базового ядра спецификаций (базового профиля) для конкретного типа ресурсов, свобода детализации которого остается за разработчиком нужного профиля. Таким образом, **понятие компоненты повторного использования, принятое для программ, распространяется на метаданные**: базовый профиль есть компонента повторного использования для определенного множества типов ресурсов. **Определены профили для программных компонент и веб-сервисов**

3) Аспект интерфейса выделен как самостоятельный в большинстве рассматриваемых нами проектов.

4) Аспект паспортных данных объекта спецификации представлен дескрипторами стандарта, называемого Дублинским ядром (Dublin Core, сокращенно Dc): название, ID (однозначный идентификатор ресурса, поддерживаемый определенным инструментом, как, например, *Uniform Resource Identify (URI)*[18], необязательные атрибуты: дата, состояние, версия, права доступа, неформальное описание, контекст использования. автор, почтовый адрес или e-mail, телефон, дата создания или срок годности, правила приобретения и тому подобные.

5) В метаданных прямо используются модели предметных областей (ПрО), к сфере которых относятся решаемые программой задачи. При этом в мире наблюдается тенденция накопления стандартизованных моделей значимых для человеческой практики ПрО (как, например, авиация, медицина, математика, химия, география и многие другие), фиксирующих состав и классификацию основных понятий, а также отношений между ними.

6) Большинство стандартизованных моделей ПрО представлены онтологиями. Онтология разрешает удерживать пользователя в максимально возможном пространстве предопределенных возможностей, смысл которых зафиксирован и понятен и разработчику, и заказчику. Очевидно, что пребывание в таком пространстве гарантирует обе стороны договора (который материализован в виде требований) от взаимных недоразумений.

7) Для представления спецификаций программ активно используются языки разметки XML, XML Schema, конкретизированные на требуемую проблемную область.

8) Для спецификации данных все рассмотренные проекты используют концепцию классов UML. При этом отличия состоят в ограничениях на типы данных, атрибутов (слотов) классов, типы отношений между классами, допустимую кратность значений атрибутов и экземпляров классов.

9) Аспектом, важным для жизнеспособности информационных ресурсов, и в том числе программ, является цена внесения изменений, необходимых в быстро меняющемся мире. Явное определение тех качеств ресурса, которые можно менять, и точек внесения изменений описывается аспектом, который будем называть *вариантностью (variability)*. Речь идет о предусмотренных на стадии проектирования возможностях, которые можно изменять локально в явно обозначенных позициях.

10) Как правило, общим для проектов является онтологический подход к спецификации аспекта функционального назначения ГОР в виде перечня дескрипторов (ключевых слов) с указанием классифицирующих их онтологий, в том числе стандартизованных, таких как онтологии продуктов и услуг, географических названий, математических, химических, медицинских терминов и др.

Смысл термина «понять программу» зависит от ответа на вопрос «**для чего**». Ответ может иметь варианты:

- **Найти** среди множества предложенных ту, которая предоставляет нужные клиенту функции, то есть **сравнить** спецификации предлагаемых программ с теми задачами, которые он хочет решать, **Понять, как** выбранную программу **использовать**: как обратиться к ней из другой программы, как развернуть ее в своей операционной среде, как настроить ее по нужным параметрам (если такая настройка предусмотрена в программе), **как изменить программу**, если она не полностью удовлетворяет потребности клиента.

При всей важности других аспектов определяющим при выборе программы является сопоставление задач клиента с функциональностью и интерфейсами программы. Исходя из этого, нами рассматривались только эти аспекты.

1. Аспект функциональности

Единица функциональности:

UML	Use Case
BPMN	Бизнес-процесс
WSDL	Операция интерфейса
UDDI	Сервис
ebXML	Сервисы объектов регистра
WSMO	Capability
RAS	Use Case

Смысл единицы функциональности и ее принадлежность к некоторой предметной области определяет так называемая ***таксономическая классификация***. Она состоит из пары характеристик, первая из которых задает систему классификации, а вторая – таксономическое значение в этой системе (аналог пары ***предметная область – дескриптор***, применяемой в информационных системах). В частности, система классификации может быть представлена онтологией, а таксономическое значение – узлом этой онтологии. Применяется в семантических дополнениях к WSDL (SAWSDL, WSDL-S), UDDI, ebXML, RAS, WSMO

В качестве дополнительных характеристик единицы функциональности предлагаются:

ее составляющие , их исполнители (программа или человек), статус (готовности) (BPMN), shared variables – переменные, значимые для поведения и используемые в логических формулах нижеследующих характеристик (WSMO), preconditions (пред-условия) определяют пространство информации до выполнения. postconditions (пост-условия) определяют пространство информации после выполнения, assumptions (предположения) определяют состояние мира до выполнения веб-сервиса, effect (эффект) определяет состояние мира после выполнения .Все условия, предположения и эффект выражаются аксиомами, представленными в языке формальной логики (WSMO)

Аспект интерфейса выделен как самостоятельный в большинстве рассматриваемых нами проектов. В нем определяются правила обращения к программе за получением объявленных в ней услуг – точки установления связи с услугой (например, сетевые адреса связывания), выполнимые в рамках услуги операции, их входные и/или выходные параметры, необходимость взаимодействия с другими программами, взаимодействие с внешними факторами и др.

ВРМН. Интерфейс бизнес-процесса описывается как цепь событий, определяющих выполнение бизнес-процесса. событие имеет ряд характеристик: **Тип события** определяет, на какой фазе процесса оно может происходить – стартовой, промежуточной или завершающей. **Характер влияния события на бизнес-процесс.** **Причина события** – внешний источник события или внутренняя ситуация процесса. **В качестве стандартизованных причин событий в сфере бизнеса используются получение или отправка сообщения извне активности, таймер, заданное условие, сигнал от устройства, исключительная ситуация и тому подобные. Для каждой из перечисленных причин предусмотрены отдельные графические маркеры событий.**

WSDL. Интерфейс описывается на двух уровнях: абстрактном и конкретном. На абстрактном уровне (синтаксическом) для компоненты описания, называемой интерфейс (interface), указываются абстрактные операции, которые может выполнить сервис (под операцией понимается простой обмен сообщениями с клиентом). Для операций указываются типы сообщений и так называемый паттерн обмена сообщениями (характер обмена – ввод, вывод, или и то, и другое). Определяются также действия при исключительных ситуациях. На конкретном уровне (уровне реализации) элемент binding (связывания) определяет точки доступа к сервису (сетевые адреса и протоколы связывания), а также уникальное имя сервиса, позволяющее создавать однозначные ссылки на компоненты описания сервиса в соответствующих хранилищах.

UDDI. Для описание интерфейса используется WSDL.

ebXML. Для описание интерфейса используется WSDL.

WSMO. Кроме характеристик, предложенных WSDL и используемых в проекте посредством механизма нефункциональных свойств, вводятся две известные характеристики интерфейса

Хореография (choreography) – описывает взаимодействие веб-сервиса и его клиента. Клиентом может быть человек, другой веб-сервис или другое приложение. Концепция хореографии базируется на абстрактной машине состояний. Ее составляющими являются: **состояние (state)** (описываемое как множество явно указанных экземпляров понятий, отношений или функций и значений их атрибутов) и **переходы (guarded transitions)** (условие изменения состояния, заданное в форме аксиом WSML, а также требуемую модификацию состояния при его истинности).

Оркестровка (Orchestration) определяет последовательность и условия вызовов других сервисов, необходимых данному для реализации его функциональности. Эта характеристика также базируется на абстрактной машине состояний. Ее составляющими являются: состояние (state) (описываемое как множество явно указанных экземпляров понятий, отношений или функций и значений их атрибутов) и переходы (guarded transitions), но переход определяет условие вызова требуемого веб-сервиса и ссылку на используемый медиатор. Заметим, что приведенные выше характеристики используют специально заданную онтологию интерфейса, отражающую внешние факторы, взаимодействующие с веб-сервисом и характерные для них события.

