



# OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

## ПОДХОДЫ К ИНТЕГРАЦИИ ПРОГРАММНЫХ СИСТЕМ В ИНТЕЛЛЕКТУАЛЬНОМ ПОМОЩНИКЕ ПРОЕКТАНТА

Боргест Н.М., Коровин М.Д.

*Самарский государственный аэрокосмический университет имени академика С.П. Королева  
(национальный исследовательский университет), г. Самара, Россия*

**borgest@yandex.ru**

**maks.korovin@gmail.com**

В работе рассмотрены основные подходы к интеграции программных систем с точки зрения их применимости к разработке интеллектуального помощника проектанта, реализуемого в предметной области предварительного проектирования самолета. Показана возможность создания программного комплекса, решающего типовые задачи проектирования в автоматическом режиме.

**Ключевые слова:** интеллектуальный помощник проектанта; интеграция; метамодель; тезаурус; матрица проекта.

### ВВЕДЕНИЕ

Современный процесс проектирования базируется на основе применения множества программных средств (промышленные САПР, СУБД, конструкторы онтологий и др.), между которыми существуют сложные алгоритмы взаимодействия. Для развития и совершенствования программной системы, а также упрощения её взаимодействия с пользователем необходима выработка общих принципов интеграции и интерфейсного взаимодействия. Наиболее эффективным методом решения данной проблемы является интеграция программных средств, которая учитывает непрерывное развитие систем и максимальное приближение интерфейса к естественному для пользователей и разработчиков.

### 1. ПРОБЛЕМЫ ИНТЕГРАЦИИ ПРОГРАММНЫХ СИСТЕМ В ИНТЕЛЛЕКТУАЛЬНОМ ПОМОЩНИКЕ ПРОЕКТАНТА

Одним из ключевых условий эффективной работы интеллектуального помощника проектанта (ИПП) является обеспечение согласованной совместной работы всех его компонентов. Данная проблема решается интеграцией программных компонентов в единую среду. Можно выделить следующие проблемы интеграции систем:

- Ускорение процессов. В динамичных областях, где “изменчивость” является самой сутью

и природой системы, задача интеграции усугубляется и превращается в серьезную проблему.

- Распределенность. Задачи, решаемые при проектировании, становятся все более комплексными, появляется логическая, организационная и географическая рассредоточенность [Бейлезон, 2010].

- Гетерогенность. При проектировании сложных технических систем, например, самолёта, нет возможности придерживаться единой платформы и полностью совместимых инструментов, поэтому приходится учитывать и поддерживать особенности нескольких платформ и искать пути взаимодействия разнородных инструментов [Шемсединов, 2011].

- Наследственность. Невозможность полностью отказаться от устаревших технологий, старого аппаратного обеспечения, которые хотя и дают вполне приемлемые показатели по надежности и производительности, никак не способствуют интеграции [Лондон и др, 2005].

- Хаотичность. Не всегда есть возможность полностью формализовать, специфицировать и структурировать данные, и часть модели процесса проектирования остается “слабосвязанной”, не поддающейся или слабо поддающейся машинной обработке, анализу, индексации и пр.

- Интерактивность. Проектант, как потребитель информации, постоянно повышает свои ожидания о скорости реакции системы, быстродействии и оперативности доставки

информации. Большинство процессов стремятся к выполнению в реальном времени.

Рассматривая гетерогенность компонентов ИПП можно выделить следующие проблемы:

- Несовместимость форматов обмена данными, различие интерфейсов. Данная проблема обычно решается путем внедрения в систему модулей, предназначенных для обмена данными внутри системы (брокеров, конвертеров).
- Концептуальная разница между компонентами системы – в случае если разработчики разных систем приняли решения, допущения и предположения, концептуально не стыкующиеся между собой. Проблема концептуальной несовместимости может решаться добавлением дополнительного уровня абстракции, которое не противоречит обоим подходам. При этом осуществляется отделение деталей реализации конкретного множества функций от их способа использования.

## 2. ДВА ПОДХОДА К ИНТЕГРАЦИИ

В настоящий момент существует два основных подхода к интеграции:

- Построение централизованной структуры, где интегрируемые системы становятся подсистемами, а взаимодействие происходит через единую точку (единую сервисную шину). (рисунок 1) В этом случае модули-подсистемы осуществляют обмен информацией по стандартизованным алгоритмам, управляемым из единого центра контроля.
- Создание структуры с брокером (посредником, не являющимся центром). Системы остаются независимыми, брокер при этом обеспечивает взаимодействие между ними (рисунок 2).

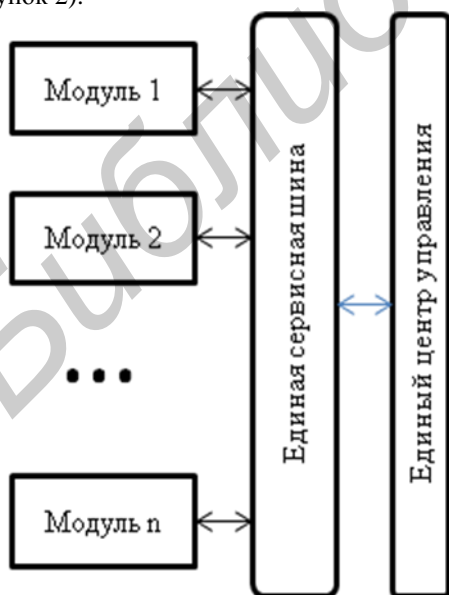


Рисунок 1 – Централизованная структура на основе единой сервисной шины

К преимуществам централизации с использованием единой сервисной шины

относятся:

- масштабируемость — возможность строить решения любого размера и информационной нагрузки;
- гибкость — возможность реализовывать и изменять интеграционные сценарии с минимальным вовлечением разработчиков системы;
- безопасность — встроенные средства аутентификации и авторизации обеспечивают контроль доступа к сервисам на уровне самой шины, избавляя разработчиков интеграционных сценариев от задач по реализации этих механизмов;
- использование открытых стандартов — позволяет уменьшить вовлеченность дорогостоящих специалистов по проприетарным технологиям;
- централизация средств контроля и администрирования — позволяет избежать «размытия» точки ответственности за интеграционные сценарии, обеспечить оперативное наблюдение и раннее оповещение в случае сбоев [Mason, 2011].



Рисунок 2 – Интеграция с использованием брокеров

Интеграция с использованием брокеров хорошо работает для небольших систем, где количество компонентов ограничено и не предполагается добавление дополнительных подсистем. Большим недостатком такого подхода является резкое возрастание необходимого количества брокеров. Так для системы из 3 компонентов достаточно 3 брокеров, а при добавлении всего 2 дополнительных модулей количество необходимых брокеров возрастает до 10. При этом каждый модуль уникален и требует индивидуального подхода. Данный недостаток делает интеграцию такого типа трудноприменимой для систем с большим количеством компонентов, где необходимы непосредственные связи между всеми компонентами. Стоит отметить, что для систем с относительно небольшим количеством модулей, которые соединены между собой последовательно и имеют ограниченное количество обратных связей (рисунок 3), метод интеграции по типу узел-узел все еще остается наиболее привлекательным, так как в этом случае затраты на построение интегральной структуры могут превысить выгоду от её внедрения.

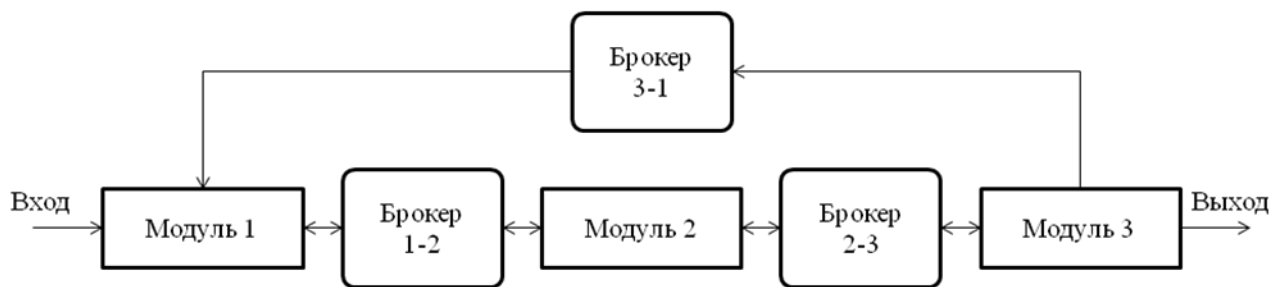


Рисунок 3 – Пример системы, для которой эффективна интеграция с использованием брокеров

### 3. ИНТЕГРАЦИЯ КОМПОНЕНТОВ В ИНТЕЛЛЕКТУАЛЬНОМ ПОМОЩНИКЕ ПРОЕКТАНТА

Интеллектуальный помощник проектанта – сложная система с развитой структурой. В силу того, что компонентами данной системы могут являться как самостоятельные программные продукты (third party applications), так и модули, специально разработанные для нее, не во всех случаях можно обеспечить семантическую совместимость всех компонентов без внедрения в систему дополнительных модулей – брокеров.

В силу особенностей предметной области ИПП разработка структур баз данных, где для всех сущностей описываются только функциональные свойства, сильно затруднена.

Сложноструктурированная предметная область приводит к огромному числу таблиц и связей, при этом динамичность предметной области требует постоянного реинжиниринга структуры базы данных. Подъем уровня абстракции решает задачу лишь частично, так как при переходе к абстрактной модели теряется специфика предметной области, а значит, что нужно хранить в одной базе данных два логически связанных слоя, один для структурных свойств модели, второй для функциональных. Роль связки должен играть мета-слой, определяющий параметры взаимно однозначного отображения одного абстрактного слоя в другом.

Таким образом, для эффективной работы ИПП необходимо построение метамодели – модели предметной области, описанной на нескольких уровнях абстракции, где каждый высший уровень полностью, целостно и непротиворечиво задает структуру данных, функциональность, отображение и связи нижних уровней [Tapscott, 2008].

Метамодель описывает не отдельную задачу, а широкий круг задач с выделением в этих задачах общих абстракций, правил обработки данных и управления процессом проектирования. К конкретной задаче метамодель адаптируется в процессе исполнения.

В процессе функционирования данные, логика и визуализация должны динамически (в реальном времени) интерпретироваться из метамодели предметной области, что позволяет модифицировать структуры хранения данных без внесения

изменений в программный код систем.

Важной является также возможность создавать несколько параллельно существующих представлений одной модели, что повышает наглядность результата.

Для обеспечения взаимодействия внутри системы необходимо обеспечивать глубокую семантическую интеграцию между компонентами [Голенков и др., 2012]. В ИПП это реализуется с помощью развитого тезауруса предметной области, который связан с матрицей проекта.

Цели использования тезауруса:

- обеспечения общей терминологии для предметной области с целью совместного использования всеми пользователями;
- получение точных и непротиворечивых определений каждого термина;
- обеспечение задания семантики с помощью множества аксиом, которые автоматически позволяют получать ответ на множество вопросов о предметной области [Боргест, 2011].

Матрица проекта (МП) содержит в себе всё описание объекта, которое соответствует определенному этапу его проектирования [Боргест, 2011]. Структура МП контурно напоминает морфологическую таблицу, позволяющую вместить все возможные реализации проектируемого объекта. В момент начала работы проектанта с системой матрица пуста. Заполнение матрицы начинается с данных технического задания на проектирование. По мере заполнения МП система осуществляет запуск соответствующих модулей и, если вмешательство проектанта не требуется, происходит дозаполнение матрицы в автоматическом режиме (рисунок 4). Появление данных в матрице инициирует выполнение сценариев её заполнения сообразно текущей задаче. Когда в матрице появляется достаточно данных для выполнения какого-либо модуля, он начинает выполняться. Таким образом, выполнение независимых модулей может идти параллельно. Структура МП формируется динамически, то есть количество и длина строк МП являются переменной величиной, зависящей от степени детализированности текущей интерпретации метамодели. На рисунке 4 представлены несколько возможных последовательных состояний матрицы проекта в процессе её заполнения.

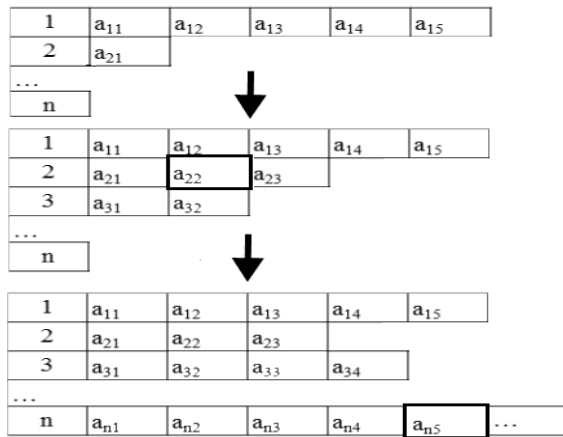
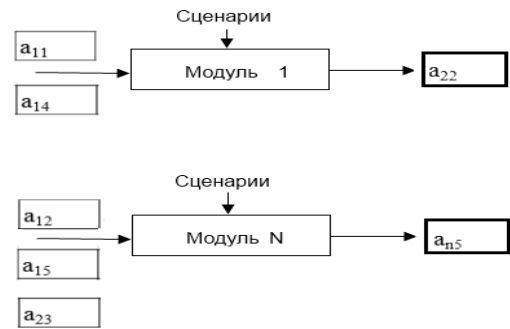


Рисунок 4 – Пример варианта заполнения матрицы проекта



## ЗАКЛЮЧЕНИЕ

Рассмотрен прототип структуры интеллектуального помощника проектанта, применяемого в предметной области предварительного проектирования самолета. Реализация подобной системы позволит автоматизировать типовые задачи проектирования, наглядно представляя проектанту весь ход выполнения работ над проектом.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Бейлезон, 2010] Бейлезон, О. Подходы к интеграции приложений Enterprise Service Bus, Компьютер пресс 6'2010.
- [Боргест, 2011] Боргест, Н.М. Модель интеллектуального интерфейса робота «конструктор самолета» / Н.М. Боргест // XI международная научная конференция имени Т.А. Таран «Интеллектуальный анализ информации ИАИ-2011», Киев, Просвіта, 2011, с. 297-302.
- [Боргест, 2011] Боргест, Н.М. Онтология проектирования самолета/ Н.М. Боргест // Искусственный интеллект, 2011, №4. с.260-265.
- [Голенков и др., 2012] Голенков, В.В. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования// Голенков В.В., Гулякина Н.А. // Материалы международной научно-технической конференции OSTIS-2012, Минск, БГУИР, 2012 - с.23-52
- [Лондон и др, 2011] Лондон Дж., Лондон К. Управление информационными системами. – 7-е изд., сер. «Классика МВА» / Пер. с англ. под ред. Д.Р. Трутнева – СПб.: Питер; 2005 – 912 с.
- [Шемседдинов, 2011] Шемседдинов Т. Meta Modelling, 2011 <http://blog.meta-systems.com.ua/> 02.12.2012
- [Mason, 2011] Ross Mason Understanding Enterprise Application Integration - The Benefits of ESB for EAI, 2011
- [Tapscott, 2008] Don Tapscott Grown Up Digital: How the net generation is changing your world, 2008

## APPROACHES TO THE INTEGRATION OF SOFTWARE SYSTEMS IN THE INTELLECTUAL DESIGNER ASSISTANT

Borgest N.M., Korovin M.D.

Samara State aerospace University, Samara,  
Russian Federation  
[borgest@yandex.ru](mailto:borgest@yandex.ru)

[maks.korovin@gmail.com](mailto:maks.korovin@gmail.com)

The paper describes the main approaches to the integration of software systems in terms of their applicability to the usage in the intellectual designer assistant, that is implemented in the preliminary design of the aircraft domain. The possibility to create a software package capable of solving typical complex designing problems in automatic mode is shown.

## INTRODUCTION

The modern design process involves a large number of various software systems (CAD, DBMS, etc). The interactions between those systems grow more and more complicated. To increase the efficiency of the process a unifying approach is proposed.

## MAIN PART

To solve the problem of the creating a unified design software complex we suggest using the following proposals:

- Module structure, that allows to add and remove components without the need to reengineer the whole system.
- Usage of the meta-model as a basis for creating different models based on various interpretations of the meta-model.
- Deep semantic integration between the components of the system based on a complex thesaurus of the subject area.
- Matrix of the project as a storage for the design variables of the project. The structure of the matrix of the project dynamically changes to represent the detail level of the current meta-model interpretation.

## CONCLUSION

The paper presents an approach to create a prototype of an intellectual designer assistant, applied to the subject area of the preliminary design of an aircraft. The realization of such a system will allow automating of typical designing tasks.