

Применение подхода открытых систем в информационных системах библиотек

Application of an Open-System Approach in Information Library Systems

Уживання принципу відкритих систем в інформаційних системах бібліотек

Рустам Фарук Халабия

*Московский государственный университет приборостроения и информатики,
Москва, Россия*

Rustam Khalabiya

Moscow State University for Instrument Making and Information Science, Moscow, Russia

Рустам Фарук Халабія

Московський державний університет приладобудування та інформатики, Москва, Росія

Рассматривается подход открытых систем для информатизации библиотечных процессов. В докладе освещен вопрос понятия технологии открытых систем и свойства таких систем. Приведены преимущества и аргументы данного подхода. Выделены перспективные направления развития этого подхода в информационных системах библиотек.

In this paper, the author examines the open-system approach to the informatization of library processes and explains the notion and characteristics of the open-system technology. He describes the strong points and arguments in favor of this approach. He outlines the prospective trends of the development of this approach in the information library systems.

Розглянуто підхід відкритих систем для інформатизації бібліотечних процесів. У доповіді висвітлено питання поняття технології відкритих систем і властивості таких систем. Наведено переваги та аргументи цього підходу. Виокремлено перспективні напрямки розвитку цього підходу в інформаційних системах бібліотек.

Перед библиотеками всегда стояли и будут стоять задачи информатизации внутрибиблиотечных процессов. Тем более что они являются бесконечными, всегда планируются и разделяются, как правило, на несколько этапов.

Кроме того, финансирование такой модернизации идет из государственного бюджета и оно очень низкое, особенно в условиях международного финансового кризиса. Компьютерное оборудование и программное обеспечение постоянно совершенствуется либо, что еще хуже устаревает быстрыми темпами.

Применение подхода открытых систем в настоящее время является основной тенденцией в области информационных технологий и средств вычислительной техники, поддерживающих эти области. Он может разрешить многие проблемы развития и модернизации процессов информатизации во многих областях народного хозяйства, в том числе и в библиотечной деятельности.

Под открытыми системами будем понимать системы, которые состоят из компонентов, взаимодействующих друг с другом через согласованный набор международных стандартов информационных технологий и профилей функциональных стандартов, которые специфицируют интерфейсы, службы и поддерживающие форматы, чтобы обеспечить интероперабельность и мобильность приложений, данных и персонала.

Общие свойства открытых систем обычно формируются следующим образом:

- **расширяемость/масштабируемость** (*extensibility/scalability*);
- **мобильность (переносимость)** (*portability*);
- **интероперабельность** (способность к взаимодействию с другими системами) (*interoperability*);
- **дружественность к пользователю, в т. ч. – легкая управляемость** (*driveability*).

Эти свойства, взятые по отдельности, были присущи и предыдущим поколениям информационных систем и средств вычислительной техники. Новый взгляд на открытые системы определяет

ся тем, что эти черты рассматриваются в совокупности, как взаимосвязанные, и реализуются в комплексе.

Подход открытых систем пользуется успехом только потому, что обеспечивает следующие преимущества:

- новые возможности сохранения сделанных вложений благодаря свойствам эволюции, постепенного развития функций систем, замены отдельных компонентов без перестройки всей системы;
- освобождение от зависимости от одного поставщика аппаратных или программных средств, возможность выбора продуктов из предложенных на рынке при условии соблюдения поставщиком соответствующих стандартов открытых систем;
- дружелюбность среды, в которой работает пользователь, мобильность персонала в процессе эволюции системы;
- возможность использования информационных ресурсов, имеющихся в других системах (организациях);
- возможность использования разных аппаратных платформ;
- возможность совместного использования прикладных программ, реализованных в разных операционных системах;
- возможности использования готовых программных продуктов и информационных ресурсов;
- новые возможности разделения труда, благодаря повторному использованию программ (*reusability*);
- развитые инструментальные среды и системы программирования;
- возможности модульной организации программных комплексов благодаря стандартизации программных интерфейсов.

Это последнее свойство открытых систем позволяет пересмотреть традиционно сложившееся дублирование функций в разных программных продуктах, из-за чего системы, интегрирующие эти продукты, непомерно разрастаются по объему, теряют эффективность. Известно, что в той же области обработки данных и текстов многие продукты, предлагаемые на рынке (текстовые редакторы, настольные издательства, электронные таблицы, системы управления базами данных) по ряду функций дублируют друг друга, а иногда и подменяют функции операционных систем. Кроме того, замечено, что в каждой новой версии этих продуктов размеры их увеличиваются на 15%.

Библиотечные системы преобразуются в распределенные системы, содержащие несколько рабочих мест на персональных компьютерах и серверов в локальной сети, избыточность программных кодов из-за дублирования возрастает многократно. Идеология и стандарты открытых систем позволяют по-новому взглянуть на распределение функций между программными компонентами систем и значительно повысить тем самым эффективность. Частично этот подход обеспечивает компенсацию затрат ресурсов, которые приходится платить за преимущества открытых систем относительно закрытых систем, ресурсы которых в точности соответствуют задаче, решаемой системой.

В связи с применением подхода открытых систем весьма перспективным направлением представляется объектно-ориентированный стиль проектирования и программирования.

Объектно-ориентированное программирование – это относительно новый подход к разработке программных систем. Этот подход строится на следующих основных принципах:

- данные и процедуры объединяются в программные объекты;
- для связи объектов используется механизм послышки сообщения;
- объекты с похожими свойствами объединяются в классы;
- объекты наследуют свойства других объектов через иерархию классов.

Объектно-ориентированные системы обладают следующими основными свойствами:

1. Инкапсуляция (скрытие реализации) – данные и процедуры объекта скрываются от внешнего пользователя, и связь с объектом ограничивается набором сообщений, которые «понимает» объект.
2. Полиморфизм (многозначность сообщений) – одинаковые сообщения по-разному понимаются разными объектами, в зависимости от их класса.

3. Динамическое (позднее) связывание – значение имени (область памяти для данных или текст программы для процедур) становится известным только во время выполнения программы.
4. Абстрактные типы данных – объединение данных и операций для описания новых типов, позволяющие использовать новые типы наравне с уже существующими.
5. Наследование – позволяет при создании новых объектов использовать свойства уже существующих объектов, описывая заново только те свойства, которые отличаются.

Заметим, что основные свойства открытых систем хорошо поддерживаются объектно-ориентированным подходом к реализации системы.

Рассмотрим отдельные аспекты этой поддержки.

1. Мобильность

Инкапсуляция позволяет хорошо скрыть машинно-зависимые части системы, которые должны быть реализованы заново при переходе на другую платформу. При этом гарантируется, что остальная часть системы не потребует изменений.

При реализации новых машинно-зависимых частей многое может быть взято из уже существующей системы благодаря механизму наследования.

2. Расширяемость

Наследование позволяет сэкономить значительные средства при расширении системы, поскольку многое не нужно создавать заново, а некоторые новые компоненты можно получить, лишь слегка изменив старые. Кроме повторного использования, увеличивается также надежность, поскольку используются уже отлаженные компоненты. Возможность конструирования абстрактных типов данных для создания новых средств – обеспечивается самим понятием класса, объединяющего похожие объекты с одинаковым набором операций.

3. Интероперабельность

Способность системы взаимодействовать с другими системами хорошо поддерживается принципом посылки сообщения и соответствующими понятиями полиморфизма и динамического связывания. В сообщении объекту (возможно удаленному) передается имя действия, которое должно быть им выполнено, и некоторые дополнительные аргументы сообщения. Как это действие выполнять – знает и решает только сам объект – получатель сообщения. От него только требуется выдать в ответ результат. Совершенно очевидно, что разные объекты будут по-разному реагировать на одинаковые сообщения (полиморфизм). Кроме того, очень удобно выбирать способ реализации в последний момент – при ответе на сообщение, в зависимости от текущего состояния системы (динамическое связывание). Для того, чтобы разные системы могли обмениваться сообщениями, необходима либо единая трактовка всех типов данных, в том числе абстрактных, либо индивидуальная процедура преобразования сообщения для каждой пары неодинаковых взаимодействующих систем. Простота понятия абстрактных типов данных в объектно-ориентированных системах существенно облегчает разработку такой процедуры.

4. Дружественность

Удобство взаимодействия человека с системой требует от последней наличия всех трех вышеуказанных качеств. Мобильность необходима ввиду быстрой смены старых и появления новых устройств, в частности, средств мультимедиа. Расширяемость требуется для разработки программной поддержки новых парадигм общения человека с машиной. Интероперабельность просто рассматривает человека как другую систему, с которой открытая система должна уметь взаимодействовать.

Интеграция компонентов в открытой системе должна следовать профилям стандартов на интерфейсы этих компонент.

Профиль составляют набор согласованных стандартов интерфейсов компонент на каждом уровне системы и обеспечивают их совместимость.

Для определенности рассмотрения интерфейсов компонент и проведения необходимых анализов их реализуемости можно использовать модель среды открытых систем. Эта модель, как правило, должна содержать пять групп компонентов:

- управление (*Management*) – функции системной администрации, безопасности, управления ресурсами, конфигурацией, сетевое управление;
- пользовательский интерфейс (*User Interface*) – интерфейс пользователя с прикладными программами и со средой разработки приложений;
- системные интерфейсы для программ (*Service Interface for Programs*) – интерфейсы между прикладными программами и между прикладными программами и операционной системой, например API (*Application Programs Interface*);
- форматы информации и данных;
- интерфейсы коммуникаций.
- Аппаратная часть открытых систем ориентирована на:
 - среду рабочих станций,
 - среду серверов процессов,
 - среду серверов данных,
 - среду транзакций,
 - среду реального времени,
 - среду суперкомпьютеров.

Кроме указанного набора профилей по классам аппаратно-программных средств существует необходимость формирования вертикальных профилей открытых систем, ориентированных на проблемно-ориентированную библиотечную область.

В качестве таких первоочередных задач применения открытых систем в библиотеках можно назвать:

- информационные системы (системы информационного обслуживания) с удаленным доступом к ресурсам,
- системы автоматизации библиотек различных типов,
- системы автоматизации научных исследований в библиотеках,
- системы передачи данных.

Подход открытых систем обеспечивает слишком много преимуществ, чтобы можно было игнорировать библиотечным сообществом.

Для реализации этого подхода:

Во-первых, необходимо выполнить ряд научных проектов, связанных с анализом реализации международных стандартов в наших условиях, выбором и разработкой профилей стандартов открытых систем по областям их применения, как технической основы информационной инфраструктуры библиотек.

Во-вторых, требуется выработать и согласовать стандарты интерфейсов для разработки или приобретения аппаратных и программных средств.