ТЕХНОЛОГИЯ ВЫСОКОУРОВНЕВОЙ РАЗРАБОТКИ КОМПОЗИТНЫХ ПРИЛОЖЕНИЙ В РАМКАХ КОНЦЕПЦИИ ВИРТУАЛЬНЫХ МОДЕЛИРУЮЩИХ ОБЪЕКТОВ

П.А. Смирнов, С.В. Ковальчук

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики
smirnp@niuitmo.ru

Рассмотрены существующие проблемы workflow-композиции, предложен метод их решения на основе виртуальных моделирующих объектов с семантическим описанием, а также описана реализация инструментальной среды, воплощающей предложенное решение.

Введение

Концепция потока задач (Workflow, WF [1]) позволяет разрабатывать композитные приложения с применением различных программно-аппаратных ресурсов. Использование распределенных систем (грид, кластер и т.д.) открывает возможности по решению комплексных научных задач, требующих знаний из различных предметных областей, обработки больших объемов данных, а также использования разнообразных методов и средств компьютерного моделирования. При этом платформа распределенной системы в общем случае должна обеспечивать управление процессом исполнения отдельных сервисов с целью обеспечения эффективного использования ресурсов и минимизации общего времени решения задач. Благодаря композиции информационных и вычислительных (как программных, так и аппаратных) ресурсов становится возможным взаимодействие географически удаленных друг от друга групп ученых и, как следствие, построение виртуальных лабораторий для организации междисциплинарных исследований – в соответствии с принципами электронной науки eScience [2]. Одной из очевидных проблем, затрудняющих проведение такого рода исследований, является высокая сложность инструментов композиции, зачастую требующих от ученых знаний и навыков в области информационных технологий, а также данных о корректности применения тех или иных доступных в системе ресурсов.

1. Обзор уровней композиции

На сегодняшний день предложены несколько подходов [3] по автоматизации построения потоков задач (WF), в которые можно условно разделить на три уровня:

1. Композиция из набора доступных сервисов. Первый и самый низкий уровень, на котором пользователю предлагаются программные и аппаратные ресурсы в виде «как есть», т.е. без каких-либо унифицированных и абстрактных описаний. Основная проблема данного уровня – сложность композиции из-за многообразия разнородных ресурсов: отладка взаимодействий между сервисами, порядок вызова, маршруты передачи данных – все эти задачи ложатся на пользователя и требуют от него соответствующих знаний. Для композиции на данном уровне применяются два подхода – последовательный (interleaved) и монолитный (monolithic) [3], различающиеся степенью взаимодействия с сервисами. Последовательный подход обращается к сервисам прямо во время композиции, предлагая к работе только доступные и рабочие варианты. Монолит-

ный подход использует только имеющиеся описания сервисов, полагая, что все сервисы развернуты правильно, находятся в рабочем состоянии и запуск WF априори завершится успешно, а получаемый на данном этапе WF-сценарий называется конкретным (concrete workflow, CWF) и может быть запущен соответствующим (зависит от WF-языка) менеджером задач, например Condor DAGMan [4]. На сегодняшний день существуют инструменты для автоматической композиции на уровне сервисов, использующие базы знаний и технологии искусственного интеллекта (см., например, [5]).

2. Сервисы с абстрактным описанием. Второй уровень композиции дополняет предыдущий наличием некоторого формата описания сервисов и ресурсов, имеющихся в системе. Композиция на данном уровне идет с использованием поэтапного (staged) подхода [3]. Поэтапный подход обычно характеризуется разделением на логический и физический уровни. Логический уровень представляет собой композицию типов, взятых из описаний сервисов. Получаемый на логическом этапе WF-сценарий носит название абстрактного потока задач (abstract WF, AWF). На физическом этапе (как правило, после запуска WF на выполнение) AWF обрабатывается исполняющим (execution) механизмом системы управления workflow, например Pegasus [6], который автоматически распределяет задачи на доступные в системе ресурсы и вызывает конкретные экземпляры сервисов, т.е. происходит трансформация абстрактного WF в конкретный. Дальнейшая работа по исполнению происходит по аналогии с первым уровнем, описанным выше.

3. WF-шаблоны. Третий уровень подразумевает использование WF-шаблонов – заранее составленных и классифицированных частей WF, предлагающих одно или множество решений текущей задачи. Сам по себе шаблонный подход [3] подразумевает использование некоторой заготовленной заранее базы шаблонов, однако на практике шаблоны создаются пользователями либо с нуля, либо путём модификации уже существующих. Недостаток у пользователя знаний о смежных предметных областях вызывает трудности при попытке проведения междисциплинарных исследований, поэтому заблаговременное формирование базы шаблонов на основании экспертных знаний – вполне логичное требование шаблонного подхода. На уровне WF-шаблонов работают такие системы, как CAT/KANAL [6] и WINGS [7], в которых благодаря иерархической организации базы шаблонов предоставляется возможность полуавтоматической композиции.

2. Виртуальные моделирующие объекты

Обзор условно выделенных уровней в подходах к композиции WF указывает на несколько существующих на сегодняшний день проблем. Во-первых, это недостаток у пользователя знаний об особенностях доступных ему программных и вычислительных ресурсов, соответственно непонимание, какой ресурс лучше решает ту или иную задачу. Вторая проблема заключается в трудностях проведения междисциплинарных исследований, осложняющихся в случае отсутствия у пользователя достаточных знаний о соседних предметных областях. Для решения этих проблем нами предлагается *четвертый уровень*, основанный на динамической генерации WF-шаблонов на основе композиции виртуальных объектов. Под виртуальным моделирующим объектом (virtual simulation object, VSO) понимается некоторая совокупность расчетных моделей и характеристик, представляющая виртуальный прототип объекта реального мира. Виртуальный объект имеет определенную внутреннюю структуру, задекларированную в виде семантического описания (метаописания), представляющего собой формализованные знания предметной области. Такое описание делает возможным автоматическую композицию соответствующих объектов и моделей внутри них, что впоследствии определяет порядок взаимодействия и передачу данных между ними. Разработкой образов объектов за-

нимаются эксперты предметной области, а благодаря некоторому унифицированному формату структуры появляется возможность объединить объекты разных разработчиков в единый каталог. Каталог представляет собой распределенную базу знаний, доступную во время работы различным специалистам, что решает проблему проведения мультидисциплинарных исследований. К тому же использование экспертных оценок тех или иных методов решения задач дает возможность оценивания конечного композитного приложения по какому-либо критерию [8].

3. Программная реализация

Практической частью данной работы является разработка решения, воплощающего концепцию виртуальных объектов [9]. Функционально решение состоит из двух компонентов:

Конструктор (VSO construct) – инструментарий для работы конечного пользователя – специалиста, проводящего моделирующие расчетные исследования путём композиции виртуальных объектов из одной или нескольких предметных областей. Графический интерфейс максимально упрощает работу, делая процесс композиции наглядным и не требуя от пользователя никакой технической информации касательно вычислительных ресурсов. Для построения корректных вычислительных схем достаточно иметь лишь необходимые наборы входных данных и базовое понимание предмета. Система автоматически объединяет соответствующие параметры, оставляя пользователю возможность конфигурировать их значения вручную. В результате наполнения среды объектами, указания им свойств, организации взаимодействия друг с другом пользователь запускает построение сценария расчетных задач, а затем производит запуск построенного WF в распределенной исполнительной платформе CLAVIRE [10]. Генерация потока задач производится платформой с использованием специального проблемно-ориентированного языка EasyFlow [11] и представляет собой последовательность шагов и параметров для запуска расчетных пакетов. Процесс выполнения WF и результаты расчетов отражаются в интерфейсе CLAVIRE.

Редактор виртуальных объектов (VSO editor) – ориентирован на использование специалистом-экспертом предметной области, в задачи которого входит разработка, редактирование образов виртуальных объектов, декларирование их свойств, моделей, сценариев исполнения, привязки пакетов, параметров и т.д.,– результатом работы эксперта в редакторе является метаописание, автоматически сгенерированное в формате OWL и пригодное для чтения в любых известных OWL-редакторах (напр. Protege). Графический интерфейс редактора позволяет в полной мере создавать, редактировать, сохранять, скачивать, загружать файлы в формате OWL, а также работать с внутренней структурой объектов, а именно: задавать сценарии, варианты реализации, добавлять пакеты, привязывать переменные пакетов к значениям метапараметров и т.д. В конечном итоге, описания объектов попадают в каталог, который используется в работе конструктора. Взаимодействие компонентов инструментария VSO с системами платформы CLAVIRE представлено на рис. 1.

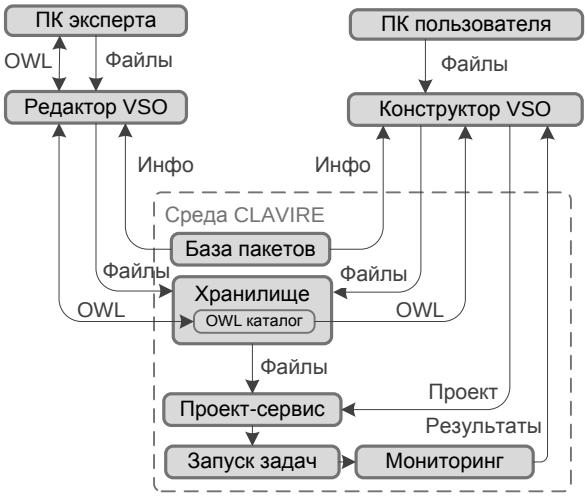

Рис. 1. Взаимодействие компонентов VSO с системами платформы CLAVIRE

4. Сценарий использования

Работа пользователя по композиции объектов происходит на самом верхнем – четвертом уровне. В конструкторе пользователю предлагается каталог объектов для выбора и добавления их в рабочую область. В рабочей области объекты представлены в виде графических элементов с некоторым набором свойств и внутренней структурой, содержащей наборы расчетных моделей, а также варианты их реализации (сценарии). Для каждой используемой модели пользователь должен выбрать сценарий и реализацию, после чего он получит доступ к параметрам реализующих её пакетов – необходимыми переменными, представленными отдельными графическими элементами. Эквивалентные параметры автоматически объединятся с параметрами других моделей, наглядно демонстрируя маршруты пересылки выходных данных одних моделей/объектов на вход другим. После построения требуемой конфигурации объектов пользователь запускает генерацию WF-сценария. Модель каждого объекта вычисляется рядом моделирующих операций (шагов), содержащих информацию о запускаемом пакете и требуемых входных параметрах. Работа автоматической системы (а не пользователя) по динамической генерации WF-шаблонов происходит по аналогии с третьим уровнем композиции, представленным выше. Следующий этап – запуск сгенерированного WF в распределенной среде (кластер, грид и т.д.) – происходит в интерфейсе CLAVIRE.

Заключение

Обозначенная концепция виртуальных объектов предлагает новый уровень разработки композитных приложений в распределенных средах. Переход на данный уровень позволит решить проблемы проведения междисциплинарных исследований, а также обеспечит пользователей дополнительной информацией касательно выбора оптимальных решений для поставленных задач. Разработанный базовый инструментарий позволяет оценить удобство композиции виртуальных объектов благодаря наглядному графическому интерфейсу и некоторым автоматическим функциям. В перспективе – внедрение критериев экспертных оценок эффективности тех или иных решений и реализация механизма автоматического выбора соответствующих требованиям вариантов.

Литература

1. Gil Y. From Data to Knowledge to Discoveries: Scientific Workflows and Artificial Intelligence // Scientific Programming. 2008. Vol. 17, Issue 3. P. 1–25.
2. Hey T., Tansley S., Tolle K. The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, Redmond Washington, 2009.
3. Agarwal V. et al. Understanding Approaches for Web Service Composition and Execution // IBM Research Report. 2007. – [http://domino.watson.ibm.com/library/cyberdig.nsf/papers/9E61D97ED1DDA2A28525733D00444D0D/$File/RI07005.pdf].
4. Directed Acyclic Graph Manager – [http://research.cs.wisc.edu/condor/dagman/].
5. Ozorhan E.K., Kuban E.K., Cicekli N.K. Automated composition of web services with the abductive event calculus // Information Sciences. 2010. №180. P. 3589-3613.
6. Kim J., Gil Y., Spraragen M. Principles For Interactive Acquisition And Validation Of Workflows // Journal of Experimental & Theoretical Artificial Intelligence. 2010. Vol. 22. P. 103-134.
7. Gil Y., Ratnakar V., Deelman E. Wings: Intelligent Workflow-Based Design of Computational Experiments // Intelligent Systems, IEEE. January 2010.
8. Ковальчук С.В., Маслов В.Г. Интеллектуальная поддержка процесса конструирования композитных приложений в распределенных проблемно-ориентированных средах // Известия вузов. Приборостроение. 2011. №10. С. 29–35.

9. Kovalchuk S.V., Smirnov P.A., Kosukhin S.S., Boukhanovsky A.V. Virtual Simulation Objects Concept as a Framework for System-Level Simulation // Proceedings of IEEE e-Sceince Conference. 2012. CD-ROM. ISBN 978-1-4673-4465-4.
10. Бухановский А.В., Васильев В.Н., Виноградов В.Ю., Смирнов Д.А., Сухоруков С. С., Яппаров Т.Г. CLAVIRE: перспективная технология облачных вычислений второго поколения // Изв. вузов. Приборостроение. 2011. Т. 54, № 10. С. 7–14.
11. Князьков К.В., Ларченко А.В. Предметно-ориентированные технологии разработки приложений в распределенных средах // Изв. вузов. Приборостроение. 2011. Т. 54, № 10. С. 40–49.