

Эволюционная разработка системы визуального планирования проектов на основе объектно-ориентированного каркаса

¹ А.С. Аничкин <anton.anichkin@ispras.ru>

^{1,2} С.В. Морозов <serg@ispras.ru>

^{1,3} В.А. Семенов <sem@ispras.ru>

^{1,2} О.А. Тарлапан <oleg@ispras.ru>

¹ Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25

² Московский государственный университет имени М.В. Ломоносова, 119991 ГСП-1, Москва, Ленинские горы

³ Московский физико-технический институт, 141700, Московская область, г. Долгопрудный, Институтский пер., 9

Аннотация. В статье описывается практический опыт разработки перспективной системы визуального планирования проектов на основе объектно-ориентированного каркаса. Используемый каркас представляет собой систему классов и интерфейсов, предназначенных для программной реализации моделей, методов и приложений теории расписаний. Благодаря наличию готовых компонентов для решения типовых задач, а также предусмотренным механизмам их конфигурирования и расширения, создание приложений осуществляется относительно просто. Применение каркаса позволило реализовать в целевой системе необходимый функционал проектного планирования, а также обеспечить его последующее развитие путем обобщения условий задач и расширения арсенала алгоритмов, применяемых для их решения.

Ключевые слова: теория расписаний; календарно-сетевое планирование; программная инженерия; объектно-ориентированное программирование.

DOI: 10.15514/ISPRAS-2017-29(5)-12

Для цитирования: Аничкин А.С., Морозов С.В., Семенов В.А., Тарлапан О.А. Эволюционная разработка системы визуального планирования проектов на основе объектно-ориентированного каркаса. Труды ИСП РАН, том 29, вып. 5, 2017 г., стр. 239-256. DOI: 10.15514/ISPRAS-2017-29(5)-12

1. Введение

Создание программных приложений теории расписаний представляет собой серьезную проблему, поскольку требует со стороны разработчиков как математической квалификации, необходимой для формализации прикладных задач и построения эффективных алгоритмов решения, так и знаний и опыта в области программной инженерии, необходимых для проектирования, реализации и интеграции сложных программных систем.

Универсальные математические библиотеки [1, 2, 3] позволяют решить некоторые типовые задачи теории расписаний известными алгоритмами, однако плохо приспособлены для прикладных задач, условия которых могут варьироваться в процессе эволюции целевого приложения. В подобных случаях вопросы выбора и реализации алгоритмов, релевантных вычислительной сложности решаемой прикладной задачи, становятся наиболее критичными [4]. С другой стороны, практика разработки программ заново для каждого приложения также является неприемлемой в силу значительных ресурсов, необходимых для реализации современных моделей и алгоритмов теории расписаний. Еще больших затрат требует конфигурирование и адаптация разработанных программ для согласованной работы в составе целевого приложения вместе с графическими и информационными компонентами.

Подход, предложенный в работе [5], предполагает создание и всестороннее применение единой программно-инструментальной среды для реализации моделей, методов и приложений теории расписаний. Данная среда сочетает в себе функции математической библиотеки и программного инструментария.

С одной стороны, такое сочетание предполагает наличие готовых к использованию программных компонентов для задания условий и решения типовых задач теории расписаний и, в частности, индустриально значимых задач высокой размерности в постановке проектного планирования Generally Constrained Project Scheduling Problem (GCPSP) [6, 7]. Поскольку многие задачи теории расписаний редуцируются к данной обобщенной постановке, среда предоставляет необходимые точные и приближенные средства решения.

С другой стороны, организация инструментальной среды в виде объектно-ориентированного каркаса обеспечивает повторное использование имеющихся компонентов при программной реализации новых моделей, методов и приложений теории расписаний при относительно низких затратах на доработку. При этом возможности развития, адаптации и конфигурации компонентов не препятствуют построению эффективных приложений, релевантных условиям и сложности решаемых прикладных задач.

Данные факторы предопределили выбор программно-инструментальной среды в качестве основного средства для построения системы визуального планирования проектов на основе ранее существовавшего приложения визуального моделирования Synchrono [8]. Функции исходного приложения, главным образом, ограничивались возможностями визуализации проектных

планов и календарно-сетевых графиков на диаграмме Ганта и в окнах просмотра трехмерных сцен. Также приложение предоставляло средства верификации графиков, выявления пространственно-временных конфликтов и генерации сопутствующих отчетов и видеоматериалов.

Принцип работы Synchro состоял в консолидации календарно-сетевых графиков, импортируемых из традиционных систем управления проектами, таких как Oracle Primavera, Microsoft Project, Asta Powerproject, и трехмерных моделей, подготовленных в популярных CAD-системах, таких как AutoCAD, Revit, Sketchup, Microstation. В результате подобной консолидации формируется единая пространственно-временная модель проекта, которая затем может использоваться для визуализации, анализа и верификации. При обнаружении пространственно-временных конфликтов календарно-сетевой график проекта может быть скорректирован средствами приложения или с использованием сторонних систем в результате экспорта и импорта проектных данных.

Отсутствие средств построения расписаний являлось принципиальным недостатком исходного приложения, поскольку любая коррекция календарно-сетевого графика при обнаружении конфликтов могла потенциально нарушить его согласованность. В этом случае требовалось экспортировать календарно-сетевой график в стороннее приложение соответствующей функциональности, строить в нем согласованное расписание, а затем импортировать график в Synchro. Поскольку такая последовательность не гарантировала отсутствие новых пространственно-временных конфликтов в результирующем графике, требовались многократные действия. Наличие развитых средств построения расписаний превратило бы Synchro в полноценную систему управления проектами, причем с возможностями многофакторного планирования и визуального анализа.

В статье описывается практический опыт эволюционной разработки перспективной системы визуального планирования проектов на основе существующего приложения Synchro с использованием программно-инструментальной среды для реализации моделей, методов и приложений теории расписаний, который может быть востребован при создании других приложений. В разделе 2 кратко рассматриваются общие вопросы организации программно-инструментальной среды в виде объектно-ориентированного каркаса и методологии построения приложений теории расписаний на его основе. Процесс разработки целевой системы визуального планирования проектов обсуждается в разделе 3. Некоторые результаты вычислительных экспериментов представлены в разделе 4, в котором проводится сравнение показателей эффективности построенной системы с популярными системами управления проектами. В заключении подводятся итоги проведенной работы и даются рекомендации по разработке и развитию приложений теории расписаний на основе каркаса.

2. Организация и назначение каркаса

Разработанный каркас представляет собой систему классов (в дальнейшем, учитывая его практическую реализацию на языке Си++, будем использовать принятые термины «класс», «конкретный класс», «абстрактный класс» и «интерфейс»). В организации каркаса (см. рисунок 1) выделим следующие группы или пакеты классов:

- пакет классов решателей (Solvers), реализующих общие алгоритмические схемы решения задач GCPSP (Schedulers), а также эвристики для поиска приближенных решений (Heuristics);
- пакет классов математических объектов (Mathematics), предназначенных для задания условий и получения результатов решения задач в обобщенной постановке GCPSP;
- пакет классов математической редукции (Reductions), обеспечивающих сведение прикладных задач составления расписаний к постановке GCPSP и соответствующую интерпретацию прикладных данных;
- пакет классов прикладных данных (Project Data), используемых для представления условий и результатов решения задач проектного планирования RCPSP в расширенных постановках.

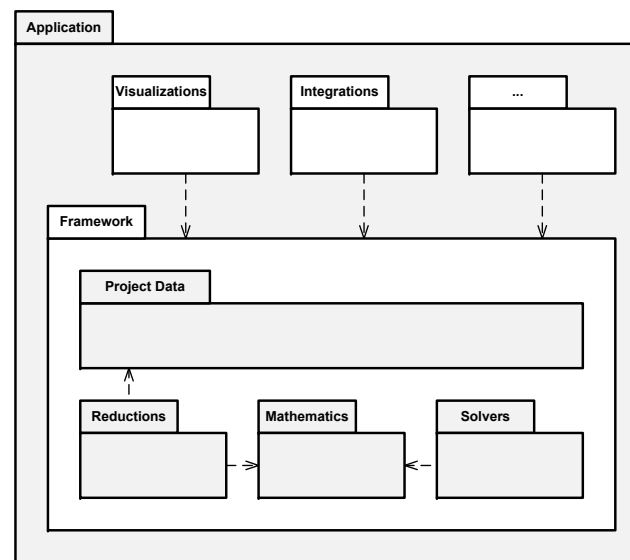


Рис. 1. Организация объектно-ориентированного каркаса для разработки приложений теории расписаний.

Fig. 1. Organization of the object-oriented framework for development of scheduling theory applications

Пакеты и классы каркаса подробно рассматриваются в [5]. Заметим, что часть из них реализуется в виде конкретных классов, допускающих непосредственное конструирование объектов. Другая часть представляет собой интерфейсы или абстрактные классы, которые по существу определяют точки расширения каркаса и позволяют предоставить их альтернативные реализации. Примечательно, что некоторые функции каркаса, в частности общие алгоритмические схемы, реализуются на уровне абстрактных классов без конкретизации частных условий решаемых задач и применяемых алгоритмов. Такой способ организации и реализации каркаса является рациональным с точки зрения повторного использования компонентов.

Сделаем некоторые комментарии, поясняющие представленную организацию и определяющие методологию построения целевых приложений на основе каркаса. Поскольку теория расписаний охватывает довольно много классов задач со своими алгоритмами, в состав каркаса включены, прежде всего, компоненты для решения обобщенных задач планирования в математически нейтральной постановке GCPSP, в которой определяются лишь тип целевой функции и вид алгебраических ограничений. Для того чтобы воспользоваться решателями, входящими в пакет Solvers, требуется предоставить собственные реализации данных абстракций, наследуя их от соответствующих интерфейсов пакета Mathematics. Тем не менее, поскольку многие задачи теории расписаний редуцируются к задачам проектного планирования RCPSP, более простым способом разработки приложений оказывается интерпретация условий исходных прикладных задач в терминах RCPSP, решение их имеющимися средствами и конвертация результатов в представление исходных задач. Пакет каркаса Project Data содержит необходимый набор классов для задания условий задач проектного планирования в расширенных постановках, что делает такую схему разработки приложений довольно простой. Более того, при создании систем проектного планирования и управления этот пакет определяет развитую модель данных, которая может быть положена в основу всей системы. В этом случае применимы и средства составления расписаний, поскольку пакет Reductions предоставляет необходимые реализации целевых функций и алгебраических ограничений, порождаемых постановками RCPSP.

Таким образом, разработка типового приложения с функциями проектного планирования может быть сведена к следующим работам:

- использование пакета Project Data для представления и хранения проектных данных, а также для задания условий задач планирования;
- конфигурирование классов решателей соответствующими целевыми функциями, ограничениями и эвристиками для эффективного решения задач планирования;
- разработка графического интерфейса пользователя целевого приложения для ввода, редактирования, визуализации проектных данных, в том числе с использованием текстовых отчетов, графиков,

диаграмм.

В частных случаях могут потребоваться дополнительные усилия для развития имеющихся пакетов:

- расширение пакета классов Project Data для представления специальных условий планирования и их сведения к расширенной постановке RCPSP;
- расширение пакета классов Reductions для приведения специальных условий планирования к обобщенной математической постановке GCPSP;
- расширение пакета классов Solvers для реализации новых алгоритмов и эвристик с учетом особенностей прикладных задач.

3. Разработка и развитие системы визуального планирования проектов

Разработанный каркас и связанная с ним методология разработки программных приложений теории расписаний были успешно апробированы в ходе построения системы визуального планирования проектов Synchro [8]. Система строилась на основе ранее существовавшего приложения визуального моделирования с основными функциями, перечисленными во введении. На рисунке 2 приведен снимок экрана, иллюстрирующий главное окно и основные элементы графического интерфейса пользователя системы Synchro, включая диаграмму Ганта и окна просмотра трехмерных сцен.

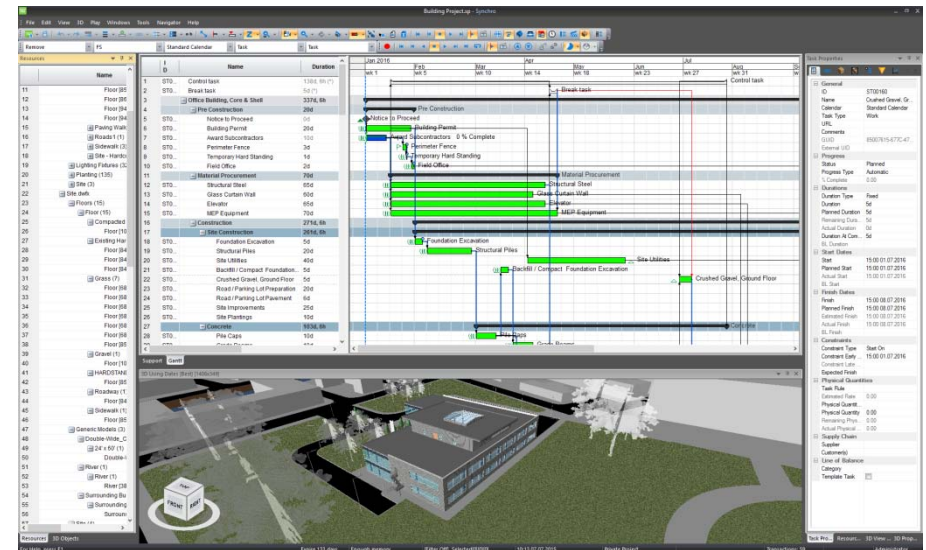


Рис. 2. Главное окно графического интерфейса пользователя системы Synchro.
Fig. 2. The main window of the graphical user interface of the Synchro system

Как отмечалось выше, отсутствие собственных средств построения расписаний являлось одним из главных недостатков приложения, поскольку ограничивало пользователя в разрешении обнаруженных конфликтов в результате согласованной коррекции календарно-сетевых графиков. Наличие подобных средств превращало бы приложение в полноценную систему управления проектами, причем с возможностями визуального моделирования и многофакторного планирования.

Учет различных факторов осуществления проектной деятельности также являлся одним из принципиальных требований, предъявляемым к развиваемому приложению. При составлении расписаний должны учитываться не только временные условия, отношения предшествования, ресурсные ограничения и календарные правила, характерные для задач проектного планирования RCPSP, но и специфические требования пространственно-временной согласованности проектных работ, их финансового и логистического обеспечения. Данные требования важны для масштабных индустриальных программ, в которых риски технологических и организационных ошибок чрезвычайно высоки, а сроки и бюджеты жестко ограничены. Примерами специфических требований могут служить условия привлечения инвестиционных средств, ограничения по поставкам материалов, правила размещения и использования оборудования, особенности монтажа элементов конструкций возводимого сооружения, условия резервирования рабочих зон при организации проектных работ.

Заметим, что по мере развития целевой системы набор ограничений может пересматриваться в сторону обобщения и расширения, поэтому средства составления расписаний должны быть реализованы надлежащим образом, допуская поддержку новых типов ограничений без каких-либо серьезных изменений существующих программных компонентов.

Данные требования удалось удовлетворить с помощью представленного каркаса для построения приложений теории расписаний. Каркас не только предоставил готовые к использованию программные компоненты для задания условий и решения задач проектного планирования, но и обеспечил возможность добавления и поддержки новых видов ограничений в рамках описанной выше методологии.

Рассмотрим процесс эволюционной разработки системы визуального планирования проектов Synchro более подробно.

Так как каркас внедрялся в существующее приложение, а не использовался для создания новой системы, возникла необходимость развития унаследованной модели данных. Заметим, что приложение Synchro реализовано в рамках объектно-ориентированной парадигмы на языке Си++ и использует объектно-ориентированную модель данных для представления проекта, работ, связей, календарей, ресурсов, трехмерных объектов сцены. Упрощенная реализация данных концептов с использованием ограниченного набора атрибутов

диктовалась функциями визуального моделирования, для которых достаточно иметь данные, подлежащие отрисовке на диаграмме Ганта или в окнах просмотра трехмерных сцен. В частности, в классе работ определялись атрибуты имени, длительности, времен начала и конца, однако отсутствовали структуры индивидуального использования ресурсов. Класс календарей реализовывал простую модель рабочей недели с исключениями, но не применял более общую модель регулярных и исключительных правил. Класс ресурсов использовался, главным образом, в качестве делегата трехмерных объектов сцены, однако не предусматривал задание ресурсных атрибутов, таких как тип, статус возобновимости и количество доступных единиц.

В силу указанных причин было принято решение заменить соответствующие классы приложения каркасными реализациями. Такая замена не являлась трудоемкой и в большинстве случаев заключалась в коррекции некоторых названий методов доступа к атрибутам классов. Для поддержания прежней функциональности визуального моделирования и реализации новых функций проектного планирования, класс проектов Project, агрегирующий трехмерные данные (объекты, текстуры, камеры, анимации), был унаследован от соответствующего класса каркаса, агрегирующего необходимые проектные данные. Аналогичная схема консолидации применена в классе ресурсов, что позволило объединить данные, необходимые для ресурсного планирования, и данные, необходимые для их связи с трехмерными объектами.

При описанной схеме наследования в приложении Synchro стало возможным составлять расписания непосредственно средствами каркаса. В частности, появилась возможность расчета критического пути и оценки временных резервов в постановке задачи проектного планирования без ресурсов. Также стали доступны алгоритмы составления расписаний в расширенных постановках. Важно, что для подобных целей не потребовалась разработка дополнительных конвертеров исходных данных и результатов расчетов, поскольку алгоритмические реализации Solvers рассчитаны для работы с проектными данными при надлежащей конфигурации компонентов Project Data, Reductions и Mathematics.

Для задания дополнительных параметров планирования, выполнения алгоритмов и визуализации расчетов потребовалось расширить графический интерфейс пользователя. В частности, в окне параметров проекта добавлены элементы для установки планируемых дат начала и завершения проекта, необходимых для составления расписания и расчета временных резервов работ. В окне параметров работ добавлены элементы для установки временных ограничений, количеств используемых ресурсов, задержек между взаимосвязанными работами. В диалоге задания опций предусмотрен раздел настроек для алгоритмов составления расписаний, который позволяет, например, установить правила выравнивания дат работ, пороговые значения временных резервов для интерпретации работ как критических и субкритических, способы разрешения нарушенных ограничений в процессе

актуализации проектных данных. Наконец, в панель инструментов добавлена кнопка для запуска пересчета расписания. На рисунке 3 представлен экранный снимок приложения Synchro с описанными развитыми функциями проектного планирования.

Заметим, что ревизии подверглись и средства визуализации результатов расчетов. В диаграмме Ганта появилась возможность отобразить критический путь, наложенные временные ограничения, а также фактические взаимосвязи, определяющие даты планируемых работ. В таблицу проектных работ (слева от диаграммы Ганта) добавлены столбцы со значениями временных резервов, которые в сочетании с универсальными средствами сортировки и фильтрации позволяют пользователю выделить относительно небольшое, но исключительно важное подмножество работ, влияющих на ход и успешное завершение всего проекта к планируемой дате. Данная возможность особенно важна для управления масштабными индустриальными программами, в которых детальный контроль выполнения каждой индивидуальной работы не представляется возможным.

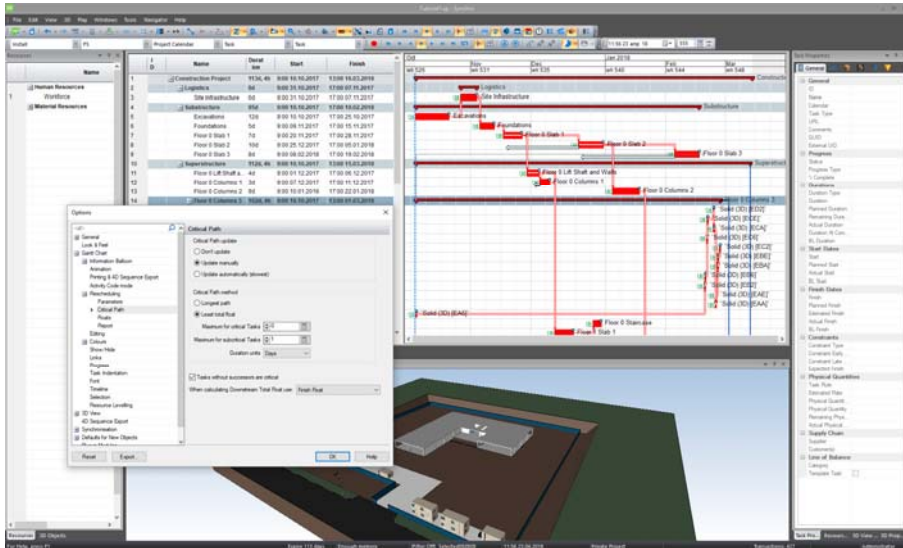


Рис. 3. Главное окно графического интерфейса пользователя системы Synchro с развитыми возможностями проектного планирования.

Fig. 3. The main window of the graphical user interface of the Synchro system with advanced features for project planning

Поскольку планирование ресурсов является одним из ключевых элементов проектного управления, в графический интерфейс пользователя были добавлены графики потребления ресурсов, которые позволяют пользователю идентифицировать конфликты в календарно-сетевом графике, связанные с

превышением доступного количества единиц того или иного ресурса. Сгенерированные приложением расписания являются согласованными в том смысле, что при корректно заданных условиях наложенные ограничения, в том числе и ресурсные, оказываются удовлетворенными. Однако при задании или коррекции календарно-сетевых графиков непосредственно пользователем часть ограничений может быть нарушена и необходимы средства их идентификации. На рисунке 4 приведён пример графиков потребления ресурсов по времени, причем слева представлены кривые, устанавливающие превышение доступного количества ресурсов в графике, а справа — кривые полностью согласованного графика.

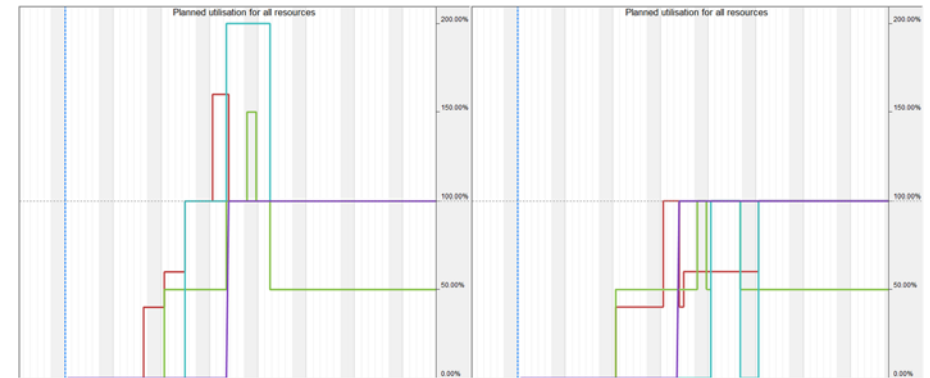


Рис. 4. Примеры графиков потребления ресурсов в несогласованном (слева) и согласованном (справа) календарно-сетевом графике.

Fig. 4. Examples of resource utilisation charts in the uncoordinated (left) and coordinated (right) calendar-network schedule

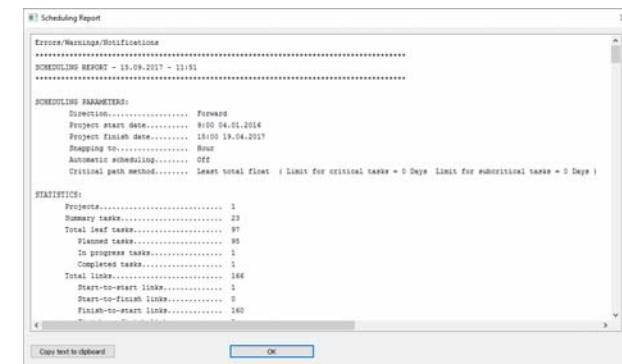


Рис. 5. Пример отчёта о построении календарно-сетевых графиков.

Fig. 5. An example calendar-network schedule construction report

Наконец, для верификации условий задач проектного планирования и генерации отчетов был разработан и добавлен в систему соответствующий

инструмент. На рисунке 5 показано окно для просмотра текстовых отчетов, в которое выводятся общие статистические данные о проекте (количество работ, ресурсов, взаимосвязей, календарей и т.п.), параметры проектного планирования, а также некорректно заданные условия, связанные с наличием циклических взаимосвязей между работами, превышением потребляемых ресурсов индивидуальными работами, переопределенными временными условиями.

Перечисленные выше функции календарно-сетевое планирования являются традиционными для систем управления проектами. Использование компонентов каркаса лишь упростило и ускорило их реализацию в составе целевой системы Synchro, которая в результате объединила в себе функционал визуального моделирования и проектного планирования.

Вместе с тем, с консолидацией проектных данных появились качественно новые возможности для решения задач планирования в расширенных постановках с учетом пространственных факторов. Такими факторами могут быть рабочие зоны, которые в случае пересечения или перегруженности порождают конфликты на проектной площадке и мешают осуществлению проектных работ [9]. Другими пространственными факторами, влияющими на реализуемость графика, могут быть коллизии размещаемого оборудования, конфликтность планируемых путей доставки материалов, отсутствие опор или подвесов при монтаже конструкций [10]. Математическая формализация задачи проектного планирования с учетом перечисленных пространственных факторов, а также метод её решения приводятся в работе [7].

Важно, что данные пространственные факторы должны учитываться вместе с другими видами ограничений в рамках общей вычислительной схемы составления расписания. Введенный класс задач проектного планирования, а также применяемая схема последовательной диспетчеризации позволяют осуществить это в результате задания специальных условий и приведения их к обобщенной математической постановке GCPSP. С этой целью в пакет каркаса Project Data были добавлены необходимые классы представления рабочих зон и трехмерных объектов, а в реализации класса Project учтены новые пространственные условия. Для данных условий были разработаны и включены в пакет Reductions соответствующие классы приведения к алгебраическим ограничениям общего вида в рамках постановки GCPSP. Для приоритизации работ с назначенными рабочими зонами была разработана новая эвристика, учитывающая требуемый объем рабочей зоны, а ее реализация включена в состав пакета Solvers. Заметим, что для подобных целей можно было бы использовать и стандартную эвристику, основанную на уровнях потребления ресурсов, однако интерпретация объема рабочей зоны в качестве ресурса не всегда корректна [11].

Рассматривались возможности использования других вычислительных схем и алгоритмов проектного планирования, в частности, параллельной схемы

генерации расписаний, одно- и многопроходных схем, схем сэмплирования [12]. Несмотря на их поддержку каркасом, в целевой системе был использован компонент, реализующий алгоритм последовательной диспетчеризации с расширенным набором эвристик. В силу невысокой вычислительной сложности данный алгоритм хорошо себя зарекомендовал при решении широкого класса задач проектного планирования высокой размерности [7], что являлось принципиальным требованием к разрабатываемой целевой системе. Поддержка возможностей выбора специализированных алгоритмов для решения частных классов задач была принята нецелесообразной.

Заметим, что использование объектно-ориентированного каркаса значительно упростило развитие приложения. Сам каркас предоставил готовые классы проектных данных и решатели задач построения расписаний в обобщенной постановке GCPSP. Адаптация каркаса к соответствующим прикладным задачам свелась лишь к реализации классов в пакете Reductions для поддержки специальных типов условий, возникающих в задачах проектного планирования. Трудозатраты (в строках программного кода) реализации данных классов приведены в таблице 1.

Табл. 1. Оценка трудозатрат развития приложения.

Table 1. An estimate of labor costs of application development

Тип ограничения	Кол-во строк кода	Доля от общего кол-ва строк
Учет предшествования работ	211	0,16%
Учет календарей	374	0,28%
Явные временные ограничения	170	0,13%
Обязательные временные ограничения	153	0,11%
Выравнивание начала или конца работ	97	0,07%
Учет использования ресурсов	669	0,49%
Учет использования рабочих зон	2033	1,5%

Таким образом, реализация каждого из типов ограничений заключалась в создании одного нового класса, описываемого в среднем 530 строками программного кода, что составляет 0,39% от общего количества строк кода инструментальной среды.

4. Сравнительный анализ производительности системы

Достигнутая общность в программной реализации алгоритмов составления расписаний, а также принятая параметризация условий задач проектного планирования в расширенных постановках могут существенно влиять на 250

производительность целевой системы. Поэтому важным было убедиться, что негативный эффект от обобщенной реализации программных компонентов каркаса не столь существенен и разработанная система визуального планирования Synchro может конкурировать с популярными системами управления проектами при решении индустриально значимых задач высокой размерности.

Оценка производительности систем производилась по критерию затраченного процессорного времени на составление расписания в зависимости от размерности задачи (числа переменных и ограничений). В качестве тестов использовались синтезированные наборы проектных данных с разным количеством работ и связей между ними. Поскольку синтезированные планы имели фиксированные соотношения глубины и кратности связей, размерность задач в каждом тестовом наборе определялась лишь числом работ. Планы реальных индустриальных проектов использовались лишь для подтверждения полученных результатов. Тестирование проводилось на одном и том же компьютере с типовой конфигурацией (процессор: Intel Core i7, количество ядер: 4, объем оперативной памяти: 16 Гб, операционная система: MS Windows 10 x64), поэтому для проводимого сравнительного анализа систем значение имели лишь относительные показатели производительности.

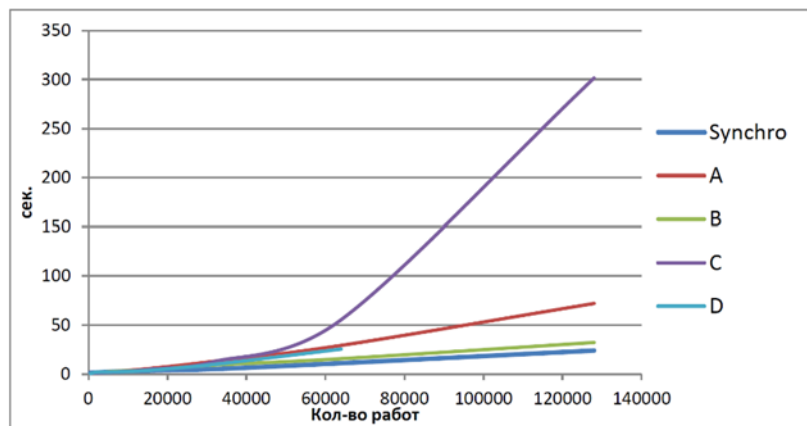


Рис. 6. Зависимость процессорного времени составления расписания в постановке CPM от размерности задачи.

Fig. 6. The dependence of computation time of schedule construction in CPM formulation on the problem dimensionality

Первая серия испытаний предназначалась для анализа производительности систем в зависимости от размерности задач планирования в постановке расчета критического пути (CPM). Для каждого теста в наборе составлялось расписание с помощью пяти различных систем, включая Synchro, и замерялось затраченное

процессорное время. Поскольку используемые в экспериментах системы управления проектами являются коммерческими и их упоминание в результатах сравнительного анализа запрещается в соответствии с лицензионными соглашениями, мы присвоили данным системам символические имена А, В, С, D. Результаты испытаний качественно не менялись при переходе к новому тестовому набору, поэтому на рисунке 6 приведены типовые графики зависимости процессорного времени от количества работ в проектом плане. Заметим, что кривая производительности для системы D обрывается при достижении размерности тестовых данных, составляющей 64000 работ. Это связано с невозможностью данной системы обрабатывать проектные планы большей размерности. С ростом размерности задач планирования система Synchro демонстрировала наилучшие результаты. Вторая серия испытаний предназначалась для аналогичных целей, но расписание строилось в постановке проектного планирования с учетом ресурсов RCPSP. Поскольку системы реализуют приближенные алгоритмы, имеющие разную вычислительную сложность и обеспечивающие разное качество найденных решений, сравнительный анализ производительности здесь не всегда корректен.

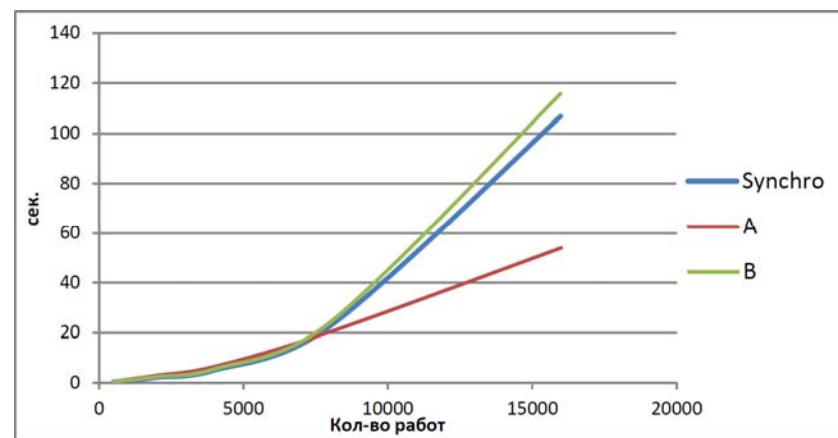


Рис. 7. Зависимость времени поиска расписания от размерности плана (с учётом ресурсов).

Fig. 7. The dependence of schedule search time on the dimensionality of the plan (considering resources)

На рисунке 7 представлены характерные результаты сравнения производительности популярных систем управления проектами. Системы С и D не участвовали в данном испытании по причине отсутствия в них соответствующих алгоритмических средств. Более тщательный анализ показывает, что алгоритм, реализованный в системе А, демонстрирует линейную сложность от размерности задачи, а в двух других системах, включая

Synchro — квадратичную. Это объясняется тем, что первый алгоритм строит согласованное расписание путем решения задачи в постановке расчета критического пути, а затем — последовательно выравнивает уровни потребления ресурсов. Это может быть выполнено за линейное время, однако качество найденного решения в общем случае будет хуже вследствие возможного увеличения длительности проектов за счет выравнивания ресурсов. Более качественный результат (меньшая длительность планируемого проекта) достигается при поиске оптимального расписания приближенными методами, например, с помощью реализованного в объектно-ориентированном каркасе классического алгоритма последовательной диспетчеризации, который имеет квадратичную сложность, поскольку выбор приоритетной работы на каждом шаге требует применения эвристики для всех работ в активном списке. По сравнению с системой В, также реализующей алгоритм поиска расписания квадратичной сложности, разработанная система демонстрирует лучшую производительность.

Тем самым, предоставляя широкие функциональные возможности, разработанная система Synchro не уступает по производительности популярным системам управления проектами при решении задач планирования в аналогичных постановках и аналогичными методами.

5. Заключение

Таким образом, применение объектно-ориентированного каркаса позволило не только ускорить разработку системы визуального планирования проектов, но и обеспечило ее динамичное функциональное развитие, причем с относительно небольшими затратами. Важно, что сами классы каркаса не претерпели каких-либо изменений, а для расширения функций системы потребовалась лишь реализация специализированных классов с предопределенными интерфейсами. Главным образом, доработка была обусловлена необходимостью обобщения условий решаемых задач и настройки алгоритмических компонентов. Эволюционные возможности среды позволили со временем поддержать развитые наборы временных условий, взаимосвязей задач, моделей привлечения ресурсов, пространственных ограничений и, тем самым, обеспечить конкурентные преимущества целевой системы над традиционными системами управления проектами.

Проведенные серии экспериментов показали, что достигнутая общность в реализации программных компонентов не препятствует высокой производительности, которая оказалась сопоставима с аналогичными показателями популярных систем управления проектами.

Успешная апробация каркаса в ходе разработки, сопровождения, развития и коммерческого использования целевой системы Synchro на протяжении десяти лет показала правильность принятых проектных решений, а также

перспективность использования каркаса для эволюционной разработки приложений теории расписаний.

Список литературы

- [1]. Kolisch R., Sprecher A. PSPLIB — A project scheduling library. *European Journal of Operational Research*, vol. 96, issue 1, 1997, pp. 205-216.
- [2]. Kolisch R., Hartmann S. Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update. *European Journal of Operational Research*, vol. 174, 2006, pp. 23-37.
- [3]. Lemmen R. Modeling Resource Alternatives in Project Scheduling. *Munich University of Applied Sciences*, March 29, 2005.
- [4]. Лазарев А. А., Гафаров Е. Р. Теория расписаний. Задачи и алгоритмы. МГУ им. М. В. Ломоносова, Москва, 2011 г., 222 с.
- [5]. Аничкин А.С., Семенов В.А. Объектно-ориентированный каркас для программной реализации приложений теории расписаний. *Труды ИСП РАН*, том 29, вып. 3, 2017 г., стр. 247-296. DOI: 10.15514/ISPRAS-2017-29(3)-14.
- [6]. Аничкин А.С., Семенов В.А. Об обобщенной математической постановке задач проектного планирования. *ИТНОУ: Информационные технологии в науке, образовании и управлении* (под ред. Глориозова Е.Л.), вып. 2, 2017 г., стр. 74-86.
- [7]. Аничкин А.С., Семенов В.А. Математическая формализация задач проектного планирования в расширенной постановке. *Труды ИСП РАН*, том 29, вып. 2, 2017 г., стр. 231-256. DOI: 10.15514/ISPRAS-2017-29(2)-9.
- [8]. Интернет-ресурс: «Synchro Software». Официальный Интернет-сайт продукта Synchro, <http://synchro ltd.com>
- [9]. Аничкин А.С., Казаков К.А., Семенов В.А. Календарно-сетевое планирование промышленных проектов с учетом перегруженности рабочих зон. *Информационные и математические технологии в науке и управлении: Труды XX Байкальской Всероссийской конференции*, в 3 томах (под ред. Массель Л.В.), том 1, 2015 г., стр. 7-14. Иркутск, ИСЭМ СО РАН.
- [10]. Семенов В.А., Аничкин А.С., Морозов С.В., Тарлапан О.А., Золотов В.А. Комплексный метод составления расписаний для сложных промышленных программ с учетом пространственно-временных ограничений. *Труды ИСП РАН* (под ред. Иванникова В.П.), том 26, вып. 1, 2014 г., стр. 457-482. ISSN 2220-6426.
- [11]. Yeoh K.W., Chua David K.H. Mitigating Workspace Congestion: A Genetic Algorithm Approach. *Proceedings of the EPPM 2012 Conference*, 2012, pp. 107-118.
- [12]. Kolisch R., Hartmann S. Heuristic algorithms for solving the resource-constrained project scheduling problem — Classification and computational analysis. Chapter in the book «Handbook on recent advances in project scheduling», ed. Weglarz J., 1999, pp. 147-178.

Evolutionary development of a visual planning system using object-oriented framework

¹ A.S. Anichkin <anton.anichkin@ispras.ru>

^{1,2} S.V. Morozov <serg@ispras.ru>

^{1,3} V.A. Semenov <sem@ispras.ru>

^{1,2} O.A. Tarlapan <oleg@ispras.ru>

¹ *Ivannikov Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*

² *Lomonosov Moscow State University,*

GSP-1, Leninskie Gory, Moscow, 119991, Russia

³ *Moscow Institute of Physics and Technology (State University),*

9 Institutskiy per., Dolgoprudny, Moscow Region, 141700, Russia

Abstract. The article describes the practical experience of developing a prospective visual planning system based on an object-oriented framework. The used framework is a system of classes and interfaces intended for software implementation of models, methods and applications of scheduling theory. Due to the availability of ready-to-use components for the solution of typical problems as well as the mechanisms for their configuration and extension, development of applications becomes a relatively simple process. The application of the framework allows to implement the necessary functional for project planning in the target system as well as to provide its subsequent evolution by generalizing the statements of the problems and expanding the arsenal of algorithms used to solve them. The described experience can be claimed when building other applications of scheduling theory. The paper discusses the general issues of organizing a software toolkit in the form of the object-oriented framework, a methodology for creation of scheduling theory applications on its basis, as well as a process of developing a target system for visual planning of projects based on the methodology and the framework. Results of computational experiments comparing the performance of the developed system with some popular project management systems are also presented in the paper. Recommendations on the development and evolution of scheduling theory applications based on the framework are summarized in conclusion.

Keywords: scheduling theory; project planning and scheduling; software engineering; object-oriented programming.

DOI: 10.15514/ISPRAS-2017-29(5)-12

For citation: Anichkin A.S., Morozov S.V., Semenov V.A., Tarlapan O.A. Evolutionary development of a visual planning system using object-oriented framework. *Trudy ISP RAN/Proc. ISP RAS*, vol. 29, issue 5, 2017. pp. 239-256 (in Russian). DOI: 10.15514/ISPRAS-2017-29(5)-12

References

- [1]. Kolisch R., Sprecher A. PSPLIB — A project scheduling library. *European Journal of Operational Research*, vol. 96, issue 1, 1997, pp. 205-216.
- [2]. Kolisch R., Hartmann S. Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update. *European Journal of Operational Research*, vol. 174, 2006, pp. 23-37.
- [3]. Lemmen R. Modeling Resource Alternatives in Project Scheduling. *Munich University of Applied Sciences*, March 29, 2005.
- [4]. Lazarev A. A., Gafarov E. R. Scheduling theory. Tasks and algorithms. *Lomonosov Moscow State University, Moscow*, 2011, 222 p. (in Russian).
- [5]. Anichkin A.S., Semenov V.A. Object-oriented framework for software development of scheduling applications. *Trudy ISP RAN/Proc. ISP RAS*, vol. 29, issue 3, 2017, pp. 247-296 (in Russian). DOI: 10.15514/ISPRAS-2017-29(3)-14.
- [6]. Anichkin A.S., Semenov V.A. About the generalized mathematical formulation of project scheduling problems. *ITNOU: Informacionnye tehnologii v nauke, obrazovanii i upravlenii/ITSEM: Information technologies in science, education and management* (ed. Gloriozov E.L.), issue 2, 2017, pp. 74-86 (in Russian).
- [7]. Anichkin A.S., Semenov V.A. Mathematical formalization of project scheduling problems. *Trudy ISP RAN/Proc. ISP RAS*, vol. 29, issue 2, 2017, pp. 231-256 (in Russian). DOI: 10.15514/ISPRAS-2017-29(2)-9.
- [8]. Internet: «Synchro Software». Official website of the product Synchro, <http://synchro ltd.com>
- [9]. Anichkin A.S., Kazakov K.A., Semenov V.A. Calendar-network planning of industrial projects taking into account the congestion of workspaces. *Proceedings of the 20th Baikal All-Russian Conference*, in 3 volumes (ed. Massel L.V.), vol. 1, 2015, pp. 7-14. Irkutsk, MESI SB RAS.
- [10]. Semenov V.A., Anichkin A.S., Morozov S.V., Tarlapan O.A., Zolotov V.A. Effective method for scheduling complex industrial programs under spatio-temporal constraints. *Trudy ISP RAN/Proc. ISP RAS* (ed. Ivannikov V.P.), vol. 26, issue 1, 2014, pp. 457-482. ISSN 2220-6426.
- [11]. Yeoh K.W., Chua David K.H. Mitigating Workspace Congestion: A Genetic Algorithm Approach. *Proceedings of the EPPM 2012 Conference*, 2012, pp. 107-118.
- [12]. Kolisch R., Hartmann S. Heuristic algorithms for solving the resource-constrained project scheduling problem — Classification and computational analysis. Chapter in the book «Handbook on recent advances in project scheduling», ed. Weglarz J., 1999, pp. 147-178.