

АДАПТИВНЫЙ ПОДХОД К РЕАЛИЗАЦИИ СЛОЖНЫХ СИСТЕМ МОДЕЛИРОВАНИЯ

М.В. Гордин

Г.С. Иванова

А.В. Пролетарский

М.В. Фетисов

gordinmv@bmstu.ru

gsivanova@bmstu.ru

pav@bmstu.ru

fetisov.michael@bmstu.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Приведен подход, позволяющий не только эффективно выполнить импортозамещение в сфере современных программных систем моделирования и проектирования производственных процессов в различных отраслях промышленности, но и компенсировать разрыв глобальных связей, выстраивающих эффективные структуры разделения труда при разработке современных программных систем. Для решения этих задач предложено реализовать адаптивную систему моделирования, способную настраиваться на различные предметные области за счет существенной автоматизации разработки предметно-ориентированных языков, на которых должны описываться модели в соответствующих предметных областях. Предложено также разделить разработку на следующие сферы компетенции и ответственности: разработка расширяемого ядра адаптивной системы моделирования и общих языковых и визуальных (графических) компонентов; разработка расширений для конкретной предметной области, в которые могут входить как языковые, так и визуальные (графические) компоненты; наполнение конкретного расширения базой отраслевых стандартов; разработка непосредственно модели производства. Применение общей платформы в виде адаптивной системы моделирования позволяет использовать расширения из разных предметных областей при формировании сложных моделей, в результате чего могут быть достигнуты технологические прорывы на стыке различных отраслей промышленности. Кратко описаны некоторые риски, связанные с реализацией и сопровождением сложных систем моделирования в Российской Федерации. Рассмотрены аналоги предлагаемых решений

Ключевые слова

Предметно-ориентированный язык, проблемно-ориентированный язык, моделирование производственных процессов, имитационное моделирование, предметная область, отрасль производства, система моделирования, адаптивная система моделирования

Поступила 01.07.2022

Принята 21.07.2022

© Автор(ы), 2022

Введение. В настоящее время глобализация, которая совсем недавно достигала своего максимального уровня, начинает ослабевать [1]. Видно, как разрушаются старые, формировавшиеся веками экономические отношения между регионами и странами, вводятся санкции, нарушаются логистические связи. Если ранее виной этому называли пандемию COVID-19, то сейчас, когда влияние этого фактора ослабевает, а кризис только усугубляется, есть основание полагать, что причина деградации кроется глубже и процесс распада глобального мира может быть достаточно глубоким [2].

Чем сложнее автоматизируемое производство, предметная область или отрасль, тем больше ресурсов необходимо затратить на разработку и сопровождение системы автоматизации. Чтобы не потерять контроля над этим процессом, формируются иерархические кластеры автоматизации, в которых производится разделение компетенций и ответственности между исполнителями. Правильное разделение труда позволит достигнуть оптимальной структуры производства и сопровождения. Организации, которые смогли создать такую оптимальную структуру, оказываются наиболее эффективными, имеют меньше накладных расходов на поддержание внутренних процессов и в результате получают больше прибыли. Значит, именно они выживают. Соображения на эту тему высказывал еще А. Смит в XVIII в. Распад глобального миропорядка нарушает эти структуры, следовательно, возможности разработки и сопровождения сложных производств, в том числе сложных систем моделирования, будут существенно затруднены.

Цель настоящей работы — сформировать подход, позволяющий благодаря введению дополнительных горизонтальных, межотраслевых сфер компетенции и ответственности сохранить глубину иерархии, необходимую для реализации и сопровождения сложных систем моделирования. Этот подход должен позволить не только реализовать проекты импортозамещения в Российской Федерации и при этом не потерять качество при замещении сложных систем, но и нарастить его за счет кооперации на стыке отраслевых задач.

Адаптивная система моделирования (АСМ) — это оригинальный и уникальный инструментальный, который позволяет разрабатывать предметно-ориентированные языки (ПОЯ), более естественные для специалистов соответствующих предметных областей при описании моделей их производств [3].

Схема разделения компетенции и передачи ответственности при автоматизации производства некоторого продукта с использованием АСМ приведена на рис. 1. Разработчик АСМ компетентен и ответственен за ее

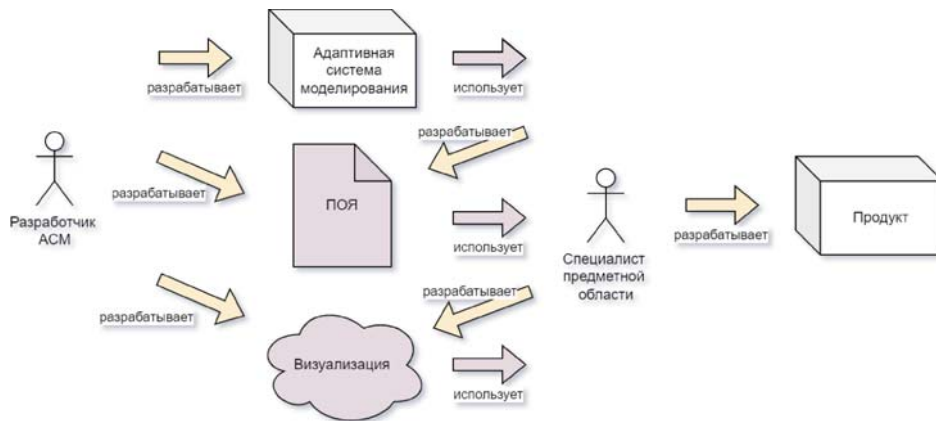


Рис. 1. Схема разделения компетенции и передачи ответственности при автоматизации производства продукта с использованием АСМ

разработку, а также за ПОЯ и систему визуализации моделей, которые будут использоваться специалистом предметной области при разработке продукта. При этом специалист предметной области должен так участвовать в разработке ПОЯ и средств визуализации, чтобы эффективно использовать АСМ при разработке продукта.

По сути, при разработке ПОЯ и средств визуализации разработчик АСМ и специалист предметной области делят компетенцию и ответственность. Степень этого разделения должна быть оговорена в каких-либо регламентных документах и согласована обеими сторонами.

Адаптивность предлагаемой системы моделирования заключается в ее способности как бы настраиваться на работу с конкретной предметной областью. При работе с конкретной предметной областью можно использовать, помимо своих ПОЯ и средств визуализации, универсальные ПОЯ и средства визуализации. К таким средствам относятся, например, ПОЯ, описывающие систему обыкновенных дифференциальных уравнений и отображение графиков и диаграмм (рис. 2). Адаптивная система моделирования должна предоставлять возможным образом совмещать специализированные ПОЯ с универсальными и средствами визуализации.

Интерфейс АСМ при работе с моделью, описываемой системой обыкновенных дифференциальных уравнений, приведен на рис. 3 (в примере — аттрактор Лоренца). Можно предположить, что и ПОЯ описания динамических систем, и средства визуализации, позволяющие строить графики и диаграммы изменения состояния динамической системы, могут быть использованы для описания многих аспектов моделирования более сложных

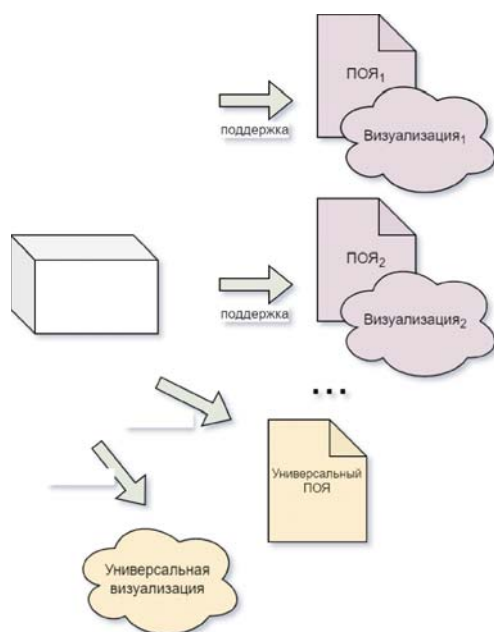


Рис. 2. Поддержка предметных областей и универсальные реализации

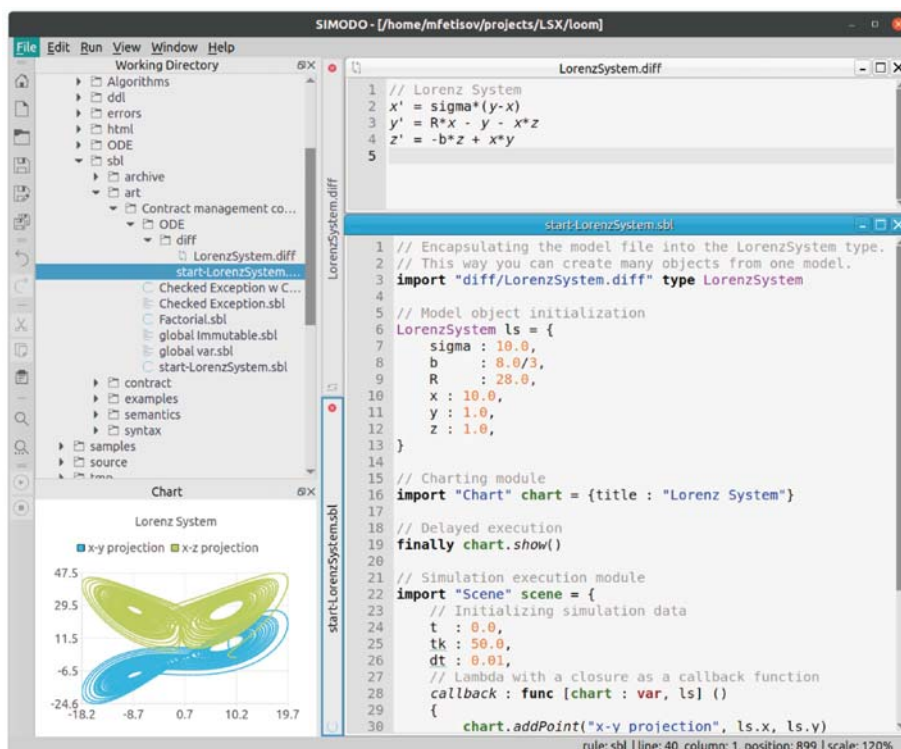


Рис. 3. Скриншот внешнего вида АСМ при работе с моделью, описываемой системой обыкновенных дифференциальных уравнений

систем в различных предметных областях. Скриншот показывает работу с прототипом АСМ, разработанным на кафедре «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана [4]. Отметим, что АСМ должна иметь открытую программную архитектуру, чтобы минимизировать риски, связанные с разработкой и сопровождением ПОЯ и средств визуализации для конкретной предметной области [5].

Императив ПОЯ. Важность и первичность использования ПОЯ при разработке конечного продукта предметной области обусловлена рядом преимуществ, например возможностью использования универсальных средств визуализации.

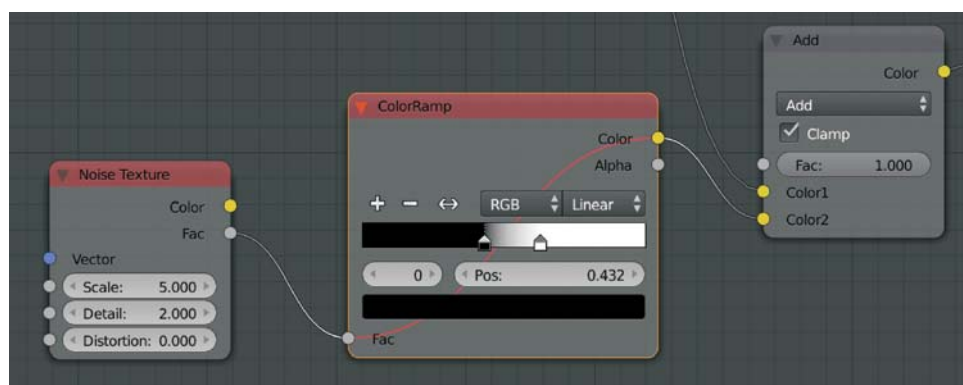
Другим преимуществом является возможность удобной коллективной работы со средствами управления версиями и репозиториями кода, а также с использованием таких современных инструментов, как непрерывная интеграция и многих других, с ней связанных.

Параллельно с разработкой и сопровождением ПОЯ для конкретной предметной области может вестись разработка и сопровождение средств визуализации, которые опираются на структуру кода, записанную с помощью ПОЯ, т. е. код является первичным описанием предметной области, а схемы и диаграммы — это его графическая интерпретация. Причем схему можно править, что будет отражаться на структуре кода. Такое разделение используется во многих предметных областях и показало свою состоятельность, выдержав проверку временем.

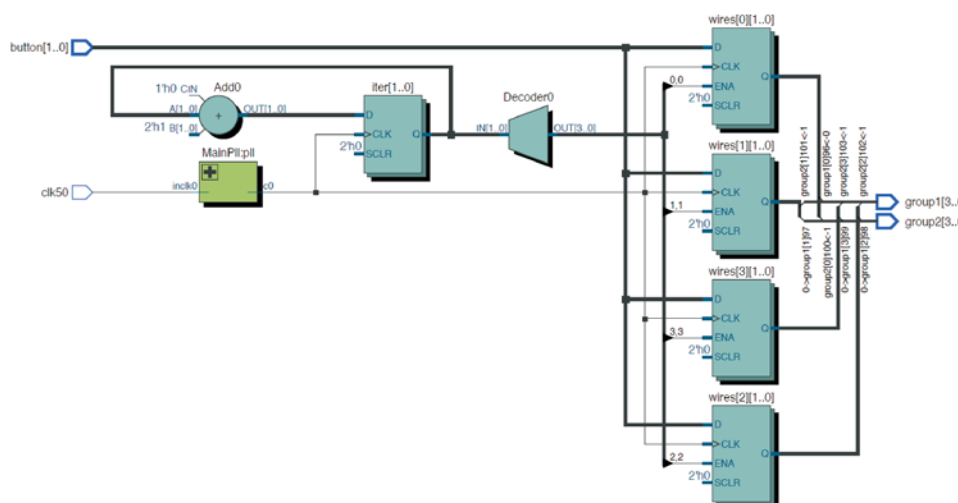
Примеры схем при построении структуры обработки изображения с использованием шейдеров в Blender [6], а также схематическое описание электрической схемы с использованием системы System Verilog [7] приведены на рис. 4. За «капотом» работы с моделями в этих системах скрывается первичный ПОЯ описания этих моделей. При этом приведенные схемы итеративны, позволяют изменять параметры элементов, выполнять запуск и отладку моделей.

В качестве еще одного примера удачной реализации разделения языковой и графической составляющих моделирования стоит привести такие универсальные системы моделирования, как MATLAB, Modelica и многие другие, которые в своих наименованиях имеют название базового языка, но зачастую ассоциируются именно с графическими инструментами, позволяющими наглядно представлять модели.

Следует отметить также такую особенность языковой реализации модели, как большую универсальность. На ПОЯ, как правило, можно описать сложную модель, которую невозможно представить в графическом виде. Это диверсифицирует процесс разработки комплексных моделей.



a



б

Рис. 4. Работа с шейдерами в Blender (*a*) и моделью в System Verilog (*б*)

Современные средства редактирования кода позволяют непосредственно при его вводе или модификации выполнять не только лексический и синтаксический анализ и подсвечивать его структуру или ошибки, но и выполнять семантический анализ с точки зрения конкретной предметной области, что приводит к выявлению определенного класса проблем на самом раннем этапе формирования моделей [8].

Сферы компетенции и ответственности при разработке и сопровождении сложных отраслевых систем моделирования. Реализация продукта в какой-либо отрасли связана с разработками сложных моделей, которые зачастую должны соответствовать довольно объемной номенклатуре отраслевых стандартов. Эту номенклатуру необходимо каким-то образом ввести (на ПОЯ) в виде отдельных модулей и элементов схем

визуализации. Это отдельная сложная задача, для ее решения необходим дополнительный уровень компетенции и ответственности, кроме тех, что приведены на рис. 1.

Доработанная схема компетенции и ответственности с добавлением промежуточного уровня приведена на рис. 5. Специалист по отраслевым стандартам, а в реальности организация, формирует базу модулей и графических компонентов, которые используются специалистами предметной области для формирования моделей и в итоге разработки конечных продуктов. Таким образом, при разработке инструментов формирования моделей в рамках какой-то сложной отрасли необходимо задействовать несколько сфер компетенции и ответственности — задействовать разработчика АСМ, специалиста по отраслевым стандартам и, наконец, специалиста предметной области в рамках отрасли.

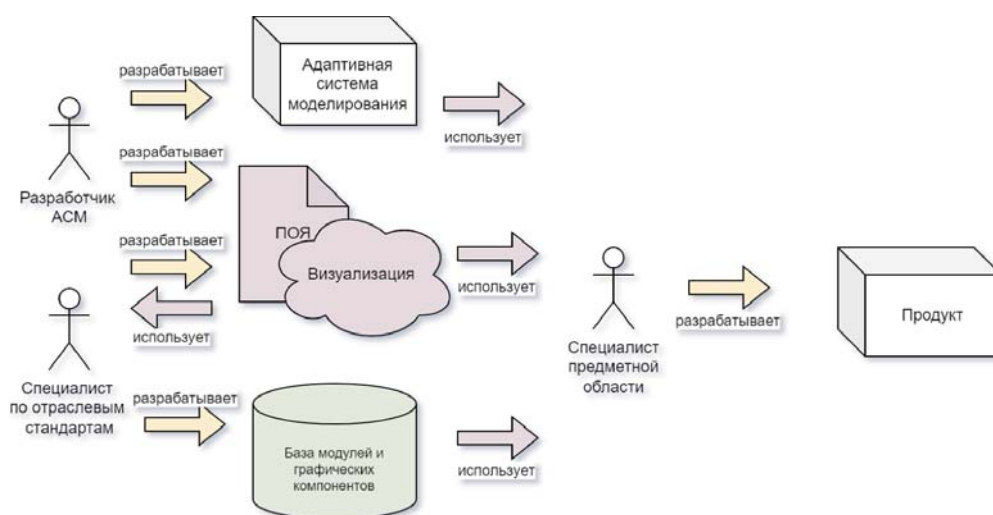


Рис. 5. Формирование базы модулей и графических компонентов специалистом по отраслевым стандартам

Компетенции, необходимые при разработке АСМ, связаны с фокусированием на общих требованиях к системе: ее быстродействию, надежности, переносимости и настраиваемости. Компетенции в разработке отраслевых расширений связаны с хорошим знанием предметной области и пониманием ее особенностей. Компетенции при разработке базы отраслевых стандартов предполагают хорошее знание регламентов и стандартов отрасли, а также владение номенклатурой существующих на рынке агрегатов.

В результате продукт будет представлять собой три независимые части, которые могут иметь собственные ценовые политики, обособленные страте-

гии маркетинга и продвижения на рынке, а также независимые схемы развития. Разработчики этих частей продукта могут быть независимыми организациями, работающими в разных темпах и на различных принципах.

Отметим, что предлагается не конкретный способ разделения, а принцип дифференциации компетенций и ответственностей. В каждой конкретной предметной области или отрасли структура разделения может быть различной.

Межотраслевая кооперация с использованием АСМ. Использование в качестве общей платформы АСМ при разработке моделей существенно упрощает межотраслевую кооперацию, поскольку ПОЯ, визуализация, база модулей и компонентов, реализованные для каждой отрасли, будут изначально совместимы. Это касается универсальных ПОЯ и визуализаций. Схема такого взаимодействия между двумя предметными областями приведена на рис. 6.

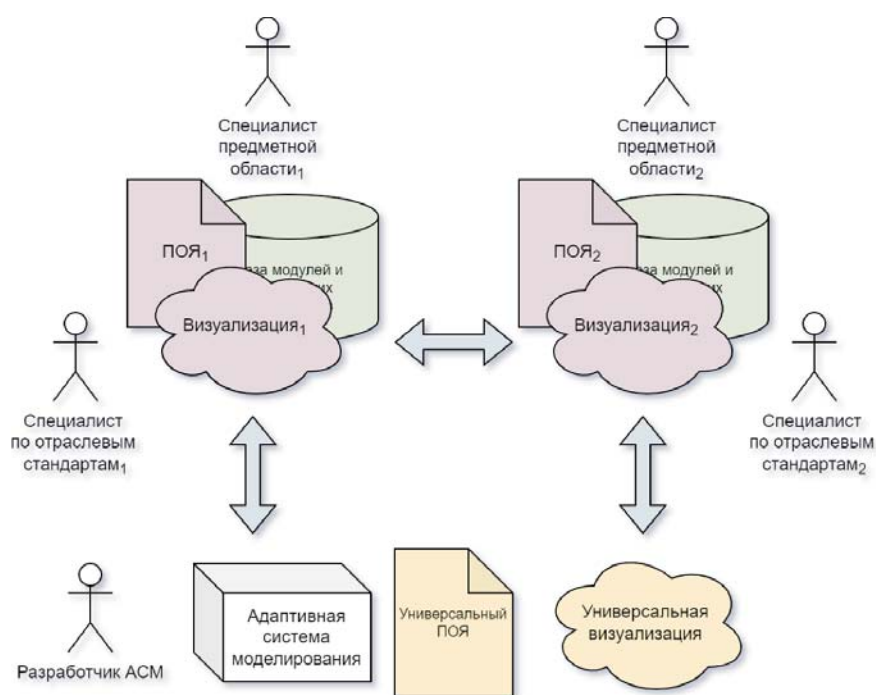


Рис. 6. Схема существенно упрощенного с использованием АСМ в качестве общей платформы межотраслевого взаимодействия

Чем больше будет реализовано расширений АСМ для различных отраслей и предметных областей, тем будет больше возможностей для взаимодействия и интеграции между ними, а значит, тем чаще будут возникать ситуации возможного прорыва в смежных областях науки и произ-

водства. Указанный подход дает такую возможность, но что касается практики взаимодействия — это отдельная задача.

Замыкание отраслей внутри себя — это одна из возможных негативных тенденций отечественного научно-производственного комплекса, обусловленная определенной изоляцией отраслей, особенно характерно проявляющейся в государственных корпорациях. Авторы настоящей работы надеются, что предлагаемый подход может стать примером реализации горизонтальных связей между отраслями, нейтрализующим эту негативную тенденцию.

Осторожное государственное регулирование в этой сфере необходимо, так как отрасли, предоставленные сами себе, будут все сильнее замыкаться в своих задачах и зачастую разрабатывать собственные стандарты и программы моделирования, во многом повторяющие базовые возможности, но совершенно не совместимые друг с другом.

Целесообразно посмотреть в сторону опыта СССР, в котором эту проблему решали с помощью создания специальных институтов, занимающихся задачами, общими для многих отраслей народного хозяйства. Эти институты имели необходимые компетенции в своей области и ответственность за те продукты, которые выпускали. В современных условиях было бы оптимальным решением создавать такие сферы компетенции и ответственности на базе профильных вузов, например, в НИИ при соответствующих факультетах. Это позволило бы совместить задачи исследований и разработки в передовых сферах науки и производства с обучением новых кадров, компетентных в этих областях.

Аналоги АСМ. Основной функцией АСМ является возможность работы со многими ПОЯ, построенными на основе базового языка. Таким образом, совокупность ПОЯ в АСМ представляет собой иерархию диалектов базового языка, имеющих семантические особенности, т. е. расширяющих базовую операционную семантику. Для оптимальной реализации такого подхода используется существенная унификация лексики, синтаксиса и семантики внутри АСМ. Такая унификация во многом противоречит классическому представлению о построении компиляторов. Например, реализацию встроенного генератора компиляторов на базе общего автомата разбора для автоматически формируемых таблиц грамматики [5] критиковал Н. Вирт в работе [9]. Он предлагал использовать распространенные внешние генераторы, например YACC [10], Bison [11] или др. Однако в условиях компактного описания потенциально большого числа диалектов ПОЯ, их частой уточняющей правки канонический подход представляется излишне громоздким.

В основном по этой причине прямых аналогов АСМ не существует. Однако есть некоторые разработки, в которых можно увидеть проявление похожих идей и тенденций. Например, российская система AnyLogic [12] — программное обеспечение для имитационного моделирования бизнес-процессов в сферах логистики, здравоохранения, производства и банковской деятельности, а также любых других процессов, которые можно представить в виде последовательности операций. Инструмент имеет современный графический интерфейс и позволяет использовать язык Java для разработки моделей.

Можно также отметить тенденцию в унификации построения компиляторов со стороны универсальных языков. В качестве примеров можно рассмотреть GCC (GNU Compiler Collection) и LLVM (Low Level Virtual Machine). Принцип работы данных программных инфраструктур заключается в следующем: на фронтенде (верхнем уровне компилятора) происходит чтение исходного кода программы, его синтаксический разбор и порождение абстрактного синтаксического дерева, на бэкенде (нижнем уровне компилятора) данное дерево конвертируется в некоторое промежуточное представление, которое оптимизируется и переводится в программу на языке ассемблера [13]. Такой подход позволяет реализовать компиляторы со многих языков программирования, реализуемые на фронтенде. Все эти компиляторы формируют промежуточное представление, а дальнейшая обработка унифицирована.

В GCC абстрактное синтаксическое дерево конвертируется в RTL (Register Transfer Language), являющийся промежуточным внутренним представлением, близким к языку ассемблера, в LLVM существует промежуточный ассемблер, который можно трансформировать во время компиляции [14].

Существуют альтернативные решения для реализации компиляторов и ПОЯ, которые предлагают дополнительные возможности и расширения. Например, Xtext — фреймворк для разработки языков программирования и ПОЯ позволяет генерировать не только парсер и компилятор, но и полноценную настраиваемую интегрированную среду разработки IDE (Integrated Development Environment). Предметно-ориентированный язык описывается на языке Xtext с вставками на Java, исходный код создается на языке Java [15].

Tree-sitter — средство для создания парсеров на языке C по описанию грамматики на языке, напоминающем РБНФ (расширенная форма Бэкуса — Наура). Данное средство позволяет строить конкретное синтаксическое дерево по исходному файлу и эффективно обновлять его при ре-

дактировании исходного файла [16]. Эта система достаточно ограничена: например, нет возможности помещать семантические вставки в описание грамматики.

Domain-Specific Language Designer — специализированное решение Visual Studio для проектирования ПОЯ. Microsoft предлагает создание и редактирование определения ПОЯ средствами графического интерфейса: на схеме модели предметной области показываются элементы и отношения в модели. Генератор кода принимает на вход данное определение и в качестве выходных данных создает исходный код на языке C# [17].

Meta Programming System (MPS) — система метапрограммирования, разработанная JetBrains [18], реализует парадигму языково-ориентированного программирования [19], является средой разработки языков и в то же время IDE для разрабатываемых языков.

Приведенные решения предназначены для проектирования универсальных языков программирования, без учета особенностей той или иной предметной области, либо же для проектирования ПОЯ, но как имеющие достаточно ограниченный функционал. В сообществе информационных технологий преобладает мнение, что «...ПОЯ — это язык программирования с ограниченными выразительными возможностями, ориентированный на некую конкретную предметную область...» [20]. На самом деле, это может быть совсем не так. В адаптивной системе моделирования ПОЯ — это язык программирования, расширяющий семантику базового универсального языка. Чаще всего в конкретном ПОЯ семантика базового языка будет ограничена, но это не является обязательным. Базовый язык АСМ, в свою очередь, должен быть построен так, чтобы его семантика могла легко расширяться — это очень интересная, но выходящая за рамки настоящей статьи, тема для обсуждения.

Идеи, связанные с реализацией системы моделирования, позволяющей совмещать различные предметные области, также появляются в западном научном сообществе. Похожий подход предложен в системах принятия решений [21, 22], что может быть полезно для использования и в АСМ.

Заключение. Использование АСМ как единой платформы при моделировании и разработке продуктов для ряда предметных областей и отраслей позволит добиться, с одной стороны, разделения труда, необходимого для достижения нужного уровня качества при формировании сложных моделей, а с другой, — нарастить качество самой АСМ в условиях массового использования. Кроме того, использование АСМ как общей платформы при моделировании и реализации продуктов в разных обла-

стях позволит существенно упростить межотраслевое взаимодействие, что может способствовать технологическим прорывам в смежных областях народного хозяйства.

ЛИТЕРАТУРА

- [1] Глазьев С. Последний виток либеральной глобализации. *izborsk-club.ru: веб-сайт*. URL: <http://izborsk-club.ru/11870> (дата обращения: 01.05.2022).
- [2] Смирнов Г. Неисповедимые пути глобализации. *kommersant.ru: веб-сайт*. URL: <http://www.kommersant.ru/doc/5295109> (дата обращения: 01.05.2022).
- [3] Иванова Г.С., Жильцов А.И., Фетисов М.В. и др. Адаптивная система моделирования. *Автоматизация. Современные технологии*, 2020, № 11, с. 500.
- [4] Адаптивная система моделирования SIMODO Stars. *bmstu.codes: веб-сайт*. URL: <http://bmstu.codes/lx/simodo/stars> (дата обращения: 01.05.2022).
- [5] Иванова Г.С., Фетисов М.В., Малкина Т.А. и др. Унификация работы с предметно-ориентированными языками и открытая программная архитектура в адаптивной системе имитационного моделирования. *Динамика сложных систем — XXI век*, 2021, т. 15, № 3, с. 36–47.
DOI: <https://doi.org/10.18127/j19997493-202103-03>
- [6] Blender. *blender.org: веб-сайт*. URL: <http://www.blender.org> (дата обращения: 01.06.2021).
- [7] 1800–2017 — IEEE standard for SystemVerilog — unified hardware design, specification, and verification language. DOI: <https://doi.org/10.1109/IEEESTD.2018.8299595>
- [8] Ivanova G.S., Fetisov M.V. The concept of contract management in the base language of the adaptive modeling system. *SUMMA*, 2021.
DOI: <http://doi.org/10.1109/SUMMA53307.2021.9632114>
- [9] Вирт Н. Построение компиляторов. М., ДМК Пресс, 2010.
- [10] BYACC — Berkeley Yacc — generate LALR(1) parsers. *invisible-island.net: веб-сайт*. URL: <https://invisible-island.net/byacc/byacc.html> (дата обращения: 01.05.2022).
- [11] GNU Bison. *gnu.org: веб-сайт*. URL: <http://www.gnu.org/software/bison> (дата обращения: 01.05.2022).
- [12] AnyLogic. *anylogic.ru: веб-сайт*. URL: <http://www.anylogic.ru> (дата обращения: 01.05.2022).
- [13] Гриффитс А. GCC. Настольная книга пользователей, программистов и системных администраторов. М., Диасофт, 2004.
- [14] Lattner C. LLVM. In: The architecture of open source applications. Lulu.com, 2012, pp. 155–170.
- [15] Xtext. Language engineering for everyone! *eclipse.org: веб-сайт*. URL: <http://www.eclipse.org/Xtext/index.html> (дата обращения: 01.05.2022).
- [16] Tree-sitter. *tree-sitter.github.io: веб-сайт*. URL: <http://tree-sitter.github.io/tree-sitter> (дата обращения: 01.05.2022).

- [17] SDK моделирования для Visual Studio — доменные языки. *docs.microsoft.com: веб-сайт*. URL: <http://docs.microsoft.com/ru-ru/visualstudio/modeling/modeling-sdk-for-visual-studio-domain-specific-languages> (дата обращения: 01.05.2022).
- [18] MPS — Meta programming system. *jetbrains.com: веб-сайт*. URL: <http://www.jetbrains.com/ru-ru/mps> (дата обращения: 01.05.2022).
- [19] Дмитриев С. Языково-ориентированное программирование. *rsdn.org: веб-сайт*. URL: <http://rsdn.org/article/philosophy/LOP.xml> (дата обращения: 01.05.2021).
- [20] Фаулер М. Предметно-ориентированные языки программирования. М., Вильямс, 2011.
- [21] Roci M., Salehi N., Amir S., et al. Towards circular manufacturing systems implementation: a complex adaptive systems perspective using modelling and simulation as a quantitative analysis tool. *Sustain. Prod. Consum.*, 2022, vol. 31, pp. 97–112. DOI: <https://doi.org/10.1016/j.spc.2022.01.033>
- [22] Roci M., Salehi N., Amir S., et al. Multi-method simulation modelling of circular manufacturing systems for enhanced decision-making. *MethodsX*, 2022, vol. 9, art. 101709. DOI: <https://doi.org/10.1016/j.mex.2022.101709>

Гордин Михаил Валерьевич — канд. техн. наук, и.о. ректора МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Иванова Галина Сергеевна — д-р техн. наук, профессор кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Пролетарский Андрей Викторович — д-р техн. наук, профессор, декан факультета «Информатика и системы управления» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Фетисов Михаил Вячеславович — старший преподаватель кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Просьба ссылаться на эту статью следующим образом:

Гордин М.В., Иванова Г.С., Пролетарский А.В. и др. Адаптивный подход к реализации сложных систем моделирования. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2022, № 3 (140), с. 18–33. DOI: <https://doi.org/10.18698/0236-3933-2022-3-18-33>

ADAPTIVE APPROACH TO IMPLEMENTATION OF COMPLEX SIMULATION SYSTEMS

M.V. Gordin

G.S. Ivanova

A.V. Proletarskiy

M.V. Fetisov

gordinmv@bmstu.ru

gsivanova@bmstu.ru

pav@bmstu.ru

fetisov.michael@bmstu.ru

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The paper presents an approach potentially capable of implementing efficient import substitution in the field of modern software systems for simulating and designing manufacturing processes in various industries, as well as of compensating for the destruction of international connections that made it possible to build efficient labor division structures for developing modern software systems. To solve these problems, we propose to implement an adaptive simulation system that can adjust to different domains by means of considerable automation concerning development of domain-specific programming languages describing models in the relevant subject areas. We also propose to split development into the following competence and responsibility zones: developing an extensible core for the adaptive simulation system and shared language and visual (graphic) components; developing domain-specific extensions, which may include both language and visual (graphic) components; populating a specific extension with a database of industry standards; developing a manufacturing model specifically. Using a shared platform represented by the adaptive simulation system allows for reusing extensions from different domains so as to form complex simulations, resulting in possible technological breakthroughs at the juxtaposition of various industries. The paper also briefly describes certain risks associated with the implementation and maintenance of complex simulation systems in the Russian Federation. We consider alternatives to the solutions proposed

Keywords

Domain-specific programming language, problem-oriented programming language, manufacturing process simulation, simulation, domain, industry, simulation system, adaptive simulation system

Received 01.07.2022

Accepted 21.07.2022

© Author(s), 2022

REFERENCES

- [1] Glazyev S. Posledniy vitok liberalnoy globalizatsii [The last round of liberal globalization]. *izborsk-club.ru: website* (in Russ.). Available at: <http://izborsk-club.ru/11870> (accessed: 01.05.2022).

- [2] Smirnov G. Neispovedimye puti globalizatsii [Inscrutable ways of globalization]. *kommersant.ru: website* (in Russ.). Available at: <http://www.kommersant.ru/doc/5295109> (accessed: 01.05.2022).
- [3] Ivanova G.S., Zhiltsov A.I., Fetisov M.V., et al. Adaptive modeling system. *Avtomatizatsiya. Sovremennye tekhnologii*, 2020, no. 11, p. 500 (in Russ.).
- [4] Adaptivnaya sistema modelirovaniya SIMODO Stars [SIMODO Stars adaptive modeling system]. *bmstu.codes: website* (in Russ.). Available at: <http://bmstu.codes/lx/simodo/stars> (accessed: 01.05.2022).
- [5] Ivanova G.S., Fetisov M.V., Malkina T.A., et al. Unification of working with domain-specific languages and open software architecture in an adaptive modeling system. *Dinamika slozhnykh sistem — XXI vek* [Journal Dynamics of Complex Systems — XXI Century], 2021, vol. 15, no. 3, pp. 36–47 (in Russ.). DOI: <https://doi.org/10.18127/j19997493-202103-03>
- [6] Blender. *blender.org: website*. Available at: <http://www.blender.org> (accessed: 01.06.2021).
- [7] 1800–2017 — IEEE standard for SystemVerilog — unified hardware design, specification, and verification language. DOI: <https://doi.org/10.1109/IEEESTD.2018.8299595>
- [8] Ivanova G.S., Fetisov M.V. The concept of contract management in the base language of the adaptive modeling system. *SUMMA*, 2021. DOI: <https://doi.org/10.1109/SUMMA53307.2021.9632114>
- [9] Wirth N. Compiler construction. New York, Addison-Wesley, 1996.
- [10] BYACC — Berkeley Yacc — generate LALR(1) parsers. *invisible-island.net: website*. Available at: <http://invisible-island.net/byacc/byacc.html> (accessed: 01.05.2022).
- [11] GNU Bison. *gnu.org: website*. Available at: <http://www.gnu.org/software/bison> (accessed: 01.05.2022).
- [12] AnyLogic. *anylogic.ru: website*. Available at: <http://www.anylogic.ru> (accessed: 01.05.2022).
- [13] Griffith A. GCC: the complete reference. New York, McGraw-Hill Osborne Media, 2002.
- [14] Lattner C. LLVM. In: The architecture of open source applications. Lulu.com, 2012, pp. 155–170.
- [15] Xtext. Language engineering for everyone! *eclipse.org: website*. Available at: <http://www.eclipse.org/Xtext/index.html> (accessed: 01.05.2022).
- [16] Tree-sitter. *tree-sitter.github.io: website*. Available at: <http://tree-sitter.github.io/tree-sitter> (accessed: 01.05.2022).
- [17] SDK modelirovaniya dlya Visual Studio — domennyye yazyki [Modeling SDK for Visual Studio — domain languages]. *docs.microsoft.com: website* (in Russ.). Available at: <http://docs.microsoft.com/ru-ru/visualstudio/modeling/modeling-sdk-for-visual-studio-domain-specific-languages> (accessed: 01.05.2022).
- [18] MPS — Meta programming system. *jetbrains.com: website*. Available at: <http://www.jetbrains.com/ru-ru/mps> (accessed: 01.05.2022).

[19] Dmitriev S. Yazykovo-orientirovannoe programmirovaniye [Language-oriented programming]. *rsdn.org: website* (in Russ.).

Available at: <http://rsdn.org/article/philosophy/LOP.xml> (accessed: 01.05.2021).

[20] Fowler M. Domain-specific languages. New York, Addison-Wesley, 2011.

[21] Roci M., Salehi N., Amir S., et al. Towards circular manufacturing systems implementation: a complex adaptive systems perspective using modelling and simulation as a quantitative analysis tool. *Sustain. Prod. Consum.*, 2022, vol. 31, pp. 97–112.

DOI: <https://doi.org/10.1016/j.spc.2022.01.033>

[22] Roci M., Salehi N., Amir S., et al. Multi-method simulation modelling of circular manufacturing systems for enhanced decision-making. *MethodsX*, 2022, vol. 9, art. 101709. DOI: <https://doi.org/10.1016/j.mex.2022.101709>

Gordin M.V. — Cand. Sc. (Eng.), Acting Rector, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Ivanova G.S. — Dr. Sc. (Eng.), Professor, Department of Computer Systems and Networks, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Proletarskiy A.V. — Dr. Sc. (Eng.), Professor, Dean of Faculty of Information Technology and Control Systems, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Fetisov M.V. — Senior Lecturer, Department of Computer Systems and Networks, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Please cite this article in English as:

Gordin M.V., Ivanova G.S., Proletarskiy A.V., et al. Adaptive approach to implementation of complex simulation systems. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2022, no. 3 (140), pp. 18–33 (in Russ.).

DOI: <https://doi.org/10.18698/0236-3933-2022-3-18-33>