

СОЗДАНИЕ КИБЕРФИЗИЧЕСКИХ СИСТЕМ: ПРОБЛЕМЫ ПОДГОТОВКИ ИТ-СПЕЦИАЛИСТОВ

АЛЕКСЕЙ ПЛАТУНОВ

aeplatonov@itmo.ru

ВАСИЛИЙ ПИНКЕВИЧ

vupinkevich@itmo.ru

Двойственная природа киберфизических систем (КФС) порождает борьбу прикладных и компьютерных специалистов за лидерство в междисциплинарных проектах. Преодолеть эту проблему возможно за счет подготовки «системообразующих» ИТ-специалистов, готовых к погружению в самые разные прикладные области. Как формировать такого специалиста — тема данной статьи.

ВВЕДЕНИЕ

На наших глазах происходит глубокая интеграция и трансформация различных отраслей хозяйства, областей деятельности, сегментов науки и техники на платформе инфокоммуникационных технологий (ИКТ, или просто ИТ). Это радикально меняет жизнь человека, среду его обитания, способы общения. Также меняется и сама вычислительная техника.

Наибольшую долю рынка ИКТ традиционно занимают задачи прикладного характера, направленные на внедрение вычислительной техники и получение от нее реальной отдачи. Вместе с тем естественно возникает и стремительно развивающееся разделение труда в инфокоммуникационной (ИК) индустрии и то, что область ИКТ еще не оформилась как традиционная наука, несет в себе

значительные технологические риски и угрозы. Отсутствует даже единый термин, отражающий все аспекты деятельности в ИК-области.

Тенденция к созданию глубоко интегрированных прикладных систем на ИК-платформе предполагает системный подход к проектированию с равными приоритетами разработчиков всех ключевых профилей. Сегодня такие комплексные системы ассоциируются с КФС (или CPS), социокиберфизическими системами (СКФС, SCPS) и «Интернетом вещей» (IoT).

В данной статье мы обсудим общие технические и технологические проблемы создания КФС и IoT-систем прежде всего со стороны их «вычислительной» природы и вопросов подготовки ИТ-специалистов нового поколения для успешных разработки и использования таких систем.

КФС, IoT, ВСТРАИВАЕМЫЕ СИСТЕМЫ

Согласно одному из официальных определений, КФС — это созданные в результате инженерной деятельности системы, которые основаны на бесшовной интеграции вычислительных алгоритмов и физических компонентов и зависят от этой интеграции (National Science Foundation, США; здесь и далее приводится авторский перевод терминов и определений). КФС могут быть небольшими и автономными, как, например, искусственная поджелудочная железа, или очень большими, сложными и распределенными (региональная энергосистема).

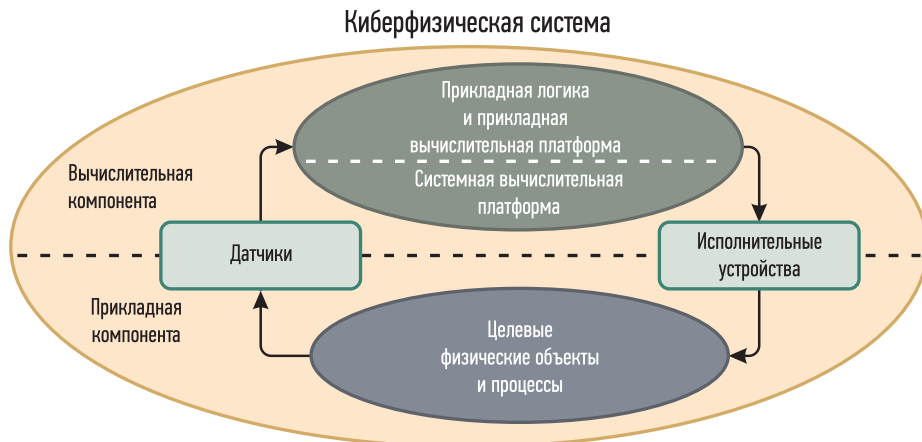
Как отмечают многие специалисты (например, в [1, 2]), к появлению таких категорий систем, как КФС и IoT, привело естественное развитие встраиваемых вычислительных систем (BcC, ES), сетевых, или распределенных, BcC (PBcC, NES) и беспроводных сенсорных сетей (WSN).

Все входящие в состав КФС вычислительные и коммуникационные средства, включая BcC, коммуникационную инфраструктуру, базы данных, облачные сервисы, мобильные приложения, мы будем называть вычислительной компонентой КФС, а «физическую» составляющую — прикладной (рис. 1).

Можно выделить две основные трактовки того, что следует относить к КФС:

- «Узкая» трактовка: это новый отдельный класс систем, ИК-платформа которых характеризуется

РИС. 1. ▽
Обобщенная структура КФС



тесной связью с объектом мониторинга и/или управления, распределенностью, интеллектуальными алгоритмами (AI, Big Data, VR/AR), широким использованием вычислительных сетей различного масштаба. Требуют «особых» методов проектирования.

- «Широкая» трактовка: это новое поколение технических систем, которое характеризуется качественным изменением всей технологии проектирования и внедрением новых технических и технологических решений в части ИК-платформы (в том числе AI, IoT, Big Data, VR/AR) для получения качественно новой функциональности. Включает узкую трактовку и не отрицает возможности проектировать системы любого назначения и масштаба по новым принципам.

В статье мы будем придерживаться широкой трактовки, считая ее более перспективной. Например, узкая трактовка относит к КФС системы автоматизации производства, отвечающие критериям инициативы «Индустрия 4.0», но исключает «простые» системы автоматизации, хотя они могут отличаться только набором функций, а не принципом проектирования.

ВсС остаются наиболее характерными представителями вычислительной компоненты КФС. Их можно определять как вычислительные системы (ВС), непосредственно взаимодействующие с физическим объектом мониторинга и/или управления, объеди-

ненные с таким объектом логикой работы и, часто, единой конструкцией [3]. Это иногда приводит к ложному восприятию: либо ВсС (без физического объекта) отождествляются с КФС, либо системы, в которых вычислительная компонента представлена в виде традиционной ВсС (без «модных» технологий, но с использованием методов бесшовной интеграции), считают «неполноценными» КФС, забывая, что мода проходит.

IoT не является обязательным компонентом КФС и обычно используется в качестве унифицированной коммуникационной среды в открытых распределенных системах автоматизации. По определению стандарта ISO/IEC 20924:2018, IoT — это инфраструктура взаимосвязанных и взаимодействующих объектов, людей, систем и информационных ресурсов вместе со службами, которые обрабатывают информацию и реагируют на сведения из физического и виртуального мира.

Появление КФС, на наш взгляд, связано с двумя основными факторами:

- отрасль ИКТ достигла состояния, которое позволяет широко внедрять ИК-компоненты в системы и процессы практически любых масштаба и назначения;
- созданы доступные массовому разработчику методы и технологии, дающие возможность эффективно и относительно дешево проектировать комплексные системы с большой долей ИК-компонентов.

Благодаря разнообразию областей применения КФС востребованным становится качественно новый по сложности уровень автоматизации, позволяющий проектировать и массово производить системы, которые иначе было бы невозможно создать в принципе (например, беспилотные автомобили). В таких, на первый взгляд, далеких от компьютеров системах вычислительная компонента начинает играть ведущую роль, а обязанности ИТ-специалиста расширяются до полноценного системного инженера.

Сегодня реализация сбалансированного киберфизического подхода остается скорее искусством, чем отработанной технологией. Существующие методики и маршруты проектирования тяготеют либо к физикоцентричному, либо к компьютероцентричному подходу. В первом случае основные усилия направлены на разработку прикладной компоненты КФС с ограниченным учетом возможностей и рисков при создании вычислительной компоненты, что может привести к критическим проблемам с надежностью и эффективностью системы в целом. Во втором случае вычислительная компонента создается до начала глубокого погружения в прикладную функциональность. К данным крайностям обычно тяготеют специалисты без достаточного кругозора, вынужденно играющие роль системного инженера в проектах КФС.

На рис. 2 показана эволюция управляющих ВС, их элементной

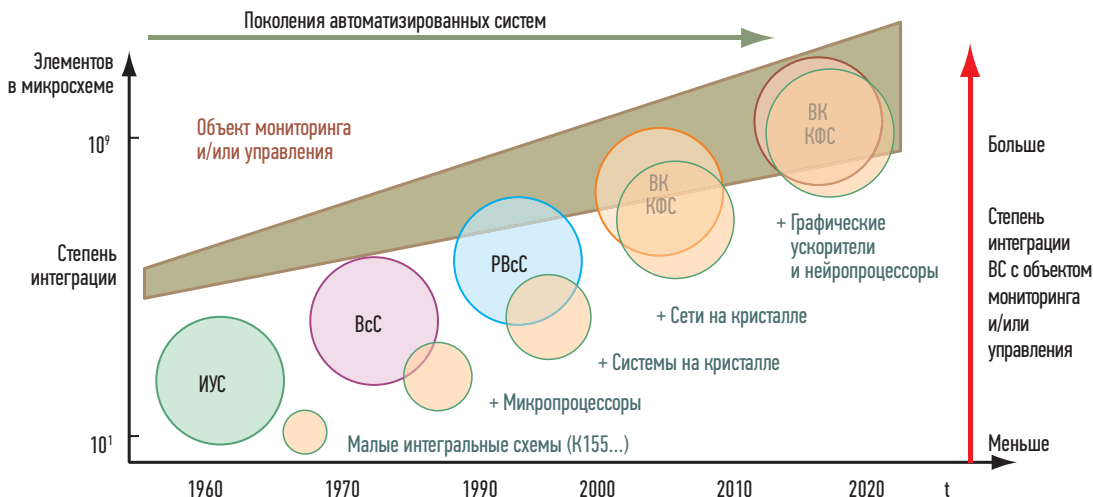
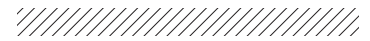


РИС. 2. ← Эволюция вычислительной компоненты КФС (ИУС — информационно-управляющие системы, ВК — вычислительная компонента)



базы, рост степени их интеграции с объектом мониторинга и/или управления. Сегодня методы проектирования элементной базы все больше приближаются к методам проектирования управляющих ВС, особенно на уровне вычислительной архитектуры. Современная система на кристалле может «закрывать» значительную часть вычислительных задач КФС.

ПРОБЛЕМЫ ПРОЕКТИРОВАНИЯ ВЫЧИСЛИТЕЛЬНОЙ КОМПОНЕНТЫ КФС

В ИК-отрасли есть общие проблемы, которые профильные ИТ-специалисты часто даже не ощущают, но последствия от их нерешенности радикально ухудшают условия и результаты проектирования, что особенно сильно заметно в сфере создания КФС.

Отставание методов системного проектирования в ИК-отрасли

Необходимость массового проектирования КФС приводит к резкому росту междисциплинарности и сложности проектов «в пересчете» на один коллектив. Вместе с новыми прикладными вычислительными технологиями это обостряет проблемы роста ИК-индустрии. Хроническое отставание методов и инструментальных средств разработки ВС не позволяет в полной мере использовать технологический потенциал ИК-отрасли в новых проектах.

Из-за объективных трудностей проектирования со сквозной оптимизацией технических решений по множеству вариантов во всех стеках технологий задачу сводят либо к крупноблочному проектированию из готовых компонентов и по готовым шаблонам, либо к заказному проектированию с использованием ряда навязанных или выбранных без должного обоснования макро- и микроархитектурных решений. Так, в рамках используемых сейчас индустриальных методов процесс проектирования КФС жестко разделен на ряд фаз, которые поддерживаются инструментами (САПР). Соответственно, существует узкая специализация разработчиков, которая формируется в вузе и совершенствуется в процессе

практической деятельности. Можно выделить три сегмента деятельности, непосредственно относящихся к созданию вычислительной компоненты КФС:

- проектирование систем на кристалле (включая технологии ASIC, ASIP, FPGA);
- проектирование специализированных контроллеров и ВС на их основе;
- проектирование систем автоматики на базе готовых программируемых логических контроллеров, SCADA-систем, технологий информационных систем.

Внутри сегментов процессы проектирования выстроены в виде эффективных и почти «бесшовных» маршрутов, степень автоматизации которых постоянно растет от рутинных операций уровня инженерной реализации к уровням проектирования, которые сегодня слабо формализованы. В каждом сегменте представлены канонические решения, позволяющие с приемлемым качеством покрывать значительную часть сегодняшних задач. Однако при кажущемся благополучии ситуации использование ограниченного числа типовых проектных ИК-платформ для множества разнообразных задач снижает качество создаваемых продуктов. Использование существующих методов и средств проектирования не позволяет выполнять адекватный анализ вариантов и влиять на внутреннее, глубинное устройство платформы, чтобы эффективно решать многие вопросы, среди которых:

- нехватка вычислительной мощности;
- большое энергопотребление;
- сложность программирования задач реального времени (РВ);
- сложность реализации и программирования гетерогенных многопроцессорных структур;
- низкая информационная безопасность.

Необходимо констатировать, что сегодня отсутствуют эффективные языковые средства и способы документирования, дающие возможность даже просто описать структуру, принципы и механизмы функционирования современных ВС с многоуровневой и гетерогенной организацией, которыми (во многом умозрительно) оперирует

ИТ-специалист, детализируя архитектуру КФС. Следствие — высокая непрозрачность программных и аппаратных средств и технологий, создаваемых в ИК-отрасли. Описания технических решений называются «раздробленными» на части по технологическому принципу, во многих случаях сводятся только к конечным техническим документам (принципиальные электрические схемы, тексты программ, технические описания сложн-функциональных микросхем).

Анализ таких документов для получения достоверной информации о работе системы или ее части («реверс-инжиниринг») крайне трудоемкий сам по себе и не способствует получению цельной картины для специалиста-проектировщика (архитектора), вынужденного работать с полным стеком технологий. Таким образом, результаты заказного проектирования системы или ее компонентов, полученные с использованием существующих низкоуровневых технологий проектирования и программирования, плохо масштабируются и малопригодны для повторного использования. Ситуацию усугубляет часто проявляющееся «плоское» мышление специалиста — нежелание или невозможность оперировать уровнями представлениями вычислительной системы и ее частей.

Тем не менее практика показывает, что при наличии команды высококвалифицированных разработчиков возможно получать эффективные решения с высокой степенью повторного использования при умеренных затратах, что подтверждает отсутствие адекватных методик проектирования, доступных массовому разработчику. Поэтому крайне необходимы и по-прежнему требующими научной активности остаются технологии и инструменты, принадлежащие ко всем основным задачам создания как специализированных ВС, так и вычислительных платформ широкого назначения. Приведем примеры таких областей:

- формальные методики проектирования;
- системная архитектура;
- модельно-ориентированная инженерия;
- многоядерные и многопроцессорные платформы;

- модели вычислений;
- смешанные системы с критическими функциями (mixed-critical systems);
- системы жесткого РВ.

Принцип Коцепа и адаптивность КФС

Пример кардинально различающихся точек зрения, определяющих наполнение и ход проекта КФС, сформулирован в так называемом принципе Коцепа [4]:

- Многие (прогнозируемые) свойства, которые мы приписываем системам (детерминизм, своевременность реакции, надежность, безопасность), на самом деле являясь не свойствами реализованной системы, а скорее свойствами модели системы.
- Опираясь на выбранные модели, мы можем делать выводы о свойствах реализации системы. Достоверность этих выводов зависит от качества модели, а модель всегда является приблизительной.

На практике система почти никогда полностью не совпадает с ее восприятием инженером (модель системы), хотя бы потому, что нет полной информации об исползованных в проекте сторонних компонентах. Важно не приписывать безоговорочно реализации системы свойства модели, а напротив, рассчитывать на возможность проявления неожиданных свойств системы, в том числе при поиске причин ее неадекватного функционирования.

При создании КФС, имея дело со сложной, интеллектуальной автоматикой, необходимо закладывать в проект значительный запас гибкости, дополнительные страхующие механизмы и решения, реализуя принцип адаптивности не только на уровне прикладных алгоритмов, а по возможности на всех уровнях организации системы. Сегодня создание таких адаптивных систем — открытая научная и техническая проблема, решение которой в области вычислительной компоненты КФС требует пересмотра парадигмы проектирования.

Приоритеты в проектировании КФС

По замыслу авторов термина «КФС», усилия научно-технического

профильного сообщества должны быть в приоритетном порядке направлены на решение проблем вычислительной платформы КФС, на развитие системного и параллельного (совместного) стиля проектирования, а также на вопросы подготовки кадров [2, 5].

Однако у значительной части специалистов существует иллюзия отсутствия фундаментальных проблем в области проектирования ВС, что приводит к увеличению массы незрелого и непрозрачного продукта и смещению приоритетов в исследованиях в ИК-сфере. Освоение и развитие системных, кросс-уровневых, сквозных технологий и инструментов проектирования в области вычислительной техники отодвигается на второй план. Вместо этого значительные силы в последнее время оказались направлены на развитие новых, безусловно актуальных, областей (IoT, Big Data, AI, Digital Industry, VR/AR и др.), за счет чего значительно сократилось финансирование и снизилась престижность работ, решающих важнейшие открытые проблемы фундамента вычислительной техники.

Для повышения качества проектирования КФС должен быть восстановлен баланс в направлениях исследований и в подготовке ИТ-специалистов.

ПОДГОТОВКА СПЕЦИАЛИСТОВ ДЛЯ ПРОЕКТИРОВАНИЯ КФС

Структура компьютеринга

Для обозначения всей области ИКТ мы считаем удачным используемый в документах АСМ и IEEE [6] термин «компьютинг», который определяется как целенаправленная деятельность, требующая компьютеров, использующая компьютеры и создающая компьютеры [7, 8]. С течением времени область компьютеринга и сообщество ИТ-специалистов распадается на отдельные сегменты, которые все больше изолируются друг от друга. Приведем определения основных направлений компьютеринга (в скобках указаны документы-источники из [6]):

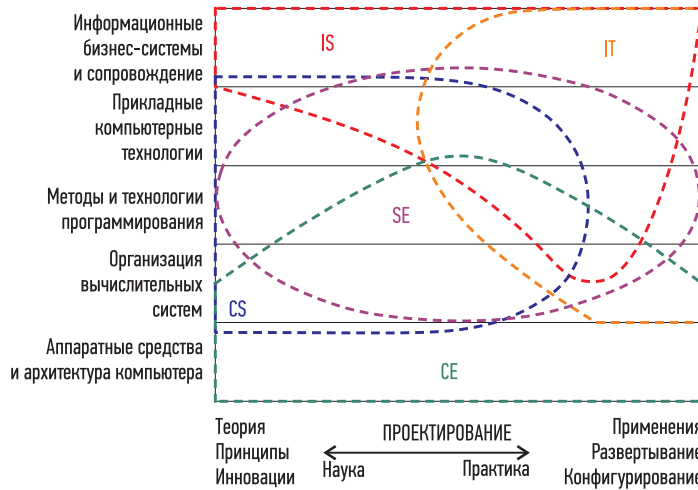
- Компьютерная инженерия (CE) охватывает науку и технологию проектирования, конструиро-

вания, внедрения и обслуживания программных и аппаратных компонентов современных вычислительных систем, оборудования с компьютерным управлением и сетей интеллектуальных устройств (CE2016).

- Информатика (CS) охватывает широкий диапазон: от теоретических и алгоритмических основ компьютеринга до передовых разработок в области робототехники, компьютерного зрения, интеллектуальных систем, биоинформатики и в других перспективных областях (CC2005).
- Программная инженерия (ПИ, SE) — применение систематического, дисциплинированного, поддающегося количественной оценке подхода к разработке, эксплуатации и обслуживанию программного обеспечения (ПО), то есть приложение инженерии к ПО (SE2014).
- Информационные технологии (IT) — это изучение и использование системных методик выбора, разработки, применения, интеграции и администрирования безопасных вычислительных технологий, позволяющих пользователям достигать своих личных, организационных и социальных целей (IT2017).
- Информационные системы (IS) — область, специалисты которой сосредоточены на интеграции решений ИТ и бизнес-процессов для удовлетворения информационных потребностей предприятий, что позволяет последним эффективно и действительно достигать своих целей. Упор делается на технологии создания, обработки и распространения информации (CC2005).
- Кибербезопасность (CSEC) — компьютерная дисциплина, в которой задействованы технологии, люди, информация и процессы, обеспечивающие гарантии деятельности в контексте существования и действий противников (CSEC2017).

Состав направлений компьютеринга продолжает расширяться. CSEC выделяется с 2017 г. С 2020 г. кандидатом на отдельное направление считается наука о данных (DS). На рис. 3 показаны зоны ответственности ИТ-специалистов, принадлежащих основным

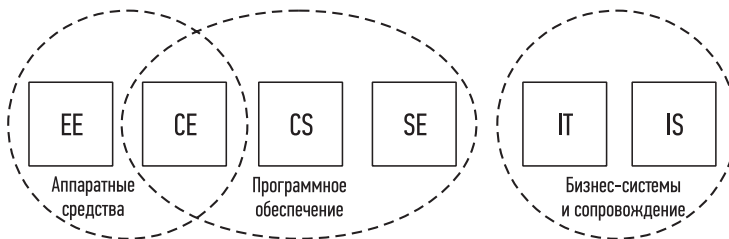
РИС. 3. ► Зоны ответственности специалистов основных направлений компьютеринга [7]



До 1990-х:



После 1990-х:



2020:

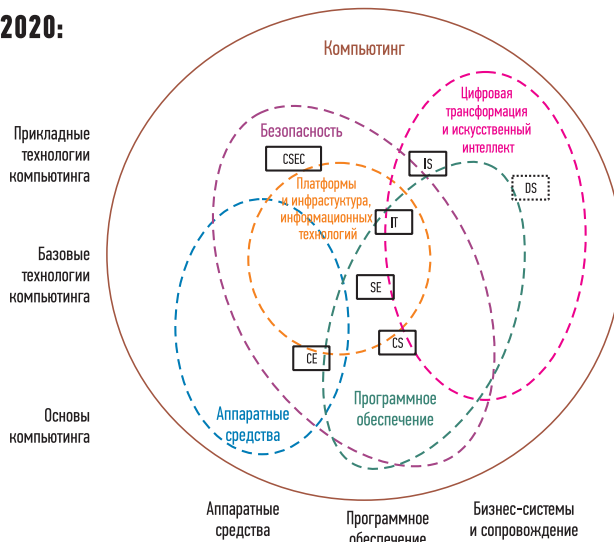


РИС. 4. ► Эволюция областей деятельности внутри компьютеринга (EE — электронная инженерия) [7]

направлениям компьютеринга [7], а на рис. 4 — изменение соотношения этих направлений во времени.

Проблемы массового применения программных технологий

Сегодня программные технологии активно выходят за пределы «чистого» программирования: проектирование встроенного ПО (embedded software), проектирование аппаратных средств (языки описания аппаратуры, HDL), архитектурное проектирование (языки описания архитектуры, ADL) и т. д.

Прикладные области науки и техники (например, автоматика, робототехника, компьютерная графика, физика) активно применяют сегменты компьютеринга IS, SE, CS, основанные на программных технологиях. Это создает иллюзию «низкого порога вхождения» для работы в ИК-отрасли, поэтому прикладные специалисты мигрируют в компьютеринг. Аналогичная ситуация с миграцией в смежные сегменты компьютеринга наблюдается среди специалистов по аппаратным средствам ВС. В результате в сфере ИКТ на всех позициях высок процент игроков-самоучек без профильного образования и с ограниченным профессиональным кругозором.

К типовым проблемам применения программных технологий перечисленными категориями специалистов можно отнести следующие:

- программисты, как правило, не экономят процессорное время и ресурсы памяти, что приводит к написанию медленных и больших по размеру программ;
- программисты часто считают, что все системы идеальны; такой взгляд приводит к катастрофическим результатам при сбоях оборудования, прикладного или системного ПО (вспомним принцип Коцепа);
- непонимание принципов многопоточных вычислений и параллельной работы процессов;
- часто подготовка программистов не включает глубокое изучение моделей вычислений и архитектуры, системной инженерии, архитектурного проектирования ПО.

Негативное влияние на сферу ИКТ оказывают общие проблемы программирования и ПИ:

- преобладают методологии, направленные на организацию менеджмента проектов, а не на синтез технических решений;
- существует большое число технологий, дублирующих друг друга, но практически отсутствуют инструменты их сравнения; при этом темпы смены технологий высокие и продолжают расти;
- программирование и ПИ часто не разделяются;
- сложность фиксации фактов ошибок, их диагностики в работе как ПО, так и ВС в целом;
- хроническое перегревание рынка труда.

Профиль ИТ-специалиста для КФС

Направления компьютеринга частично пересекаются, но по сути сильно различаются и образуют собственные парадигмы, в которых существуют и способны генерировать решения соответствующие специалисты. Внутри парадигмы специалист обычно замыкается в узкой профессиональной нише, из которой выходит с трудом. За пределы парадигмы он не выходит почти никогда.

Полезно выделять компетенции ИТ-специалистов, относящиеся к созданию заказной программно-аппаратной платформы (CS, SE, VC, SE в части системного ПО) и к предметно-ориентированным программным компьютерным технологиям (CS, SE, IS, CSEC).

Зоны ответственности, структуру и охват профессиональной области специалиста демонстрирует известная I-, T-, π-модель. «Ножки» символа обозначают узкие области специализации, горизонтальная «полка» — кругозор и возможность взаимодействия со смежными областями. Для создания вычислительной компоненты КФС необходимы как минимум T-, а лучше π-специалисты, не находящиеся в плену шаблонов, навязываемых ограниченностью сферы ответственности. Из π-специалистов формируется элита проектирования — архитекторы. I- и T-специалисты выступают носителями экспертных знаний в конкретных узких областях деятельности и полезны прежде всего на этапе реализации проекта.

Для создания КФС ИТ-специалисты из разных областей компью-

тинга должны обладать общей системой понятий (нуждается в развитии и внедрении), кросс-доменным, кросс-уровневым языком спецификации технических решений, быть способны к эффективному распределению вычислительных задач по компонентам платформы, интеграции гетерогенных платформ и компонентов и т. п. Для будущих создателей вычислительной компоненты КФС необходимо обеспечить всестороннее образование в сфере компьютеринга для создания достаточно широкой горизонтальной «полки» и значительное погружение в несколько тем, желательно далеко разнесенных друг от друга (например, ПО и программируемая логика).

Желание определить область компетенций для подготовки системообразующего КФС-специалиста заставляет приоритетно объединять направления SE (в том числе VC) и IS, дополняя их разделами из CS, SE, CSEC, IT, теории автоматического управления, мехатроники. Это необходимо для проектирования с гораздо более глубоким погружением в организацию вычислительной компоненты КФС, чем это обычно делают в сфере IS при создании бизнес-систем.

Также необходимо обеспечить знакомство будущих специалистов с прикладной физико- и бизнес-аналитикой, моделями описания соответствующих процессов, технологиями программирования, тестирования, соответствующими разделами математики. Обычно ИТ-специалисты в основном взаимодействуют с интеграторами или с прикладными специалистами, а не напрямую с объектом, что ограничивает возможность полноценно учитывать предметную область.

Очевидно, что это ставит сложную задачу формирования обозримой учебной программы с правильным балансом знаний и расстановкой акцентов. Можно утверждать, что для создания такой учебной программы не хватает зрелости как самой области КФС, так и компактного интегрированного представления необходимых компетенций в вопросах:

- быстрого погружения в конкретные прикладные области (навыки системной аналитики в широком

диапазоне, от физики до бизнес-процессов);

- комплексного (full-stack) проектирования специализированных вычислительных систем.

ПРИМЕР ПРОФИЛЯ ПОДГОТОВКИ КФС-СПЕЦИАЛИСТА

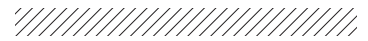
Сегодня представлен целый ряд предложений по наполнению учебной программы подготовки КФС-специалиста. Они различаются объемом (от вводных курсов до полноценных магистерских программ), наполнением (зависит от трактовки термина КФС), степенью связи с компьютерингом. Подробный обзор учебных программ представлен в [9].

Широту взглядов на наполнение области КФС и ее взаимосвязь со смежными дисциплинами, в том числе на взаимоподчиненность и связь с сегментами компьютеринга, демонстрируют документы комитетов ТК 194 «Кибер-физические системы» Росстандарта РФ и ISO/IEC JTC 1/SC41 “Internet of Things and digital twin”.

В [10] перечисляются следующие разделы подготовки КФС-специалиста:

- математика, статистика;
- электроника, компьютерная архитектура;
- компиляторы, операционные системы;
- встроенное ПО;
- оптимизация;
- модели вычислений, формальные методы;
- Интернет и веб-ПО, беспроводная связь;
- информационная безопасность;
- системы управления, системная инженерия;
- человеко-машинное взаимодействие;
- работа в команде, управление проектами;
- управление развертыванием и конфигурирование;
- процессный подход, проблемы жизненного цикла;
- стандарты, экологичность и др.

Одним из наиболее полных и интересных документов, предлагающих развернутый набор компетенций специалиста по КФС, можно считать [7]. Приведем примеры из раздела “Foundations for a CPS curriculum”:



- Базовые вычислительные концепции, помимо тех, что рассматриваются во вводных курсах по программированию, такие как ВСС, структуры данных, теория автоматов и ПИ.
- Организация вычислений с учетом особенностей и ограничений объектов физического мира, что предполагает понимание:
 - свойств физического мира;
 - организации и особенностей ВСС РВ;
 - ограничений вычислительных ресурсов, таких как энергопотребление и размер памяти.
- Владение дискретной математикой и такими разделами непрерывной математики, как дифференциальные уравнения, вероятностные и случайные процессы, линейная алгебра.
- Проектирование вычислительной компоненты КФС, объединяющее:
 - применение датчиков, исполнительных устройств, алгоритмов управления, коммуникационных протоколов и систем, вычислений, предполагающее их междоменное и межуровневое (cross-cutting) взаимодействие;
 - учет центральной роли взаимодействия между физическими и вычислительными аспектами;
 - контроль над коммуникационными сетями, получением данных от датчиков, обработкой сигналов и выдачей управляющих воздействий с учетом ограничений РВ.
- Моделирование неоднородных и динамических систем, объединяющих управление, вычисления и связь, с акцентом на неопределенность свойств объекта и гетерогенность системы, включая такие методы, как линейное и нелинейное моделирование, стохастические модели, дискретно-событийные и гибридные модели, а также связанные с ними методологии проектирования, основанные на оптимизации, теории вероятностей и динамическом программировании.
- Методы проектирования КФС, особенно для критически важ-

ных по безопасности, надежности и отказоустойчивости применений, требующие охвата фаз жизненного цикла от технического задания до тестирования, сертификации и сопровождения, включая вопросы формальной верификации и валидации, а также возможности адаптации, позволяющие обеспечивать эволюционирование системы.

ВЫВОДЫ

Для повышения качества проектирования КФС нам видится необходимым вести работу по следующим направлениям:

- Повышение качества проектирования вычислительной компоненты КФС.

ВС в КФС являются заказными в силу уникальности требований. Их проектирование остается крайне трудозатратным процессом с низкой степенью повторного использования и, как правило, недостаточно высоким качеством. Мы видим возможности решения этих проблем в усилении этапов архитектурного/микроархитектурного проектирования, формализации абстракций и инструментов проектирования на базе единой системы понятий в сфере компьютеринга. Необходимо существенно повысить прозрачность и степень повторного использования всех проектных решений, принимаемых до этапа реализации. Требуется увеличивать глубину проработки функциональной и логической организации системы с качественным содержательным документированием. Специалисты должны осваивать и развивать соответствующие методики и инструменты.

- Повышение качества подготовки ИТ-специалистов.

Необходимо усиливать учебные программы ИТ-специалистов в части знаний и навыков по опорным технологиям вычислительной техники. Например, необходимо знакомить прикладных ИТ-специалистов с базовыми механизмами системной вычислительной платформы.

- Подготовка КФС-специалистов.

Сегодня в проектировании КФС центральная роль все чаще переходит от прикладных специалистов к ИТ-специалистам. Однако существует явный дефицит КФС-специалистов с глубоким понима-

нием компьютеринга. Эксперименты по подготовке таких специалистов в ведущих вузах продолжают уже около 10 лет [7]. Наиболее удачные, на наш взгляд, наборы компетенций представлены выше. На повестке дня стоит создание реальных образовательных программ, позволяющих с приемлемым качеством охватить весь необходимый объем знаний.

Направление проектирования КФС активно развивается в Университете ИТМО. На мегафакультете компьютерных технологий и управления представлены основные направления компьютеринга и другие важнейшие сегменты проектирования КФС (автоматика, мехатроника, «умные производства»). Специалисты мегафакультета развивают методы проектирования КФС, участвуют в создании конкретных систем. Совместно с ведущими научными и промышленными фирмами-партнерами университет формирует магистерские образовательные программы, направленные на подготовку специалистов по различным аспектам создания КФС, IoT и ВСС. ●

ЛИТЕРАТУРА

1. Lee E. A., Seshia S. A. Introduction to Embedded Systems. A Cyber-Physical Systems Approach. Second Edition. MIT Press, 2017.
2. A 21st Century Cyber-Physical Systems Education. National Academies of Sciences, Engineering, and Medicine. Washington, DC: The National Academies Press, 2016.
3. Платунов А. Е. Встраиваемые системы управления // Control Engineering Россия. 2013. № 1.
4. Kopetz H. Real-Time Systems. Design Principles for Distributed Embedded Applications. Springer US, 2011.
5. Cyber-Physical Systems. <https://ptolemy.berkeley.edu/projects/cps/>.
6. Association for Computing Machinery. Curricula Recommendations. www.acm.org/education/curricula-recommendations.
7. Computing Curricula 2020: Paradigms for Global Computing Education. Association for Computing Machinery, IEEE Computer Society. 2020 December 31.
8. Перекатов В.И. Эталоны компьютерной науки в американском высшем образовании (1968–2001). Ч.1, 2 // Высшее образование сегодня: Реформы. Нововведения. Опыт = Higher education today. 2003. №1; 2003. №2.
9. Marwedel P., Mitra T., Grimheden M. E., Andrade H. A. Survey on Education for Cyber-Physical Systems // IEEE Design & Test. 2020. Vol. 37. № 6.
10. Grimheden M. E., Törnqvist M. Towards curriculum guidelines for Cyber-Physical Systems // Proceedings of the WESE'14: Workshop on Embedded and Cyber-Physical Systems Education. October 2014. Article № 7.