

МАРКО БЕТТОНИ

(Базельский технологический институт, Швейцария)

Кант и кризис программного обеспечения¹. Предложения по построению программных систем, ориентированных на человека²

Что я могу знать?
Что я должен делать?
На что я могу надеяться?

И. Кант, 1787

1. Введение

До сих пор в компьютерной науке в целом, и искусственном интеллекте³ в частности, доминируют технологически ориентированные подходы. Однако это порождает внутренние трудности при разработке систем, адаптированных к человеческой способности мышления. В моем поиске идей для построения систем программного обеспечения, ориентированных на человека, я столкнулся с работами Канта⁴, в частности, с его моделями познания и рассуждений⁵.

2. Кризис программного обеспечения: факты, недостатки и предпринимаемые шаги

Примерно 20 лет назад, когда я приобретал свой первый опыт в использовании больших компьютерных систем и программных продуктов в *Computer Center of ETH* в Цюрихе, я быстро понял, что качество этих продуктов значительно ниже, чем то, в котором я нуждался. Однако большинство пользователей, по-видимому, было вполне этим качеством удовлетворено.

Для меня до сих пор ситуация реально не улучшилась, хотя программное обеспечение стало значительно более мощным. Мало того, для большинства пользователей положение ухудшилось — их требования возросли, а качество программного продукта возросло незначительно.

Этот кризис программного обеспечения, т. е. тот факт, что качество современных систем программного обеспечения в общем⁶ является недостаточным, недавно был отмечен А. Сэйджем и О. Палмером⁷. Согласно их мнению, в настоящее время только 2% производимого программного продукта могут быть эффективно использованы сразу же после поставки, 3% могут быть использованы только после доводки, 19% требуют существенной переработки, 47% уже через достаточно короткое время после поставки вообще не используются — 21% в силу

своей непригодности, 26% в силу своей некорректности — и 29% настолько плохи, что даже не поставляются.

Многие исследования продуктивности систем программного обеспечения, которые анализируют это положение, упоминают следующие главные дефекты: 1) программное обеспечение является слишком дорогим, 2) оно часто превышает планируемую стоимость, 3) его поставки часто запаздывают, 4) документация очень бедна, 5) его нельзя согласовать с остальным программным обеспечением, 6) его нельзя перенести в новую среду или 7) улучшить для того, чтобы оно развивалось вместе с потребностями пользователя, 8) его эксплуатация затруднительна и чревата ошибками, 9) характеристики работы нестабильны, 10) они не соответствуют описаниям, 11) возможности программного обеспечения не соответствуют тому, что ожидается и требуется, 12) программное обеспечение трудно использовать и 13) системные и программные требования, используемые для развития систем программного обеспечения, не соответствуют потребностям пользователей.

Эти же исследователи связывают упомянутые дефекты с этапами производства программного обеспечения, т. е. с: а) методами и средствами разработки, б) методологией, в) управлением проектированием. Дисциплина, которая объединяет все эти три области и имеет своей задачей разработку способов и средств улучшения качества программного продукта, называется *программной инженерией* (Software Engineering).

Однако, как показывает современный кризис, программная инженерия не добилась успеха в преодолении указанных выше трудностей. По моему мнению, одной из важных причин этого является неполный анализ существующего положения дел. Обычно такой анализ сводится к исследованию дефектов и соответствующих шагов по их преодолению, в то время как подходы, которые стоят за этими шагами, почти никогда не рассматриваются. Однако именно анализа этих подходов нельзя избежать, если вы хотите улучшить шаги процедуры, которые зависят от этих подходов.

3. Причины кризиса программного обеспечения

Многие дефекты программного обеспечения, которые мы рассматривали выше, имеют дело с подходами и мерами, существующими в области *методов определения (definition)*. В этой области было затрачено много усилий, как это показывают распространенные методы объектно-ориентированных и основанных на знаниях разработок программного обеспечения. Эти методы должны вносить вклад в улучшение качества программного обеспечения путем создания возможностей построения систем, адаптированных к человеческой способности мышления.

Их основная сила состоит в том, что главную роль в них

играет *моделирование*⁸. Однако подход к моделированию, а равно и лежащий в его основе подход к знаниям, не были в достаточной степени критически проанализированы в рамках этих методов. Именно поэтому до сих пор было невозможно использовать главную силу этих двух методов; по-моему, это негласное методологическое основание того, что многие серьезные дефекты программного обеспечения все еще существуют, несмотря на широкую распространенность этих методов.

Два примера помогут продемонстрировать установившийся в настоящее время подход к программной инженерии:

1. В своем докладе по проекту «Информационная модель контроля за процессом производства» — проекту, который был разработан смешанным методом: одновременно и объектно-ориентированным, и основанным на знаниях — автор пишет: «Если, так сказать, «приподнять крышу фабрики», то обнаружатся уже готовые объекты с различными значениями их характеристик, т. е. с различными состояниями»⁹.

2. Автор книги «Разработка объектно-ориентированных программ», всемирно известный эксперт Б. Майер высказывает позицию по поводу спорного вопроса о том, каким образом могут быть обнаружены системные объекты. Он пишет: «И в физической, и в абстрактной реальности объекты моделируются и ждут, чтобы их прочли»¹⁰.

Для этих двух авторов моделирование — это простое «обнаружение и выбор» объектов, которые ожидают нас! Однако если объекты только ожидают того, чтобы быть обнаруженными и выбранными, то это необходимо влечет за собой гипотезу, согласно которой *порядок вещей*, зафиксированный в нашем знании, существует независимо от нас¹¹.

Эта позиция, которую я со ссылкой на Канта назову *познавательным догматизмом* (3, 98—99), влечет за собой или, по крайней мере, обосновывает убеждение в том, что хорошая модель должна быть точной копией этого порядка вещей. Если этот подход к моделированию используется в программной инженерии, т. е. в отношениях между разработчиком, пользователем и системой программного обеспечения, то это порождает следующие неявные требования:

1) пользователь должен объяснить разработчику «порядок вещей»;

2) разработчик должен воспроизвести этот «порядок вещей» в системе;

3) система должна отобразить этот «порядок вещей» пользователю.

Критический взгляд на практику, напротив, показывает, что в рамках существующих объектно-ориентированных и основанных на знаниях методов мы не сможем выполнить эти три требования. Каждое из трех требований ставит нас перед пропастью:

1) пропасть между пользователем и разработчиком: пользователь испытывает большие затруднения в формулировании требований и экспертного знания таким образом, чтобы разработчик был способен понять их;

2) пропасть между разработчиком и системой программного обеспечения: разработчик испытывает большие затруднения при встраивании в системную модель экспертного знания и требований пользователя;

3) пропасть между системой программного обеспечения и пользователем: система недостаточно хорошо адаптирована к способности и способам мышления пользователя.

Это показывает, что объектно-ориентированные и основанные на знаниях методы не могут достичь своей цели — обеспечить возможность систем, которые адаптированы к человеческой способности мышления. А это привело меня к предположению, согласно которому главная методологическая причина этого состоит в *обнаружении и выборе* как подходе к моделированию.

4. *Подход к моделированию, ориентированный на человека*

Потребность в разработке систем программного обеспечения, ориентированных на человека, т. е. систем, которые в наибольшей возможной степени адаптированы к нашей способности мышления, влечет за собой необходимость разработки нового подхода к моделированию. Если такой подход не удастся создать, то разработчик будет придерживаться традиционной установки: «Я создаю машину»¹², а это, в свою очередь, усилит господство технологически ориентированных, основанных на машинном представлении понятий.

Основания моего подхода к моделированию были сформулированы уже более 200 лет назад Кантом в его «Критике чистого разума» следующим образом: «До сих пор считали, что всякие наши знания должны сообразоваться с предметами. При этом, однако, кончались неудачей все попытки через понятия что-то априорно установить относительно предметов... Поэтому следовало бы попытаться выяснить, не разрешим ли мы задачи метафизики более успешно, если будем исходить из предположения, что предметы должны сообразоваться с нашим познанием» (3, 87).

Для моих целей идея Канта о том, что объекты должны сообразовываться с нашей способностью познания, составляет магистральный путь к его «Критике». Я вывожу отсюда основной принцип познания следующим образом: «Порядок вещей и систем, который воплощается в нашем знании, зависит от нас и создается нами: он связан (сообразуется) с нашим способом обработки знаний (нашей способностью мышления)». Эта позиция, близкая к «радикальному конструктивизму» Эрнста фон Глазерфельда¹³, также является базисом моего принципа моде-

лирования: «Хорошая модель не является картинкой независимого порядка, но *работающей*¹⁴ (*жизнеспособной*) *формализацией* порядка, который мы сами порождаем в нашем познании». Посредством двух этих принципов я могу теперь переформулировать предшествующие примеры следующим образом:

1. «Если мы «подымаем крышу над фабрикой», то мы не найдем там готового порядка, но мы получим только тот порядок, т. е. системы, объекты, характеристики и состояния, которые мы сами создадим в процессе познания».

2. «Физическая реальность не моделируется в готовом виде в терминах объектов. Объекты даны нам как объекты опыта только после того, как мы зафиксировали порядок для них в процессе познания: мы можем «прочитать» объекты только после того, как мы сами «написали» их».

В методах (системах), основанных на знаниях, модель называется *базой знаний* и составлена главным образом из так называемых *фактов* и *правил*. Традиционный подход к моделированию идентичен с объектно-ориентированными методами¹⁵: моделирование понимается как *обнаружение и выделение* *положений дел* («*Sachverhalte*»), которые существуют независимо от нас. Мой подход к моделированию, который я называю «конструктивно ориентированное моделирование», рассматривает факты и правила, принадлежащие базе знаний как *работающие формализации «положений дел»*, а эти положения дел как «жизнеспособные конструкты, т. е. как нечто, что мы создаем (конструируем) сами при помощи нашего мышления и при помощи организации наших взаимодействий с окружением некоторым способом, который также работает как *возможный... путь*».

5. Обновление и обогащение разработок программного обеспечения с помощью философии Канта

Принцип конструктивно ориентированного моделирования составляет первый строительный блок моего метода определения для систем программирования: в моей работе как программиста и инженера по знаниям я уже много раз испытал, что этот принцип действительно работает. Второй строительный блок, тесно связанный с первым, состоит в следующей гипотезе:

«Приближение в компьютере к тому способу, при помощи которого мы порождаем «порядок вещей», является необходимым приспособлением систем программного обеспечения к человеческой способности мышления».

Модель того способа, при помощи которого мы — посредством обработки нашего знания — создаем порядок вещей, должна удовлетворять двум основным требованиям. Это:

1. Обработка знания должна рассматриваться прежде всего как *синтетически-конституирующая* (а не аналитически-трансформирующая) процедура.

2. Любая система (*живая или неживая*), которая сначала выполняет синтетические акты, должна моделироваться как *организм*, а не как машина (организм как условие синтеза).

В этом пункте компьютерная технология могла бы быть улучшена при помощи Канта¹⁶. В «Аналитике понятий» Кант разрабатывает теорию умственной деятельности, которая — согласно моей интерпретации — выполняет первое требование, а в «Критике способности телеологического суждения», по моему мнению, в достаточной степени разработаны основания для выполнения второго требования, и их можно использовать для проектирования искусственных *неживых* организмов.

6. Заключение

Нам еще предстоит длинный путь, пока мы доберемся с помощью Канта до человечески-ориентированных методов определения для систем программного обеспечения. Моей проблемой в этой статье прежде всего было стремление показать направление, по которому мы должны двигаться, в надежде, что эта постановка проблемы может вдохновить компьютерщиков, инженеров и исследователей Канта начать интереснейшее междисциплинарное сотрудничество.

¹ Речь идет не о деловом кризисе на рынке программного обеспечения, а о качественном кризисе.

² Переработанный вариант доклада, прочитанного на VI Кантовских чтениях (секция 4: Логическое кантоведение), проведенных Кантовским обществом России и Калининградским университетом 21—24 сентября 1993 г. в г. Светлогорске Калининградской области, Россия.

³ Искусственный интеллект (ИИ) означает здесь ту «дисциплину компьютерной науки, чьей главной целью является реализация в компьютере когнитивных функций, процессов и процедур, которые в настоящее время лучше выполняются людьми» (Rich E. Artificial Intelligence. New-York: McGraw-Hill, 1983).

⁴ Кант различает у человека три когнитивных измерения (3, 661), которые следующим образом используются в моем подходе: 1) измерение *знания* (ментальные аспекты типа способности мышления), 2) измерение *долженствования* (этико-практические аспекты, к примеру, участие в деятельности, организации труда и т. п.), 3) измерение *надежды* (результаты, которых я могу ожидать, если буду делать то, что должен).

⁵ Кант И. Критика чистого разума; Критика способности суждения.

⁶ Для ориентированных на клиента, на предметную область и стандартные системы программного обеспечения.

⁷ Sage A., Palmer J. Software Systems Engineering. N. Y.: John Wiley & Sons, 1990.

⁸ Coad P., Yourdon E. Object-oriented analysis. Englewood Cliffs. N. Y.: Prentice Hall, 1990.

⁹ Dangelmeier W. Objektorientierter Modellierungsansatz fuer eine regelbasierte Fertigungssteuerung//Zeitschrift für wirtschaftliche Fertigung und Automatisierung ZwF. 1993. Bd. 88. H. 5. S. 222.

¹⁰ Meyer B. Objektorientierte Softwareentwicklung. München: Hanser Verlag, 1990. S. 55.

¹¹ Эта гипотеза составляет — насколько я понимаю — основание существующих наук (производство знаний) и технологий (применение знаний).

¹² Wirth N. Data structures and algorithms//Scientific American. Vol. 251. N 3. P. 48—57.

¹³ Von Glaserfeld. The Construction of Knowledge. Salinas: Inter-systems Publications, 1988.

¹⁴ Формализация которой соответствует цели, для которой она была использована.

¹⁵ «Реальный мир составлен из объектов и отношений. Соответствующий символический мир составлен из символов объектов и символов отношений» (Lusti M. Wissensbasierte Systeme: Algorithmen, Datenstrukturen und Werkzeuge. Mannheim: BI, 1990).

¹⁶ В настоящее время ИИ все же способен выполнить два эти требования, поскольку существующие подходы либо ограничены трансформационной парадигмой (аналитически трансформативной в символическом ИИ и эмерджентно-трансформативной в коннекционизме), либо строго утверждают, что синтетически-конститутивная обработка может иметь место только в живых организмах (Bourgine P., Varela (eds.) Proc. 1-st European Conference on Artificial life (ECAL 91). Cambridge: MIT Press, 1992).

Перевод с английского В. Н. Брюшинкина.

А. Н. ТРОЕПОЛЬСКИЙ

(Калининградский государственный университет)

Кант и проблема практического применения философской теории

Как показывает жизнь, вопрос об отношении теории к практике в смысле пригодности ее применения для решения чисто практических задач постоянно подвергается сомнению. В этой связи работа Канта «О поговорке «Может быть, это и верно в теории, но не годится для практики» (1793) не потеряла своей актуальности и в наши дни.

Специфической особенностью этой работы является то, что в ней данная проблема решается как на метатеоретическом, так и на конкретно-теоретическом уровнях. К сожалению, здесь эти уровни представлены, если так можно выразиться, обратно пропорционально их теоретической значимости, т. е. наиболее ценный в плане решения поставленной проблемы метатеоретический уровень представлен вступлением, состоящим из двух страниц, а конкретно-теоретический уровень, хотя и важный в плане осмысления кантовского обоснования природы морали, государственного и международного права, все же нельзя, на мой взгляд, рассматривать как решение проблемы нахождения общих признаков, необходимо присущих всякой теории, имеющей практическое применение. Более того, в метатеоретической части работы Кант наряду с убеждением в практической применимости математики, механики, медицины, агрономии и других эмпирических дисциплин в соответствующих областях практики все же допускает мысль о возможности практической неэффективности философской теории, за исключением ее морального фрагмента. В связи с этим Кант утверждает: «Совершенно ина-