

12. Martirosyan A.A., Martirosyan K.V., Chernyshev A.B. Application of Fourier series in distributed control systems simulation, *Proceedings of the 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2019*, 2019, pp. 609-613.
13. Pershin I.M. Sintez sistem s raspredelennymi parametrami [Synthesis of systems with distributed parameters]. Pyatigorsk: Izd-vo RIA-KMV, 2002, 212 p.
14. Chernyshev A.B., Mayransaev Z.R. Problemy garmonicheskoy linearizatsii sistem s raspredelennymi parametrami [Problems of harmonic linearization of systems with distributed parameters], *Obozrenie prikladnoy i promyshlennoy matematiki* [Review of Applied and Industrial Mathematics], 2019, Vol. 26, No. 1, pp. 31-42.
15. Chernyshev A.B., Nazartsev M.S., Mayransaev Z.R. Garmonicheskaya linearizatsiya raspredelennykh sistem [Harmonic linearization of distributed systems], *Sovremennaya nauka i innovatsii* [Modern science and innovation], 2018, No. 4 (24), pp. 94-101.
16. Mayransaev Z.R., Chernyshev A.B. Formirovanie temperaturnogo polya v rezul'tate tochechnykh impul'snykh vozdeystviy [Formation of the temperature field as a result of point pulse effects], *Evrasiyskoe Nauchnoe Ob'edinenie* [Eurasian Scientific Association], 2018, No. 12-2 (46), pp. 82-84.
17. Chernyshev A.B., Tkachenko I.V., Mayransaev Z.R. Cylindrical criterion of absolute stability of distributed systems, *Современная наука и инновации* [Modern science and innovation], 2020, No. 4 (32), pp. 34-40.
18. Martirosyan A.A., Martirosyan K.V., Chernyshev A.B. Methods of distributed systems' structured modeling, *Proceedings of the 2016 IEEE North West Russia Section Young Researchers in Electrical and Electronic Engineering Conference, ElConRusNW 2016*, 2016, pp. 283-289.
19. Martirosyan A.V., Chernyshev A.B., Martirosyan K.V. Calculation of the first switch-on time of distributed objects control action, *Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2020*, 2020, pp. 750-754.
20. Martirosyan K.V., Chernyshev A.B., Martirosyan A.V., Tatyana K.V. Formation of the anterior heating function under the action of uniformly distributed sources, *Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2020*, 2020, pp. 755-760.

Статью рекомендовал к опубликованию к.т.н. А.Н. Попов.

Майрансаев Зураб Русланович – Северо-Кавказский федеральный университет, институт сервиса, туризма и дизайна (филиал); e-mail: sarmat007@inbox.ru; г. Пятигорск, Россия; тел.: +78793973927; кафедра систем управления и информационных технологий; аспирант.

Чернышев Александр Борисович – e-mail: chalbor@rambler.ru; кафедра систем управления и информационных технологий; профессор.

Mayransaev Zurab Ruslanovich – North Caucasus Federal University, Institute of Service, Tourism and Design (branch); e-mail: sarmat007@inbox.ru; Pyatigorsk, Russia; phone: +78793973927; the department of management systems and information technologies; postgraduate student.

Chernyshev Alexandr Borisovich – e-mail: chalbor@rambler.ru; the department of management systems and information technologies; professor.

УДК 519.711

DOI 10.18522/2311-3103-2021-2-189-199

Д.Е. Чикрин, А.А. Егорчев

СРАВНЕНИЕ МЕТОДОЛОГИЙ ПРОЕКТИРОВАНИЯ СВЕРХУ–ВНИЗ И СНИЗУ–ВВЕРХ ПРИ РАЗРАБОТКЕ СИСТЕМ ADAS

Выбор типа основной методологии проектирования оказывает значительное влияние на качество итогового продукта, в том числе и на его способность к дальнейшему развитию и масштабированию. В статье рассматриваются особенности стандартных мето-

долгий проектирования снизу–вверх и сверху–вниз применительно к системам ADAS (системам автоматизированного (беспилотного) управления автомобилем), показывается, что использование "чистых" методологий неприемлемо при проектировании указанных систем и требуется создание новой совмещённой методологии проектирования. Для этого рассмотрены особенности и ограничения подхода сверху–вниз: ориентация подхода на максимальное соответствие разрабатываемой системы предъявляемым к ней требованиям; методологическая строгость подхода; трудность тестирования системы в процессе разработки; чувствительность к изменениям требований к разрабатываемой системе. Рассмотрены особенности и ограничения подхода снизу–вверх: возможность итеративной разработки с получением промежуточного результата; возможность использования стандартных компонентов; масштабируемость и гибкость системы разрабатываемой системы; возможность несоответствия функций подсистем требованиям, которое может проявляться только на поздних этапах разработки; возможная несогласованность при разработке отдельных подсистем и элементов. Рассмотрены особенности и факторы разработки систем ADAS: повышенные требования по надёжности и безопасности работы системы; разнородность используемых компонентов. Выделены два этапа развития ADAS-систем: этап интенсивной разработки и этап экстенсивной эволюции. Рассмотрена применимость той или иной методологии относительно различных аспектов разработки и эволюции систем ADAS, таких как: определение требований; композиционный морфизм; масштабируемость и расширяемость; стабильность и устойчивость и время разработки; способность к развитию. В результате сравнения методологий делается вывод о том, что существуют аспекты разработки и развития технической системы, в которых наблюдается значительное преимущество одной или другой из методологий. В должной степени эволюция системы может быть обеспечена только при использовании подхода снизу–вверх. Однако, для сложных систем критически важным является определение изначальных требований к системе, что может быть достигнуто только с применением методологии сверху–вниз.

Методология проектирования; сверху–вниз; снизу–вверх; система ADAS; беспилотное транспортное средство; проектирование сложных систем.

D.E. Chickrin, A.A. Egorchev

TOP-DOWN VS BOTTOM-UP METHODOLOGIES FOR ADAS SYSTEM DESIGN

Selection of the principal design methodology has a significant impact on final product quality, including its evolvability and scalability. The article discusses the features of traditional bottom-up and top-down design methodologies in the context of ADAS (driver assistance and automated driving systems). Necessity of the combined design methodology is shown due to unacceptability of "pure" methodologies for design of this kind of systems. For this purpose, the features and limitations of the top-down approach are considered: commitment to maximum compliance of the developed system with its requirements; methodological rigor of the approach; difficulty of system testing in the process of the development; sensitivity to changes in requirements. The features and limitations of the bottom-up approach are considered: possibility of iterative development with obtaining intermediate results; possibility of using standard components; scalability and flexibility of the developed system; possibility of discrepancy of functions of subsystems to requirements, which may appear only at later stages of development; possible inconsistency in development of separate subsystems and elements. The features and factors of ADAS system development are considered: increased requirements for reliability and safety of the system; heterogeneity of used components. Two stages of ADAS-systems development are distinguished: the stage of intensive development and the stage of extensive evolution. The applicability of one or another methodology to various aspects of ADAS system development and evolution (such as: requirements definition; compositional morphism; scalability and extensibility; stability and sustainability; cost and development time; development capability) is considered. A comparison of the

methodologies concludes that there are aspects of technical system design and development in which there is a significant advantage of one or the other of the methodologies. Only the bottom-up approach can ensure the proper evolution of the system. However, for complex systems, it is critical to define the initial requirements for the system, which can only be achieved using the top-down methodology.

Design methodology; bottom-up; top-down; ADAS; self-driving vehicle; complex systems design.

Введение. Корреляция между особенностями методологии проектирования и качеством итоговой системы достаточно подробно показана применительно к программному обеспечению [1]. Современные же системы ADAS представляют собой сложную смесь программных и аппаратных компонент, но в то же время строятся по тем же принципам, что и сложные программные комплексы различного назначения, так как представляют собой автоматизированную систему управления исполнительными механизмами автомобиля [2–4]. Методология сверху–вниз является классической при осознанном (запланированном ранее) создании практически любой крупной аппаратно–программной или программной системы. Действительно, при заказе и планировании создания сложных систем любого рода возникает желание максимально детализировать требования к ним – чтобы получить именно то, что требуется на текущий момент времени. В свою очередь, для создания прототипов и получения быстрых результатов практически всегда используется методология снизу–вверх, использующая блоки – «чёрные ящики» с фиксированной, но уже реализованной функциональностью, позволяющие быстро достичь работающей в первом приближении системы и получить первые результаты и обратную связь от испытателей системы и заказчика работ.

1. Методологии сверху–вниз и снизу–вверх. При разработке программно–аппаратных систем активно применяется как методология сверху–вниз (top–down), так и снизу–вверх (bottom–up).

Методология сверху–вниз подразумевает проведение процесса разработки начиная с определения требований к разрабатываемой системе. Разрабатываемая система описывается с точки зрения того, что необходимо получить в результате, а не каким образом. При этом полное соответствие требованиям достигается на высоких уровнях абстракции, а при продвижении вниз могут возникать проблемы, связанные с технической невозможностью или сложностью конечной реализации запланированной функциональности составных частей системы. Такие проблемы могут быть исправлены двумя способами: 1) внедрением на уровне реализации обходных решений, которые, как правило, снижают эффективность системы и/или повышают стоимость разработки; 2) возвратом на более высокие уровни абстракции системы и переработкой декомпозиции [1]. Так как проблемы в реализации низкоуровневых компонент возникают на поздних этапах разработки, то и изменение становится очень дорогим – потенциально придётся реализовывать заново часть функциональности, и это не говоря о самом процессе повторного дизайна системы [5].

Подход сверху–вниз подразумевает реализацию подсистем, максимально соответствующих поставленным требованиям, что обычно приводит к полной реализации всей функциональности системы с нуля – повторное использование ранее разработанных и/или универсальных компонентов обычно невозможно, так как технические и функциональные возможности уже разработанных систем как правило не учитываются при декомпозиции («что получить, а не как») и могут не соответствовать поставленным требованиям.

Таким образом, можно выделить следующие особенности подхода сверху–вниз:

1. (+) система в полной мере соответствует поставленным изначально требованиям;
2. (+) подход более эффективен с методологической точки зрения;
3. (–) сколько-нибудь законченная система может быть получена только в конце процесса разработки – затрудняется тестирование и оценка получаемого результата;
4. (–) разрабатываемая система нетолерантна к изменению требований в процессе разработки – в худшем случае при изменениях весь процесс разработки необходимо начинать заново.

Методология снизу–вверх широко применяется в робототехнике, встраиваемых системах и сенсорных сетях [6]. При использовании подхода снизу–вверх на начальном этапе разработки определяется набор базовых систем и технологий. С каждой новой итерацией разработки низкоуровневые компоненты связываются и объединяются в более сложные системы, которые начинают проявлять новые свойства, превосходящие свойства простой совокупности отдельных компонентов – проявляется свойство эмерджентности.

Основная проблема методологии снизу–вверх – возможное несоответствие разрабатываемого продукта требованиям. Но при этом, как показано в [1, 7–9], требования к продукту неизбежно меняются, причём начинается этот процесс ещё во время разработки. С другой стороны, использование методологии снизу–вверх позволяет быстро получать прототипы системы, которые могут использоваться для постепенной верификации системы на соответствие требованиям, и, что более важно, для уточнения и обновления самих требований.

Особенности методологии снизу–вверх:

1. (+) возможна итеративная разработка с получением промежуточного результата, который можно использовать для тестирования, уточнения требований и особенностей реализации;
2. (+) активное использование стандартных компонентов позволяет снизить временные и финансовые издержки при разработке;
3. (+) высокая масштабируемость и гибкость системы: базовые элементы системы независимы, что позволяет заменять элементы и добавлять новые;
4. (–) возможные несоответствия функций подсистем высокоуровневым требованиям (недостаточная производительность вычислительных устройств, недостаточная точность сенсоров и т.д.), обнаружить которые возможно только на поздних этапах разработки при реализации взаимодействия высокоуровневых подсистем;
5. (–) при независимой разработке подсистем возможны ситуации, когда некоторые элементы, имеющие схожую функциональность, реализуются независимо и различными способами. При этом усложняется их обслуживание и увеличивается стоимость их разработки.

2. Особенности ADAS–систем. Системы автоматизированного управления транспортным средством – сложные гетерогенные системы управления автомобилем, которые могут быть как вспомогательными для водителя (например, круиз–контроль, автоматическая парковка и т.д.), так и полностью заменять его [10]. Базовыми высокоуровневыми функциями таких систем, как и любых технических систем, являются восприятие окружающей обстановки, принятие решений (планирование) и управление исполнительными механизмами автомобиля. Так как системы беспилотного вождения в качестве базы используют автомобили, к ним предъявляются повышенные требования по надёжности и безопасности работы. В связи с этим, помимо реализации трёх базовых функций, системы такого рода должны иметь возможность контролировать работу системы и обеспечивать отка-

зоустойчивость (путём резервирования подсистем и аварийных режимов), фиксацию системных событий и возможность наблюдения и контроля со стороны человека [11].

Развитие систем ADAS происходит на стыке автомобильной индустрии и робототехники, целесообразно рассмотреть подходы к проектированию, принятые в этих сферах.

Автомобильную промышленность отличают высокие требования к безопасности, устойчивости работы и надёжности всех компонентов системы – такие свойства системы возможно получить при проектировании системы по методологии сверху–вниз. Но в последние десятилетия в автомобилестроении наблюдается стремление к подходу снизу–вверх – активно развиваются модульные платформы, подразумевающие повторное использование базовых компонентов в различных моделях автомобилей. При этом сама платформа разрабатывается в соответствии со всеми требованиями к безопасности и после выпуска не меняется – изменение платформы потребует перестройки процессов производства и возможной потери совместимости с компонентами. Таким образом, процесс разработки в современной автомобильной промышленности строится в соответствии с обеими методологиями – платформа, спроектированная в строгом соответствии с требованиями и актуальная для нескольких поколений автомобилей строится по методологии сверху–вниз, а при проектировании самих автомобилей используется эта базовая платформа и набор дополнительных взаимозаменяемых компонентов в различных комбинациях и вариациях, что соответствует подходу снизу–вверх.

В современной автомобильной промышленности при разработке автомобиля используется множество готовых компонентов от сторонних производителей. При этом соблюдение стандартов в автомобильной промышленности (AUTOSAR, OSEK/VDX, FlexRay, CAN) абсолютно критично, т.к. данные стандарты ориентированы как раз на обеспечение совместной работы оборудования различных производителей, что решает одну из основных проблем подхода снизу–вверх – обеспечение корректного взаимодействия между подсистемами [11–14].

В ADAS–системах компоненты значительно сложнее и разнороднее, а значит и проблема стоит острее, чем для традиционных автомобильных (неинтеллектуальных) систем и узлов. В этой части ADAS–системы близки к робототехническим системам. Робототехнические системы представляют собой комплексы гетерогенных систем с совершенно разным функциональным назначением – различные сенсоры, актуаторы, вычислительные модули. Процессы разработки в робототехнике тяготеют к методологии снизу–вверх: как правило, синтез систем производится из уже готовых компонентов, которые сами по себе достаточно сложны; полная разработка таких подсистем в соответствии с требованиями при применении подхода сверху–вниз радикально увеличивает продолжительность и стоимость разработки.

В качестве примера подобных гетерогенных подсистем можно привести подсистему восприятия беспилотного автомобиля, которая представляет собой совокупность разнородных сенсоров: лидаров, сонаров, радаров, камер, спутниковых и инерциальных систем навигации и т.д. (см., например, [15, 16]). На рис. 1 приведён пример сенсорной подсистемы автомобиля, оснащённого системой ADAS.

Большая часть компонентов сенсорной подсистемы предлагается сторонними производителями, которые как правило, специализируются на разработке и производстве какого-то конкретного типа сенсоров, обладают значительной экспертизой в предметной области и инвестируют в активное развитие этих технологий. Самостоятельная разработка систем такого рода непозволительно затратна даже для крупных автоконцернов, в связи с чем можно сделать вывод, что использование сторонних компонентов в ADAS–системах неизбежно.

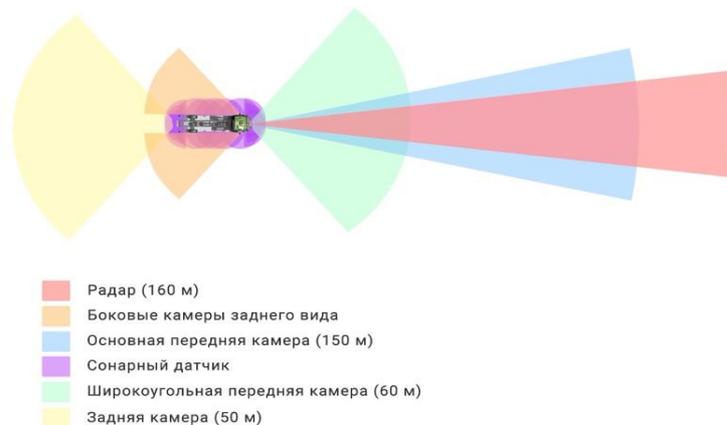


Рис. 1. Сенсорные подсистемы беспилотного автомобиля

Получение требуемой функциональности при минимизации стоимости – одна из ключевых задач при проектировании ADAS–систем. Активное развитие ADAS и робототехнических систем привело к значительному прогрессу в области как аппаратных компонентов, в частности, сенсоров, так и программных решений. Разнообразие решений позволяет соблюсти баланс между стоимостью и функциональностью систем в широком диапазоне степеней автоматизации. Постоянный прогресс в сфере компонентов также приводит к улучшению технических характеристик предлагаемых устройств. В связи с этим ADAS–системы должны иметь возможность адаптироваться и масштабироваться для сохранения актуальности системы – при этом с обязательным обеспечением поддержки стандартных интерфейсов.

3. Влияние выбора методологии на разрабатываемую ADAS– систему

3.1. Определение аспектов процессов разработки и эволюции системы.

Если рассмотреть систему ADAS как систему с фиксированным (монокристаллическим) аппаратным ядром, модифицируемым программным ядром и атомарными периферийными компонентами, можно выделить два этапа развития системы – интенсивной разработки с реализацией/модификацией ядра и экстенсивной эволюции (с перестроением связей между компонентами и/или изменением состава периферийных компонентов). Для выбора оптимальной методологии разработки ADAS–систем необходимо провести сравнение влияния рассмотренных методологий на различные аспекты как процесса разработки системы, так и процесса её эволюции. В первую очередь необходимо определить аспекты, по которым будет производиться сравнение.

Необходимость корректного определения требований при проектировании сложных систем рассматривается в [17, 18]. Описание композиционного морфизма систем (возможности системы по изменению своей внутренней структуры) и его влияния на функциональный и эволюционный потенциал системы приводится в [1, 19]. Также в [19] как важные параметры систем рассматриваются стоимость системы (включающая стоимость разработки, производства и обслуживания), стабильность. Аспекты эволюции системы рассматриваются в статье [20], выделяют следующие критерии эволюционной способности системы:

- 1) общность системы – заведомое соответствие системы не только поставленным требованиям, но и потенциально возможным;
- 2) адаптируемость – способность системы к адаптации к изменяющимся требованиям посредством перестроения связей между компонентами (при неизменности состава системы);

3) масштабируемость – возможность вертикального масштабирования системы (увеличения производительности используемых компонентов) при увеличении нагрузки;

4) расширяемость – возможность адаптации системы к изменившимся требованиям посредством добавления новых компонентов, не входивших ранее в состав системы.

В работе [9] авторы выделяют два пути эволюционного развития системы – статический (без изменения структуры системы, адаптация только посредством заложенной избыточности) и динамический (изменение структуры, добавление либо замена компонентов) на основе рассмотренных в [20] аспектов. К статическому пути авторы относят общность систем, к динамическому – адаптируемость, масштабируемость и расширяемость. Вполне очевидно, что возможность статической эволюции системы всецело зависит от целеполагания и не может рассматриваться в контексте методологий проектирования. В связи с этим рассмотрим только аспекты динамической эволюции систем.

3.2. Влияние выбора методологии на различные аспекты процессов разработки и эволюции системы

3.2.1. Постановка задачи и определение требований. Использование методологии сверху–вниз является неизбежным при постановке задачи на начальной стадии проектирования (утверждения эскизного проекта) любых сложных технических систем – корректная формулировка начального технического задания является важнейшим этапом проектирования.

3.2.2. Композиционный морфизм. Композиционный морфизм возможно рассматривать как расширение фактора адаптируемости системы – адаптируемость подразумевает возможность модификации системы под влиянием внешних факторов, тогда как композиционный морфизм характеризует структурную гибкость системы в целом.

Подход снизу–вверх по своей концепции позволяет получить систему, состоящую из набора ортогональных (независимых и непересекающихся по функциональности) компонентов, и функциональность системы определяется системой связей между компонентами. При этом выбор низкоуровневых ортогональных компонентов в методологии снизу–вверх производится на ранних этапах разработки, и решение задачи адаптации системы для соответствия требованиям производится в большинстве случаев посредством корректировки связей между ними. Следовательно, системы, разработанные в соответствии с подходом снизу–вверх, по определению полиморфны – при неизменности их состава (набора подсистем и компонентов) такие системы обладают большим многообразием внутренних структур (систем связей). Очевидно, что подобная внутренняя гибкость структуры системы повышает адаптационные и эволюционные возможности системы. Согласно [9], высокая способность системы к адаптации посредством модификации структуры связей реализуема при использовании модульной архитектуры, открытых (непроприетарных) стандартов и стандартных интерфейсов, что соответствует методологии снизу–вверх.

В соответствии с методологией сверху–вниз системы разрабатываются с ориентацией на требования и сценарии использования [1], и фиксированная система связей (единственная, наиболее оптимально соответствующая изначально заданному набору требований) является равноправным и неотъемлемым компонентом системы. Системы такого рода мономорфны – возможен только один способ объединения подсистем, причём зачастую связи описываются раньше формализации

требований к функциональности подсистем при декомпозиции. Эволюция таких систем происходит при изменении требований, что приводит к проведению полного цикла проектирования заново – в результате формируется новая система, также имеющая монолитную структуру.

Для систем ADAS мономорфная структура слабо применима, т.к. активное развитие сферы требует постоянной эволюции разрабатываемых систем.

3.2.3. Масштабируемость и расширяемость. Помимо эволюции с сохранением состава системы и редактированием системы связей, в процессе развития системы может возникнуть потребность в замене каких-либо периферийных компонентов (например, при выходе на рынок устройств с улучшенными характеристиками). Движущим фактором эволюции такого рода являются подсистемы и компоненты, а конкретнее – их развитие (независимое от ADAS-системы, в которой они используются). Так как независимые подсистемы могут существовать только в рамках методологии снизу-вверх, то обеспечить масштабируемость ADAS-систем возможно только с использованием этого подхода.

Согласно определению из работы [20] расширяемость представляет собой модификацию системы путём добавления новых компонентов, что подразумевает, аналогично масштабируемости, развитие системы, движимое компонентами, и может быть реализовано только с применением методологии снизу-вверх. В контексте систем ADAS может быть вызвано выходом на рынок нового типа сенсоров или новых программных решений.

Масштабирование и расширение системы в рамках подхода сверху-вниз выражается в формировании обновлённых требований к системе и полной переработке системы по полному циклу проектирования, что подразумевает глобальные изменения в системе, так как подсистемы в концепции методологии сверху-вниз не являются независимыми и тесно связаны с другими компонентами. Изменение характеристик отдельных подсистем, не контролируемое целеполаганием ко всей системе в целом, может оказать непредсказуемое влияние на внутрисистемные взаимодействия и функционирование системы [8, 21].

3.2.4. Стабильность, устойчивость. Стабильность системы здесь корректно рассматривать с точки зрения необходимости внесения изменений. Требования к системам меняются, особенно это актуально для ADAS-систем. В условиях активного развития их программной и аппаратной базы, когда растущая функциональность потенциальных компонентов позволяет постоянно повышать требования к системе, системы, спроектированные по методологии сверху-вниз, обладает меньшим адаптационным ресурсом, а вносимые изменения более деструктивны для работающей системы. В соответствии с подходом снизу-вверх системы разрабатываются с учётом необходимости корректировки реализации системы с целью удовлетворения требований и обладают достаточной гибкостью внутренней структуры для неструктурной модификации, в том числе и в процессе эволюции системы. Таким образом, при использовании подхода снизу-вверх можно добиться более устойчивого функционирования системы в условиях изменяющихся требований, а подход сверху-вниз позволяет получить высокостабильную систему, но лишь при условии, что требования будут оставаться неизменными – в этом случае ни о какой эволюции системы говорить не приходится.

3.2.5. Стоимость и время разработки. В случае разработки достаточно сложных систем подход снизу-вверх обладает значительными преимуществами в части стоимости и времени разработки, так как позволяет (и даже подразумевает) использование сторонних и разработанных ранее компонентов. Как было показано ранее, разработка ADAS-систем невозможна без использования сторонних компонентов в виду их высокой сложности, соответственно применение методологии

сверху–вниз в ADAS–системах с точки зрения стоимости и времени разработки нецелесообразно. При этом процесс разработки в методологии снизу–вверх итеративен, так как подразумевает поэтапную адаптацию получаемого результата для соответствия требованиям, тогда как подход снизу–вверх предполагает полное проектирование системы, а затем полную её реализацию оптимальным способом. Таким образом, если рассматривать стоимость разработки не с практической, а с методологической точки зрения, предпочтительнее использование методологии сверху–вниз.

Однако, вышесказанное правомерно только в случае оптимального течения процесса разработки. Отклонения неизбежны при проектировании системы по обеим рассматриваемым методологиям. В методологии снизу–вверх невозможность конечной реализации запланированной функциональности, как было показано, проявляется на поздних этапах разработки. Также нельзя исключать вероятность неточного/некорректного определения изначальных требований [5], что может быть обнаружено при тестировании на поздних этапах разработки или даже конечными пользователями при использовании уже готовой системы. В обоих случаях процесс переработки системы потребует значительных финансовых и временных затрат, сопоставимых с разработкой системы заново. При использовании методологии снизу–вверх неизбежно несоответствие системы требованиям, но гибкость структуры системы и возможность частых итеративных уточнений позволяет адаптировать систему с минимальными издержками.

3.2.6. Способность к развитию и процесс развития (эволюции). По рассмотренным выше аспектам развития системы можно сделать обобщённые выводы по влиянию методологий сверху–вниз и снизу–вверх на эволюционный потенциал разрабатываемой системы. Для процесса эволюции ADAS–систем характерны постоянные, но незначительные по масштабам системы в целом обновления существующих подсистем и периодическое внедрение новых типов периферийных компонентов (как программных, так и аппаратных). В рамках подхода снизу–вверх изменения такого рода могут быть произведены с минимальными временными и финансовыми затратами при условии либо: 1) сохранения обратной совместимости интерфейсов заменяемых/обновляемых компонентов; либо 2) добавления компонентов в пределах заложенной коммуникационной и вычислительной избыточности монолитного ядра. В противном случае эволюция системы становится интенсивной и требуется обновление ядра. В случае же подхода сверху–вниз система по определению оптимальна и максимально соответствует поставленным требованиям, следовательно, в такой системе не заложена избыточность для масштабирования. Помимо этого, в таких системах эволюция может происходить только исходя из изменений требований, что приводит к необходимости перепроектирования системы. Сценарий покомпонентного развития функциональности ADAS–системы в рамках подхода сверху–вниз невозможен.

3.3. Результаты. Сравнительная таблица. В табл. 1 обобщены выводы по влиянию выбора методологии на процесс разработки и эволюции системы. Здесь символ «+» показывает, что методология полностью удовлетворяет приведённому аспекту, «–» – совершенно не удовлетворяет, «+/-» – в большей степени удовлетворяет, «-/+» – в некоторой степени удовлетворяет. Как видно из результатов, существуют аспекты процесса развития системы, в которых наблюдается значительное преимущество только одной из методологий.

Таблица 1

Сравнение влияния методологий сверху–вниз и снизу–вверх на различные аспекты системы

	сверху–вниз	снизу–вверх
Постановка задачи, определение требований	+	–
Композиционный морфизм	–	+
Масштабируемость, расширяемость	–	+
Стабильность, устойчивость	–/+	+/-
Стоимость, время разработки	–/+	+/-
Способность к развитию (эволюции)	–	+

Выводы. В работе исследуются достоинства и недостатки подходов проектирования снизу–вверх и сверху–вниз применительно к системам ADAS. Рассматривается влияние выбора методологии проектирования на процессы как проектирования, так и эволюции системы.

Оба традиционных подхода проектирования (сверху–вниз и снизу–вверх) обладают недостатками, отклоняющими течение процесса разработки от оптимального направления при возникновении ошибок проектирования, которые в обоих случаях проявляются лишь на поздних этапах разработки: в частности, для методологии сверху–вниз возможно несоответствие требований техническим возможностям на низких уровнях; в методологии снизу–вверх точное соответствие требованиям практически невозможно. При этом существуют аспекты процессов разработки и эволюции системы, в которых наблюдается значительное преимущество каждой из методологий.

При проектировании систем автоматизированного управления транспортным средством требуется обеспечить масштабируемость и модифицируемость системы в процессе её эволюции. Особенно это актуально в настоящее время – наблюдается постоянный прогресс в развитии сенсорных систем, робототехники, алгоритмов автоматизации, и недостаточная гибкость требований и самих процессов разработки приводит к неизбежному и быстрому устареванию систем. В должной степени эволюция системы может быть обеспечена только при использовании подхода снизу–вверх. Однако, для сложных систем критически важным является определение изначальных требований к системе, что может быть достигнуто только с применением методологии сверху–вниз. Формирование обобщённого подхода к проектированию с учётом проведённого анализа планируется как продолжение исследований на тему.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Pizka M., Bauer A.* A brief top-down and bottom-up philosophy on software evolution, *Proceedings. 7th International Workshop on Principles of Software Evolution, 2004*, [S.l.]: IEEE, 2004, pp. 131-136.
2. *Li L., Wen D., Zheng N., Shen L.* Cognitive Cars: A New Frontier for ADAS Research, *IEEE Transactions on Intelligent Transportation Systems*, 2012, Vol. 13, No. 1, pp. 395-407. Doi: 10.1109/TITS.2011.2159493.
3. *Ziebinski A., Cupek R., Grzechca D., Chruszczyk L.* Review of advanced driver assistance systems (ADAS), *Proceedings of the International Conference of Computational Methods in Sciences and Engineering 2017 (ICCMSE-2017). AIP Conference Proceedings*, Vol. 1906, [S.l.]: [AIP], 2017, Art. no. 120002, 4 p. Doi: 10.1063/1.5012394.
4. *Jumaa B.A., Abdulhassan A.M., Abdulhassan A.M.* Advanced Driver Assistance System (ADAS): A Review of Systems and Technologies, *International Journal of Advanced Research in Computer Engineering & Technology*, 2019, Vol. 8, No. 6, pp. 231-234.
5. *Sangiovanni-Vincenelli A., Damm W., Passerone R.* Taming Dr. Frankenstein: Contract-based design for cyber-physical systems, *European journal of control*, 2012, Vol. 18, No. 3, pp. 217-238.

6. Crespi V., Galstyan A., Lerman K. Top-down vs bottom-up methodologies in multi-agent system design, *Autonomous Robots*, 2008, Vol. 24, No. 3, pp. 303-313.
7. Wan-Kadir W.M.N., Loucopoulos P. Relating evolving business rules to software design, *Journal of Systems architecture*, 2004, Vol. 50, No. 7, pp. 367-382.
8. Borches P.D., Bonnema G.M. Coping with System Evolution-Experiences in Reverse Architecting as a Means to Ease the Evolution of Complex Systems, *INCOSE International Symposium*, 2009, Vol. 19, No. 1, pp. 955-969.
9. Christian J., Olds J. A quantitative methodology for identifying evolvable space systems, *1st Space Exploration Conference: Continuing the Voyage of Discovery*. [S.l.]: [s.n], 2005, pp. 2543.
10. Bernhart W., Winterhoff M. Autonomous Driving: Disruptive Innovation that Promises to Change the Automotive Industry as We Know It, *Energy Consumption and Autonomous Driving*. Cham: Springer, 2016, pp. 3-10.
11. Jo K. et al. Development of autonomous car – Part I: Distributed system architecture and development process, *IEEE Transactions on Industrial Electronics*, 2014, Vol. 61, No. 12, pp. 7131-7140.
12. Heinecke H. Automotive system design-challenges and potential, *Proceedings of the conference on Design, Automation and Test in Europe-Volume 1*. [S.l.]: IEEE Computer Society, 2005, pp. 656-657.
13. Benvenuti L. et al. Contract-based design for computation and verification of a closed-loop hybrid system, *International Workshop on Hybrid Systems: Computation and Control*. Berlin: Springer, 2008, pp. 58-71.
14. Sangiovanni-Vincentelli A. Quo vadis, SLD? Reasoning about the trends and challenges of system level design, *Proceedings of the IEEE*, 2007, Vol. 95, No. 3, pp. 467-506.
15. Thakurdesai H.M., Aghav J.V. Autonomous cars: technical challenges and a solution to blind spot, *Advances in Computational Intelligence and Communication Technology. Proceedings of CICT 2019*. Advances in Intelligent Systems and Computing, Vol. 1086. Singapore: Springer, 2021, pp. 533-547. Doi: 10.1007/978-981-15-1275-9_44.
16. Ziebinski A., Cupek R., Erdogan H., Waechter S. A Survey of ADAS Technologies for the Future Perspective of Sensor Fusion, *Computational Collective Intelligence. ICCCI 2016. Lecture Notes in Computer Science*, Vol. 9876. Cham: Springer, 2016, pp. 135-146. Doi: 10.1007/978-3-319-45246-3_13.
17. Boehm B.W. Verifying and validating software requirements and design specifications, *IEEE software*, 1984, Vol. 1, No. 1, pp. 75.
18. Arnaut B.M., Ferrari D.B., e Souza M.L.O. A requirements engineering and management process in concept phase of complex systems, *2016 IEEE International Symposium on Systems Engineering (ISSE)*. [S.l.]: IEEE, 2016, pp. 1-6.
19. Sztipanovits J. et al. Toward a science of cyber-physical system integration, *Proceedings of the IEEE*, 2012, Vol. 100, No. 1, pp. 29-44.
20. Rowe D., Leaney J., Lowe D. Defining systems evolvability-a taxonomy of change, *Change*, 1994, Vol. 94, pp. 541-545.
21. Watson J.D. et al. Optimization of excess system capability for increased evolvability, *Structural and Multidisciplinary Optimization*, 2016, Vol. 53, No. 6, pp. 1277-1294.

Статью рекомендовал к опубликованию к.т.н. П.А. Кокунин.

Чикрин Дмитрий Евгеньевич – Казанский федеральный университет; e-mail: dmitry.kfu@gmail.com; г. Казань, Россия; тел.: +79172727100; к.т.н.; доцент; директор Института вычислительной математики и информационных технологий.

Егорчев Антон Александрович – e-mail: eanton090@gmail.com; тел.: +78432337609; научный сотрудник Института фундаментальной медицины и биологии.

Chickrin Dmitry Evgenevich – Kazan Federal University; e-mail: dmitry.kfu@gmail.com; Kazan, Russia; phone: +79172727100; cand. of eng. sc.; associate professor; director of the Institute of Computer Mathematics and Information Technologies.

Egorchev Anton Aleksandrovich – e-mail: eanton090@gmail.com; phone: +78432337609; research fellow of the Institute of Fundamental Medicine and Biology.