

УДК 007.52

Дата подачи статьи: 03.10.18

DOI: 10.15827/0236-235X.126.306-312

2019. Т. 32. № 2. С. 306–312

Теоретико-множественная модель функционального подхода к интеллектуализации процессов управления зданиями и сооружениями

*Р.В. Душкин*¹, зам. генерального директора по интеллектуальным транспортным системам и аппаратно-программному комплексу «Безопасный город»,
roman.dushkin@gmail.com

¹ Компания «ВойсЛинк», г. Москва, 127322, Россия

В статье описан функциональный подход к управлению внутренней средой интеллектуального здания. Такой подход подразумевает использование и передачу между периферийным оборудованием и центральными инженерными системами неизменяемых состояний, отсутствие побочных эффектов при осуществлении управляющих воздействий и построение децентрализованной сети для вычислений и принятия решений на стороне оконечных устройств на базе интернета вещей. Это позволяет одновременно получить все выгоды различных парадигм рассмотрения процессов управления и интеллектуализации. Кроме того, появляется возможность эмерджентно проявить новые свойства общего подхода для повышения степени управляемости и эффективности эксплуатации объектов управления.

Рассмотрены математические основы организации распределенной вычислительной среды для реализации функционального подхода. Кратко описываются возможные сценарии управления в различных режимах работы инженерных систем интеллектуальных зданий. Путем описания теоретико-множественной модели интеллектуализированной системы управления зданием подводится теоретический базис под рекомендацией об изменении парадигмы построения общей АСУ интеллектуальным зданием от автоматизации инженерных процессов к интеллектуализации управления и созданию системы искусственного интеллекта, осуществляющей полный и автономный цикл управления зданием.

Применение функционального подхода в совокупности с новыми технологиями для управления такими объектами, как интеллектуальные здания, позволяет перевести эксплуатацию этих зданий на более высокий уровень доступности сервисов, устойчивости, экологичности и всестороннего развития не только самого объекта управления, но и иерархии его надсистем – муниципалитета, региона, государства.

Ключевые слова: функциональный подход, интеллектуальное здание, интернет вещей, децентрализация, искусственный интеллект, интеллектуализация.

Развитие технологий открывает широкие возможности для решения новых задач автоматизации и интеллектуализации технологических процессов и процессов управления. Появление и всестороннее распространение концепции умных элементов инфраструктуры (умных дорог, домов, городов) предполагает использование для автоматизации таких процессов новых принципов, выводящих объекты автоматизации на более высокие уровни эффективности и инновационности [1].

Однако вопросам теоретического осмысления и практического применения новых технологий в автоматизации и интеллектуализации устоявшихся процессов уделяется мало внимания. При проектировании и разработке сложных систем управления часто исповедуется принцип «если работает, не надо трогать», а потому при проектировании используются уже испытанные подходы, а в практику внедряются

морально и технологически устаревшие решения [2].

Такая ситуация характерна, например, для автоматизации и интеллектуализации процессов жизнеобеспечения, технологических процессов и процессов управления интеллектуальными зданиями [3]. Автоматизация этих процессов осуществляется на основе перевода части операций в составе технологического процесса на средства вычислительной техники или периферийную автоматику, при этом кибернетический принцип новых задач практически не используется [4].

Однако для автоматизации и интеллектуализации разнообразных процессов уже разработаны, апробированы в различных областях и успешно применяются такие технологии, как

- машинное обучение;
- распределенные вычисления и распределенные БД;

- одноранговые пиринговые сети и интернет вещей;
- облачные, туманные вычисления и вычисления на стороне оконечного оборудования [5–9].

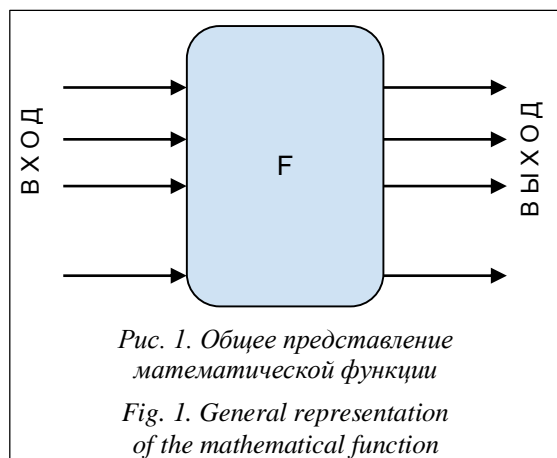
Совмещение всех этих технологий для повышения эффективности процессов управления и принятия решений возможно в рамках функционального подхода, основанного на принципах функциональной парадигмы программирования, теории категорий и базового математического понятия функции [10].

Теоретические основы

Функциональный подход разработан в рамках декларативного программирования и основан на понятии функции [11]. Функциональное программирование оперирует функциями как базовыми примитивами для описания алгоритмов. Это позволяет рассматривать вычислительные процессы в качестве последовательности вызовов функций, выраженных через атомарные действия и вызов других функций, в том числе и самой себя, при организации рекурсивных вызовов [12].

Математическая функция представляет собой черный ящик с множеством входов и выходов, который реализует некоторое вычисление, то есть преобразование входных значений в выходные [13]. Функции обладают двумя важными качествами – детерминированностью и чистотой [14]. Фактически эти свойства подразумевают, что функция работает только с выделенной для ее работы памятью, ничего не получая из внешней памяти и ничего не записывая в нее, – только вход, выход и черный ящик вычислений между ними (рис. 1).

Вычисления организованы в виде цепочек выполняемых функций, когда одни, закончив



свои вычисления, передают результаты следующим. Главная особенность таких цепочек в том, что функции, не зависящие друг от друга, могут безопасно вычисляться в параллельном режиме, так как ни одна из них гарантированно не повлияет на память, используемую другой [10]. Это позволяет избежать многих проблем, возникающих в рамках параллельных или конкурентных вычислений, основанных на обычной вычислительной модели [15].

Другой важной особенностью чистого функционального подхода является то, что функции можно рассматривать с точки зрения математики, в которой разработано большое количество методов для анализа функций [16]. Например, в рамках этого подхода в программировании имеется возможность оптимизации программ на уровне исходного кода [17]. Также можно анализировать программы, описывающие вычислительные процессы, состоящие из функций и их вызовов, на предмет корректности определений, оптимальности задействования вычислительных ресурсов, возможности распараллеливания вычислений и т.д. [18]. Это позволяет уменьшить стоимость эксплуатации систем, реализованных в рамках функционального подхода, так как часто ошибку можно быстро локализовать до конкретной функции, оперативно обследовать ее и исправить. Если система должна функционировать непрерывно, локализованную проблему и функцию, где она проявляется, можно быстро заменить на корректную, пустив поток технологического процесса по обходному пути [19].

При рассмотрении вопросов автоматизации технологических процессов раскрывается важная роль различных устройств ввода-вывода. К устройствам ввода обычно относятся разного рода датчики, в то время как устройства вывода – это, как правило, исполнительные устройства. Эти типы устройств предназначены для непосредственного взаимодействия со средой, значит, в них нарушаются свойства детерминированности и чистоты. Ввод информации из среды всегда связан с недетерминированностью, а вывод ее в среду – это побочный эффект. Таким образом, использование устройств взаимодействия со средой в процессе выполнения технологического процесса нарушает базовые принципы функционального подхода [20].

Естественным методом описания технологических процессов могут являться диаграммы потоков данных [21], на которых указываются элементы системы управления технологиче-

ским процессом, между которыми по тем или иным регламентам осуществляется передача потоков информации, материальных ресурсов или управленческих воздействий. Пример такой диаграммы абстрактного характера приведен на рисунке 2.

Проблема заключается в том, что любое устройство ввода-вывода по своей природе является недетерминированным, а также часто использует в своей работе побочные эффекты. В принципе, любая операция ввода не детерминирована, а любая операция вывода представляет собой побочный эффект. В рамках функционального подхода эта проблема решается при помощи теоретико-категориального понятия монады [22], в которую «упаковывается» нечистый код и которая, таким образом, скрывает, инкапсулируя недетерминированность и нечистоту, все проблемные зоны кода и, как следствие, технологических процессов. Тем самым монады возвращают в системы с использованием ввода-вывода возможность полноценного применения функционального подхода со всеми его положительными качествами [10].

Таким образом, диаграмма потоков данных технологического процесса может быть изоморфно отображена на диаграмму вызова функций. Каждое устройство ввода и вывода, каждая технологическая операция, входящая в технологический процесс, представляют собой функцию. Для инкапсуляции недетерминированности и побочных эффектов используется монада, скрывающая от разработчика системы и ее пользователей сквозную передачу состояния о среде, в которой функционирует система, предоставляя всю мощь функционального подхода для разработки и эксплуатации.

Модель процесса управления

Процесс управления внутренней средой интеллектуального здания основан на применении функционального подхода к сценариям различных типов. Сценарий представляет собой набор операций и условий их выполнения. Операции выполняются отдельными подсистемами и элементами интеллектуальной системы управления, которая проверяет условия применения и запускает соответствующие ветви сценария, если условие выполнено. Функциональный подход требует, чтобы операции в рамках выполняемых сценариев рассматривались в качестве функций, а все недетерминированные аспекты и побочные эффекты были инкапсулированы в монаду [10]. В связи с этим далее модель управления внутренней средой интеллектуального здания рассматривается и описывается с позиций теории множеств и теории категорий.

Теоретико-множественная модель процесса управления внутренней средой интеллектуального здания M представляет собой следующий кортеж: $M = \langle I, S, E \rangle$, где I – механизм вывода и принятия решений; S – множество сценариев; E – история состояний внутренней среды здания.

Механизм вывода и принятия решений I представляет собой набор информационных массивов и программный комплекс, выполняющий следующие функции:

- реализация сценариев из множества S ;
- объяснение полученных результатов (в случае необходимости);
- визуализация текущего и прогнозного состояний объектов управления;
- предоставление сводной и конкретизированной отчетности.

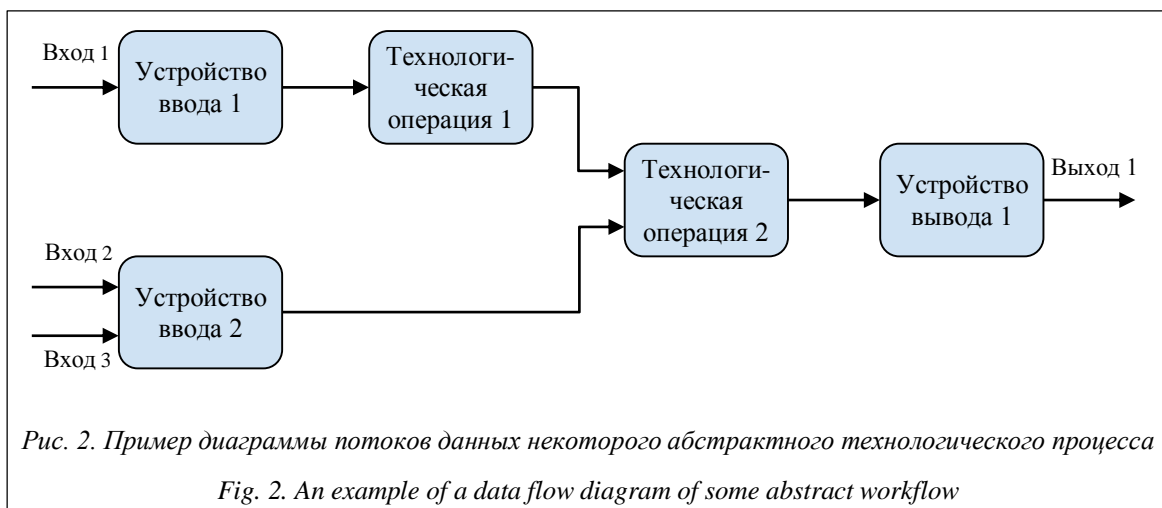


Fig. 2. An example of a data flow diagram of some abstract workflow

С теоретико-множественной точки зрения механизм вывода и принятия решений I является кортежем следующего вида: $I = \langle Kb, Im, Xp, Vz \rangle$, где Kb – база знаний; Im – универсальная машина вывода; Xp – модуль объяснения результатов вывода и принятия решений; Vz – модуль визуализации и отчетности.

База знаний Kb является гибридной и представляет собой тройку: $Kb = \langle Sn, Pr, Rp \rangle$, где Sn – семантическая сеть проблемной области управления эксплуатацией интеллектуального здания; Pr – множество продукционных правил, описывающих конкретные правила базового жизненного цикла здания; Rp – набор установленных форм отчетности для выдачи конечному пользователю или во внешние системы.

Универсальная машина вывода Im представляет собой программное средство, осуществляющее машинный вывод на базе знаний Kb с учетом текущих и исторических фактов о состоянии внутренней среды интеллектуального здания E и последовательностей шагов текущих активированных сценариев из множества сценариев S .

Модуль объяснения результатов вывода и принятия решений Xp также является программным средством, которое принимает на вход результаты работы универсальной машины вывода Im и с учетом знаний из базы знаний Kb объясняет полученный вывод и подготовленное реализованное решение.

Модуль визуализации и отчетности Vz является программным средством, которое принимает результаты работы универсальной машины вывода Im , модуля объяснения результатов вывода и принятия решений Xp и множества текущих и исторических фактов о состоянии внутренней среды интеллектуального здания E , а возвращает набор визуализированных показателей в виде графиков и диаграмм, а также набор отчетности по установленной форме. Набор визуализированных показателей определяется всем объединенным множеством параметров и сводных характеристик, принимаемых с подключенных к системе управления датчиков и исполнительных устройств. Набор форм отчетности определяется базой знаний Kb .

Множество сценариев S представляет собой объект следующего вида: $S = \{s_i\}$, $i = 1, \dots, N$, где s_i – один сценарий, состоящий из идентификатора, типа (фиксации состояния – S_1 , реактивные – S_2 , проактивные – S_3) и направленного графа, состоящего из условий и технологических операций, то есть каждый сценарий представляет собой кортеж $s = \langle id, tp, G \rangle$.

История состояний внутренней среды здания E представляет собой множество кортежей вида $E = \{ \langle t, p, v \rangle_i \}$, $i = 1, \dots, P$, где t – временная отметка; p – наименование параметра внутренней среды; v – значение параметра внутренней среды. Все приведенные соображения можно объединить в рамках графической модели, представляющей общую архитектуру системы управления интеллектуальным зданием (рис. 3).

Стрелки на рисунке показывают следующие действия:

- от сенсоров в универсальную машину вывода Im поступают значения параметров внутренней среды интеллектуального здания;
- универсальная машина вывода Im берет из семантической сети Sn и набора продукционных правил Pr все необходимые знания для обработки значений параметров внутренней среды;
- универсальная машина вывода Im выбирает один из типов сценариев $\{S_1, S_2, S_3\}$ для обработки полученного из внутренней среды значения;
- сценарии S получают и передают в историю состояний E значения параметров внутренней среды;
- универсальная машина вывода Im передает в модули Xp и Vz результаты выполнения выбранного сценария;
- модуль объяснения Xp получает из набора продукций Pr все необходимые знания для объяснения полученного универсальной машиной вывода Im результата;
- модуль объяснения Xp передает в модуль Vz результаты объяснений;
- модуль визуализации Vz получает из набора форм отчетности Rp необходимый формат вывода результатов;
- модуль визуализации Vz передает на исполнительные устройства или во внешние системы сгенерированную отчетность по установленному формату.

Материалы и методы

Представленная модель процесса управления была реализована на языке Haskell в инструментальном средстве Haskell Platform версии 8.6.3 для операционной системы Windows 64 bit. В составе этого инструментального средства использовался компилятор указанного языка программирования Glasgow Haskell Compiler версии 8.6.2.

В качестве языка представления семантической сети использован язык Resource Description Framework (RDF) с версией специфика-

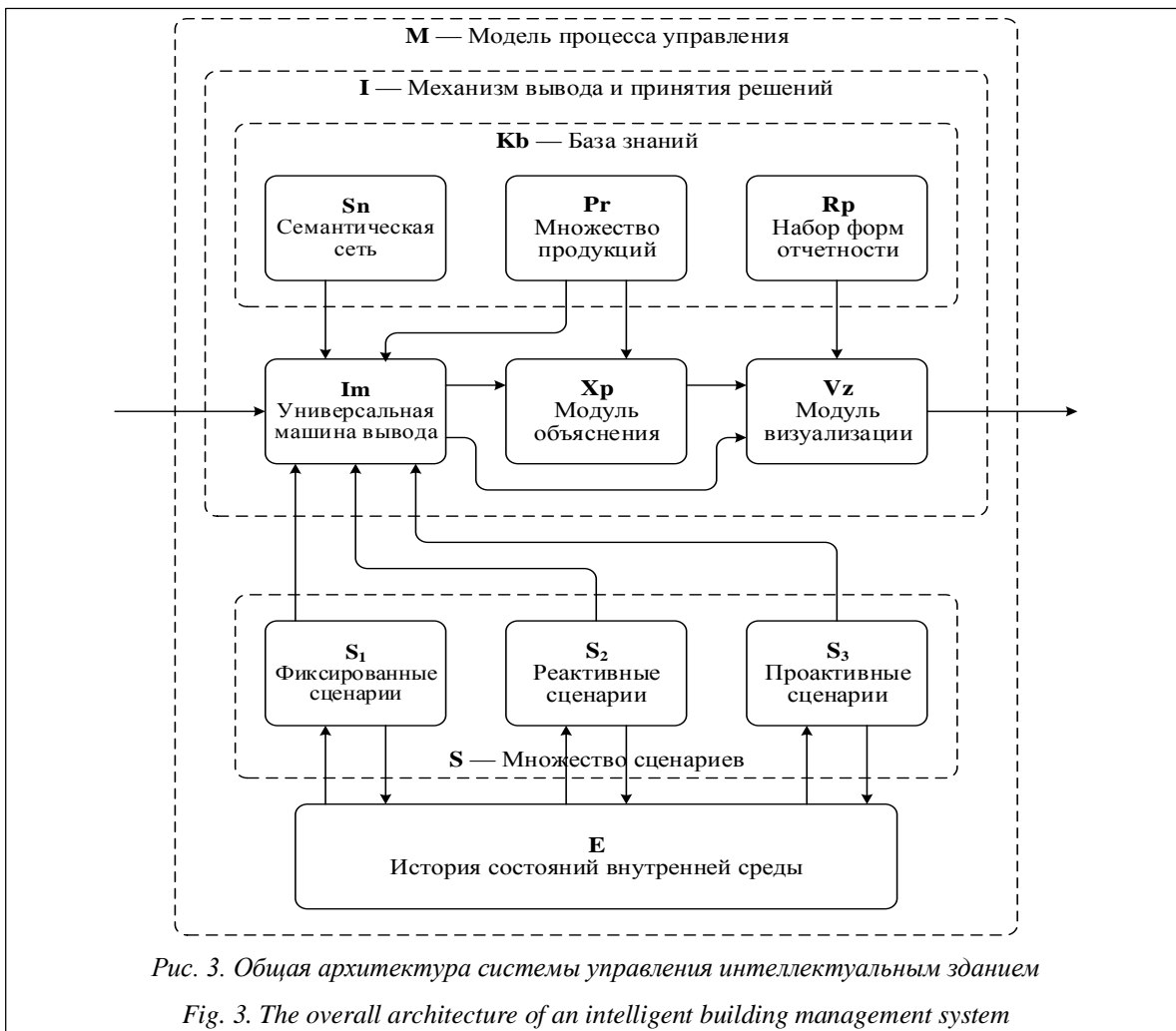


Рис. 3. Общая архитектура системы управления интеллектуальным зданием

Fig. 3. The overall architecture of an intelligent building management system

ции 1.1, для представления продукционных правил – язык разметки XML со специально подготовленным для этого описанием структуры данных, а для представления сценариев – язык программирования Haskell.

Поскольку Haskell является чистым функциональным языком, вся работа с устройствами ввода-вывода инкапсулировалась в монаду IO, а передача состояния между симуляциями технологических процессов была инкапсулирована в монаду State. Это позволило сохранить чистоту и детерминированность для функций, реализующих сценарии управления.

Результаты симуляции подтвердили обоснованность разработанной теоретической модели.

Выводы

Таким образом, использование функционального подхода при автоматизации техно-

логических процессов в интеллектуальном здании позволит достичь повышения эффективности управления технологическими процессами и степени устойчивости внутренней среды интеллектуального здания, снижения времени на локализацию и устранение аварийных ситуаций, а также расходов на эксплуатацию как самой системы управления, так и всего интеллектуального здания в целом.

Однако для получения более конкретных значений показателей эффектов, которые могут быть достигнуты при внедрении функционального подхода в управление внутренней средой интеллектуального здания, требуются дополнительные исследования.

Предполагается, что дальнейшие исследования позволят масштабировать полученные результаты на уровень комплекса зданий, муниципалитета, населенного пункта, региона и даже всего государства. В свою очередь, такое масштабирование позволит увеличить степень устойчивости и жизнеспособности зданий,

населенных пунктов и даже регионов и, как следствие, снизит нагрузку на экологию.

Литература

1. Cameron B.T. Using responsive evaluation in strategic management. *Strategic Leadership Review*, 2014, no. 4, vol. 2, pp. 22–27.
2. Писаренко Н.Л., Длигач А.А. Стратегическое управление. СПб: Питер, 2008. 327 с.
3. Asadullah M., Ahsan Raza. An Overview of home automation systems. *Proc. Conf. IEEE*, 2016, pp. 27–31. DOI: 10.1109/ICRAI.2016.7791223.
4. Кутейников А.В. Академик В.М. Глушков и проект создания принципиально новой (автоматизированной) системы управления советской экономикой в 1963–1965 гг. // *Экономическая история. Обзорение*. Вып. 15. 2011. С. 139–156.
5. Girard J.P., Girard J.L. Defining knowledge management: Toward an applied compendium. *OJAKM*, 2015, no. 3, iss. 1, pp. 111–124.
6. Рассел С., Норвиг П. Искусственный интеллект: современный подход; [пер. с англ. и ред. К.А. Птицына]. М.: Вильямс, 2006. 1408 с.
7. Bentaleb A., Yifan L., Xin J., et al. Parallel and distributed algorithms. 2016. URL: <https://www.comp.nus.edu.sg/~rahul/allfiles/cs6234-16-pds.pdf> (дата обращения: 08.09.2018).
8. Kranenburg R. The Internet of things: a critique of ambient technology and the all-seeing network of RFID. *Pijnacker, Telstar Media*, 2008, 62 p.
9. Antonopoulos N., Lee G. *Cloud computing: principles, systems and applications*. Springer, 2010, 379 p.
10. Душкин Р.В. Функциональное программирование на языке Haskell. М.: ДМК Пресс, 2006. 608 с.
11. Cheung L. Is functional programming better for your startup? *InfoWorld*, 2017. URL: <https://www.infoworld.com/article/3190185/is-functional-programming-better-for-your-startup.html> (дата обращения: 16.02.2018).
12. Guy C., Mauny M. *The functional approach to programming*. UK, Cambridge Univ. Press, 1998, 479 p.
13. Johnston J. *The allure of machinic life: cybernetics, artificial life, and the new AI*. Massachusetts, MIT Press, 2008, 477 p.
14. Душкин Р.В. *Справочник по языку Haskell*. М.: ДМК Пресс, 2008. 544 с.
15. Magnoni L. Modern Messaging for Distributed Systems. *JPCS*, 2015, vol. 608, no. 1, art. 012038. DOI: 10.1088/1742-6596/608/1/012038.
16. Friedman A. *Foundations of modern analysis*. Dover Publ., 2010, 129 p.
17. Memeti S., Pllana S., Binotto A., Kołodziej J., Brandic I. Using meta-heuristics and machine learning for software optimization of parallel computing systems: a systematic literature review. *Springer*, 2018, arXiv:1801.09444. DOI: 10.1007/s00607-018-0614-9.
18. O’Sullivan B., Goerzen J., Stewart D. *Real world Haskell*. O’Reilly Publ., 2008, 710 p.
19. Душкин Р.В., Коптев А.П. Автоматизация деловых процессов при помощи Единого комплекса автоматизированных систем управления предприятием // *ИНТЕХМЕТ-2008: сб. тезис. докл. I Междунар. науч.-практич. конф.* 2008. СПб: Изд-во СПГТИ, 2008. С. 33–34.
20. Lipovača M. *Learn you a Haskell for great good!* SF, No Starch Press, 2011, 490 p.
21. Goodland M., Riha K. *History of SSADM – an Introduction*. 1999.
22. Levent E. *Value recursion in monadic computations*. Oregon Graduate Institute Publ., 2002, 162 p.

Software & Systems
DOI: 10.15827/0236-235X.126.306-312

Received 03.10.18
2019, vol. 32, no. 2, pp. 306–312

A set-theoretic model of a functional approach to intellectualization of building and construction management processes

R.V. Dushkin¹, Deputy Director General on Intelligent Transportation Systems and a hardware-software package Smart City, roman.dushkin@gmail.com

¹ *VoiceLink Ltd, Moscow, 127322, Russian Federation*

Abstract. The paper describes a functional approach to managing the internal environment of an intelligent building. Such approach implies using and transferring unchangeable states between the peripheral equipment and central engineering systems, the absence of side effects during control actions, as well as the construction of a decentralized network of computations and decision making on an end-user device side based on the Internet of things. This allows simultaneous obtaining of all benefits of different consideration paradigms of management processes and intellectualization. Additionally, there is an possibility of emergent showing new features of the general approach to increase the degree of controllability and the operation efficiency of controlled objects.

The paper considers mathematical foundations of the organization of a distributed computing environment for implementation of a functional approach. It also briefly describes possible control scenarios in various operation modes of engineering systems of intelligent buildings. The authors describe a set-theoretic model of an intellectualized building management system. This model is a theoretical basis for recommendations on changing the paradigm of constructing a general automated control system for an intelligent building from the automation of engineering processes to the management intellectualization and the creation of an artificial intelligence system that implements a complete and autonomous cycle of building management.

The functional approach complete with new technologies (artificial intelligence, decentralized databases, the Internet of things) for managing such objects as intelligent buildings, allows transferring the operation of such buildings to a higher level of service availability, sustainability, environmental friendliness and comprehensive development of not only a controlled object, but also the hierarchy of its supersystems (municipality, region, state).

Keywords: functional approach, intelligent building, Internet of things, decentralization, artificial intelligence, intellectualization.

References

1. Cameron B.T. Using responsive evaluation in strategic management. *Strategic Leadership Review*. 2014, no. 4, vol. 2, pp. 22–27.
2. Pisarenko N.L., Dligach A.A. *Strategic Management*. 2008, 327 p.
3. Asadullah M., Raza A. An overview of home automation systems. *2nd Intern. Conf. on Robotics and Artificial Intelligence (ICRAI)*. 2016. DOI: 10.1109/ICRAI.2016.7791223.
4. Kuteynikov A.V., Academician V.M. Glushkov and the project of creating a fundamentally new (automated) system for managing the Soviet economy in 1963–1965. *Economic History. Review*. 2011, iss. 15, pp. 139–156 (in Russ.).
5. Girard J.P., Girard J.L. Defining knowledge management: Toward an applied compendium. *OJAKM*. 2015, no. 3, iss. 1, pp. 111–124.
6. Russell S., Norvig P. *Artificial Intelligence: Modern Approach*. Prentice Hall Publ., 2nd ed., 2002, 1132 p. (Russ. ed.: K.A. Ptitsyn, 2nd ed., Moscow, Williams Publ., 2006, 1408 p.).
7. Bentaleb A., Yifan L., Xin J., et al. *Parallel and Distributed Algorithms*. 2016.
8. Kranenburg R. *The Internet of Things: A Critique of Ambient Technology and the All-Seeing Network of RFID*. Pijnacker, Telstar Media Publ., 2008, 62 p.
9. Antonopoulos N., Lee G. *Cloud Computing: Principles, Systems and Applications*. Springer Publ., 2010, 379 p.
10. Dushkin R.V. *Functional Programming in the Haskell Language*. Moscow, DMK Press, 2006, 608 p.
11. Cheung L. Is functional programming better for your startup? *InfoWorld*. 2017. Available at: <https://www.infoworld.com/article/3190185/is-functional-programming-better-for-your-startup.html> (accessed February 16, 2018).
12. Guy C., Mauny M. *The Functional Approach to Programming*. Cambridge, UK, Cambridge Univ. Press, 1998, 479 p.
13. Johnston J. *The Allure of Machinic Life: Cybernetics, Artificial Life, and the New AI*. Cambridge, Massachusetts, MIT Press, 2008, 477 p.
14. Dushkin R.V. *Reference Book on the Haskell Language*. Moscow, DMK Press, 2008, 544 p.
15. Magnoni L. Modern Messaging for Distributed Systems. *JPCS*. 2015, vol. 608, no. 1, art. 012038. DOI: 10.1088/1742-6596/608/1/012038.
16. Friedman A. *Foundations of Modern Analysis*. Dover Publ., 2010, 129 p.
17. Memeti S., Pllana S., Binotto A., Kołodziej J., Brandic I. Using meta-heuristics and machine learning for software optimization of parallel computing systems: a systematic literature review. *Computing*. Springer Publ., 2018. arXiv:1801.09444. DOI: 10.1007/s00607-018-0614-9.
18. O’Sullivan B., Goerzen J., Stewart D. *Real World Haskell*. O’Reilly Publ., 2008, 710 p.
19. Dushkin R.V., Koptev A.P. Automation of business processes using the Uniform complex of enterprise automated control systems. *Proc. 1st Intern. Sci. and Pract. Conf. INTEKhMET-2008*. St. Petersburg, SPGGI Publ., 2008, pp. 33–34 (in Russ.).
20. Lipovača M. *Learn You a Haskell for Great Good!* San Francisco, No Starch Press, 2011, 490 p.
21. Goodland M., Riha K. *History of SSADM – an Introduction*. 1999.
22. Levent E. *Value Recursion in Monadic Computations*. Oregon Graduate Institute Publ., 2002, 162 p.