УДК 004.05
DOI: 10.15827/0236-235X.128.682-689

Дата подачи статьи: 22.07.19
2019. Т. 32. № 4. С. 682–689

Модель и алгоритм выбора программной архитектуры для систем Интернета вещей

Ю.В. Ядгарова [1], *аспирант, y.v.yadgarova@gmail.com*

[1] *Московский государственный технический университет им. Н.Э. Баумана, г. Москва, 105005, Россия*

В статье приведены аналитическая модель оценки стоимости и алгоритм выбора базового шаблона программных архитектур и тактик проектирования для систем Интернета вещей. Обобщено понятие IoT-технологий, сделан обзор параметров качества программных систем, выделены основные значимые параметры применительно к системам Интернета вещей, приведены методы их достижения. Необходимые параметры качества программных систем достигаются реализацией базового шаблона программной архитектуры и сопутствующих тактик проектирования.

В работе представлена аналитическая модель зависимости трудоемкости проекта, рассчитанной по методике COCOMO II, от используемых элементов программной архитектуры. Приведен алгоритм поиска базового шаблона архитектуры и тактик проектирования. Указанный алгоритм построен на основе локального поиска при решении задачи удовлетворения ограничений с минимизацией функции трудоемкости, при этом в расчет принимаются предпочтения пользователя при выборе шаблона. Модель и алгоритм позволяют выбирать наиболее подходящие для конкретного типа проекта шаблоны архитектуры и тактики на ранних этапах проектирования.

Указанный подход позволяет сократить ошибки в построении программной архитектуры на начальном этапе при выборе шаблона IoT-архитектуры. Рассмотрено применение данного подхода в проекте разработки системы гибкого управления рабочими пространствами. Применение подхода целесообразно для достижения требуемых параметров качества системы и минимизации ошибок при выборе программной архитектуры на начальных стадиях проекта, что в конечном итоге ведет к снижению его стоимости. Подход может также применяться при создании работоспособных прототипов в сжатые сроки.

Ключевые слова*: качество систем Интернета вещей, архитектура Интернета вещей, проектирование архитектуры ПО, параметры качества архитектуры ПО, тактики проектирования, шаблоны архитектуры, CSP-задача.*

Термин «Интернет вещей» был впервые упомянут в 1999 г. Кевином Эштоном [1]. В последние годы технологии Интернета вещей набирают популярность во многих сферах науки и техники, однако зачастую пользователи данной технологии не могут дать им четкое определение.

На международном уровне данная концепция приобретает сформировавшиеся черты, что можно проследить по появлению классификации типов подобных систем, стандартов взаимодействия устройств, а также по выделению фундаментальных характеристик технологии в соответствии с рекомендациями *Международного союза электросвязи* (МСЭ-Т). Однако в основе построения большинства типов систем IoT (Internet of things) так или иначе лежат принципы системного анализа.

Автор данной статьи будет придерживаться определения МСЭ-Т и считать Интернет вещей глобальной инфраструктурой для информационного общества, которая обеспечивает возможность предоставления более сложных услуг путем соединения друг с другом (физических и виртуальных) вещей на основе существующих и развивающихся функционально совместимых информационно-коммуникационных технологий [2].

Обзор публикаций

Наибольшую популярность системы Интернета вещей приобрели в 2010–2017 гг., когда удешевились оконечные устройства, появились технологии для безопасной и быстрой передачи данных, а также активно развивались Интернет и протоколы взаимодействия. Данные факторы благотворно сказались на появлении большого числа относительно недорогих и простых оконечных устройств, которые в короткие сроки можно было собрать в готовое решение. Наряду с простыми решениями на рынке стали появляться профессиональные системы с достаточно сложной и разветвленной

логикой, к возникновению которых причастны уже целые компании и международные комитеты (например, Концепция промышленности 4.0 и Industrial Internet Consorcium, Консорциум промышленного Интернета).

Проблемы выбора и построения систем Интернета вещей наиболее полно описаны в зарубежной литературе. Среди основных исследований в данной области можно выделить работы [3–6]. Проблемы разработки систем Интернета вещей в России и теоретические основы данной технологии изложены в [7, 8], а перспективы становления технологии, а также потенциальные вопросы стандартизации, адресации и сетевых коммуникаций – в [9].

Некоторые исследователи сходятся во мнении, что для приложений Интернета вещей не может быть единого стандарта программной архитектуры, так как сам по себе термин не несет какую-либо контекстную информацию для принятия даже предварительных проектировочных решений [10, 11]. Однако в данной работе предпринята попытка опровергнуть такое мнение путем выявления общих нефункциональных требований к различным типам IoT-решений и создания средства поддержки проектирования для конкретных типов систем Интернета вещей. Подобная методика применительно к системам промышленного Интернета вещей была разработана Консорциумом индустриального Интернета. Среди опубликованных документов можно выделить [12] – стандарт, регламентирующий уровни построения, сложность, технические детали и особенности внедрения подобных систем на производстве. Указанный документ принимали к сведению большинство западных компаний при разработке систем промышленного Интернета вещей, что показало высокую применимость методологии на практике.

Модель выбора шаблона программной архитектуры

Задача выбора базового шаблона архитектуры и тактик для систем Интернета вещей состоит в нахождении такой совокупности компонентов и связей между ними для базового шаблона и множества тактик, реализация которых приведет к достижению требуемых функциональных параметров и параметров качества системы при минимизации трудоемкости.

Полагаем, что критерием достижения параметра качества программной системы является реализация в выбранной архитектуре множества тактик проектирования $T = \{t_i\}$, $i = 1, \ldots, n$, отвечающих за упомянутый параметр качества. При этом некоторые шаблоны программных архитектур по своему исходному построению удовлетворяют определенным параметрам качества. Данное условие представим булевой переменной x_{ij}, определяющей соответствие базового шаблона архитектуры i требуемому параметру качества j:

$$x_{ij} = \begin{cases} 1 \mid \text{реализация тактик для параметра} \\ \quad \text{качества не требуется,} \\ 0 \mid \text{реализация тактик для параметра} \\ \quad \text{качества необходима.} \end{cases}$$

Соответствие базовых типов архитектур основным параметрам качества систем отражено в таблице 1.

В качестве оценки трудоемкости среди множества аналитических методик оценки выбрана наиболее применимая к начальной стадии проекта – COCOMO II (англ. **Co**nstructive **co**st **mo**del – конструктивная модель стоимости) [13, 14].

Для решения задачи выбора шаблона архитектуры и тактик проектирования введем понятие *вектора параметров качества* системы. Им будем называть совокупность параметров качества, зависящую от типа системы и требований: $S_t = S\{i.\ t\}$, где $i = 1, \ldots, k$ – параметр качества; $t = 1, \ldots, n$ – тип системы. Каждый параметр качества характеризуется либо реализацией его в базовом шаблоне архитектуры, либо суммой тактик проектирования, необходимых для его достижения на базовом шаблоне: $S_i = \begin{cases} x_{ij}, \\ \sum_{k=1}^{m} T_k, \end{cases}$, где T_k – тактика проектирования; m – количество тактик, необходимое для реализации параметра; $x_{ij} = 1$, если реализация тактики проектирования для достижения не требуется, $x_{ij} = 0$ в обратном случае.

Тактика проектирования, в свою очередь, может быть представлена как модификация элементов архитектурного шаблона (компонентов и связей между ними).

Однако реализация одних и тех же тактик проектирования для различных базовых шаблонов архитектуры может существенно отличаться, следовательно, отличаться будет и трудоемкость реализации. Для учета данного ограничения введем понятие типа изменения шаблона архитектуры тактикой, отражающее сущность модификации шаблона архитектуры и, следовательно, влияющее на трудоемкость. Каждая тактика может вносить от одного до не-

Таблица 1

Соответствие типов архитектур параметрам качества

Table 1

Compliance of architectures types with quality parameters

Тип архитектуры	Параметр качества					
	Способность к взаимодействию	Модифицируемость	Производительность	Доступность	Надежность	Информационная безопасность
Клиент-сервер	1	0	0	0	0	1
Точка-точка	1	0	0	0	1	0
Каналы и фильтры	0	0	1	0	0	0
Событийно-ориентированная	0	1	0	0	0	0
Доска объявлений	0	1	0	1	1	0
Сервисно-ориентированная	1	0	0	0	0	0
Брокер	1	1	0	0	0	0
Архитектура слоев	0	1	0	0	0	0
Архитектура микросервисов	1	1	0	1	1	0

скольких типов изменений. Подсчитаем с помощью методики COCOMO II относительную трудоемкость реализации добавления компонента в базовый шаблон архитектуры, состоящий из *n* компонентов.

Трудоемкость проекта задается формально как функция зависимости от количества тысяч строк кода:

$$Q = \alpha \cdot (KLOC)^b \cdot EAF, \quad (1)$$

где Q – трудозатраты, выраженные в человеко-месяцах; EAF – фактор корректировки трудозатрат в зависимости от среды; α – фактор, зависящий от детальности оценки (постоянная величина, для предварительной оценки равная 2,94); $b = B + 0,01 \cdot \sum_{j=1}^{5} SF_j$, где SF_j – факторы масштаба.

Тогда исходную трудоемкость реализации можно представить как $Q = \alpha \cdot (KLOC_0)^{b_0} \cdot EAF$. Так как детальность оценки и среда остаются неизменными, реализация дополнительного компонента зависит от количества тысяч строк кода $KLOC$ реализации и фактора масштаба b. Трудоемкость готового проекта, выраженная через сумму трудоемкостей компонентов:

$$Q = \sum_{i=1}^{n} Q_i = \alpha \cdot EAF \cdot \left(\sum_{i=1}^{n} KLOC_i \right)^b.$$

Выразим также трудоемкость разработки одного модуля через трудоемкость всего проекта: $Q_i = \alpha \cdot \left(KLOC/n \right)^b \cdot EAF$, где n – количество модулей в проекте. Тогда принимаем $t = KLOC/n$, где t – усредненное количество строк кода на один модуль. Переписывая выражение (1), получаем трудоемкость проекта:

$$Q = \alpha \cdot EAF \cdot n \cdot t^b.$$

Количество строк кода является величиной, зависящей от типа изменения шаблона. Фактор масштаба, используемый в настоящей зависимости, по сути, также зависит от типа изменения шаблона (так как риски, архитектура и т.д. зависят от типа изменения). В результате получаем $Q = \alpha \cdot EAF \cdot n(m) \cdot t^{b(m)}$, (2)

где m – тип изменения шаблона по таблице 2.

Рассчитаем для одного из типов изменений модифицированную зависимость для трудоемкости по методике COCOMO II. При добавлении нового совместимого компонента в шаблон и наличии базового шаблона архитектуры с известными величинами n и t добавление еще одного компонента означает, что добавляется элемент со средней трудоемкостью t, а также связи между компонентами, что приводит к выражению $\Delta_{доб.} = \alpha \cdot EAF \cdot (t + t_{св})^{b(m)}$.

Функция $f = b(m)$ в данном случае представляет изменение фактора масштаба в зависимости от типа изменения шаблона. Рассматривая фактор масштаба по методике COCOMO II, получаем базовое значение: $E = B + 0,01 \sum_{j=1}^{5} SF_j$, где SF_j – фактор масштаба.

Добавление компонента в существующий шаблон предполагает изменение факторов прецедентности, рисков и архитектуры (сработанность команды в какой-то степени), так как предполагает реализацию остальных компонентов.

Тогда, считая изменения факторов прецедентности от низкого до высокого, получаем фактор масштаба $n(Доб_{кш}) = 1,49$.

Получим результирующую функцию:

$$\Delta_{доб.} = \alpha \cdot EAF \cdot (t + t_{св})^{1,49}.$$

Таблица 2

Типы изменения шаблона тактикой

Table 2

Template modification types by tactics

Тип изменения	Трудоемкость	Описание
Добавление компонента в шаблон, $Доб_{кш}$	Низкая	Добавление нового совместимого компонента в шаблон, связь с другими. Структура остается прежней
Модификация компонента, $Мод_{к}$	Средняя Высокая	Меняется структура компонентов шаблона. Влияет на остальные компоненты
Реализация внутри компонента, $Вн_{к}$	Низкая	Меняется один из компонентов шаблона
Дублирование компонента, $Дуб_{к}$	Низкая	Дублируется один из компонентов, добавляются связи
Добавление компонента вне шаблона, $Доб_{квш}$	Высокая	Добавляется компонент, по структуре не подходящий шаблону

На рисунке показан график зависимости относительной трудоемкости от вариации $t_{св}$ в диапазоне от 0,1t до t.

Аналогично высчитываются зависимости для остальных типов из таблицы 2.

Таким образом, общая трудоемкость проекта для выбранного базового типа состоит из базовой относительной трудоемкости реализации шаблона (выраженной через величину t) и суммы относительных трудоемкостей n тактик проектирования:

$$Q = Q_б(t) + \sum_{i=1}^{n} Q_i(t,m) = \\ = \alpha \bullet EAF \bullet \left(n \bullet t + \sum_{i=1}^{n} Q_i(t,m) \right), \quad (3)$$

где m – тип изменения по таблице 2. Величина t зависит как от шаблона архитектуры, так и от реализации компонента и рассчитывается с помощью метода функциональных точек для каждого конкретного проекта. Таким образом, задача сводится к нахождению такой комбинации базового шаблона архитектуры и тактик, при которой относительная величина трудоемкости является минимальной: $Q \to \min$.

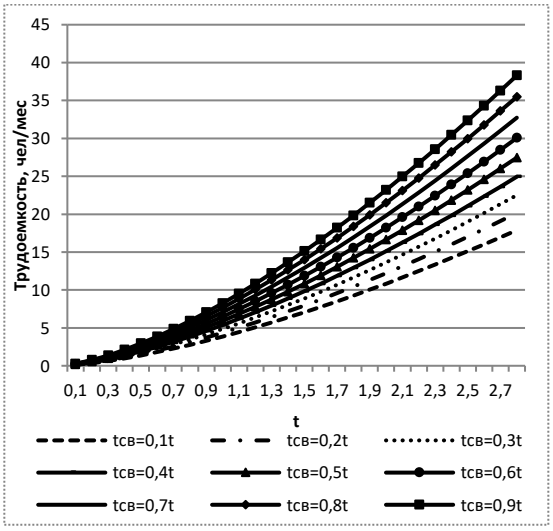

График зависимости относительной трудоемкости при изменении t и $t_{св}$

The chart of dependency of relative labor intensity when changing t and $t_{св}$

Алгоритм выбора шаблона архитектуры и тактик проектирования

Задача выбора базового шаблона и сопутствующих тактик проектирования относится к классу комбинаторных задач структурного синтеза с минимизацией целевой функции стоимости. На задачу накладываются следующие ограничения:

– структурные ограничения на комбинацию тактик и шаблонов: использование не более двух основных шаблонов вместе, ограничение тактик, если они реализованы в шаблоне (табл. 1);

– ограничения на качество: для достижения минимального уровня параметра качества реализация как минимум одной тактики проектирования, адресующей параметр качества;

– ограничение предпочтения: в области построения архитектуры большую роль играют предпочтения (опыт) команды с определенным шаблоном/тактиками проектирования;

– минимизация относительной стоимости решения, рассчитанной по методике COCOMO.

В качестве метода решения указанной задачи был выбран локальный алгоритм поиска, а начального значения для поиска – шаблон архитектуры на основе пользовательских предпочтений. В случае отсутствия предпочтений вы-

биралась случайная комбинация, удовлетворяющая множеству ограничений.

Необходимым условием для решения задачи конфигурации является определение вектора достигаемых параметров качества системы, описанных ранее.

При фиксированном количестве параметров качества состояние задачи определяется множеством конфигураций $X = \left\{ X_{ij}^{ш}, X_{ik}^{т} \right\}$ базовых шаблонов и сопутствующих тактик проектирования ($X_{ij}^{ш}$ – включенность *j*-го шаблона в *i*-ю конфигурацию, $X_{ik}^{т}$ – включенность *k*-й тактики в *i*-ю конфигурацию), удовлетворяющих ограничениям. Областью определения конфигурации X_i является множество шаблонов и тактик: $D_i = \left\{ D_j^{ш}, D_k^{т} \right\}$, областью значений – множество совместимых присваиваний, определяющих достижение вектора параметра качества. Представлением решения задачи конфигурации служит таблица, в строках которой содержатся варианты конфигурации, в столбцах – шаблоны и тактики проектирования (табл. 3).

В качестве алгоритма решения данной задачи был выбран локальный алгоритм поиска (так как можно достаточно точно определить локальную окрестность точки путем присвоения начальной конфигурации исходя из предпочтений пользователя). В качестве эвристики используется эвристика с минимальными конфликтами, которая находит ближайшую конфигурацию-преемник исходя из условия минимизации количества конфликтов с уже присвоенными переменными. Перечислим основные шаги алгоритма.

1. Ввод вектора параметров качества системы. На данном этапе среди основных важных параметров качества систем Интернета вещей выделяются наиболее существенные для данного типа системы.

2. Ввод предпочтений пользователя в части шаблонов архитектур. Введенные значения используются для определения начальной конфигурации решения на этапе поиска.

3. Формирование начальной конфигурации решения. В случае наличия ограничений предпочтения поиск начинается с удовлетворения данных ограничений, в случае их отсутствия выбирается случайная начальная конфигурация.

4. Для выбранной конфигурации вычисление относительной трудоемкости *Q* по методике COCOMO II.

5. Формирование окрестности путем выделения набора конфигураций с замененным элементом (тактикой или шаблоном). На данном этапе применяется эвристика с минимальными конфликтами.

6. Для элементов окрестности вычисление относительной трудоемкости по методике COCOMO II. Выбирается новая начальная конфигурация по условию $Q \to \min$.

Результаты исследования и их обсуждение

В таблице 4 представлен пример относительной оценки шаблона архитектуры Брокер, выполненной для системы гибкого управления рабочими пространствами.

Вектор параметров качества для систем Интернета вещей указанного типа будет включать надежность (Н), производительность (П), модифицируемость (Мод) и информационную безопасность (ИБ) как первичные параметры. Особенностью выбранного шаблона архитектуры является высокий уровень модифицируемости и способности к взаимодействию. При

Таблица 3

Представление решения

Table 3

Submitted solutions

Конфигурация	Шаблоны и тактики проектирования									
	Клиент-сервер	Точка-точка	Другие шаблоны	Архитектура микросервисов	Пинг-эхо	Пульсация	Обработка исключений	Голосование	Активная избыточность	Другие тактики
Конфигурация 1	1	0	…	0	1	0	0	1	0	…
Конфигурация 2	1	0	…	0	0	1	0	1	0	…
Конфигурация 3	0	1	…	0	0	1	0	0	1	…
….	…	…	…	…	…	…	…	…	…	…

Таблица 4

Оценка шаблона архитектуры Брокер

Table 4

Evaluation of the Broker architecture pattern

Тактика	Тип изменения шаблона	Относительная стоимость
Увеличение вычислительной эффективности (П)	$Мод_к$	1,19*t*
Уменьшение накладных расходов (П)	$Мод_к$	1,19*t*
Управление темпом событий (П)	$Доб_{кш}$	3,857*t*
Распараллеливание (П)	$Мод_к$, $Дуб_к$	1,428*t*
Копии данных (П)	$Дуб_к$	0,238*t*
Улучшение доступных ресурсов (П)	$Мод_к$	1,19*t*
Политика планирования (П)	$Доб_{кш}$, $Мод_к$	5,047*t*
Пинг-эхо (Н)	$Мод_к$	1,19*t*
Обработка исключений (Н)	$Мод_к$	1,19*t*
Голосование (Н)	$Дуб_к$, $Доб_{кш}$	4,095*t*
Активная избыточность (Н)	$Дуб_к$	0,238*t*
Пассивная избыточность (Н)	$Дуб_к$, $Мод_к$	1,458*t*
Повторное включение компонента: затемнение (Н)	$Дуб_к$, $Мод_к$	1,458*t*
Аутентификация пользователей (ИБ)	$Доб_{кш}$	3,857*t*
Авторизация пользователей (ИБ)	$Доб_{кш}$	3,857*t*
Сохранение конфиденциальности данных (ИБ)	$Мод_к$	1,19*t*
Сохранение целостности (ИБ)	$Мод_к$	1,19*t*

этом для достижения вектора параметров качества необходимо дополнительно реализовать тактики, адресующие надежность, производительность и безопасность. Список данных тактик с относительной стоимостью реализации ($n = 5$, а $t_{мод.} = 0{,}2t$) частично представлен в таблице 4.

Подставляя данные значения в (3), получим выражение для относительной оценки трудоемкости: $Q = \alpha \bullet EAF \bullet (n \bullet t + 33{,}863 \bullet t)$, которое можно использовать в оценке и выборе результирующей архитектуры.

Заключение

В настоящей статье был представлен метод оценки качества программной архитектуры систем Интернета вещей на ранних стадиях проектирования. Проведен аналитический обзор, затрагивающий определение существующих IoT-технологий, выделены основные значимые атрибуты качества систем IoT, а также приведены методы их оценки, влияющие на выбор программной архитектуры системы. Сформулирована аналитическая модель выбора программной архитектуры исходя из требуемых параметров качества системы при минимизации трудоемкости проекта. Построен алгоритм выбора базового шаблона архитектуры и тактик проектирования, построенный на решении задачи CSP.

Метод применен для оценки архитектуры построения системы гибкого управления рабочими пространствами. Дальнейшие направления исследований в данной области будут затрагивать обобщение данного метода на системы, выходящие за рамки области IoT, а также особенности применения метода на практике в процессе создания прототипов программных систем.

Литература

1. Sundmaeker H., Guillemin P., Friess P., Woelffle S. Vision and challenges for realising the Internet of Things. CERP IoT, 2010, 230 p.
2. Рекомендация МСЭ-Т Y.2060. Обзор Интернета вещей. 2012. 22 с. URL: https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11559&lang=ru (дата обращения: 12.03.2019).
3. Bass A., Bauer M., Fiedler M., Kramp T., van Kranenburg R., Lange S., Meissner S. Enabling Things to Talk. Springer-Verlag GmbH, 2013, 325 p.
4. Xia F., Yang L.T., Wang L., Vinel A. Internet of things. Int. J. of Commun. Syst. 2012, vol. 25, no. 9, pp. 1101–1109. DOI: https://doi.org/10.1002/dac.2417.

5. Weber R.H. Internet of things – new security and privacy challenges. Computer Law & Security Review, 2010, vol. 26, iss. 1. DOI: https://doi.org/10.1016/j.clsr.2009.11.008.

6. Wortmann F., Flüchter K. Internet of things. Business & Inform. Syst. Eng., 2015, vol. 57, no. 3, pp. 221–224.

7. Росляков А.В., Ваняшин С.В., Гребешков А.Ю. Интернет вещей. Самара: Изд-во ПГУТИ, 2014. 200 с.

8. Черняк Л. Интернет вещей: новые вызовы и новые технологии // Открытые системы. СУБД. 2013. № 4. С. 14–18.

9. Алгулиев Р., Махмудов Р. Интернет вещей // Информационное общество. 2013. № 3. С. 42–48.

10. Gubbi J., Marusicet S., Buyya R., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems. 2013, vol. 29, no. 7, pp. 1645–1660. DOI: 10.1016/j.future.2013.01.010.

11. Khan R., Zaheer R., Khan S.U., Khan S. Future internet: the internet of things architecture, possible applications and key challenges. Proc. FIT Conf., 2012, pp. 257–260. DOI: 10.1109/FIT.2012.53.

12. Lin S.W., Martin R.A., Miller B.W., Durand J. et al. Industrial internet reference architecture technical report. IIC, 2015.

13. Boehm B.W. Software engineering economics. Broy M., Denert E. (eds.): Pioneers and Their Contributions to Software Engineering, 2001, pp. 99–150. DOI: 10.1007/978-3-642-48354-7_5.

14. Boehm B., Horowitz E., Madachy R., Clark B., Westland J.C., Selby R. Cost models for future software life cycle processes: COCOMO 2.0. Annals of Software Engineering. 1995, vol. 1, no. 1, pp. 57–94. DOI: 10.1007/BF02249046.

Software & Systems
DOI: 10.15827/0236-235X.128.682-689

Received 22.07.19
2019, vol. 32, no. 4, pp. 682–689

Model and algorithm of selecting software architecture for IoT systems

Yu.V. Yadgarova [1], *Postgraduate Student, y.v.yadgarova@gmail.com*

[1] *Bauman Moscow State Technical University, Moscow,105005, Russia*

Abstract. The paper presents an analytical model for cost estimation and an algorithm for selecting a basic pattern of software architecture, as well as design tactics for the IoT (Internet of things) systems. It generalizes the concept of IoT technologies, reviews software systems quality parameters, emphasizes the main significant parameters applicable to IoT systems and proposes methods to achieve them. The necessary quality parameters of software systems are achieved by implementing the basic pattern of software architecture and related design tactics.

The paper introduces the analytical model of dependency of project labor intensity on the used elements of software architecture. The project labor intensity is calculated according to the COCOMO II method. The search algorithm for the basic architecture pattern and design tactics is presented. The indicated algorithm is built on the basis of a local search when solving the problem of satisfying constraints with minimizing the labor intensity function. Selecting a pattern, user preferences are also taken into account. The model and algorithm allow selecting the most appropriate architecture patterns and tactics for a particular type of project at early stages of their design.

Indicated approach allows reducing errors in software architecture design at the initial stage of IoT-architecture pattern selection. The paper considers implementation of this approach in the project of flexible workspace management system development. It is advisable to use the approach to achieve the required system quality parameters and minimize errors when selecting software architecture at the initial stages of the project, finally reducing the project cost. The approach can also be used to develop fully functional prototypes on a tight schedule.

Keywords: quality of IoT systems, IoT architecture, software architecture design, quality parameters of software architecture, design tactics, architecture pattern, CSP-task.

References

1. Sundmaeker H., Guillemin P., Friess P., Woelffle S. *Vision and Challenges for Realising the Internet of Things*. CERP IoT, 2010, 230 p.
2. *Recommendation MSE-T Y.2060. Internet of Things Review*. 2012, 22 p. Available at: https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11559&lang=ru (accessed 12.03.2019).
3. Bassi A., Bauer M., Fiedler M., Kramp T., van Kranenburg R., Lange S., Meissner S. *Enabling Things to Talk*. Springer-Verlag GmbH, 2013, 325 p.
4. Xia F., Yang L.T., Wang L., Vinel A. Internet of Things. *Int. J. of Commun. Syst.* 2012, vol. 25, no. 9, pp. 1101–1109. DOI: 10.1002/dac.2417.
5. Weber R.H. Internet of Things – New Security and Privacy Challenges. *Computer Law & Security Review*. 2010, vol. 26, iss. 1. DOI: 10.1016/j.clsr.2009.11.008.
6. Wortmann F., Flüchter K. Internet of things. *Business & Information Systems Engineering*. 2015, vol. 57, no. 3, pp. 221–224. DOI: 10.1007/s12599-015-0383-3.
7. Roslyakov A.V., Vanyashin S.V., Grebeshkov A.Yu. *Internet Veshchey*. Samara, PSUTI Publ., 2014, vol. 340, 200 p.
8. Chernyak L. Internet of things: new challenges and new technologies. *OSP. DMS*. 2013, no. 4, pp. 14–18.
9. Alguliev R., Makhmudov R. Internet of things. *Information Society*. 2013, no. 3, pp. 42–48.
10. Gubbi J., Marusicet S., Buyya R., Palaniswami M. Internet of things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*. 2013, vol. 29, no. 7, pp. 1645–1660. DOI: 10.1016/j.future.2013.01.010.
11. Khan R., Zaheer R., Khan S.U., Khan S. Future internet: the internet of things architecture, possible applications and key challenges. Frontiers of Information Technology. *Proc. 10th Intern. Conf. IEEE, FIT*, 2012, pp. 257–260. DOI: 10.1109/FIT.2012.53.
12. Lin S.W., Martin R.A., Miller B.W., Durand J., et al. Industrial internet reference architecture. *IIC, Tech. Rep*. 2015. DOI: 10.13140/RG.2.2.18076.90249.
13. Boehm B.W. Software engineering economics. Broy M., Denert E. (eds.): *Pioneers and Their Contributions to Software Engineering*, 2001, pp. 99–150. DOI: 10.1007/978-3-642-48354-7_5.
14. Boehm B., Horowitz E., Madachy R., Clark B., Westland J.C., Selby R. Cost models for future software life cycle processes: COCOMO 2.0. *Annals of Software Engineering*. 1995, vol. 1, no. 1, pp. 57–94. DOI: 10.1007/BF02249046.

Для цитирования

Ядгарова Ю.В. Модель и алгоритм выбора программной архитектуры для систем Интернета вещей // Программные продукты и системы. 2019. Т. 32. № 4. С. 682–689. DOI: 10.15827/0236-235X.128.682-689.

For citation

Yadgarova Yu.V. Model and algorithm of selecting software architecture for IoT systems. *Software & Systems*. 2019, vol. 32, no. 4, pp. 682–689 (in Russ.). DOI: 10.15827/0236-235X.128.682-689.