

Koshevoy N. D., Kalashnikov E. E., Kostenko E. M., Cherapaschuk G. A.

MODELING AND OPTIMIZATION OF WEIGHT-MEASURING SYSTEM FOR CONTINUOUS DOZING OF DRY MATERIALS

In this article the mathematical models of a weight measuring system have been developed with the use of optimum price expenses experiment planning. The optimum modes of its operation have been determined.

**Key words:** experiment planning, price expenses, mathematical model, weight measuring system, optimum modes.

УДК 004.03

Кудерметов Р. К.

Канд. техн. наук, заведующий кафедрой Запорожского национального технического университета

## ЗАДАЧА ОПТИМИЗАЦИИ ДЛЯ ИЕРАРХИЧЕСКИХ СИСТЕМ НА ОСНОВЕ СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ

Сформулирована задача оптимизации для системы, которая имеет иерархическую многоуровневую структуру и построена с использованием сервис-ориентированной архитектуры. Определены целевые функции каждого из уровней такой системы и показано, что эта система имеет межуровневую функцию качества. Рассмотрены свойства межуровневой функции качества при условии, что система обладает свойством монотонности.

**Ключевые слова:** сервис-ориентированная архитектура, иерархическая многоуровневая система, постулат сатисфакции, межуровневая функция качества.

### ВВЕДЕНИЕ

Термин «Сервис-ориентированная архитектура» (Service-oriented architecture, SOA) имеет много определений. Одни определения учитывают технические аспекты SOA, другие – информационно-сетевые, третьи подчеркивают преимущества для бизнеса. В данной статье наиболее подходящим будет одно из официальных бизнес-определений фирмы IBM: «SOA предлагает возможность гибкой работы с элементами бизнес-процессов и лежащей в их основе IT-инфраструктурой как безопасными, стандартизированными компонентами (службами), которые можно использовать многократно и комбинировать при изменении приоритетов бизнеса» [1].

Из этого определения следует, SOA – это компонентная модель, в которой компонентами являются сервисы, и эти сервисы взаимодействуют в компьютерной среде, в общем случае с использованием компьютерной сети. Ключевым понятием для SOA является *интероперабельность*, т. е. способность взаимодействовать. Интероперабельность достигается за счет использования стандартов, разработанных организациями W3C, OASIS и WS-I. Организация WS-I определила так называемый базовый профиль интероперабельности, который называется Web-Services Interoperability Basic Profile и включает в себя стандарты:

– XML и XML Schema 1.0 (XSD). Для определения моделей и моделей обмена данными между сервисами;

– SOAP 1.1 (Simple Object Access Protocol). Протокол, определяющий правила, по которым сервисы обмениваются сообщениями, записанными на языке XML;

– WSDL 1.1 (Web Services Description Language). Язык описания Web-сервисов, который включает описание реализуемых сервисом методов, адресную и др. информацию;

– UDDI 2.0. (Universal Description, Discovery and Integration). Определяет правила для отображения информации о сервисах в распределенном UDDI-реестре сервисов, размещенных в Интернете.

В определениях SOA особенно часто выделяют свойство слабой связанности сервисов друг с другом, что обусловлено, прежде всего, использованием вышеназванных стандартов. Поэтому для организации некоторого процесса с применением SOA могут привлекаться независимые поставщики сервисов – провайдеры. Еще одна характерная особенность архитектуры SOA – возможность генерации новых, комплексных, сложных сервисов из более простых сервисов. Процессы формирования таких сервисов и управление ими получили названия оркестровка и хореография сервисов (Web-services orchestration,

choreography) [2, 3]. Таким образом, сервисы могут синтезироваться, модифицироваться, быть распределенными во времени и пространстве и принадлежать разным организациям.

### ПОСТАНОВКА ЗАДАЧИ

Наряду с совершенствованием технологий создания и использования Web-сервисов, интеграции их в предприятия, кооперации и виртуальные организации, в настоящее время развиваются методы и технологии управления сервисами. Они охватывают вопросы управления жизненным циклом сервисов, создание новых сервисов из множества других сервисов, образование сложных сервисов *ad hoc* (на лету) для решения специфических задач, в том числе одноразовых, использование онтологий для определения номенклатуры сервисов и т. д. [3].

Вместе с тем, важным является исследование структурных свойств таких сервисов и использующих их организаций. Прежде всего, для реализации некоторого полезного процесса в организации нужно убедиться, что необходимые сервисы существуют. Также необходимо определить, насколько использование сервисов эффективно с точки зрения организации.

Как в любом предприятии, в организации, использующей сервисы, возможны конфликты между участниками организации, которые обусловлены в первую очередь несопадением их интересов (целей). В данном случае положение усугубляется ввиду основного положительного свойства SOA – слабой связанности. Действительно, если координатор, управляющий основным процессом в организации, преследует свои цели, то у провайдеров сервисов могут быть совсем иные цели.

Далее под системой будем понимать объединение процесса и организации, которая управляет процессом, совокупности провайдеров сервисов или просто сервисов. Структура такой системы показана на рис. 1.

Здесь показан некоторый процесс и сервисы, которые используются для обеспечения некоторого подпроцесса. Система представлена в виде иерархической структуры: сервисы первого уровня взаимодействуют с подпроцессом и, в свою очередь, могут использовать сервисы второго уровня.

Введем следующие обозначения для сервисов: уровни в иерархии сервисов будем обозначать с помощью верхнего индекса, первым нижним индексом будем обозначать номер сервиса в рамках уровня, т. е. тип сервиса на данном уровне, а вариант сервиса определенного типа – вторым нижним индексом. Та-

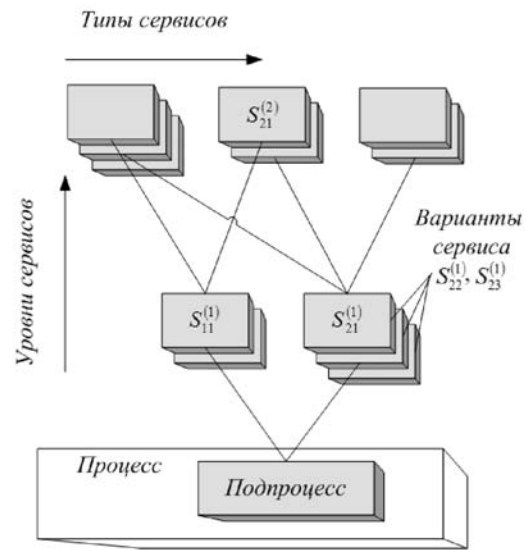


Рис. 1. Иерархическая организация системы с использованием SOA

ким образом, сервис будет обозначаться как  $S_{jk}^{(i)}$ , где  $i = 1, \dots, I$  – уровень сервиса,  $j = 1, \dots, J_i$  – тип сервиса  $i$ -го уровня и  $k = 1, \dots, K_{j_i}$  – вариант сервиса  $j$ -го типа  $i$ -го уровня.

Координатор процесса для каждого из подпроцессов *должен* выбрать необходимые типы сервисов из множества  $S^{(1)} = \{S_1^{(1)}, \dots, S_{J_1}^{(1)}\}$  с целью организации подпроцесса, а для каждого  $j$ -го типа сервиса *может* выбрать вариант из множества  $S_j^{(1)} = \{S_{j1}^{(1)}, \dots, S_{jK_j}^{(1)}\}$  с целью достижения некоторой эффективности организации процесса.

Для провайдера  $j$ -го типа сервиса 1-го уровня *может* потребоваться  $p_j$  сервисов 2-го уровня. Тогда он *должен* использовать эти  $p_j$  сервисов из множества  $S_j^{(2)} = \{S_{j1}^{(2)}, \dots, S_{j2}^{(2)}\}$ , но из каждого  $j$ -го типа сервиса 2-го уровня *может* выбрать вариант сервиса из множества  $S_{j1}^{(2)} = \{S_{j11}^{(2)}, \dots, S_{j1K_j}^{(2)}\}$ .

Если говорить о целях, которые преследуют все участники такой системы, то не исключено, что они могут конфликтовать. Предполагается, что координатор процесса непосредственно может выбирать варианты сервисов для минимизации целевой функции своей организации и даже влиять в какой-то мере на цели провайдеров сервисов первого уровня, но, в общем случае, влиять на цели провайдеров сервисов второго и более высоких уровней и на их качество не может. Можно предположить, что такое влияние возможно, но косвенное, например, с помощью экономических или политических методов. В любом случае процесс координации целей провайдеров

сервисов может быть гораздо медленнее, чем основной процесс организации.

В данной работе ставится задача исследовать структурные свойства определенной выше системы и взаимосвязи целевых функций ее участников.

## ОСНОВНАЯ ЧАСТЬ

Для рассмотрения системы, которая включает участников с разнородными целями, необходимо определиться с тем, что является целью такой системы. Если рассматриваемая система образовалась благодаря наличию достаточного числа и/или удачному сочетанию экзогенных и эндогенных системообразующих факторов (а не целенаправленно создана), устойчива, функционирует и полезна, скажем для общества, значит она, возможно, имеет цель. Однако получение результата на выходе процесса и его эффективное достижение определяют целевую функцию не системы в целом, а локальную цель координатора или организации, осуществляющей процесс. Поэтому такую целевую функцию следует понимать как *локальную*.

Цели провайдеров сервисов также являются локальными, поскольку провайдер сервиса может и не знать целей координатора процесса и цели системы, если она есть. Более того, цели провайдеров сервисов могут быть даже противоположными целям координатора процесса, а также целям других провайдеров как на одном уровне иерархии, так и на разных уровнях. Таким образом, достижение минимумов локальными целевыми функциями провайдеров не обязательно обеспечивает минимум целевой функции координатора процесса. Исследуем, как могут быть связаны между собой целевые функции участников системы.

Рассматриваемая система имеет многоуровневую иерархическую структуру и для того, чтобы она могла существовать, должен выполняться постулат сатисфакции [4]. Смысл постулата сатисфакции (1) заключается в том, что сервисы второго уровня удовлетворяют процесс тогда, когда они удовлетворяют сервисы первого уровня.

$$(\forall s^{(2)})(\forall s^{(1)})\{[\Pi(s^{(2)}, S^{(1)}(s^{(2)})) \& P(s^{(1)}, s^{(2)})] \Rightarrow \Pi(f(s^{(1)}, S^{(2)}), s^{(2)})\}, \quad (1)$$

где  $\Pi(s^{(i)}, S^{(i)})$  – предикат « $s^{(i)}$  – есть результат сервиса  $S^{(i)}$ », т. е. предикат  $\Pi(s^{(i)}, S^{(i)})$  является истинным тогда и только тогда, когда  $S^{(i)}$  – сервис, а  $s^{(i)}$  – результат этого сервиса;  $f(s^{(1)})$  – некоторая полезная функция, которая реализуется в процессе благодаря

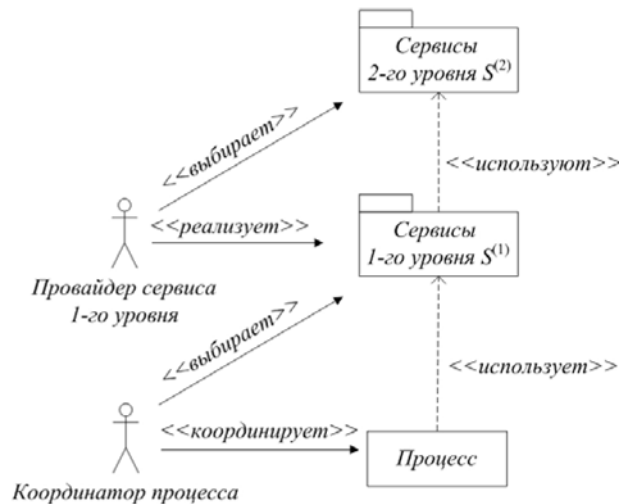


Рис. 2. UML-представление двухуровневой системы с SOA

сервису первого уровня;  $P(s^{(1)}, s^{(2)})$  – предикат, связывающий сервисы первого и второго уровней и являющийся истинным тогда и только тогда, когда существует такой результат  $s^{(1)}$  сервиса  $S^{(1)}$ , который создается благодаря результату  $s^{(2)}$  сервиса  $S^{(2)}$  (рис. 2).

Очевидно, что этот постулат можно обобщить и на случай большего числа уровней иерархии сервисов.

Другими словами, это формальное выражение того факта, что существование иерархической системы, включающей в себя элементы, которые предоставляют сервисы и сами могут зависеть от сервисов, возможно, если все составные элементы системы обеспечивают друг друга необходимыми сервисами и сервисы обеспечивают целевой процесс.

Сформулируем задачу оптимизации для координатора процесса. Вначале следует предположить, что система может состоять из нескольких уровней и/или контуров управления для осуществления основного процесса, а также содержать несколько уровней сервисов как для вспомогательных функций в процессе, так и основных. Т.е. такая система является смешанной и содержит как координирующие воздействия на элементы различных уровней иерархии и процесс, так и связи, возникающие в результате выбора подходящих сервисов провайдерами нижестоящих сервисов, а также координатором процесса таких сервисов, которые непосредственно взаимодействуют с основным процессом. Что касается координации, то такие системы подробно исследованы М. Месаровичем в [5]. В данной работе рассматривается только та составляющая системы, которая обеспечивается выбором сервисов (рис. 2).

Сделаем еще несколько очень важных уточнений: 1) в общем случае каждый тип сервиса может реализовываться как одним провайдером, так и несколькими; 2) каждый вариант одного типа сервиса также может обеспечиваться как одним, так и разными провайдерами; 3) для реализации конкретного варианта определенного сервиса может потребоваться разное число типов других сервисов. Поэтому реальная структура системы может быть различной сложности, и от нее может зависеть окончательная формулировка задачи оптимизации.

Возьмем для определенности простой вариант структуры системы, а именно, будем считать, что каждый вариант любого типа сервиса 1-го уровня использует одинаковое число типов сервисов 2-го уровня. Например, для реализации сервиса  $S_1^{(1)}$  необходимы сервисы  $\{S_1^{(2)}, S_3^{(2)}, S_4^{(2)}\}$ , а для реализации  $S_2^{(1)}$  – необходимы  $\{S_1^{(2)}, S_2^{(2)}\}$ , провайдер сервиса  $S_1^{(1)}$  может выбрать по своему усмотрению комбинации вариантов, например  $\{S_{12}^{(2)}, S_{31}^{(2)}, S_{42}^{(2)}\}$  или  $\{S_{11}^{(2)}, S_{32}^{(2)}, S_{43}^{(2)}\}$ .

Предположим, что процесс  $P$  состоит только из одного подпроцесса и будем считать, что его результат  $Y$  зависит непосредственно только от сервисов первого уровня. Если  $P$  – выходная функция процесса, то  $P: S^{(1)} \rightarrow Y$ . Функцию качества задачи, решаемой координатором, можно выразить как  $G: \bar{S}^{(1)} \times Y \rightarrow V$ , где  $\bar{S}^{(1)} = S_1^{(1)} \times \dots \times S_{J_1}^{(1)}$ ,  $V$  – некоторое множество платежей, частично или полностью упорядоченное отношением  $\leq$ .

Множеством результатов сервисов 1-го уровня является множество  $J_1$ -мерных векторов  $s^{(1)}$ , связанных отношением  $\bar{S}^{(1)}$ . Каждый  $j$ -й сервис 1-го уровня имеет  $k_j^{(1)}$  вариантов, следовательно, всего таких векторов будет  $M = \prod_{j=1}^{J_1} k_j^{(1)}$  и мощность  $|V| = M$ .

Целевую функцию координатора процесса можно задать, исходя из введенных выходной функции процесса и функции качества, следующим образом:

$$g(s^{(1)}) = G(s^{(1)}, P(s^{(1)})), \quad (2)$$

а оптимальным решением задачи координатора процесса будет такой выбор  $J_1$ -мерного вектора  $\hat{s}^{(1)}$  среди  $s_m^{(1)}$  результатов сервисов 1-го уровня ( $m = 1, \dots, M$ ), что

$$g(\hat{s}^{(1)}) = \min_{m \in M} g(s_m^{(1)}). \quad (3)$$

Теперь рассмотрим задачу одного из провайдеров сервиса 1-го уровня как задачу оптимизации. Возь-

мем для определенности сервис 1-го типа, т. е.  $S_1^{(1)}$ . Положим, что для реализации этого типа сервиса требуется множество, состоящее из  $q$  типов сервисов 2-го уровня  $\tilde{S}_1^{(2)} = \{S_{j_1}^{(2)}, \dots, S_{j_q}^{(2)}\}$ . Если каждый из этих типов имеет  $k_{j_r}$  вариантов,  $r = 1, \dots, q$ , то можно рассмотреть отображение  $h_1: \bar{S}_1^{(2)} \rightarrow V_1$ , где  $\bar{S}_1^{(2)} = S_{j_1}^{(2)} \times \dots \times S_{j_q}^{(2)}$ ;  $V_1$  – множество платежей провайдера сервиса  $S_1^{(1)}$ , частично или полностью упорядоченное отношением  $\leq$ , мощность  $|V_1| = N_1$ ,  $N_1 = \prod_{r=1}^q k_{j_r}$ . Тогда задачу провайдера сервиса  $S_1^{(1)}$  можно записать как

$$h_1 = h_1(s_1^{(2)}), \quad (4)$$

где  $s_1^{(2)}$  –  $q$ -мерные векторы результатов сервисов, связанных отношением  $\bar{S}_1^{(2)}$ , а его оптимальным решением будет такой выбор вектора  $\hat{s}_1^{(2)}$  среди  $s_{1n_1}^{(2)}$  ( $n_1 = 1, \dots, N_1$ ), что

$$h_1(\hat{s}_1^{(2)}) = \min_{n_1 \in N_1} h_1(s_{1n_1}^{(2)}). \quad (5)$$

Таким образом, каждый  $p$ -й сервис 1-го уровня  $S_p^{(1)}$  имеет целевую функцию  $h_p = h_p(s_p^{(2)})$  при реализации результата  $s_p^{(1)}$ ,  $p = 1, \dots, J_1$  и  $h_p: \bar{S}_p^{(2)} \rightarrow V_p$ . Областью значений функций  $h_p$  будут соответствующие множества  $V_p$ , частично или полностью упорядоченные отношением  $\leq$  множества платежей провайдеров сервисов  $S_p^{(1)}$ ,  $|V_p| = N_p$ ,  $N_p = \prod_{r=1}^{q_p} k_{j_r}$ , где  $q_p$  – число сервисов 2-го уровня, необходимых для реализации  $p$ -го сервиса 1-го уровня.

Тогда, рассматривая систему в целом и обозначив совокупность областей значений функций  $h_p$  как  $\bar{V}_p = V_1 \times \dots \times V_{J_1}$ , можно построить отношение

$$\Omega \subseteq \bar{V}_p \times V, \quad (6)$$

которое связывает множества платежей провайдеров 1-го уровня и координатора.

Отметим, что в системе, построенной на принципах координации, где на месте результатов сервисов выступают координирующие сигналы, возможны такие векторы локальных затрат, для которых не существует соответствующих координирующих сигналов [5]. Т. е., если бы  $s^{(2)}$  представляли собой искомые для решения задачи управляющие сигналы, то возможны векторы  $(v_1, \dots, v_{J_1})$  в  $\bar{V}_p$ , не являющиеся результатом любых  $s^{(2)}$ .

Для существования системы на основе использования сервисов должен выполняться постулат сатисфакции. Предположим, что  $\Omega$  – функция, обозначим ее как  $\omega$ , тогда

$$\omega: \bar{V}_p \rightarrow V. \quad (7)$$

Таким образом, мы предположили существование межуровневой функции качества

$$g(s^{(1)}) = \omega(h_1(s^{(2)}), \dots, h_{J_1}(s^{(2)})), \quad (8)$$

т. е. функции зависимости целевой функции координатора от целевых функций провайдеров сервисов для системы, построенной с использованием сервисов для организации некоторого процесса. В следующей теореме доказывается существование межуровневой функции качества для такой системы.

**Теорема 1.** Система, для которой выполняется постулат сатисфакции (1), имеет межуровневую функцию качества.

*Доказательство.* Одним из условий постулата сатисфакции является истинность предиката  $\mathbf{P}(s^{(1)}, s^{(2)})$ , т. е. существование такого результата  $s^{(1)}$  сервиса  $S^{(1)}$ , который создается благодаря результату  $s^{(2)}$  сервиса  $S^{(2)}$ . Значит, если система, созданная на базе сервисов, существует, такой предикат истинен для каждого сервиса 1-го уровня и каждого необходимого для его реализации набора сервисов 2-го уровня, т. е.  $(\forall s^{(2)})(\forall s^{(1)})\mathbf{P}(s^{(1)}, s^{(2)})$ .

Это означает, что всегда (если система существует) имеется функциональная зависимость результата сервиса 1-го уровня от необходимого набора результатов сервисов 2-го уровня  $s^{(1)} = \varphi(s^{(2)})$ . Из утверждения постулата  $(\forall s^{(1)})\mathbf{P}(f(s^{(1)}), S^{(2)})$  следует, что существует функциональная зависимость  $y = \psi(s^{(1)})$  выхода процесса от результатов каждого сервиса 1-го уровня. Отсюда вытекает функциональная зависимость выхода процесса от сервисов 2-го уровня, т. е. композиция функций  $y = \psi(\varphi(s^{(2)}))$  или  $\psi \circ \varphi: s^{(2)} \rightarrow Y$ .

Составим список пар отображений  $[h_p \circ \varphi_p: s_p^{(2)} \rightarrow V_p; g \circ \psi \circ \varphi: s_p^{(2)} \rightarrow V]$  для всех  $p = 1, \dots, J_1$ , он будет представлять собой таблицу соответствия каждого вектора на отношении  $\bar{V}_p$  элементу множества  $V$  и, таким образом, задающую межуровневую функцию качества  $\omega: \bar{V}_p \rightarrow V$ . ■

Рассматриваемая система в особых случаях может обладать свойством монотонности, когда цели провайдеров совпадают с целью координатора процесса. Согласно [5] система является монотонной,

если ее межуровневая функция качества является монотонной. Для нашего случая это означает, что целевая функция координатора процесса  $g(s^{(1)})$  монотонно связана с целевыми функциями  $h_j(s^{(2)})$  каждого провайдера сервисов 1-го уровня ( $j = 1, \dots, J_1$ ). Для такой системы, по аналогии с монотонными координируемыми системами [5] и на основании существования межуровневой функции качества, можно доказать следующую теорему.

**Теорема 2.** Для монотонной системы, построенной на основе сервис-ориентированной архитектуры, целевая функция координатора процесса достигает минимума всякий раз, когда достигают минимума целевые функции провайдеров сервисов, т. е.  $g(s^{(1)}) = \min g$ , если  $(\forall j)(h_j(s^{(2)}) = \min h_j)$ ,  $j = 1, \dots, J_1$ .

*Доказательство.* Монотонная система имеет монотонную межуровневую функцию качества  $g(s^{(1)}) = \omega(h_1(s^{(2)}), \dots, h_{J_1}(s^{(2)}))$ , поэтому уменьшение каждой из целевых функций провайдеров  $h_j(s^{(2)})$  не будет увеличивать целевую функцию координатора, а при строгой монотонности уменьшение любых целевых функций провайдеров будет вызывать уменьшение целевой функции координатора, отсюда вытекает справедливость данной теоремы. ■

Отметим, что системы, подобные рассматриваемой, не всегда могут обладать свойством монотонности. Это связано со свойством слабой связанности элементов сервисов, что, с одной стороны, придает гибкость таким системам, но с другой – возможную несогласованность целей ее участников.

Полученные в данной работе результаты могут быть распространены на системы с большим, чем два, числом уровней иерархии и для случаев, когда варианты типов сервисов используют разное число других сервисов.

## ВЫВОДЫ

Таким образом, для систем, построенных с использованием концепции SOA, возможна постановка задач оптимизации как для координатора процесса, для осуществления которого собственно система создана или образовалась, так и для провайдеров сервисов, которыми пользуются процесс и другие сервисы. Благодаря постулату сатисфакции для таких систем существует межуровневая функция качества, с помощью которой можно исследовать степень согласованности целей участников или элементов системы.

## СПИСОК ЛІТЕРАТУРЫ

1. Компас в мире сервис-ориентированной архитектуры (SOA): ценность для бизнеса, планирование и план развития предприятия / [Биберштейн Н., Боуз С., Джонс К. и др.]. – М. : КУДИЦ-ПРЕСС, 2007. – 256 с.
2. Service oriented architecture Modeling Language (SoaML) – Specification for the UML Profile and Metamodel for Services (UPMS). Revised Submission. OMG document: ad/2008-08-04. – [Электронный ресурс] – Режим доступа: <http://www.omg.org/>
3. Cox D. E. Management of the service-oriented architecture life cycle / D.E. Cox, H. Kreger // IBM Systems Journal. – 2005. – Vol. 44, No. 4. – Pp. 709–726.
4. Кудерметов Р. К. Концептуальная и формальная модели систем с сервис-ориентированной архитектурой / Р. К. Кудерметов // Науковий вісник Чернівецького університету : збірник наук. праць. – Вип. 446: Комп'ютерні системи та компоненти. – Чернівці : ЧНУ, 2009. – С. 94–99.
5. Месарович М. Теория иерархических многоуровневых систем / М. Месарович, Д. Мако, И. Такахара. – М. : Мир, 1973. – 344 с.

Надійшла 17.08.2009

Кудерметов Р. К.

ЗАДАЧА ОПТИМІЗАЦІЇ ДЛЯ ІЄРАРХІЧНИХ СИСТЕМ  
НА БАЗІ СЕРВІС-ОРІЄНТОВАНОЇ АРХІТЕКТУРИ

Сформульовано задачу оптимізації для системи, що має ієрархічну багаторівневу структуру та побудована з використанням сервіс-орієнтованої архітектури. Визначено цільові функції кожного рівня такої системи та показано, що ця система має міжрівневу функцію якості. Розглянуто властивості міжрівневої функції за умовою, що система має властивість монотонності.

**Ключові слова:** сервіс-орієнтована архітектура, ієрархічна багаторівнева система, постулат сатисфакції, міжрівнева функція якості.

Kudermetov R. K.

OPTIMIZATION PROBLEM FOR HIERARCHICAL SYSTEMS BASED ON SERVICE-ORIENTED ARCHITECTURE

The problem of optimization is formulated for a hierarchic multilevel system which uses the foundations of Service-oriented architecture. The objective functions of each of such system levels are determined and it is shown that this system has an interlevel performance function. Properties of interlevel performance function are considered provided that the system has a monotonicity property.

**Key words:** Service-oriented architecture, hierarchical multilevel system, satisfaction postulate, interlevel performance function.

УДК 621.7.073-52

Невлюдов И. Ш.<sup>1</sup>, Шостак Б. А.<sup>2</sup>, Бабай О. Н.<sup>3</sup>, Юрьева С. В.<sup>4</sup>

<sup>1</sup>Д-р техн. наук, заведующий кафедрой Харьковского национального университета радиоэлектроники

<sup>2</sup>Канд. техн. наук, ведущий инженер ООО «НПО «Агротехника»»

<sup>3</sup>Доцент Харьковского национального университета радиоэлектроники

<sup>4</sup>Студент Харьковского национального университета радиоэлектроники

## ДИАГНОСТИКА МИКРОСХЕМ ПАМЯТИ В МИКРОПРОЦЕССОРНЫХ УСТРОЙСТВАХ

Рассмотрен новый метод диагностирования работы микросхем памяти, который позволяет проводить диагностику в процессе эксплуатации запоминающих устройств. Приведено описание принципа функционирования аппаратной части, а также описана структура программного обеспечения, позволяющего реализовать разработанный метод диагностики микросхем памяти.

**Ключевые слова:** запоминающее устройство, диагностирование, оперативное запоминающее устройство, постоянное запоминающее устройство, эмуляция, информационный сигнал, стробирующий сигнал, данные.

### ВВЕДЕНИЕ

В современных условиях производства и эксплуатации диагностика и контроль запоминающих устройств (ЗУ) является чрезвычайно актуальной задачей. И хотя в настоящее время существует большое количество методов диагностики и соответствующих программно-аппаратных комплексов, методы, позволяющие диагностировать ЗУ в процессе эксплуатации, т. е. не выпаивая микросхему из платы, недостаточно развиты. На сегодняшний момент су-

ществует несколько способов диагностирования микросхем памяти. Наиболее распространенный – тестовое диагностирование. При таком способе вначале формируется диагностическая модель объекта диагностики, генерируется набор тестовых программ и затем производится непосредственно процесс диагностирования.

Нередко применяется и еще один из способов диагностирования, основанный на измерении температуры микросхем. При таком способе с помощью

© Невлюдов И. Ш., Шостак Б. А., Бабай О. Н., Юрьева С. В., 2010