

УДК 004.03

**Р.К. Кудерметов** (канд. техн. наук, доц.), **М.Ю. Юрич** (асп.)

Запорожский национальный технический университет

кафедра компьютерных систем и сетей

[krk@zntu.edu.ua](mailto:krk@zntu.edu.ua), [mary@yurich.org.ua](mailto:mary@yurich.org.ua)

## **ОПТИМИЗАЦИЯ В СИСТЕМАХ С СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРОЙ НА ОСНОВЕ СВОЙСТВА МОНОТОННОСТИ**

Обоснована возможность декомпозиции задачи оптимизации для иерархических многоуровневых систем, построенных на основе SOA и обладающих свойством монотонности. Предложено для установления свойства монотонности системы применять моделирование системы с помощью нейронных сетей.

**сервис-ориентированная архитектура, монотонность, оптимизация, межуровневая функция качества, нейронные сети**

### ***Введение***

В условиях современной конкурентной борьбы предприятия должны быстро реагировать на изменения внешнего окружения, требований к производимой продукции и предоставляемым сервисам, оптимизировать организацию бизнес-процессов. Характерно, что современная организация бизнеса часто является географически распределенной. Все это приводит к тому, что руководителям организаций, участвующих в распределенных кооперациях, необходимо по-новому организовывать свои ИТ-инфраструктуры, прежде всего, для интеграции распределенных бизнес-процессов [1]. Причем, здесь речь идет не только о технической совместимости, а, главным образом, об интеграции постоянно изменяющихся под воздействием требований бизнес-приложений, их функциональности, размещения в хостинг-средах, степени безопасности, изменений списков партнеров (других бизнес-приложений).

Одним из решений для поддержания ИТ-инфраструктуры организаций в соответствии с высокой динамикой внутренних и внешних изменений на сегодняшний день считается использование сервис-ориентированной архитектуры (Service-Oriented Architecture, SOA). Одной из особенностей архитектуры SOA является возможность создания новых, комплексных, сложных сервисов из более простых сервисов, т.е. сервисы могут синтезироваться, модифицироваться, быть распределенными во времени и пространстве и принадлежать разным организациям [2].

## Постановка задачи. Общая структура

Рассматривая возможность образования комплексных сервисов, совокупность сервисов можно представить как иерархическую структуру. Рассмотрим пример двухуровневой организации сервисов для обеспечения некоторого процесса, представленной на рис. 1. Каждый сервис может оказаться необходимым для другого сервиса, причем для образования комплексного сервиса может потребоваться несколько типов сервисов и, кроме того, каждый провайдер нового сервиса может выбрать наиболее подходящий для него вариант типа сервиса. На рис. 1 введены следующие обозначения.

Уровни в иерархии сервисов обозначены с помощью верхнего индекса, первый нижний индекс обозначает номер сервиса в рамках уровня, т.е. *тип* сервиса на данном уровне, а второй нижний индекс - *вариант* сервиса данного типа. Таким образом, сервис обозначается 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  $S_{jk}^{(2)} = \{S_{jk1}^{(2)}, \dots, S_{jkK_j}^{(2)}\}$ .

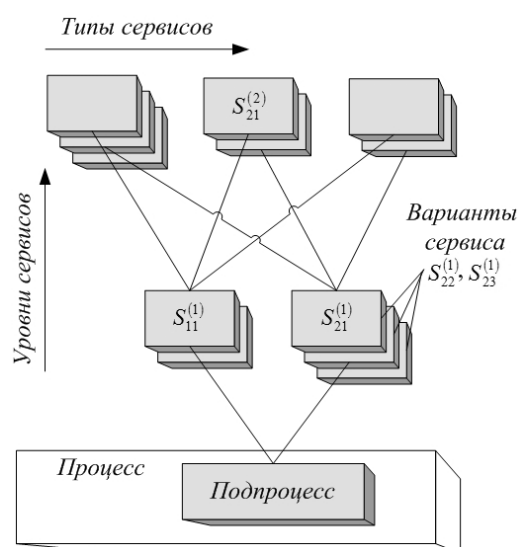


Рисунок 1 - Иерархическая организация системы с использованием SOA

Отметим, что в случае более сложного взаимодействия сервисов, схему системы все равно можно представить в виде иерархической структуры, подобной рис. 1. Действительно, если, к примеру, два сервиса обеспечивают друг друга, то всегда можно выполнить их декомпозицию на более простые сервисы, т.к. непосредственно связанные сервисы должны находиться в причинно-следственных отношениях.

Для состоятельности организации сервисов с целью обеспечения некоторого процесса должен соблюдаться *постулат самисфакции*, т.е. сервисы нижних уровней должны удовлетворять процесс и, в свою очередь, удовлетворяться сервисами следующего, верхнего в иерархии уровня.

Возможность выбора вариантов типов сервисов на каждом уровне иерархии сервисов порождает задачу оптимизации (например, минимизации затрат), которая обеспечивает оптимальность организации целевого процесса создания продукта или комплексной услуги.

В данной работе рассматриваются особенности решения задачи оптимизации, зависящие от описанной выше структуры организации, основанной на использовании SOA.

### **Использование свойства монотонности для декомпозиции задачи оптимизации**

Предположим, что *все* сервисы 1-го уровня используют результаты *всех* сервисов 2-го уровня. Это предположение, не теряя общности, упрощает индексацию для дальнейшего изложения.

Целевую функцию координатора процесса можно записать следующим образом

$$g = g(S^{(1)}), \quad (1)$$

где  $S^{(1)} = S_1^{(1)} \times \dots \times S_J^{(1)}$ ,  $J$  - индексы типов сервисов 1-го уровня. Оптимальным решением задачи координатора процесса будет такой выбор  $J$ -мерного вектора  $\hat{S}^{(1)}$  среди  $S^{(1)}$ , что

$$g(\hat{S}^{(1)}) = \min g(S^{(1)}). \quad (2)$$

Целевыми функциями провайдеров сервисов 1-го уровня будут

$$h_j = h_j(S^{(2)}), \quad (3)$$

где  $S^{(2)} = S_1^{(2)} \times \dots \times S_I^{(2)}$ ,  $I$  - индексы типов сервисов 2-го уровня. Оптимальным решением для провайдера  $j$ -го сервиса 1-го уровня будет такой  $I$ -мерный вектор  $\hat{S}^{(2)}$  среди  $S^{(2)}$ , что

$$h_j(\hat{S}^{(2)}) = \min h_j(S^{(2)}). \quad (4)$$

В [3] доказано, что при условии выполнения постулата сатисфакции для системы, построенной на основе SOA, существует межуровневая функция качества

$$g(S^{(1)}) = \omega(h_1(S^{(2)}), \dots, h_J(S^{(2)})). \quad (5)$$

В случае, если эта функция обладает свойством монотонности, система является монотонной. Это означает, что уменьшение затрат провайдеров сервисов не увеличивает затрат координатора, а при строгой монотонности вызывает уменьшение этих затрат и, таким образом, целевая функция координатора процесса будет достигать минимума, если достигают минимума функции провайдеров сервисов. Свойство монотонности системы значительно упрощает поиск оптимальных решений для целевых функций координатора процесса и провайдеров сервисов и сводится к независимому решению задач оптимизации (2) и (4).

### **Пример поиска оптимальных решений**

Предположим, что система построена с использованием SOA, имеет структуру, показанную на рис. 2, и включает следующие типы сервисов (индексы обозначают варианты сервисов):

-  $M = \{M_1, M_2, M_3\}$  - сервисы, предоставляющие математические пакеты (например, Matlab, Matematica, Statistica и т.п.) для решения задач пользователей;

-  $I = \{I_1, I_2, I_3, I_4, I_5\}$  - сервисы Internet-провайдеров, обеспечивающих доставку заданий от процесса к сервисам  $M$  ;

-  $S = \{S_1, S_2, S_3\}$  - сервисы долговременного хранения результатов, полученных с помощью сервиса  $M$  ;

-  $H = \{H_1, H_2, H_3, H_4\}$  - хостинг, предоставляемый некоторыми организациями для сервиса  $M$  ;

-  $T = \{T_1, T_2, T_3\}$  - средства телекоммуникации для реализации сервиса  $I$  .

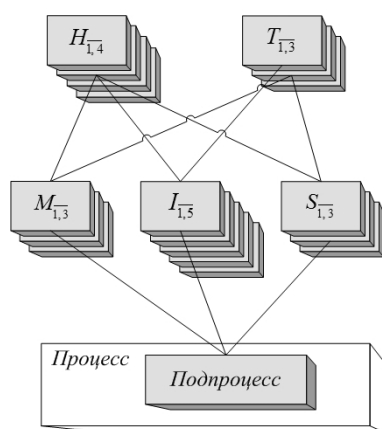


Рисунок 2 – Структура примера системы с использованием SOA

Стоимость предоставляемых сервисов может зависеть от разных факторов, например, от версий программного обеспечения и качества оборудования, от объема пересылаемых и хранимых данных, скорости пересылок и т.п. Однако, провайдеры сервисов нижних уровней не могут учитывать образование цены сервисов верхнего уровня – они просто могут этого не знать. Это относится и к процессу, использующему сервисы, относительно сервисов, уровень которых выше сервисов первого уровня в иерархии. По этой причине некоторые характеристики сервисов также не могут быть учтены, такие, например, как скорость и объем пересылок данных, а также время, затрачиваемое на реализацию услуг сервисами верхних уровней.

Пусть цены потребления вариантов названных сервисов в некоторых денежных единицах образуют следующую таблицу 1, где  $V_i$  - вариант:

Таблица 1

	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$R$
$M$	30	40	50	-	-	30
$I$	50	55	60	65	55	50
$S$	20	15	20	-	-	15
$H$	10	7	8	15	-	7
$T$	4	5	8	-	-	4

Целевой функцией координатора процесса и ее решением, очевидно, будет

$$g = \min_{i=1,3} M + \min_{i=1,5} I + \min_{i=1,3} S = 30 + 50 + 15 = 95. \quad (6)$$

В свою очередь, целевые функции провайдеров сервисов первого уровня и их решения:

$$h_M = \min_{i=1,4} H + \min_{i=1,3} T = 7 + 4 = 11; \quad h_I = \min_{i=1,4} H + \min_{i=1,3} T = 7 + 4 = 11; \quad h_S = \min_{i=1,4} H + \min_{i=1,3} T = 7 + 4 = 11. \quad (7)$$

### **Моделирование системы помощью нейронных сетей**

Для анализа свойств более сложных структур систем, построенных на основе SOA, целесообразно применить подходящие методы моделирования. На наш взгляд, здесь уместно воспользоваться моделями в виде нейронных сетей, поскольку мы ни о каких свойствах системы, кроме монотонности, не предполагаем, но имеем некоторые сведения о поведении системы в виде табл.1, (6) и (7), которые можно использовать как обучающее множество [4]. Другим фактором в пользу использования моделей нейронных сетей является то, что некоторые из них обладают свойством монотонности.

Если нейронная сеть монотонна, то этого оказывается достаточно для получения практических результатов, без проведения более детального

изучения свойств моделируемой системы [5, 6]. Свойством монотонности обладают многослойные (слоистые) нейронные сети, удовлетворяющие следующим дополнительным требованиям к элементам и связям. Каждый слой, кроме выходного, разбит на два блока – возбуждающий ( $A$ ) и тормозящий ( $B$ ). Связи между слоями также подразделяются на два типа – возбуждающие (с положительными весами) и тормозящие (с отрицательными весами). Если от блока  $A$  к блоку  $C$  ведут только возбуждающие связи, то любой выходной сигнал блока  $C$  является монотонной неубывающей функцией от любого выходного сигнала блока  $A$ . Если же эти связи только тормозящие ( $B$ ), то любой выходной сигнал блока  $C$  является монотонной невозрастающей функцией [7].

Учитывая, что мы предположили только о монотонности рассматриваемой системы, воспользуемся возможностями программного пакета Statistica Neural Network 6.1 [8], который позволяет автоматически синтезировать множество различных типов нейронных сетей на основании обучающего множества. Это позволит нам оценить, какая из архитектур нейронных сетей наиболее соответствует архитектуре нашей системы и сформулированной задаче оптимизации.

Для моделирования описанной в примере системы используем данные из столбца  $R$  табл. 1, являющиеся минимальными значениями данных в соответствующих строках, как обучающее множество нейронных сетей. В результате моделирования были синтезированы архитектуры нейронных сетей, приведенные в табл. 2. Ниже приведено их краткое описание [8].

Таблица 2 – Синтезированные архитектуры нейронных сетей

№	Модель сети	№	Модель сети	№	Модель сети	№	Модель сети
1	МП 3:3-10-8-1:1	9	МП 5:5-9-7-1:1	17	МП 5:5-6-1:1	25	МП 2:2-7-1:1
2	Линейная 2:2-1:1	10	ОРНС 5:5-3-2-1:1	18	МП 5:5-10-7-1:1	26	МП 1:1-1-1:1
3	Линейная 3:3-1:1	11	МП 2:2-3-5-1:1	19	МП 5:5-10-8-1:1	27	МП 3:3-4-1:1
4	Линейная 4:4-1:1	12	МП 2:2-8-1:1	20	Линейная 2:2-1:1	28	ОРНС 5:5-3-2-1:1
5	РБФ 5:5-1-1:1	13	ОРНС 5:5-3-2-1:1	21	Линейная 3:3-1:1	29	МП 5:5-5-1:1
6	Линейная 5:5-1:1	14	МП 5:5-9-1:1	22	Линейная 4:4-1:1	30	ОРНС 5:5-3-2-1:1
7	МП 1:1-2-1:1	15	ОРНС 5:5-3-2-1:1	23	РБФ 5:5-1-1:1	31	МП 5:5-10-3-1:1
8	МП 2:2-2-3-1:1	16	ОРНС 5:5-3-2-1:1	24	Линейная 5:5-1:1	32	МП 2:2-6-4-1:1

ОРНС – обобщенно-регрессионная нейронная сеть предназначена для решения задач регрессии. В точку расположения каждого обучающего наблюдения помещается гауссова ядерная функция. Мы считаем, что каждое наблюдение свидетельствует о некоторой нашей уверенности в том, что поверхность отклика в данной точке имеет определенную высоту, и эта уверенность убывает при отходе в сторону от этой точки. ОРНС-сеть копирует внутри себя все обучающие наблюдения и использует их для оценки отклика в произвольной точке. Окончательная выходная оценка

сети получается как взвешенное среднее выходов по всем обучающим наблюдениям, где величины весов отражают расстояние от этих наблюдений до той точки, в которой производится оценивание (и, таким образом, более близкие точки вносят больший вклад в оценку).

МП – многослойный персептрон. Каждый элемент сети строит взвешенную сумму своих входов с поправкой в виде слагаемого и затем пропускает эту величину активации через передаточную функцию, и, таким образом, получается выходное значение этого элемента. Элементы организованы в послойную топологию с прямой передачей сигнала. Такую сеть легко можно интерпретировать как модель "вход-выход", в которой веса и пороговые значения (смещения) являются свободными параметрами модели. Такая сеть может моделировать функцию практически любой степени сложности, причем число слоев и число элементов в каждом слое определяют сложность функции.

РБФ – сеть типа радиальной базисной функции имеет промежуточный слой из радиальных элементов, каждый из которых воспроизводит гауссову поверхность отклика. Поскольку эти функции нелинейны, для моделирования произвольной функции нет необходимости брать более одного промежуточного слоя. Для моделирования любой функции необходимо лишь взять достаточное число радиальных элементов. Остается решить вопрос о том, как следует скомбинировать выходы скрытых радиальных элементов, чтобы получить из них выход сети. Оказывается, что достаточно взять их линейную комбинацию (т.е. взвешенную сумму гауссовых функций). Сеть РБФ имеет выходной слой, состоящий из элементов с линейными функциями активации.

Значения результатов, предсказанные перечисленными в табл. 2 нейронными сетями, представлены в табл. 3. Здесь  $R$  – множество значений, к которым должны стремиться значения соответствующих элементов результирующих множеств  $R_i$ , полученных  $i$ -ми моделями сетей из табл. 2.

Как видно из представленных результатов, наиболее близкий результат для каждого из сервисов показали сети №№ 14, 17, 31 с архитектурой МП – многослойный персептрон (см. табл. 2, 3), относящийся к числу монотонных слоистых сетей. На рис. 3-5 показаны архитектуры этих сетей.

Тот факт, что сети именно данного класса моделей показали результаты, наиболее близкие к оптимальным значениям, подтверждает предположение о монотонности нашей системы. Как было сказано выше, в монотонных сетях обязательно присутствуют возбуждающие и тормозящие нейроны. На рис. 3-5 наиболее тормозящий нейрон (т.к. ищем

минимум) выделен черным цветом, в данном случае для цифры "4" – первого элемента из последней строки табл. 1.

Таблица 3 - Предсказанные нейронными сетями значения

№	$R$	$R_1$	$R_2$	$R_3$	$R_4$	$R_5$	$R_6$	$R_7$	$R_8$
1	30,00000	31,3339	25,09237	27,18597	32,30084	10,31029	33,10444	27,02476	43,24235
2	50,00000	50,0327	50,00000	50,00000	50,00000	46,83857	50,00000	49,73761	50,61431
3	15,00000	14,9762	15,00000	15,00000	15,00000	6,95498	15,00000	16,03828	14,64551
4	7,00000	7,0542	7,00000	7,00000	7,00000	18,20645	7,00000	6,20246	8,11514
5	4,00000	-14,9645	-8,78390	-6,25595	-2,85734	60,44108	-0,62537	1,01880	7,37450
№	$R$	$R_9$	$R_{10}$	$R_{11}$	$R_{12}$	$R_{13}$	$R_{14}$	$R_{15}$	$R_{16}$
1	30,00000	42,80782	24,00000	36,35722	33,93823	49,95610	39,25776	40,82925	31,84455
2	50,00000	50,65636	50,00000	49,81090	49,96201	50,00000	50,23906	49,94512	47,32151
3	15,00000	14,51706	15,00000	14,75941	15,02005	14,94610	14,49533	13,25202	13,28143
4	7,00000	7,75872	7,00000	7,05365	6,94576	7,05390	7,54546	8,79204	10,54976
5	4,00000	6,27106	24,00000	5,52947	5,33697	14,94076	4,96153	13,18296	12,14318
№	$R$	$R_{17}$	$R_{18}$	$R_{19}$	$R_{20}$	$R_{21}$	$R_{22}$	$R_{23}$	$R_{24}$
1	30,00000	36,77939	52,8830	21,9395	25,09237	27,18597	32,30084	10,31029	33,10444
2	50,00000	50,71081	73,7160	45,5857	50,00000	50,00000	50,00000	46,83857	50,00000
3	15,00000	14,54553	34,0009	16,2005	15,00000	15,00000	15,00000	6,95498	15,00000
4	7,00000	7,07814	27,0616	9,9217	7,00000	7,00000	7,00000	18,20645	7,00000
5	4,00000	4,64410	-19,1828	-17,4350	-8,78390	-6,25595	-2,85734	60,44108	-0,62537
№	$R$	$R_{25}$	$R_{26}$	$R_{27}$	$R_{28}$	$R_{29}$	$R_{29}$	$R_{31}$	$R_{32}$
1	30,00000	43,10790	43,31206	20,18110	24,00000	16,01726	49,95610	32,92205	43,10270
2	50,00000	49,87410	50,51582	32,58064	50,00000	50,07342	50,00000	49,99515	50,10293
3	15,00000	15,23235	16,23436	14,05664	15,00000	13,07583	14,94610	15,12774	15,22169
4	7,00000	7,08892	5,87395	11,10165	7,00000	8,71811	7,05390	6,89438	6,89540
5	4,00000	1,57770	5,87395	5,82224	24,00000	5,27637	14,94076	3,19552	4,37322

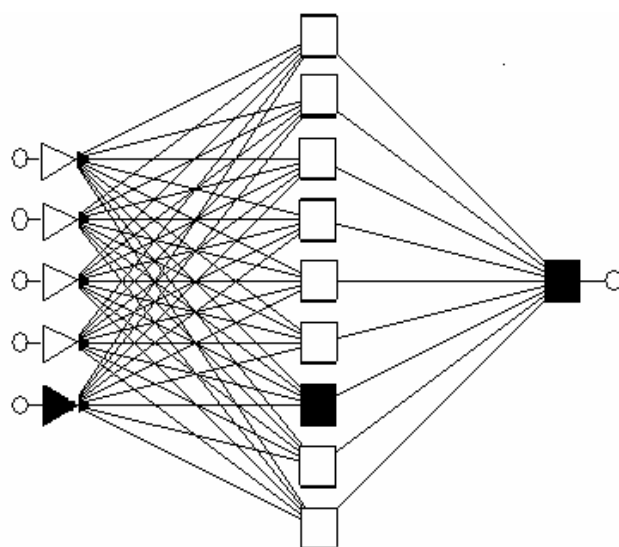


Рисунок 3 - Архитектура сети МП 5:5-9-1:1



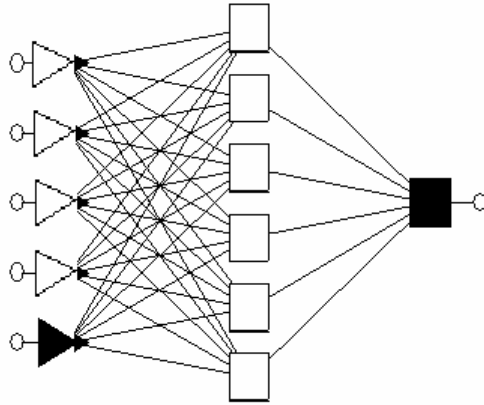


Рисунок 4 - Архитектура сети МП 5:5-6-1:1

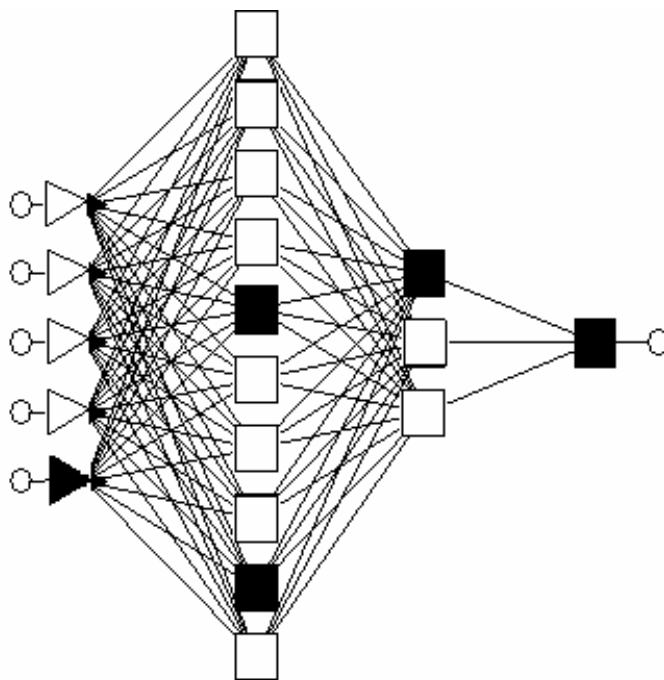


Рисунок 5 - Архитектура сети МП 5:5-10-3-1:1

### **Выводы**

Таким образом, на основании того, что система с многоуровневой иерархической структурой, построенной на основе SOA, имеет межуровневую функцию качества, показано, что в случае монотонности этой функции значительно упрощается задача оптимизации затрат конечного потребителя сервисов и провайдеров сервисов, использующих другие сервисы. Эта задача сводится к поиску локальных оптимумов для каждого провайдера сервисов и конечного потребителя. Монотонность системы можно установить путем моделирования системы с помощью нейронных сетей типа многослойный персептрон.

## Список литературы

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. Cox D.E. Management of the service-oriented architecture life cycle / D.E. Cox, H. Kreger // IBM Systems Journal, Vol 44, No 4, 2005. - pp. 709-726.
3. Кудерметов Р.К. Задача оптимизации для иерархических систем на основе сервис-ориентированной архитектуры. // Радиоелектроніка. Інформатика. Управління. – №1, 2010.
4. Луканин А.В. Нейронные сети как механизм представления лексико-семантической информации - [Электронный ресурс] // Режим доступа: <http://www.artiom.ice-lc.com/science/lexsem2004.doc>
5. Некипелов Н. Введение в RBF сети - [Электронный ресурс] // Режим доступа: <http://www.basegroup.ru/neural/rbf.htm>.
6. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. Новосибирск: Издательство Института математики, 1999. – 270 с.
7. [Тарков](#) М.С. Виды нейронных сетей и способы организации их функционирования - [Электронный ресурс] // Режим доступа: <http://www.intuit.ru/department/expert/neuro/5/>
8. Statistica. Информация с официального сайта компании StatSoft [Электронный ресурс] // Режим доступа: <http://www.statsoft.ru/>

*Надійшла до редакції 04.11.2009р. Рецензент: к. т.н., доц. Зеленьова І.Я.*

**Р.К. Кудерметов, М.Ю. Юрич**

Запорізький національний технічний університет

**Оптимізація в системах з сервіс-орієнтованою архітектурою на базі властивості монотонності.** Обґрунтована можливість декомпозиції задачі оптимізації для ієрархічних багаторівневих систем, які побудовані на базі SOA та мають властивість монотонності. Запропоновано для установлення властивості монотонності системи застосовувати моделювання системи за допомогою нейронних мереж.

**сервіс-орієнтована архітектура, монотонність, оптимізація, межрівнева функція якості, нейронні мережі**

**R.K. Kudermetov, M.Yu. Yurich.**

Zaporizhzhya National Technical University

**An optimization in systems with service-oriented architecture on the basis of monotonicity property.** The capability of decomposition of an optimization problem for hierarchic multilevel systems which based on SOA and have the monotonicity property is substantiated. For clarification of the monotonicity property of the system to apply its simulation by neuron networks is offered.

**service-oriented architecture, monotonicity, optimization, interlevel performance function, neuron networks**