

УДК 004.02

Дата подачи статьи: 06.09.14

DOI: 10.15827/0236-235X.109.017-020

СПЕЦИФИКАЦИЯ ЗАДАЧ В САМООРГАНИЗУЮЩЕЙСЯ ИНФОРМАЦИОННОЙ СИСТЕМЕ

В.В. Дрождин, к.т.н., доцент, drozhdin@yandex.ru;

А.А. Шалаев, аспирант, shell@live.ru

(Пензенский государственный университет, ул. Красная, 40, г. Пенза, 440026, Россия)

Для обеспечения самомодификации и совершенствования самоорганизующаяся информационная система должна выявлять различные внутрисистемные проблемы, для устранения которых необходимо формулировать и решать задачи. В данной статье приведена общая спецификация задачи и определены ее компоненты. Для метода решения предложено использовать семантическое и конструктивное описания. Определены правила соответствия метода решаемой задаче и рассмотрены способы формирования метода, удовлетворяющего ей.

Отмечено преимущество использования в самоорганизующейся информационной системе недоопределенных задач и предложены способы доопределения задачи в случаях неполного покрытия области определения задачи областью определения метода, неполного удовлетворения ограничений метода решения условиям задачи, отклонения от эквивалентности преобразования исходных данных в результат в методе решения и способе преобразования данных, инициированного проблемой.

Семантическую спецификацию задачи предложено представлять в виде понятия-задачи и отношения между понятиями. Понятие-задача содержит идентификатор, состав, содержание, метод решения, внешнее описание и макросвойства. Между понятиями-задачами могут устанавливаться отношения агрегации, классификации, обобщения и абстрагирования. Семантические спецификации задач, известных и решаемых в системе, могут быть использованы для формирования слоя задач в семантическом пространстве «проблема–задача–подзадача–метод решения–знания».

Таким образом, формулирование и решение задач дают возможность системе устранять возникающие проблемы в процессе ее функционирования, а использование недоопределенных задач существенно расширяет возможности системы по устранению проблем и позволяет сделать этот процесс эффективным.

Ключевые слова: самоорганизующаяся информационная система, проблема, задача, недоопределенная задача, модель задачи, модификация задачи.

Принципиальным отличием самоорганизующихся информационных систем (СИС) от автоматизированных информационных систем (АИС) является их способность к самомодификации (самодостраиванию) и совершенствованию [1]. Это означает, что СИС самостоятельно должна формировать и поддерживать в актуальном состоянии свои структуру и функциональность, сохраняя целостность, расширяя свои возможности и повышая эффективность функционирования. Для этого СИС должна выявлять проблемы, формулировать задачи, разрабатывать и апробировать методы решения задач, строить исполняемые программные модули и использовать их в работе.

Исходным для самомодификации и совершенствования СИС является выявление различных внутренних проблем. Проблема или проблемная ситуация – это внутрисистемная ситуация несоответствия, рассогласования или отклонения от нормального функционирования, отрицательно влияющая на эффективность и качество функционирования системы, для разрешения которой у системы нет конкретного решения [2].

Это заставляет СИС запустить процесс формулирования и решения задачи, в результате чего должен быть сформирован новый или модифицирован существующий исполняемый модуль, обеспечивающий дальнейшее функционирование СИС с устраненной проблемой.

Задача – это точная спецификация решения некоторой проблемы [3]. Задача включает в себя требования (цель), условия (известное) и искомое (неизвестное). Между этими элементами существуют оп-

ределенные связи и зависимости, за счет которых осуществляются поиск и определение неизвестных элементов через известные.

В общем виде задача представляется следующим образом: $z = \langle id_z, S_u, S_k, c, \mu \rangle$, где id_z – идентификатор (имя) задачи, выделяющий ее среди множества других задач; $S_u = \{s_u\}$ – область определения задачи, представляющая собой множество допустимых исходных состояний (совокупностей элементов данных) решаемой задачи; $S_k = \{s_k\}$ – область значений задачи, представляющая собой множество допустимых результирующих состояний (совокупностей элементов данных) решаемой задачи; c – ограничения, накладываемые средой на решение задачи, частью которых является соответствие q_z между исходными и результирующими состояниями, инициируемое проблемой и заданное в форме отношения $q_z = \{\{s_u, s_k\}\}$ (возможно, частичного, например, только тестовые данные) или функции $q_z: s_u \rightarrow s_k$; $\mu: s'_u \xrightarrow{f} s'_k$ – метод решения, представляющий собой определенный способ f преобразования $s'_u \in S'_u$ в $s'_k \in S'_k$; S'_u, S'_k – соответственно области определения и значений метода μ .

Ожидаемый результат ($s_k \in S_k$) – это цель решения задачи, выступающая в качестве внешнего дополнения для оценки правильности (корректности) и эффективности решения поставленной задачи. Корректность решения задачи заключается в установлении такого соответствия между s_u и s_k , чтобы $\langle s_u, s_k \rangle \in q_z$.

Метод решения задачи μ – это описание последовательности действий, определяющей процесс реше-

ния некоторой абстрактной задачи путем преобразования исходного состояния s'_u в результирующее состояние s'_k . Метод может задаваться двумя способами: семантически и конструктивно.

При семантическом описании метод представляется семантикой (содержанием) укрупненных действий, которые необходимо выполнить над исходными данными для получения требуемого результата. Семантическое описание определяет логику решения абстрактной задачи и специфицирует укрупненный процесс ее решения. Поэтому семантическое описание метода можно рассматривать как согласованную последовательность подзадач, которые необходимо решить для решения абстрактной задачи, или как формальное доказательство ее разрешимости, а для точной спецификации процесса решения абстрактной задачи предварительно необходимо разработать конструктивное описание метода решения задачи, соответствующее семантическому описанию и удовлетворяющее внутренним и внешним ограничениям.

При конструктивном описании метод задается последовательностью действий (алгоритмом) определенного уровня абстракции, позволяющей путем последовательных преобразований исходных данных получить требуемый результат. Конструктивный метод формируется путем реализации семантически заданных действий в последовательности исполняемых действий. Необходимость реализации определяется наличием внешних и внутренних условий, накладываемых на процесс решения абстрактной задачи, доступными ресурсами, необходимой эффективностью и скоростью решения абстрактной задачи, а также неоднозначностью перехода от семантически заданных действий к последовательностям исполняемых действий вследствие использования различных исполняемых операторов и последовательностей их выполнения, формирования различных структур данных для организации, хранения и обработки и др.

Кроме семантического и конструктивного описаний, метод может иметь ряд промежуточных семантических и/или конструктивных описаний, последовательно детализирующих наиболее общее семантическое описание в наиболее точное детальное описание метода.

Конструктивное описание метода более громоздко, но требует меньшей квалификации исполнителя при решении задачи, а использование только семантического описания метода приводит к увеличению времени решения задачи вследствие необходимости реализации семантически заданных действий непосредственно в процессе решения задачи.

Для обеспечения возможности решения задачи z методом μ требуется их согласованность, заключающаяся в эквивалентности законов преобразования и в соответствии областей определений и значений. Поэтому при $q_z \equiv f$, соответствии ограничений c и ограничений метода и при $S_u \subseteq S_u^n$ и $S_k \subseteq S_k^n$ имеем полностью разрешимую задачу, а при $S_u \cap S_u^n \neq \emptyset$, $S_u \setminus S_u^n \neq \emptyset$ и $S_k \cap S_k^n \neq \emptyset$, $S_k \setminus S_k^n \neq \emptyset$ – частично разрешимую.

Решение задачи осуществляется в некоторой среде, которая, с одной стороны, выступает в качестве

ограничений c , а с другой – может использоваться для доопределения задачи, что позволяет перейти от полностью определенной задачи к недоопределенной (контекстно определяемой), требующей интеллектуального подхода к решению [4, 5]. Способность решения недоопределенных задач существенно расширяет возможности системы, но вызывает необходимость разработки специальных формальных методов [6, 7]. В области информационных систем и баз данных недоопределенные вычисления использовались для многомерного анализа данных [8], а для решения математических задач на основе недоопределенных моделей разработаны универсальный решатель математических задач UniCalc [9] и объектно-ориентированная система программирования в ограничениях NeMo+ [10].

В СИС в недоопределенной задаче могут отсутствовать метод решения, требования к скорости и эффективности решения задачи, типы и объемы необходимых ресурсов и др. Это требует доопределения задачи перед ее решением, но дает системе дополнительные возможности для более качественного и эффективного решения конкретной задачи в конкретных условиях. Использование недоопределенных задач позволяет повышать эффективность и скорость получения результата для часто решаемых задач, получать результат за приемлемое время при наличии имеющихся ресурсов для задач, критичных ко времени исполнения и к объемам ресурсов, а также формировать и поддерживать структуру и функциональность системы, ориентированные на решение задач, полезных для внешней среды (например, запросы пользователей в информационной системе).

Приобретение способности решения недоопределенных задач требует наличия в системе механизмов доопределения задачи в случаях неполного покрытия области определения задачи областью определения метода, неполного удовлетворения ограничений метода условиям задачи, незначительного отклонения от эквивалентности соответствия q_z и способа преобразования f .

Несоответствие областей определений и значений задачи и метода решения может устраняться двумя способами:

- выбором метода, осуществляющего решение более общей абстрактной задачи, для которой решаемая задача будет являться частным случаем (при этом могут возникнуть проблемы с эффективностью и скоростью решения задачи);

- формированием сложного (составного) метода решения абстрактной задачи, покрывающего S_u путем объединения S_{u_i} и вызывающего метод μ_j решения частной задачи для исходных данных $s_u \in S_{u_j}$.

Несоответствие условий задачи и ограничений метода может проявляться во внешних требованиях и внутренней организации системы. К несоответствию внешних требований относятся недостаточный объем имеющихся ресурсов и низкая скорость решения задачи, отсутствие модулей для решения подзадач или их некорректное функционирование, а внутренних – увеличение объема системы, снижающее

скорость решения задач, организация данных, не допускающая их эффективную обработку, и другие.

Устранение несоответствий условий задачи и ограничений метода может осуществляться несколькими способами:

- выделить часто решаемую частную задачу (с высокой частотой входных данных) и использовать для нее наиболее эффективный метод решения;
- в ситуациях, критичных по времени или по памяти, использовать методы решения, эффективные в соответствующей ситуации;
- в случае некорректного решения задачи использовать другой корректный метод или модифицировать метод, выявив некорректно решаемые подзадачи и заменив их на корректно решаемые;
- постепенно формировать структуры данных, допускающие более эффективную обработку данных, и заменять ими структуры данных, используемые в системе.

Устранение отклонений от эквивалентности соответствия q_z и способа преобразования f может производиться двумя способами:

- выбором метода, осуществляющего решение более общей абстрактной задачи, в которой отклонение от q_z устраняется автоматически;
- модификацией используемого метода так, чтобы обеспечивалась полная эквивалентность соответствия q_z и способа преобразования f .

Указанные способы решения недоопределенных задач возможны при достаточно богатом наборе подзадач и методов их решения в различных условиях.

Для преобразования конструктивного описания метода решения задачи в исполняемый код можно использовать конструктивно-универсальный автомат [11]. При этом конструктивное описание метода будет являться шаблоном автомата, решающего требуемую подзадачу.

Семантическую спецификацию задачи целесообразно представить в виде понятия-задачи [12]: $z = \langle id_z, z_s, z_c, z_v, z_w, z_a \rangle$, где id_z – идентификатор (имя) задачи; $z_s = \{z_i\}$ – состав задачи, являющейся совокупностью подзадач, подлежащих решению для решения задачи z ; $z_c = (S_u, S_r, c)$ – содержание задачи z , включающее допустимые исходные данные и ожидаемый результат, а также ограничения c , содержащие способ соответствия q_z между исходными и результирующими состояниями; $z_v = \{\mu_j\}$ – множество возможных методов решения задачи z ; z_w – описание задачи z для пользователей (назначение задачи, изложение сущности ее решения, пример решения задачи и др.); z_a – макросвойства задачи, включающие временную и емкостную сложность и др.

Между понятиями-задачами могут устанавливаться отношения вида $r_z = \langle n_{r_z}, z_1, z_2, t_{r_z} \rangle$, где n_{r_z} – имя отношения; t_{r_z} – тип отношения между понятиями z_1 и z_2 .

В соответствии с [12] между понятиями-задачами допустимы следующие типы отношений t_{r_z} : агрегация (часть–целое) – задача z_2 является подзадачей (частью) задачи z_1 ; классификация – задача z_2 является подклассом класса задач z_1 ; обобщение (род–вид) – задача z_2 является категорией (модификацией, ча-

стным случаем) задачи z_1 ; абстрагирование – задача z_2 является конкретизацией обобщенной задачи z_1 .

Агрегация позволяет представлять задачу в виде последовательности подзадач. Например, расчет заработной платы для сотрудников подразделения является последовательностью решения подзадач: расчет начислений, расчет отчислений и расчет суммы к выдаче для каждого сотрудника.

Классификация определяет разбиение множества задач на подмножества в соответствии с некоторым признаком (основанием классификации), например задачи поиска, модификации, добавления и удаления данных.

Обобщение дает возможность определять для типовой задачи ряд частных задач, применимых в различных конкретных ситуациях. Например, для типовой задачи расчета заработной платы разработать частные задачи, учитывающие достижения и недостатки в работе сотрудников, пребывание в командировке, отпуске или на лечении.

Абстрагирование позволяет формировать различные образы задач (абстрактные задачи) путем огрубления конкретных задач или формировать конкретные задачи путем конкретизации абстрактных задач. Огрубление может осуществляться путем отбрасывания несущественных свойств и компонентов задачи, формированием обобщенной (типовой) задачи для множества конкретных задач и др. Наибольшей степенью абстракции будет обладать образ задачи, включающий ее назначение и макросвойства. К образам задач следует отнести недоопределенные задачи, а также абстрактные, решаемые формальными методами.

Семантические спецификации задач, известных и решаемых в СИС, могут быть использованы для формирования слоя задач в семантическом пространстве «проблема–задача–подзадача–метод решения–знания».

Использование недоопределенных задач позволяет СИС обрести следующие новые возможности: более точно согласовывать процесс решения задач с предсказуемо (периодически и причинно-следственно) возникающими ситуациями; получить более широкие возможности для решения задач в критических ситуациях; обеспечивать повышение эффективности и качества функционирования системы путем замены устаревших (неэффективных) реализаций семантически заданных действий и целых методов более эффективными действиями и методами.

Литература

1. Дрождин В.В. Организация и функционирование самоорганизующейся информационной системы // Новые информационные технологии и системы: тр. X Междунар. науч.-технич. конф. Пенза: Изд-во ПГУ, 2012. С. 231–233.
2. Что такое Проблема? Значение и толкование слова problema, определение термина. URL: <http://www.onlinedics.ru/slovar/log/t/problema.html> (дата обращения: 15.03.2014).
3. Что такое Задача? Значение и толкование слова zadacha, определение термина. URL: <http://www.onlinedics.ru/slovar/fil/z/zadacha.html> (дата обращения: 15.03.2014).
4. Перчик Е. Методология синтеза знаний: преодоление фактора некорректности задач математического моделирования. URL: <http://pelbook.narod.ru/read.htm> (дата обращения: 15.03.2014).

5. Нариньяни А.С. Недоопределенность в системе представления и обработки знаний // Изв. АН СССР. Техн. кибернетика. 1986. № 5. С. 3–28.
6. Пилипчук Л.А. Разреженные недоопределенные системы линейных алгебраических уравнений. Минск: Изд-во БГУ, 2012. 260 с.
7. Нариньяни А.С., Телерман В.В., Ушаков Д.И., Швецов И.Е. Программирование в ограничениях и недоопределенные модели // Информационные технологии. 1998. № 7. С. 13–22.
8. Смирнов К.Е. Многомерный анализ данных в системах недоопределенных вычислений // Программные продукты и системы. 2010. № 4. С. 71–74.

9. Web-версия универсального решателя математических задач UniCalc. URL: <http://uniserv.iis.nsk.su/unicalc> (дата обращения: 15.06.2014).
10. Расчетно-логические системы на основе недоопределенных моделей. URL: <http://www.artint.ru/projects/nemo.php> (дата обращения: 16.06.2014).
11. Дрождин В.В., Жуков М.В. Модель конструктивно-универсального автомата // Программные продукты и системы. 2009. № 4. С. 122–125.
12. Дрождин В.В., Герасимова Е.В., Зинченко Р.Е., Кондрашин М.В. Семантическая организация самоорганизующейся информационной системы // Биокосмология – нео-Аристотелизм. 2012. Т. 2. Вып. 4. С. 345–359; URL: <http://www.biocosmology.ru/> (дата обращения: 05.09.2014).

DOI: 10.15827/0236-235X.109.017-020

Received 06.09.14

TASKS SPECIFICATION IN A SELF-ORGANIZING INFORMATION SYSTEM*Drozhdin V.V., Ph.D. (Engineering), Associate Professor, drozhdin@yandex.ru;**Shalaev A.A., Postgraduate Student, shell@live.ru**(Penza State University, Krasnaya St. 40, Penza, 440026, Russian Federation)*

Abstract. To ensure self-modification and self-improvement, a self-organizing system should identify intra-system problems and address them via tasks formulation and accomplishment. The article presents general task specification and defines its components. As a method of solution, we suggest using semantic and constructive descriptions. The paper defines corresponding rules between a task and a method of its accomplishment and discusses the ways of forming a method that addresses the task at hand.

This article highlights the advantages of using under-defined tasks in a self-organizing system and suggests a number of ways to complete task definitions in the following cases: a) incomplete coverage of the task definition domain by the problem definition domain, b) solution method partial constraint satisfaction in relation to the initial task conditions, c) deviations from the transformations of the initial data into a solution method equivalence and from the data conversion methods that were initiated by the problem.

The paper proposes the semantic task specification as a concept-task and relationships between the concepts. The concept-task consists of an identifier, a composition, content, a solving method, an external description and macro properties. Possible relationships between concept-tasks include aggregation, classification, generalization and abstraction. Semantic specifications of the tasks that are known to the system and are to be resolved in it can be used to form a layer of tasks in the “problem – task – subtask – method of solution – knowledge” semantic domain.

In conclusion, the problem formulation and solution allows the system to resolve any issues that arise during the course of its operation. In addition, the use of the under-defined tasks significantly enhances system troubleshooting and makes the process more efficient.

Keywords: self-organizing information system, problem, task, under defined task, model tasks, modifying task.

References

1. Drozhdin V.V. Organisation and functioning of a self-organizing information system. *Novye informatsionnye tekhnologii i sistemy: tr. X Mezhdunar. nauchno-tekhn. konf.* [New information technologies and systems: Proc. 10th int. Science and Technical Conf.]. Penza, Penza Gos. Univ. Publ., 2012, pp. 231–233.
2. *Chto takoe Problema? Znachenie i tolkovanie slova problema, opredelenie termina* [What is a Problem? Meaning and explanation of “problem”]. Available at: <http://www.onlinedics.ru/slovar/log/t/problema.html> (accessed March 15, 2014).
3. *Chto takoe Zadacha? Znachenie i tolkovanie slova zadacha, opredelenie termina* [What is a Task? Meaning and explanation of “task”]. Available at: <http://www.onlinedics.ru/slovar/fil/z/zadacha.html> (accessed March 15, 2014).
4. Perchik E. *Metodologiya sinteza znaniy: preodolenie faktora nekorrektnosti zadach matematicheskogo modelirovaniya* [Knowledge synthesis methodology: overcoming an incorrect factor of mathematical modeling tasks]. Available at: <http://pelbook.narod.ru/read.htm> (accessed March 15, 2014).
5. Narinyani A.S. Nedoopredelennost v sisteme predstavleniya i obrabotki znaniy. *Izv. AN SSSR, Ser. Tekhn. Kibernetika* [News of AS SSR, Series Technical Cybernetics]. 1986, no. 5, pp. 3–28 (in Russ.).
6. Pilipchuk L.A. *Razrezhennyye nedoopredelennyye sistemy lineynykh algebraicheskikh uravneniy* [Sparse underdetermined simultaneous linear algebraic equations]. Minsk, BGU Publ., 2012, 260 p.
7. Narinyani A.S., Telerman V.V., Ushakov D.I., Shvetsov I.E. Programming in limitations and underdetermined models. *Informatsionnye tekhnologii* [Information Technologies]. Moscow, Mashinostroenie Publ., 1998, no. 7, pp. 13–22 (in Russ.).
8. Smirnov K.E. Mnogomerny analiz dannykh v sistemakh nedoopredelennykh vychisleniy. *Programmnye produkty i sistemy* [Software & Systems]. 2010, no. 4, pp. 71–74 (in Russ.).
9. *Web-versiya universalnogo reshatelya matematicheskikh zadach UniCalc* [UniCalc Web Version for Solving Mathematical Exercises]. Available at: <http://uniserv.iis.nsk.su/unicalc> (accessed June 15, 2014).
10. *Raschetno-logicheskie sistemy na osnove nedoopredelennykh modeley* [Computational and Logical Systems Based on Underdetermined Models]. Available at: <http://www.artint.ru/projects/nemo.php> (accessed June 16, 2014).
11. Drozhdin V.V., Zhukov M.V. A model of constructive and universal automaton. *Programmnye produkty i sistemy* [Software & Systems]. 2009, no. 4, pp. 122–125 (in Russ.).
12. Drozhdin V.V., Gerasimova E.V., Zinchenko R.E., Kondrashin M.V. Semantic organization of self – organizing information system. *Biocosmology – neo-Aristotelism*. V. Novgorod, 2012, vol. 2, no. 4, pp. 345–359 (in Russ.).