Дрождин В.В., Шалаев А.А. Программная реализация подсистемы самомодификации и самодостраивания адаптивной информационной системы. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XV Междунар. научно-техн. конф. – Пенза: ПДЗ, 2015. – С. 48-57.

УДК 004.42

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПОДСИСТЕМЫ САМОМОДИФИКАЦИИ И САМОДОСТРАИВАНИЯ АДАПТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

В.В. Дрождин, А.А. Шалаев

IMPLEMENTATION OF SELF-MODIFICATION AND SELF-EXTENSION SUBSYSTEM IN AN ADAPTIVE INFORMATION SYSTEM

V.V. Drozhdin, A.A. Shalaev

Аннотация. Рассматривается реализация подсистемы самомодификации и самодостраивания адаптивной информационной системы (АдИС), обеспечивающей высокую адекватность решения задач в различных условиях. Предложены шаблоны программных модулей, осуществляющих решение простых и сложных задач, а также разработаны алгоритмы формирования и модификации программных модулей на основе репозиториев подзадач и методов решения задач. Использование новых программных модулей осуществляется путем их трансляции в dll-модули и динамического подключения в процессе функционирования АдИС.

Ключевые слова: адаптивная информационная система, самомодификация, самодостраивание, задача, программный модуль, шаблон программного модуля, модуль dll.

Abstract. This paper discusses an implementation of the self-modification and self-extension subsystem in an adaptive information system. This subsystem allows maintaining a high level of adequacy of solving problems under different conditions. The article proposes two key things: 1) templates of program modules, which are designed to solve simple and complex problems (tasks); 2) algorithms that provide generation and modification of program modules. In order to use the program modules, the system has to compile a DLL from a program module's source code first and then link the DLL at runtime on demand.

Keywords: adaptive information system, self-modification, self-extension, task, program module, program module template, dll.

Подсистема самомодификации в адаптивной информационной системе (АдИС) решает две задачи:

- формирование программного модуля (ПМ), решающего требуемую задачу в условиях определенного контекста;
- модификация существующего ПМ для удовлетворения процесса решения задачи определенному контексту.

Формирование ПМ с требуемыми свойствами осуществляется менеджером самомодификации по заявке менеджера рабочего процесса [1], осуществляющего загрузку исполняемого программного модуля (ИПМ) в соответствии с алгоритмом на рис. 1.

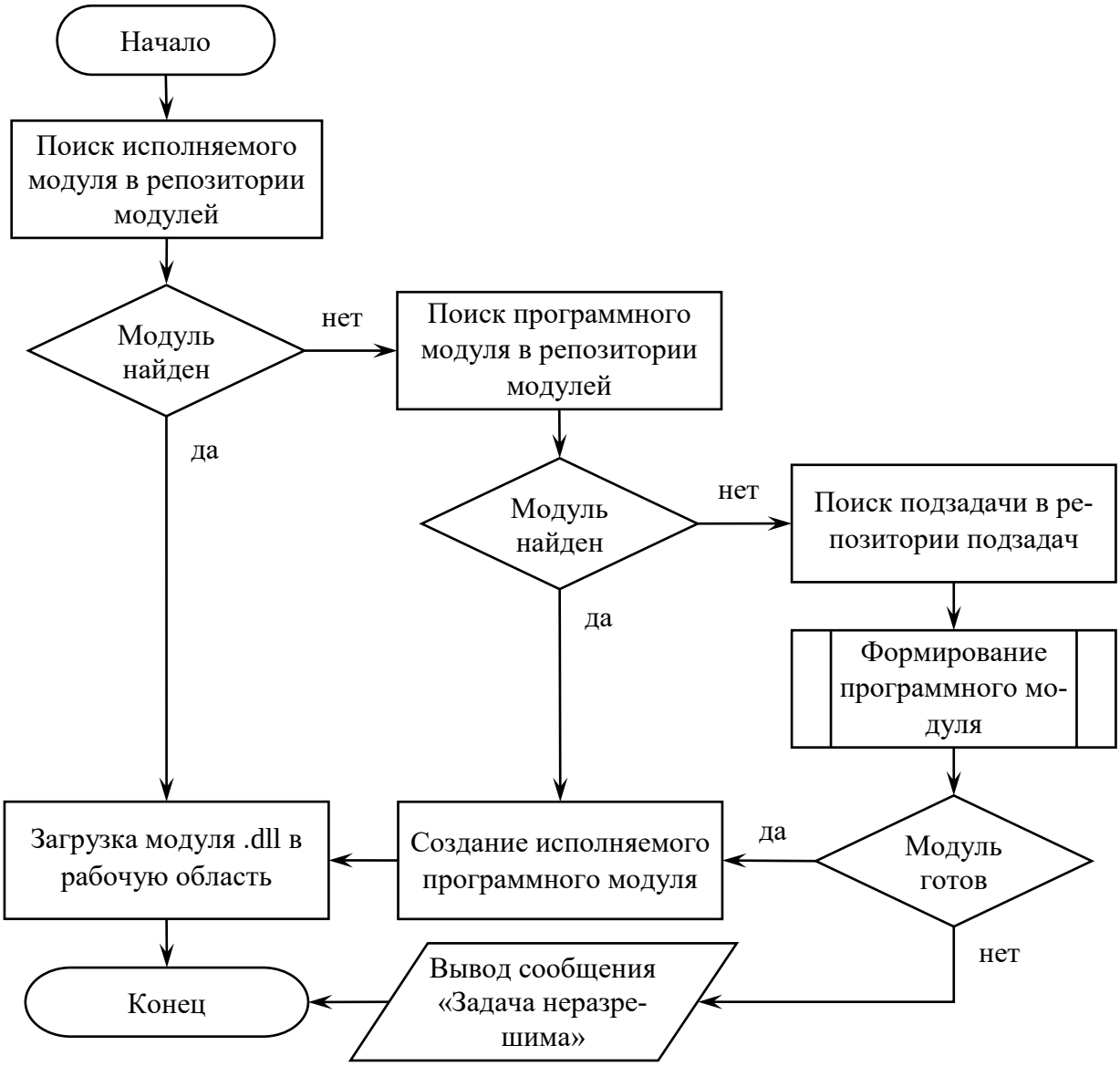

Рис. 1. Алгоритм подготовки и загрузки на исполнение ПМ

В процессе формирования ПМ менеджер самомодификации осуществляет поиск задачи и всех ее подзадач в репозитории подзадач и реализует подзадачи нижнего уровня методами, соответствующими контексту, найденными в репозитории методов (рис. 2). В случае невозможности формирования ПМ менеджер самомодификации производит соответствующую запись в лог проблем.

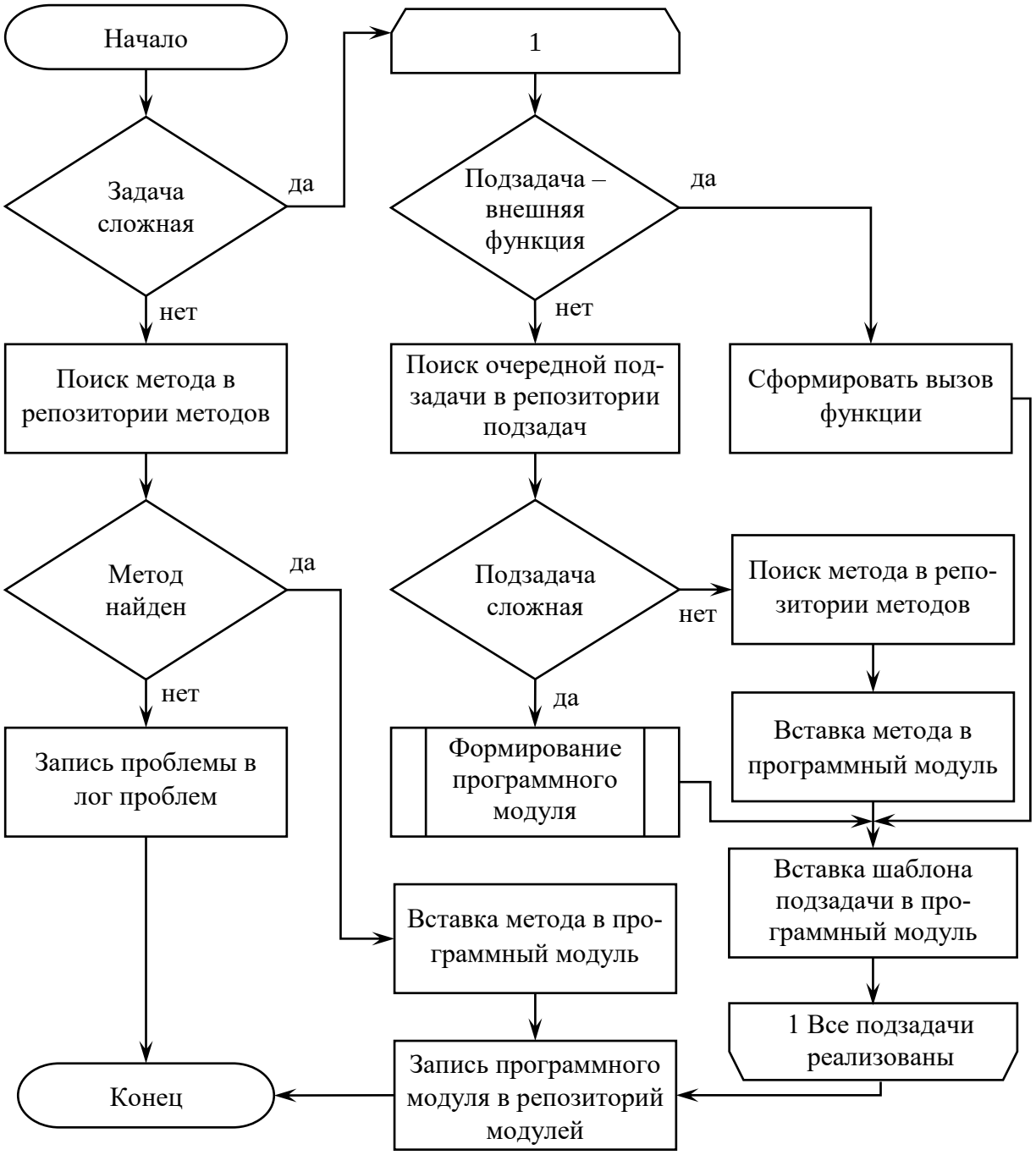

Рис. 2. Алгоритм формирования программного модуля

Формирование ПМ осуществляется в соответствии со структурой шаблонов ПМ (рис. 3, 4) с реализацией связей по управлению и по данным между подзадачами, заданными в шаблоне сложной задачи [2].

Реализация связей по управлению соответствует порядку размещения подзадач в сложной задаче или порядку обращений к подзадачам, задаваемым операторами управления: условным, выбора или циклом.

Для корректной реализации связей по данным необходимо разрешение проблем согласования входов и выходов задачи и подзадач и проблемы возникновения дубликатов переменных. Решение этих проблем целесообразно осуществить на основе использования пространств имен (namespace), соответствующих задаче и всем ее подзадачам. Тогда реализация связей по данным будет осуществляться путем добавления к входным и выходным параметрам названий соответствующих пространств имен.

```
using System;

/// <семантическое описание задачи>
namespace <имя задачи>
{
    public class <имя задачи>
    {
///     <семантическое описание метода решения задачи>
        public <тип> Execute(<входные параметры>)
        {
            <программный код метода решения задачи>
        }
    }
}
```

Рис. 3. Шаблон ПМ, решающего простую задачу

При программной реализации процесса решения сложной задачи системой взаимосвязанных ПМ возникает задача межмодульного взаимодействия.

При реализации связей по управлению и по данным с подзадачей, представленной отдельным программным модулем (объявленной как @task в семантическом описании задачи), необходимо учитывать следующие ограничения:

обращение к подзадаче осуществляется запуском ИПМ;

для передачи данных в ИПМ и получения результатов целесообразно использовать некоторый выходной.

В примере шаблона ПМ на рис. 4 для решения сложной задачи используются подзадачи 1, 2 и 3, причем подзадача 1 выполняется в случае истинности “условия 1”, подзадача 2 – в случае истинности “условия 2”, а подзадача 3 – в противном случае.

```
using System;

namespace <имя задачи>
{
    public class <имя задачи>
    {
///     <семантическое описание задачи>
        public <тип> Execute(<входные параметры>)
        {
            <объявление и инициализация входов задачи>
            <объявление выходного объекта>
            if (<условие 1>)
            {
                namespace <имя подзадачи 1>
                {
///                 <семантическое описание подзадачи 1>
                    <связывание входов подзадачи 1>
///                 <семантическое описание метода решения подзадачи 1>
                    <программный код метода решения подзадачи 1>
                    <запись выходов подзадачи 1 в выходной объект>
                }
            }
            else
            {
                if (<условие 2>)
                {
                    namespace <имя подзадачи 2>
                    {
                        <загрузка ИПМ подзадачи 2>
                        <связывание входов подзадачи 2>
                        <вызов подзадачи 2>
                        <запись выходов подзадачи 2 в выходной объект>
                        <выгрузка ИПМ подзадачи 2>
                    }
                }
                else {
                    namespace <имя подзадачи 3>
                    {
///                     <семантическое описание подзадачи 3>
                        <связывание входов подзадачи 3>
                        <объявление выходного объекта подзадачи 3>

                        namespace <имя подзадачи 3.1>
                        {
///                         <семантическое описание подзадачи 3.1>
                            <связывание входов подзадачи 3.1>
///                         <семантическое описание метода решения подзадачи 3.1>
                            <программный код метода решения подзадачи 3.1>
                            <запись выходов подзадачи 3.1 в выходной объект подзадачи 3>
                        }

                        namespace <имя подзадачи 3.2>
                        {
///                         <семантическое описание подзадачи 3.2>
                            <связывание входов подзадачи 3.2>
///                         <семантическое описание метода решения подзадачи 3.2>
                            <программный код метода решения подзадачи 3.2>
                            <запись выходов подзадачи 3.2 в выходной объект подзадачи 3>
                        }
                        <запись выходов подзадачи 3 в выходной объект>
                    }
                }
            }
            <возврат выходного объекта>
        }
    }
}
```

Рис. 4. Пример шаблона ПМ, решающего сложную задачу

Подзадача 1 является простой задачей, и ее программный код должен быть включен в ПМ при его формировании.

Подзадача 2 является внешней задачей по отношению к данной, и ее программный код будет представлен отдельным ПМ.

Подзадача 3 является сложной задачей, состоящей из последовательности решения подзадач 3.1 и 3.2, программный код которых должен быть включен в ПМ при его формировании.

Связь по данным между подзадачами сложной задачи на рис. 4 осуществляется следующим образом:

входы подзадач связываются с входами задачи или с выходами предшествующих подзадач;

выходы подзадач связываются с выходами подзадач предыдущего уровня или с выходами задачи через выходные объекты.

Шаблон ПМ на рис. 4 представляет эффективную организацию программного кода для решения задачи, однако его практическая реализация является крайне сложной, вследствие недопустимости непосредственного многоуровневого вложения пространств имен в существующих языках программирования. Поэтому для практической программной реализации процесса решения сложных задач разработан шаблон ПМ на основе объектного подхода (рис. 5).

В примере шаблона ПМ на рис. 5 решается та же задача, что и на рис. 4. При этом в класс решаемой задачи включаются:

- метод Execute – основная программа задачи, соответствующая содержимому тега implementation шаблона сложной задачи из [2];

- методы всех внутренних подзадач (в данном примере подзадачи 1, 3.1 и 3.2), соответствующие содержимому тега source шаблона метода из [2].

Особенностью реализации связи по данным является использование выходного объекта только на уровне задачи.

Еще одним принятым ограничением является обязательное указание полных имен библиотек у всех компонентов, используемых из специализированных библиотек (например, System.Math.Sin(…), Microsoft.CSharp. CSharpCodeProvider (…), YamlDotNet.Serialization. Deserializer (…)). Так как предполагается, что программный код методов решения подзадач будет разрабатывается проблемно ориентированными генераторами, то необходимость указания полных названий библиотек не является серьезным ограничением.

```
using System;

/// <семантическое описание задачи>
namespace <имя задачи>
{
    public class <имя задачи>
    {
        public <тип> Execute(<входные параметры>)
        {
            <объявление выходного объекта>
            if (<условие 1>)
            {
                <связывание входов подзадачи 1>
                <вызов подзадачи 1>
                <запись выходов подзадачи 1 в выходной объект>
            }
            else
            {
                if (<условие 2>)
                {
                    <загрузка ИПМ подзадачи 2>
                    <связывание входов подзадачи 2>
                    <вызов подзадачи 2>
                    <запись выходов подзадачи 2 в выходной объект>
                    <выгрузка ИПМ подзадачи 2>
                }
                else
                {
///                 <семантическое описание подзадачи 3>
                    <связывание входов подзадачи 3.1>
                    <вызов подзадачи 3.1>
                    <связывание входов подзадачи 3.2>
                    <вызов подзадачи 3.2>
                    <запись выходов подзадачи 3 в выходной объект>
                }
            }
            <возврат выходного объекта>
        }

///     <семантическое описание подзадачи 1>
///     <семантическое описание метода решения подзадачи 1>
        public <тип> <подзадача 1>(<входные параметры>)
        {
            <программный код метода решения подзадачи 1>
        }

///     <семантическое описание подзадачи 3.1>
///     <семантическое описание метода решения подзадачи 3.1>
        public <тип> <подзадача 3.1>(<входные параметры>)
        {
            <программный код метода решения подзадачи 3.1>
        }

///     <семантическое описание подзадачи 3.2>
///     <семантическое описание метода решения подзадачи 3.2>
        public <тип> <подзадача 3.2>(<входные параметры>)
        {
            <программный код метода решения подзадачи 3.2>
        }
    }
}
```

Рис. 5. Пример шаблона ПМ, решающего сложную задачу на основе объектного подхода

Задача самодостраивания АдИС эквивалентна задаче формирования ПМ, обеспечивающего решение задачи в требуемом контексте.

Другой важной задачей подсистемы самомодификации АдИС является модификация ПМ для обеспечения решения задач с требуемыми свойствами в заданном контексте.

Запуск процесса модификации задач, не удовлетворяющих контексту, осуществляется главным менеджером при выходе критерия оценки функционирования системы [3] за допустимый интервал $J \neq [1{,}4 \ldots 1{,}82]$.

Для модификации задач подсистема самомодификации АдИС анализирует лог модулей, выявляет неразрешимые и несвоевременно выполняемые задачи, ПМ и подзадачи и производит их корректировку путем замены неудовлетворительно решаемых подзадач на подзадачи, удовлетворяющие контексту (рис. 6). При этом метод некорректно или неэффективно решаемой подзадачи может заменяться на известный (например, более эффективный) метод или для решения подзадачи может автоматически разрабатываться сложный способ решения (рис. 7) в виде комбинации известных методов или известных методов и таблицы решений, представляемые в виде сложной подзадачи. В случае невозможности разработки нового способа решения подзадачи или формирования ПМ с требуемыми свойствами менеджер самомодификации производит соответствующую запись в лог проблем.

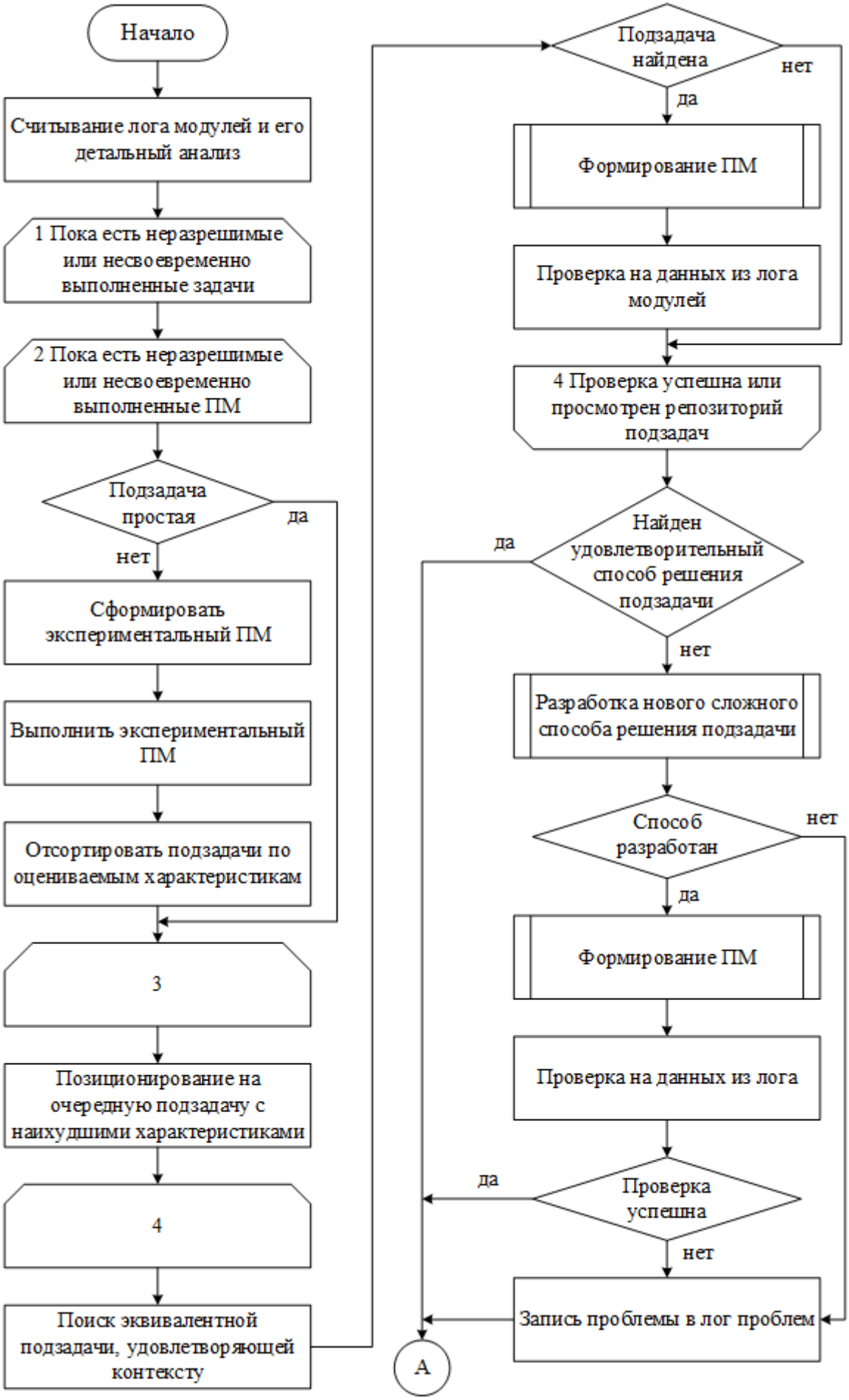

Рис. 6, а. Алгоритм модификации программного модуля

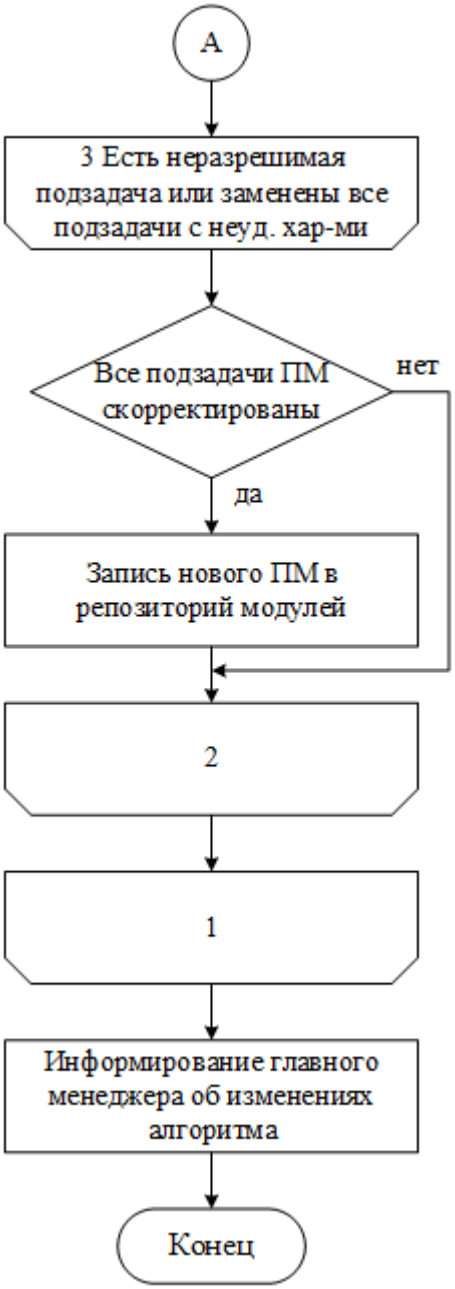

Рис. 6, б. Алгоритм модификации программного модуля

При модификации задачи или ПМ новая подзадача или сложный способ решения включаются вместо неудовлетворительно решаемой подзадачи.

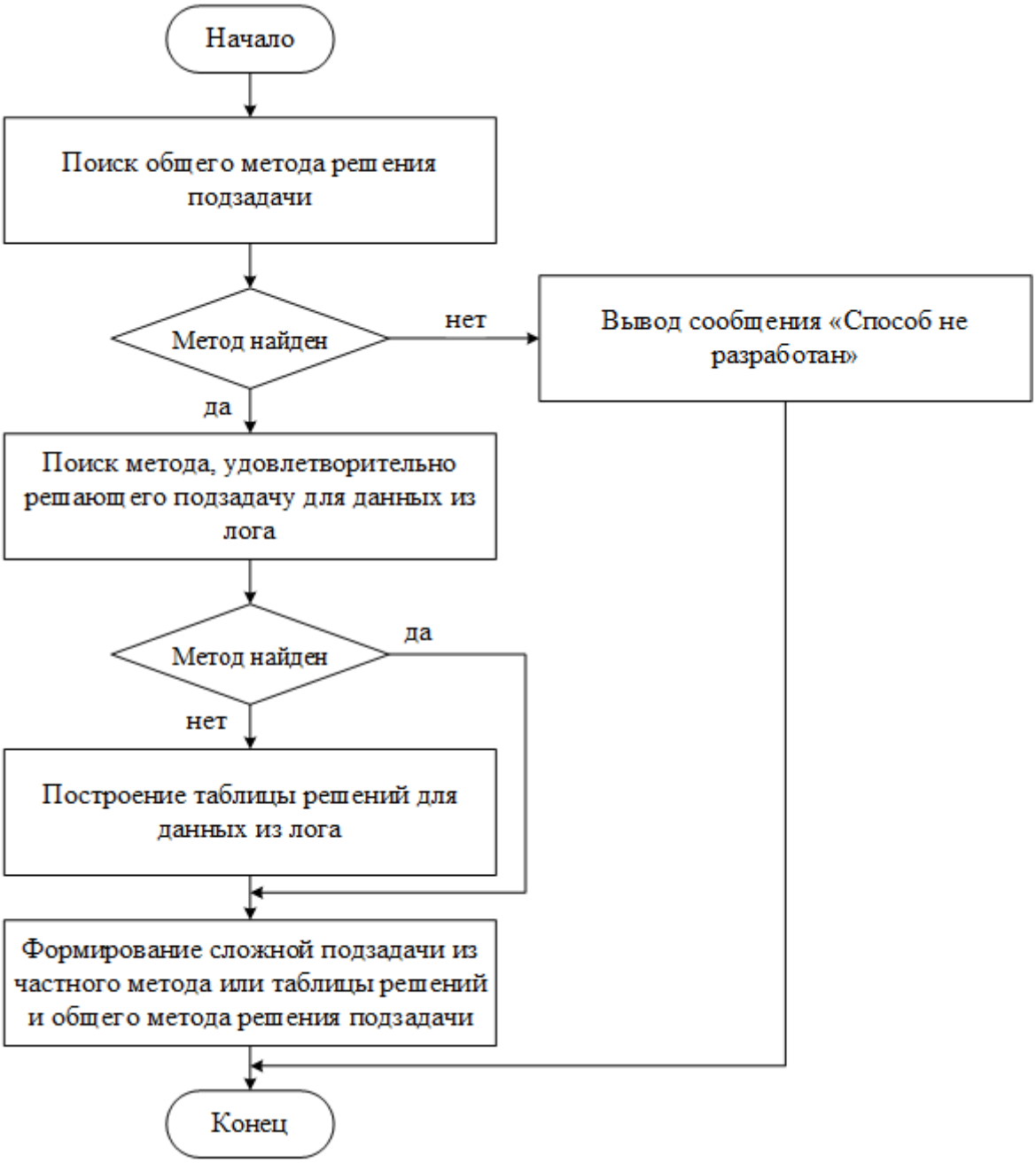

Рис. 7. Алгоритм разработки сложного способа решения подзадачи

Таким образом, разработанная подсистема самомодификации и самодостраивания позволяет АдИС корректно формировать и модифицировать ПМ, обеспечивающие решение задач с требуемыми свойствами в заданном контексте.

Библиографический список

1. Дрождин В.В., Кондрашин М.В., Симакин В.О., Шалаев А.А. Модель организации вычислительного процесса в самоорганизующейся информационной системе // Вестник УГАТУ. – 2014. – Т. 18, № 3 (64). – С. 236–241.

2. Дрождин В.В., Шалаев А.А. Язык разметки программ в самоорганизующейся информационной системе // Проблемы информатики в образовании, управлении, экономике и технике : сб. статей XIV Междунар. науч.-техн. конф. – Пенза: ПДЗ, 2014. – С. 96-104.

3. Дрождин В.В., Шалаев А.А. Модель самомодификации информационной системы // Educatio. – 2015. – № 7(14), Ч. 2. – С. 20–24.

Дрождин Владимир Викторович
Пензенский государственный университет, г. Пенза, Россия
E-mail: drozhdin@yandex.ru

Drozhdin V.V.
Penza State University, Penza, Russia

Шалаев Александр Александрович
Пензенский государственный университет, г. Пенза, Россия
E-mail: shell@live.ru

Shalaev A.A.
Penza State University, Penza, Russia