

Метод повышения эффективности процесса логического вывода в продукционных системах

Пусть $F = \{f_1, f_2, \dots, f_n\}$ – множество фактов в наборе правил ЭС, где $n \in N$ – общее количество фактов. Пусть $A = \{a_1, a_2, \dots, a_m\}$ – множество действий в наборе правил ЭС, где $m \in N$ – общее количество действий. Каждый антецедент включает известный факт из БД ЭС, следовательно, $Ant \subseteq F$, где Ant – множество антецедентов. Каждый консеквент включает известное действие из БД ЭС, следовательно, $Con \subseteq A$, где Con – множество консеквентов. Правило можно представить парой $R_k = \langle ant_k \in Ant, con_k \in Con \rangle$, где $k \in N$ – общее количество правил в базе знаний ЭС.

Аспектом (Аспект) Asp будем называть пересечение отдельных фактов в антецедентах или действий в консеквентах в наборе правил ЭС. Каждый аспект состоит из среза (Pointcut) фактов или действий. Тогда аспектом, являющимся описанием уровней контроля, будет $Asp_F \in F$, а аспектами, представляющими требования контроля состава и содержания документации, – $Asp_{A1} \in A$, контроля исходного состояния ПО – $Asp_{A2} \in A$, статического анализа исходных текстов программ – $Asp_{A3} \in A$, динамического анализа исходных текстов программ – $Asp_{A4} \in A$, требования к отчетности – $Asp_{A5} \in A$.

Заменив в антецеденте и консеквенте конкретный факт или действие выделенным аспектом, получаем представление антецедента в виде $Ant \in Asp_F$, представление консеквента в виде $Con \subseteq \langle Asp_{A1} \vee Asp_{A2} \vee Asp_{A3} \vee Asp_{A4} \vee Asp_{A5} \rangle$. При этом избавляемся от повторяющихся фактов и действий.

Выявленные пересечения фактов или действий в наборе правил образуют аспект. Множество фактов или действий, выделенных в аспект, образуют срез. Аспект внедряется в правило через антецедент или консеквент. Роль совета в данном подходе играют факт или действие, которые будут заменены элементом аспекта. Правило, в котором факт или действие будет заменено элементом аспекта, обозначается целью. Интерпретатор правил выполняет роль компоновщика.

Представим результат анализа составляющих продукционных систем и основных концепций АО-похода:

аспект – выявленные пересечения фактов или действий в наборе правил;

срез – множество фактов или действий, принадлежащих аспекту;

точка соединения – место внедрения элементов аспекта, представленное антецедентом или консеквентом;

совет – факт или действие, которые будут заменены элементом аспекта;

цель – правило, в котором факт или действие будет заменено элементом аспекта;

компоновщик – интерпретатор правил.

Приведем фрагмент набора продукционных правил после применения АО-подхода:

Правило 1

Если

<Уровень контроля → «4»>

то

<Контроль состава и содержания документации → «Выполнить контроль состава и содержания документа Спецификация (ГОСТ 19.202-78)»>

Правило 2

Если

<Уровень контроля → «4»>

то

<Контроль состава и содержания документации → «Выполнить контроль состава и содержания документа Описание программы (ГОСТ 19.402-78)»>

Правило 3

Если

<Уровень контроля → «4»>

то

<Контроль состава и содержания документации → «Выполнить контроль состава и содержания документа Описание применения (ГОСТ 19.502-78)»>

...

Правило 19

Если

<Уровень контроля → «3»>

то

<Статический анализ исходных текстов программ → «Выполнить контроль информационных объектов»>

Правило 20

Если

<Уровень контроля → «3»>

то

<Статический анализ исходных текстов программ → «Выполнить формирование перечня маршрутов выполнения функциональных объектов»>

...

Правило 39

Если

<Уровень контроля → «2»>

то

<Динамический анализ исходных текстов программ → «Выполнить контроль выполнения функциональных объектов»>

Правило 40

Если

<Уровень контроля → «2»>

то

<Динамический анализ исходных текстов программ → «Выполнить сопоставление фактических маршрутов выполнения функциональных объектов и маршрутов, построенных в процессе проведения статического анализа»>

...

Правило 57**Если**

<Уровень контроля → «1»>

то

<Динамический анализ исходных текстов программ → «Выполнить контроль выполнения функциональных объектов»>

Правило 58**Если**

<Уровень контроля → «1»>

то

<Динамический анализ исходных текстов программ → «Выполнить сопоставление фактических маршрутов выполнения функциональных объектов и маршрутов, построенных в процессе проведения статического анализа»>

Правило 59**Если**

<Уровень контроля → «1»>

то

<Отчетность → «Оформить протокол испытаний»>

Внедрение АО-похода не требует переработки структуры базы знаний и БД ЭС. В целях совместимости предлагается использовать АО-поход в интерпретаторе правил.

На рисунке 5 приведены значения скорости логического вывода в базе знаний ЭС после применения АО-подхода.

Как видим, скорость логического вывода в базе знаний ЭС после применения АО-подхода в обоих случаях увеличилась. Выделение аспектов позволило избавиться от непроизводительной проверки применимости правил.

Заключение

Представленная множеством правил продукции ЭС организует знания в виде логической связки *Факт (Посылка) ⇒ Действие (Следствие)*. Умение ЭС решать различные задачи (интерпретация данных, диагностика, мониторинг, проектирование, прогнозирование, планирование и обучение) позволило применять ЭС в различных отраслях промышленности.

В статье был рассмотрен актуальный недостаток производственных систем, который, согласно [4], выражается в более низкой эффективности процесса логического вывода по сравнению с другими моделями представления знаний.

Предложенный АО-подход в производственных системах позволил повысить скорость логического вывода в ЭС, а выделение аспектов – сократить количество операций при поиске решения и избавиться от перебора фактов и действий, что привело к повышению скорости логического вывода.



Рис. 5. Скорость поиска решения в базе знаний ЭС после АО-подхода

Fig. 5. The speed of finding solutions in the expert system knowledge base after AO approach

Литература

1. Минин А.Я. Информационные технологии в образовании. М.: Изд-во МПГУ, 2016. 148 с.
2. Корнеев И.К., Машурцев В.А. Информационные технологии в управлении. М.: ИНФРА-М, 2001. 158 с.
3. Джарратано Дж., Райли Г. Экспертные системы. Принципы разработки и программирование; [пер. с англ.]. М.: Вильямс, 2007. 1152 с.
4. Валетов В.А., Орлова А.А., Третьяков С.Д. Интеллектуальные технологии производства приборов и систем. СПб: Изд-во СПб ГУИТМО, 2008. 134 с.
5. Kiczales G., Lamping J., Mendhekar A., Maeda C., Lopes C., Loingtier J., Irwin J. Aspect-oriented programming. Proc. Europ. Conf. on Object-Oriented Programming (ECOOP), Jyvaskyla, Finland, 1997. URL: <http://people.cs.ubc.ca/~gregor/papers/kiczales-ECOOP1997-AOP.pdf> (дата обращения: 23.04.2018).
6. Rashid A. Aspect-oriented software development in practice: tales from AOSD-Europe. 2010. URL: <https://pdfs.semanticscholar.org/08d0/1e2ebec9c9904376c92d092663c2e5467ed2.pdf> (дата обращения: 15.04.2018).
7. Гончаров А.А., Семенов Н.А. Аспектно-ориентированное программирование в контексте решения вопросов повышения эффективности экономических показателей ИТ-проектов // Программные продукты и системы. 2016. № 3. С. 149–153.
8. Laddad R., Johnson R. AspectJ in action: enterprise AOP with spring applications. Manning Publ., 2009, 568 p.
9. Сафонов В.О. Aspect.NET – инструмент аспектно-ориентированного программирования для разработки надежных и безопасных программ // Компьютерные инструменты в образовании. 2007. № 5. С. 3–13.

Improving logical inference speed of production expert systems using aspect-oriented approach

A.A. Goncharov¹, Postgraduate Student, andre_goncharov@hotmail.com
N.A. Semenov¹, Dr.Sc. (Engineering), Professor, is@tstu.tver.ru

¹ Tver State Technical University, Tver, 170026, Russian Federation

Abstract. The reason for active applying of expert systems in various industries is their ability to solve problems of data interpretation, diagnosis, monitoring, design, forecasting, planning and training.

Each expert system is based on a knowledge representation model. A production model, semantic networks and frames are the most common models. The production model is the one used most frequently.

The paper considers the important shortage of production systems related to low efficiency of the logical inference process compared to other knowledge representation models.

The paper describes the proposed method of increasing the efficiency of the logical inference process in production systems based on applying an aspect-oriented approach.

The aspect-oriented approach allows identifying intersecting functional elements and providing their consolidation during architecture creation and system implementation. This approach was first introduced in 1997 and it remains popular at the present time.

As the example, the article provides a set of production rules of expert system for selecting the requirements for a given level of control according to the requirements of guidance document of undocumented features. In this case, the facts are presented in the form of values of control levels and actions are presented as requirements to the selected level of control.

The proposed aspect-oriented approach to organizing production systems provided an opportunity of increasing the speed of logical inference in expert systems. The separation of intersecting facts and actions from the set of production rules into aspects has made it possible to reduce the number of operations when searching for a solution and to eliminate exhaustive search for facts and actions.

Keywords: aspect-oriented programming, aspect-oriented approach, expert system, production rule, artificial intelligence.

References

1. Minin A.Ya. *Information Technologies in Education*. Moscow, MPGU Publ., 2016, 44 p.
2. Korneev I.K., Mashurtsev V.A. *Informat Information Technologies in Management*. Moscow, INFRA-M Publ., 2001, 158 p.
3. Giarratano J.C., Riley G.D. *Expert Systems: Principles and Programming*. 4th ed. Course Technology Publ., 2004, 288 p. (Russ. ed.: Moscow, Williams Publ., 2007, 1152 p.).
4. Valetov V.A., Orlova A.A., Tretyakov S.D. *Intellectual Technologies for the Production of Devices and Systems*. St. Petersburg, SPb GUITMO Publ., 2008, 134 p.
5. Kiczales G., Lamping J., Mendhekar A., Maeda C., Lopes C., Loingtier J., Irwin J. Aspect-oriented programming. *Proc. Europ. Conf. on Object-Oriented Programming (ECOOP)*. Jyvaskyla, Finland, 1997. Available at: <http://people.cs.ubc.ca/~gregor/papers/kiczales-ECOOP1997-AOP.pdf> (accessed April 23, 2018).
6. Rashid A. *Aspect-Oriented Software Development in Practice: Tales From AOSD-Europe*. 2010. Available at: <https://pdfs.semanticscholar.org/08d0/1e2ebec9c9904376c92d092663c2e5467ed2.pdf> (accessed April 15, 2018).
7. Goncharov A.A., Semenov N.A. Aspect-oriented programming as an approach to improve the efficiency of IT projects economic performance. *Software & Systems*. 2016, no. 3, pp. 149–153.
8. Laddad R., Johnson R. *AspectJ in Action: Enterprise AOP with Spring Applications*. Manning Publ., 2009, 568 p.
9. Safonov V.O. Aspect.NET as an aspect-oriented programming tool for the development of reliable and safe programs. *Computer Tools in Education*. 2007, no. 5, pp. 3–13 (in Russ.).