

УДК 004.9

ПРИМЕНЕНИЕ МЕТОДОВ ДЕКЛАРАТИВНОГО ПРОГРАММИРОВАНИЯ В ЗАДАЧАХ ДИАГНОСТИКИ КОМПЬЮТЕРНЫХ СИСТЕМ

Д. т. н. С. А. Нестеренко, к. ф.-м. н. П. М. Тишин, С. М. Семчишин

Одесский национальный политехнический университет

Украина, г. Одесса

knb47@mail.ru, tik88@mail.ru, sviatoslavnexus@yandex.ua

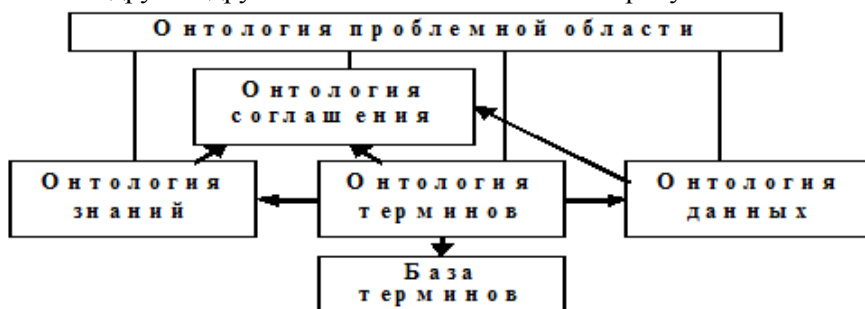
На основе методов декларативного программирования построена конфигурация интеллектуального решателя задач диагностики путем описания артефактов, используемых в процессе работы решателя. Представление конфигурации интеллектуального решателя в виде набора описаний артефактов позволяет снизить трудоемкость на разработку и сопровождение такого решателя.

Ключевые слова: декларативное программирование, диагностика, компьютерная система, управляемый интеллектуальный решатель.

Создание методов интеллектуальной диагностики (ИД) компьютерных систем является одним из передовых направлений в области разработки систем обработки информации. Обусловлено это широкой распространенностью таких систем, растущими требованиями к надежности работы и достоверности результатов. Трудоемкость разработки сложных программных систем ИД сопоставима с разработкой крупных материальных объектов [1], а их сопровождение является еще более дорогим процессом, что обусловлено использованием большого количества трудоемких методов, технологий и инструментальных средств для их развития [2]. Это требует от исследователей разработки новых, менее трудоемких подходов для решения этих проблем. В качестве одного из таких подходов предложена парадигма декларативного программирования [3].

Целью работы является разработка конфигурации управляемого интеллектуального решателя задач диагностики компьютерных систем с использованием декларативного подхода к программированию. Под конфигурацией управляемого интеллектуального решателя в данной работе понимается описание артефактов, используемых решателем, требований к нему и особенностей его работы. Таким образом, создается набор описаний, на основе которого можно динамически создавать и изменять экземпляры такого решателя, минимизируя временные и стоимостные затраты на его создание.

Для построения сложных интеллектуальных систем в настоящее время используют возможности многоагентных технологий [4]. В работе используется современная концепция проектирования интеллектуальных сервисов как совокупности процедурно-декларативных ресурсов [5]. Предлагается все компоненты интеллектуальной системы (онтологии, данные, знания, решатель задач с пользовательским интерфейсом и его программные единицы - агенты) представить в едином унифицированном формате (в форме семантических сетей). Онтологии проблемной области для интеллектуального решателя и их зависимость друг от друга схематически показаны на рисунке.



Онтология проблемной области для интеллектуального решателя

Стрелки указывают направление использования одной модели (элемента конфигурации) в качестве источника информации для формирования следующей модели.

Рассмотрим теперь содержание и структуру артефакта онтологии терминов. Артефакт онтологии терминов задает структуру словаря терминов предметной области. Язык для написания этой онтологии должен позволять перечислить все описатели сущностей, наблюдаемых в действительности (например, в технической диагностике требуется указывать текущие признаки, постоянные особенности, произошедшие события), возможные диапазоны значений этих описателей и другие базовые термины проблемной области. Фрагмент ее на языке прикладной логики может быть представлен следующим образом [6]:

сорт признаки: $\{ \}N \setminus \{ \emptyset \}$,
сорт события: $\{ \}N$,
сорт особенности: $\{ \}N$,
сорт службы: $\{ \}N$,
наблюдения = *признаки* & *события* & *особенности*,
множества значений = $\{ \}N \setminus \{ \emptyset \}$,
сорт возможные значения: *наблюдения* \rightarrow *множества значений*,
сорт по: *службы* \rightarrow $\{ \}N \setminus \{ \emptyset \}$.

признаки \cap *события* = \emptyset && *особенности* \cap *события* = \emptyset && *признаки* \cap *особенности* = \emptyset
наблюдения \cap (\cup (множество: *множества значений*) множество) = \emptyset
(наблюдение: *наблюдения*) μ (*возможные значения* (наблюдение)) ≥ 2

Причина отклонений (*по*) обозначает функцию, которая каждой службе в рамках компьютерной системы сопоставляет множество причин отклонений, описания которых представлены в знаниях. В знаниях должно присутствовать описание хотя бы одной причины отклонений.

На основе методов декларативного программирования построена конфигурация интеллектуального решателя задач диагностики путем описания артефактов, используемых в процессе работы решателя. Унифицированное представление всех компонентов системы в форме семантических сетей позволяет существенно сократить расходы на разработку и сопровождение данных систем за счет использования общих подходов, технологий и инструментальных средств, используемых при их разработке и развитии.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Обзор Microsoft // 2005. [Электронный ресурс] http://www.ict.edu.ru/ft/005126/intro_net.pdf.
2. Промыслов В. Г. и др. Практические аспекты сопровождения и модификации сложных программных систем // Труды IV Международной конференции «Идентификация систем и задачи управления» SICPRO'05. г. Москва.— М.: ИПУ РАН, 2005. С. 1151.
3. Шалфеева Е.А. Разработка модели конфигурации для декларативного программирования управляемых агентных решателей.— Владивосток: ИАПУ ДВО РАН, 2012.
4. Гаврилова Т.А. и др. Взаимодействие интеллектуальных агентов для поддержки сервера дистанционного обучения // IS&ITC.— 2000.— С. 224—227.
5. Клещев А.С. Концепция многоагентной системы в многоцелевом компьютерном банке знаний // Международная конференция по проблемам управления: Сборник трудов.— 2009.— № 4.— С.1585—1595.
6. Нестеренко С. А. и др. Разработка модели онтологии диагностики сервис-ориентированных сетевых структур на основе многосортного языка прикладной логики // Электротехнические и компьютерные системы.— 2012.— № 07(83).— С.102—108.

S. A. Nesterenko, P. M. Tishin, S. M. Semchishin

Application of declarative programming in tasks of computer systems diagnostics.

A configuration of intellectual solver, based on the hypothesis of declarative programming, has been built for diagnostic tasks by describing artifacts used by the solver. It is expected that the configuration of the intellectual solver in the form of a set of definitions will lead to reduction of the development and maintainability complexity of the solver.

Keywords: *declarative programming, diagnostics, computer system, controlled intelligent solver.*