

МОДЕЛИ И МЕТОДЫ ПРИМЕНЕНИЯ И РАЗВИТИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Харитонов А.М., Сергеев Е.С., Пигачёв П.В.

ФГБОУ ВПО «Чувашский государственный университет им .И.Н. Ульянова», Чебоксары, Россия (428015, Чебоксары, Московский пр-т, 15), e-mail: chnk@mail.ru

В статье рассмотрены вопросы разработки методов проектирования и использования автоматизированных систем производства изделий машиностроения. В качестве инструмента для формального представления предметной области предложены онтологические модели. Прикладная онтология рассматривается как база знаний автоматизированной системы производственного предприятия. В процессе формирования прикладной онтологии исследовано функционирование подсистемы автоматизированного контроля комплектующих. Работа подсистемы представлена конечным автоматом. Приводятся характеристики конечного автомата и их семантические соответствия к производственному процессу. Автомат представляет работу системы как продвижение заказа в зависимости от состояний файлов о производственном процессе и требований, заданных во входной цепочке словаря языка предметной области. Исследованы вопросы представления концептов, их отношений и функции переходов автомата. В качестве инструмента реализации конечного автомата использована платформа .NET Framework с применением компонентов набора State Machine.

Ключевые слова: автоматизированные системы, производственный процесс, база знаний, семантика, прикладная онтология, конечный автомат, реализация автомата.

MODELS AND METHODS OF APPLICATION AND DEVELOPMENT OF COMPUTER AIDED SYSTEMS

Kharitonova A.M., Sergeev E.S., Pigachev P.V.

Federal state budget educational institution of higher professional education "Chuvash State University named after I.N. Ulyanov", Cheboksary, Russia (428015, Cheboksary, Moskovsky Prospect, 15), e-mail: chnk@mail.ru

The subject of this article is elaborating the methods for designing and using manufactory enterprise computer aided systems. Ontological models are suggested as a possible ones as means for a formal interpretation of subject domain concepts and their relations. The application ontology is considered as a knowledge base of given manufactory enterprise computer aided system. The application ontology creating process was investigated on the work of the computer aided system add-in products component. The subsystem's functioning is presented by finite state machine model. Characteristics of the model and their semantic according to the manufactory process are given. The finite state machine presents the system's functioning as the order advancement in depend of state files about manufactory process and requires defined in input strings of the subject area language. Questions about a finite automat components and transitions implementation are investigated. As a tool for it the .NET Framework platform's Workflow State Machine components set is used.

Keywords: finite state machine, semantic, computer aided system, application ontology, knowledge base, manufactory process, implementation.

Исследования показывают, что предприятия часто используют морально устаревшие автоматизированные системы (АС) для обслуживания производственных процессов (ПП) на фоне богатого выбора предлагаемой рынком программной продукции. Это связано с высокой стоимостью и излишней универсальностью представленных там АС. Обеспечение отечественных пользователей моделями и методами оперативной интеграции и обработки данных с применением современных технологий позволит внедрять в эксплуатируемые системы дополнительные функциональности. В статье приводится исследование логики функционирования и взаимодействия концептов в области автоматизированных систем.

Цель работы – предложить новые модели и методы для представления производственных процессов и их управлением. Для этого решались следующие задачи: выявить эффективные модели проектирования и использования АС, разработка концептуальной модели управления производственным процессом, разработка методов реализации модели.

Роль и место онтологических моделей

При проектировании или реализации сложных систем математические и физические модели достаточно сложны и громоздки. Альтернативные модели для исследования таких задач должны обладать достаточно выразительной мощностью. Такой мощностью обладают онтологические модели (ОМ). Они основаны на идее формирования общих шаблонов представления об исследуемой области знаний, автоматизированного наполнения их конкретными фактами, компонентами и методами обработки с использованием уже имеющихся знаний. Онтологические базы являются составной частью информационных систем, в них должны аккумулироваться метаданные, представляющие собой факты о предметной области, источниках данных и свойствах этих данных. Каждой конкретной концептуализации предметной области соответствует собственная онтология, то есть логика бытия. Онтологической фиксацией можно назвать буквально: 1) процесс выбора конкретной логики предметной области и/или 2) определение смысла концептов и их отношений конкретного случая концептуализации. Логические теории и специальные языки используют различную терминологию для описания семантики предметной области. Например, в Википедии приводится такое соответствие терминов дескрипционной логики (ДЛ) и онтологического языка (OWL):

- концепт, роль, индивид и база знаний;
- класс, свойство, объект и онтология, соответственно.

При формальном представлении предметной области можно допустить следующие рассуждения. База знаний K предметной области представляется через набор утверждений общего вида $TBox$ и набор утверждений частного вида $ABox$, которые вместе составляют так называемую базу знаний или онтологию, то есть $K = TBox \cup ABox$. То, что это выражение на языке ДЛ называется также онтологией, связано с тем, что мощность дескрипционной логики позволяет задавать семантику языка множествами объектов путем интерпретации атомарных концептов и их ролей. Концептуальную схему, представленную через атомарные концепты и их отношения, можно рассматривать как онтологию верхнего уровня (ОВУ). Значения классов верхнего уровня онтологии можно выражать через подмножества понятий тезаурусов данной предметной области, допустимых в качестве значений свойств этих классов, и с использованием ограничений на языке дескрипторной логики. Можно вести

независимо тезаурус и онтологию, которые вместе формируют интенциональную (схему) и экстенциональную (данные) компоненты структурированных описаний. А возможность интерпретаций позволяет на базе ОВУ создавать прикладные онтологии, отображая концепты и их отношения в соответствующие множества классов, объектов и их свойств. Именно это обстоятельство делает онтологические модели достаточно гибкими и выразительными. Концепт, дополняющий ОМ, независимо, интенциональную или экстенциональную ее составляющую, не отвергает саму модель, а просто расширяет ее. Процесс формирования прикладной онтологии можно продолжать постоянно, при этом имеется возможность корректировать модель, обновляя концепты заменой интерпретирующих их элементов множеств.

Разработка концептуальной модели управления производственным процессом

Исследование функционирования АС и реализация методов их разработки проводились в парадигме управления подсистемой автоматизированного контроля комплектующих (АКК). Поэтому при разработке модели АС в качестве основных концептов и их отношений выделены те, которые связаны непосредственно с подсистемой АКК. Работу подсистемы можно рассматривать как управление последовательностями функций цикла выполнения очередного заказа, то есть функций по обеспечению производственного процесса (ПП) необходимыми комплектующими, исходя из требуемой даты и остатков на складах, определения дефицита, обеспечения покрываемости альтернативами, заказами на закупку или собственным производством. Выполнение функций сопровождается соответствующей регистрацией в файлах базы данных и генерацией необходимых для производственных служб документов для принятия решений. На рис.1 приводится граф конечного автомата (КА), представляющего работу исследуемой подсистемы АКК в процессе продвижения текущего заказа. КА – простейший вариант машины Тьюринга. На базе конечного автомата достаточно просто создавать распознаватели конфигураций для управления функционированием исследуемой предметной области. Конечный автомат определяется пятеркой объектов: $M = (S, R, \delta, s_0, F)$, где S – конечное множество состояний автомата; R – конечное множество допустимых входных символов; δ – функция переходов: отображение множества $(S \times R)$ в подмножество состояний $P(S)$; s_0 – начальное состояние автомата; F – множество заключительных состояний. Цикл выполнения очередного заказа включает в себя распознавание последовательностей цепочек, каждая из которых состоит из одного или нескольких шагов на конфигурациях этого автомата.

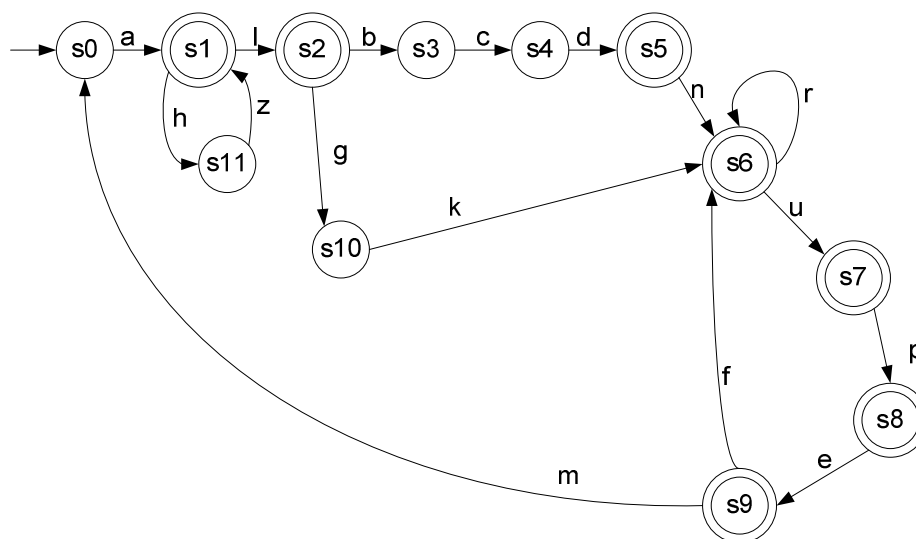


Рис.1. Конечный автомат языка функционирования подсистемы АКК

На базе этого конечного автомата выделены концепты, формируемые при создании прикладной онтологии подсистемы АКК. Приведенные ниже концепты онтологий верхнего уровня, являясь компонентами концептуальных моделей, также являются и компонентами информационной базы исследуемой предметной области.

$O = \{o0, o1, o2, \dots, oN\}$ – онтология и ее конфигурации в цикле выполнения очередного заказа, N – количество необходимых шагов выполнения заказа.

$S = \{s0, s1, s2, \dots, sM\}$ – состояние устройства управления КА.

$F = \{f0, f1, \dots, fK\}$ – функции выполнения заказа. Функцию F можно рассматривать как статическую составляющую базы знаний, элементы которой используются в процессах изменения динамической составляющей онтологической базы в цикле выполнения очередного заказа.

$R = \{a, l, b, ,c, d, n, u, p, e, m, f, g, k, h, z\}$ – алфавит КА. Концепт представлен множеством элементов, обозначенных строчными буквами латинского алфавита. Каждой букве соответствует конкретный ресурс. Наличие в заданной цепочке буквы показывает о необходимости наличия нужного ресурса в состоянии, соответствующем позиции данной буквы. В процессе управления производственным процессом персоналом формируются цепочки, определяющие, по какому пути продвигать выполнение текущего заказа.

$A = \{A1, A2, A3, \dots\}$ – файлы ПП. Элементами множества файлов являются таблицы-файлы с описанием состояний компонентов производственного процесса. Множества записей из этих таблиц используется для интерпретации одноименных концептов.

$K = \{k_a, k_l, k_b, \dots\}$ – концепты о комплектующих. Элементы множества представляют собой также концепты, интерпретируемые множествами записей из таблиц концепта A . Индексы

элементов концепта соответствуют символам алфавита КА и показывают, в каком состоянии должен формироваться концепт для анализа о наличии своего ресурса. Элементы концепта динамически формируются в процессе анализа предметной области, исходя из текущего состояния заказа и в соответствии с заданной цепочкой режима использования конечного автомата. Для их вычисления используются функции с предикатами, позволяющими анализировать наличие необходимой информации в базе данных производственного процесса для выполнения нужной функции перехода. В таблице 1 приводится фрагмент из общего формального представления элементов концепта K .

Таблица 1

Формальная семантика символов алфавита КА

Символы алфавита	Формальное представление концептов	Предикаты (двухместные)
a	$k_a = \exists r0.A9$	$r0$: запись имеет атрибут со свойством $DR \leq DT$
l	$k_l = \forall r1.k_a$	$r1$: запись имеет атрибут со свойством $TR = RP$
b	$k_b = \forall r2.k_l$	$r2$: запись имеет атрибут со свойством $(PDZ = true)$
c	$k_c = k_b \cup \forall r5.A7$	$r5$: запись имеет атрибут со свойством $SZ = true$
d	$k_d = \forall r6.A8 \cup (FV = N)$	$r6$: запись имеет атрибут со свойством $k_c.KK = A8.KK$.

Например, для выполнения функции перехода $\delta(s2, b) = s3$ во входной цепочке должен быть символ b , что требует наличия концепта $k_b = \forall r2.k_l$, где $r2$ – двухместный предикат – запись **имеет атрибут со свойством** $(PDZ = true)$. Интерпретируется множеством записей, у которых все $r2$ -последователи находятся в k_l .

Каждому участнику производственного процесса (ПП) соответствуют собственные цепочки для управления режимами производства, корректировки производственного процесса при выполнении текущего задания. Входная цепочка определяется диспетчером или другим пользователем подсистемы АКК. При наличии концепта заказ продвигается вперед в соответствующем входной цепочке направлении. Например, цепочка $\alpha = alg$ является запросом на формирование закупки для заказа, а цепочка $\beta = abcdefk$ является

запросом на продвижение заказа до поступления его на производственный склад, включая печать накладной на перемещение ТМЦ и производство продукции.

Реализация управляющей программы

Реализация управляющей программы и интерфейсов пользователей ведется в инфраструктуре технологии .NET Framework, которая предоставляет набор услуг для таких условий, когда исполнение заказа может занять долгое время, необходимы приостановки или отмены процессов, проводить анализ и изменение логики ведения процессов.

Применение ее новейших компонентов для работы с состояниями позволяет легче и быстрее создавать модели исследуемых процессов, проводить их реализации. На рисунках 2–4 приводятся фрагменты реализации конечного автомата с применением компонентов State Machine режима Workflow технологии .NET Framework.

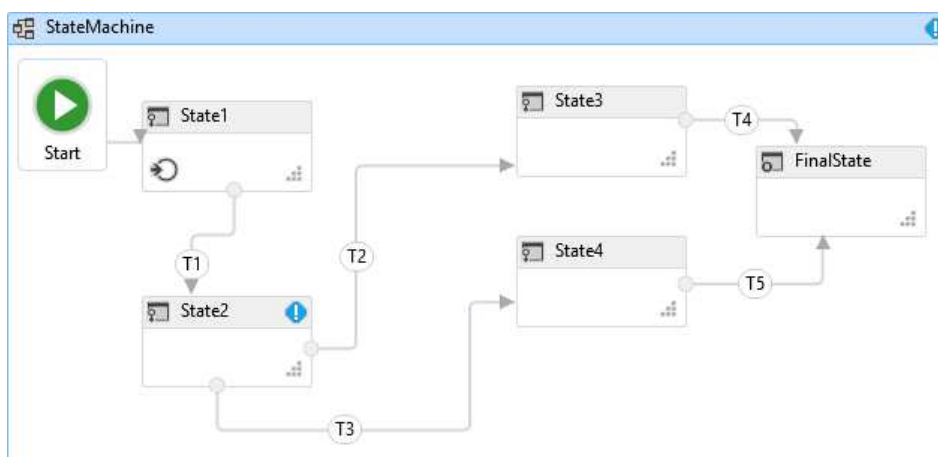


Рис. 2. Фрагмент представления состояний конечного автомата

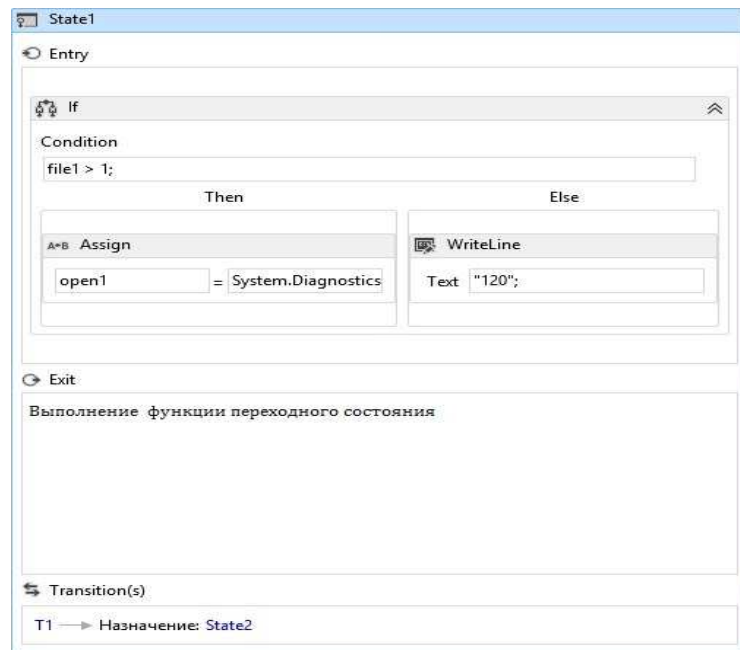


Рис.3. Переход T1 конечного автомата из состояния State 1 в состояние State 2

State1								
	Код	KOD_DE	KOBD	NAIM	CEX_IJG	CEX_POL	KOL_WO	DATA
▶	236	14622	СТД216.9-165	ШТОК	7014	5555	10	27.02.2015
	237	6441	СТД216.10-206	МУФТА	7014	5555	15	27.02.2015
*								
27.02.2015		Вывод						
	Код	KOD_DE	KOBD	NAIM	CEX_IJG	CEX_POL	KOL_WO	DATA
▶	237	6441	СТД216.10-206	МУФТА	7014	5555	15	27.02.2015
	236	14622	СТД216.9-165	ШТОК	7014	5555	10	27.02.2015
*								

Рис. 4. Результат выполнения активностей состояния

Заключение

Продолжением аспекта исследования будет построение прикладной онтологии. Проведение экспериментальных реализаций конечного автомата в процессе разработки прикладной онтологии постоянно расширяет интенциональную составляющую базы знаний предметной области, включающую функции, запросы, спецификации, технологии. Это развивает базовую составляющую автоматизированной системы, что позволит разрабатывать более эффективные методы для решения задач управления производственным процессом.

Список литературы

1. Молчанов А.Ю. Системное программное обеспечение: учебник для вузов. – СПб.: Питер, 2003. – 396 с.
2. Соколов А.П. Системы программирования: теория, методы, алгоритмы: учеб. пособие. – М.: Финансы и статистика, 2004. – 320с.
3. Иванов В.В., Иванов В.А. Модели и методы интеграции структурированных текстовых описаний на основе онтологий // Труды Казан. Школы по компьютерной и когнитивной лингвистике TEL-2008. – Казань: Казан. гос. ун-т, 2009. – 177с.
4. Харитонов А.М. Стандартные методы автоматизации обработки текстовой информации // Компьютерные технологии и моделирование: сб. науч. трудов. – Вып. 8. – Чебоксары: Изд-во Чуваш. ун-та, 2012. – С.67–72.
5. Харитонов А.М. Логические основы языков web-онтологий. Компьютерные технологии и моделирование: сб. науч. трудов. – Вып. 7. – Чебоксары: Изд-во Чуваш. ун-та, 2011. – С.65–68.
6. Харитонов А.М. Применение онтологических средств при реализации подсистемы моделирования в САУ ПП // Компьютерные технологии и моделирование: сб. науч. трудов. – Вып. 6. – Чебоксары: Изд-во Чуваш. ун-та, 2010. – С. 3–10.

Рецензенты:

Славутский Л.А., д.ф.-м.н., профессор кафедры автоматизации и управления в технических системах ФГБОУ ВПО «Чувашский государственный университет им. И.Н. Ульянова», г. Чебоксары;
Артемьев И.Т., д.ф.-м.н., профессор, зав. кафедрой математического и аппаратного обеспечения информационных систем ФГБОУ ВПО «Чувашский государственный университет им. И.Н. Ульянова», г. Чебоксары.