

СПЕЦИФИКАЦИЯ КОМПЛЕКСА МОДЕЛЕЙ ВЗАИМОДЕЙСТВУЮЩИХ ДИНАМИЧЕСКИХ СИСТЕМ

ШПАКОВ В.М.

УДК 681.3.06

Шпаков В.М. Спецификация комплекса моделей взаимодействующих динамических систем.

Аннотация. Обсуждаются преимущества представления моделей сложных динамических систем в виде иерархического комплекса взаимодействующих моделей отдельных подсистем. Предлагается холистическая структура комплекса и расширенный ситуационно-событийный формализм спецификации гибридных процессов, который позволяет обеспечить спецификацию процессов в моделях с учетом взаимодействия моделей между собой и с внешней средой. Показано, что предложенный формализм позволяет моделировать структурные изменения систем. Рассматриваются некоторые вопросы реализации подобного комплекса моделей. Приводится иллюстративный пример комплекса моделей, реализованного на основе предложенного формализма. Комплекс обеспечивает моделирование автоматической координации процессов в двух автоматических системах позиционирования ролика на плоскости.

Ключевые слова: моделирование, автоматическое управление, представление и обработка знаний.

Shpakov V.M. **Interacting dynamic Systems Model Complex Specification.**

Abstract. Advantages of compound dynamic systems presentation in the form of an interacting subsystem models complex are discussed. A holonic complex structure and expanded situation-event formalism of hybrid processes specification are proposed. The formalism enables to specify model processes taking into consideration an interaction of the processes between each other and with the environment. It is shown that the formalism proposed enables to simulate system structure changes. Some problems of the model complex realization are considered. A model complex realized in terms of the formalism proposed is produced as an illustrative example. The complex enables to simulate automatic coordination of processes, which occur in two systems of automatic positioning a roll on a plane.

Keywords: modeling and simulating, automatic control, knowledge presentation and processing.

1. Введение. Роль и значение моделирования процессов и динамических систем непрерывно повышаются во многих областях науки и техники и отраслях производства. Это связано с усложнением исследуемых и проектируемых систем, с одной стороны, и с расширением возможностей вычислительной техники — с другой.

Реальные динамические системы в большинстве случаев содержат в своем составе в качестве структурных единиц другие динамические системы. Так даже в простой автоматической системе могут быть выделены объект управления и управляющая часть системы или контроллер. Сложные системы, например, современные автоматизированные технологические комплексы, как правило, состоят из значительно-

го числа взаимодействующих друг с другом подсистем. Существуют также многоуровневые распределенные системы, содержащие большое число одинаковых подсистем. Моделирование всех таких систем наиболее удобно производить с помощью многомодельных комплексов, структура которых соответствует структуре реальной сложной системы. Возможность формирования комплексов из более простых составляющих моделей позволяет обеспечить целый ряд существенных преимуществ. Описания и представления моделей подсистем оказываются существенно проще. Выразительная способность спецификации становится значительно выше, так как для человека более естественной является спецификация относительно простых и функционально более однородных устройств с последующим объединением их в комплексы. При разбиении сложной модели на подмодели по функциональному признаку упрощается интерпретация результатов моделирования. Модели распространенных подсистем могут храниться в отлаженном виде и неоднократно использоваться в различных комплексах. Появляется возможность при моделировании в зависимости от ситуации изменять состояния моделей отдельных подсистем. Моделям могут, например, присваиваться такие значения, как *включена*—*выключена* или *имеется*—*отсутствует*. При этом расширяются возможности модельного эксперимента, может быть повышена эффективность реализации моделей и появляется возможность моделирования структурных изменений систем.

Однако реализация многомодельного подхода связана с решением ряда задач. Необходимо выбрать структуру комплекса, наиболее адекватную реальным системам и эффективную в смысле реализации. Формализм спецификации процессов в отдельной модели должен быть пригодным для модификации т. е. он должен учитывать влияние процессов в других моделях, обеспечивая связи между моделями и одного, и различных уровней иерархии. В реальных системах состояние элемента одной подсистемы любого уровня может влиять на состояние другого элемента также любой подсистемы. Поэтому в общем случае необходимо обеспечить возможность связи «каждого с каждым».

2. Спецификация процессов в отдельной модели. В общем случае поведение динамической системы определяется совокупностью происходящих в ней взаимодействующих гибридных процессов. Для спецификации этих процессов предлагается использовать ситуационно-событийный формализм, основанный на математической модели гибридного автомата [1] и реализации функций перехода с помощью правил трансформации состояний процессов [2]. Текущее состояние

совокупности гибридных процессов задается множеством вещественных переменных X , представляющих непрерывные составляющие, и множеством логических переменных W , представляющих дискретные составляющие процессов. Среди этих переменных выделяются подмножества независимых внешних воздействий: логических (V) и непрерывных (X_i). В составе множества W выделяются также подмножество Q , содержащее переменные для представления состояний дискретно-событийных процессов и режимов гибридных процессов, и подмножество предикатов от непрерывных состояний (G). Предикаты могут определять состояния дискретных и режимы гибридных процессов. В результате для представления процессов имеем следующие множества переменных

$$W = V \cup Q \cup G, \quad (1)$$

$$X = X_i \cup X_s, \quad (2)$$

где X_s — непрерывные переменные состояния.

Для спецификации процессов необходимо задать функции переходов следующих типов:

$\sigma : W \rightarrow Q \times \{False, True\}$ — функция трансформации состояний дискретно-событийных процессов и режимов гибридных процессов,

$\delta : W \times X \rightarrow X_s$ — функция трансформации непрерывных состояний для возможных режимов гибридных процессов,

$\gamma : X \rightarrow G \times \{False, True\}$ — зависимость значений предикатов от непрерывных состояний процессов.

С учетом этого модель процессов в автономной системе может быть представлена в виде следующего кортежа:

$$M = (V, Q, G, \sigma, \gamma, X, \delta, q_0, Init),$$

где $q_0, Init$ — множества дискретных и непрерывных начальных состояний соответственно.

Функция γ является логической, ее реализация связана с вычислением неравенств от непрерывных состояний. При использовании логических переменных формирование функции перехода σ можно производить с помощью логических связей между переменными, т. е. описывать область определения функции с помощью логических формул, что делает описание более наглядным. Функция перехода σ задается с помощью совокупности продукционных правил вида «Условие \rightarrow Действие», в которых в качестве условия может быть исполь-

зована логическая формула, а в качестве действия — присвоение требуемых значений определенным состояниям и режимам процессов. В общем случае для спецификации модели могут потребоваться произвольные логические формулы. В качестве условия наиболее удобно использовать элементарные конъюнкции логических переменных. Такие конъюнкции интуитивно понятным образом могут интерпретироваться как динамические ситуации. Элементарная конъюнкция всех логических переменных спецификации процесса определяет глобальную динамическую ситуацию. Элементарные конъюнкции, составленные из элементов подмножеств W , можно назвать локальными динамическими ситуациями. Вводя обозначение S_j для некоторой динамической ситуации, получим

$$S_j = s_{j_1}, \dots, s_{j_i}, \dots, s_{j_n}, \text{ где } s_{j_i} = w_{j_i} \text{ или } s_{j_i} = \neg w_{j_i}, w_{j_i} \in W, \quad (3)$$

$$n \in N_w, N_w = |W|.$$

Обозначая множество значимых локальных ситуаций S , тип функции трансформации динамических ситуаций можно определить, как $\sigma : S \rightarrow Q \times \{False, True\}$. Такая функция может быть специфицирована совокупностью правил типа «if...then», левая часть (условие) которых является локальной ситуацией, а правая (действие) — совокупностью (списком) логических переменных состояния:

$$S_j \rightarrow q'_{j_1}, \dots, q'_{j_i}, \dots, q'_{j_m}, q'_{j_i} \in Q \times \{False, True\}, m \in N. \quad (4)$$

При этом правила между собой соединены логической связкой \vee (или). Это значит, что одно и то же действие может быть произведено при выполнении различных условий, т. е. условия могут образовывать дизъюнкцию. Поскольку дизъюнкция элементарных конъюнкций представляет собой дизъюнктивную нормальную форму (ДНФ), то можно утверждать, что таким способом можно в качестве условия действия представить любую логическую формулу. Реализация функции σ производится исполняющей процедурой, которая в цикле сканирует правила, вычисляет значения условных частей правил и в зависимости от этих значений выполняет присваивание специфицированных значений переменным из исполнительской части правил.

Для реализации функции перехода δ необходимо создавать исполняемые спецификации непрерывных составляющих процесса для каждого режима. Предлагается это делать на основе использования передаточных функций элементарных динамических звеньев, функциональных преобразователей и структурных схем их соединения [3]. Спецификацию же элементарных процессов удобно производить на

основе использования транзитивных моделей представления процессов [4]. Эти модели позволяют непосредственно определять отношение следования между входным воздействием и текущим состоянием процесса, с одной стороны, и новым следующим его состоянием — с другой. При рассмотрении транзитивных моделей используется соглашение о том, что именем переменной со штрихом обозначается состояние процесса, непосредственно следующее за состоянием, обозначенным этим же именем без штриха. В случае свободного одномерного непрерывного процесса отношение следования или транзитивное отношение (transition relation) представляет собой бинарное отношение на множестве вещественных чисел вида $\tau(y, y')$ или $y' = \tau(y)$, где y — текущее состояние процесса, а y' — его следующее состояние. Транзитивное отношение τ определяется параметрами элементарного динамического звена и длительностью интервала времени Δt между y и y' . В случае вынужденного процесса оно также зависит от внешнего воздействия x . Для реализации зависимости непрерывных состояний от логики развития гибридного процесса процедуры вычисления этих отношений включаются в исполнительные части правил, условными частями которых являются требуемые значения соответствующих режимов. В принятых обозначениях эти правила имеют вид

$$S_j \rightarrow x'_k = \tau_k(x_k, x), x_k \in X_s, x \in X, \quad (5)$$

где S_j — ситуация, определяющая режим, τ_k и x_k — соответствующие отношение и состояние, x — вещественная переменная модели.

Формирование структурных динамических схем производится путем использования одних состояний в качестве входных воздействий при вычислении других состояний, а также использования сумматоров и других функциональных преобразователей. Эти правила включаются в общий цикл обновления состояний процессов. Обновление состояния процесса в цикле представляет собой реализацию транзитивного замыкания отношения следования и позволяет представить процесс как траекторию состояний.

3. Структура комплекса моделей. Сложные динамические системы, такие, например, как современные автоматизированные производственные комплексы, содержат в качестве составных частей другие динамические системы, некоторые из которых могут быть тоже сложными и так же иметь в своем составе динамические системы. Иерархические структуры такого типа получили название колониических и ха-

рактируются степенью вложенности. Концепция холонов была предложена в работе [5]. «Холос» по-гречески значит целое, а «он» означает часть. Основная идея холонов заключается в единстве двух, на первый взгляд, несовместимых сущностей. Холон представляет собой автономное целое и вместе с тем является частью другого, внешнего по отношению к данному, холона. Холон обладает способностями к автономному функционированию и к взаимодействию с другими холонами. Холоническая структура является адекватной большому числу различных биологических, общественных и технических систем. Ее использование, например, для представления производственных систем оказалось весьма эффективным и рассматривается в настоящее время как наиболее перспективное [6].

Для представления комплекса моделей динамических систем холоническая структура подходит наилучшим образом. Она соответствует большинству реальных сложных систем. Если обеспечить модели системы возможность иметь в своем составе произвольное количество моделей, таких же, как она сама, то с помощью такой структуры может быть представлен произвольный комплекс моделей любой степени вложенности. Отношения между головной моделью и моделями, входящими в ее состав, являются отношениями типа *часть—целое*, т. е. в этом случае модели нижнего уровня являются частями (подмоделями) головной модели, которую по отношению к ним можно назвать надмоделью. На рис. 1 приведена структурная схема такой рекурсивной модели, содержащей три подмодели.

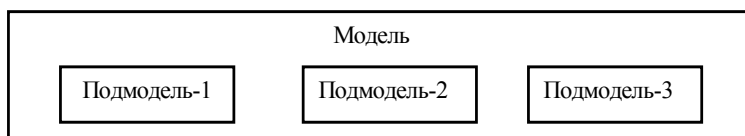


Рис. 1. Структурная схема рекурсивной модели.

Такой комплекс допускает очень эффективную программную реализацию, так как большинство универсальных языков программирования поддерживают рекурсивное описание данных и создание рекурсивных процедур. Рекурсивные структуры данных содержат в своем составе элементы такого же типа, как они сами. Аналогично, в теле рекурсивных процедур имеются точно такие же процедуры, оперирующие, конечно, другими данными. Таким образом, механизм рекурсии позволяет создавать программные объекты произвольной степени вложенности.

Для обеспечения спецификации комплекса моделей необходимо обеспечить спецификацию взаимодействия процессов различных моделей между собой. Для этого рассмотренный формализм спецификации процессов модели необходимо модифицировать следующим образом. Во-первых, надо ввести логический вектор состояния подмоделей данной модели. Эти переменные могут использоваться и в условных, и в исполнительных частях продукционных правил (2). Далее, все логические переменные данной модели, включая состояния подмоделей, должны иметь возможность участия в формировании ситуаций в своих подмоделях, в надмодели и в других моделях своего уровня вложенности. Аналогично, необходимо, чтобы все вещественные переменные модели могли быть использованы в качестве аргументов процедур вычисления транзитивных отношений в подмоделях, надмодели и других моделях одного с данной моделью уровня.

4. Спецификация процессов взаимодействия моделей комплекса. Одним из достоинств ситуационного подхода к спецификации систем являются простота и удобство модификации спецификаций [7]. Можно, например, добавлять и удалять правила, не изменяя оставшиеся правила.

Прежде всего, нужно ввести новый вектор логических переменных состояния, а именно: вектор состояния подмоделей SM . После этого множество логических переменных модели в отличие от (1) будет иметь вид $W = V \cup Q \cup G \cup SM$.

Переменные состояния подмоделей должны иметь возможность участвовать в определении ситуации своей, а также других моделей. Для обеспечения взаимодействия модели с другими моделями разных уровней вложения необходимо расширить области определения всех функций перехода $(\sigma, \lambda, \gamma)$, включив в них переменные из других моделей. Однако каждая модель комплекса может иметь свое автономное функционирование, не влияющее на другие модели. Поэтому из всего множества переменных модели необходимо выделить подмножество публичных или выходных переменных. Тогда оставшиеся переменные можно назвать автономными. Введем в описание модели вектор выходных логических переменных Z и вектор выходных вещественных переменных Y . В выходные вектора могут быть включены соответствующие переменные данной модели и координаты соответствующих выходных векторов подмоделей данной модели. Таким образом, выходные вектора модели являются подмножествами следующих множеств переменных:

$Z \subset W \bigcup_{k=1}^n Z_k$, где $Z_1 \dots Z_n$ — множества выходных логических переменных подмоделей данной модели;

$Y \subset X \bigcup_{k=1}^n Y_k$, где $Y_1 \dots Y_n$ — множества выходных вещественных переменных подмоделей данной модели.

Включая выходную переменную модели нижнего уровня в выходной вектор модели верхнего уровня, можно обеспечить использование состояния модели самого низкого уровня в формировании ситуации в модели самого верхнего уровня. Выходные переменные модели могут поступать на входы моделей одного с данной моделью уровня, а также участвовать в формировании выходного вектора надмодели. Подмодели являются составными частями модели, поэтому любые переменные состояния модели могут участвовать в формировании правил трансформации состояния подмоделей. С учетом сказанного динамическую ситуацию в модели, находящейся в составе комплекса, в отличие от (3), можно определить следующим образом:

$$S_j = s_{j_1}, \dots, s_{j_i}, \dots, s_{j_n}, \text{ где } s_{j_i} = w_{j_i} \text{ или } s_{j_i} = \neg w_{j_i},$$

$$w_{j_i} \in W \cup W_0 \bigcup_{k=1}^m Z_k,$$

$$m = |SM|, n \in N_w, N_w = |W| + |W_0| + \sum_{k=1}^m |Z_k|.$$

Здесь W — множество логических переменных данной модели, W_0 — множество логических переменных надмодели, Z_k — множество логических выходных переменных k -й подмодели, m — количество подмоделей. При этом правило трансформации логических состояний (4) не изменится. Аналогичным образом расширяется область определения функции перехода вещественных переменных. В случае модели в составе комплекса правило (5) можно представить в следующем виде:

$$S_j \rightarrow x'_k = \tau_k(x_k, x_i), x_k \in X_s, x_i \in X \cup X_0 \bigcup_{k=1}^m Y_k,$$

где X — множество всех вещественных переменных данной модели, X_0 — множество всех вещественных переменных надмодели, Y_k — множество выходных вещественных переменных k -й подмодели.

Возможность использования переменной состояния подмодели в формировании ситуаций надмодели и в исполнительных частях ее трансформационных правил позволяет существенно расширить выразительные возможности спецификаций. Изменяя в соответствии с ситуациями надмодели состояния ее подмоделей, можно моделировать изменения структуры системы, управлять структурой. Возможны различные варианты использования состояний подмоделей. Например, при выключении модели ее переменные могут принимать определенные (например, исходные) значения и продолжать влиять на ситуации в других моделях. При изменении значения переменной состояния подмодели, соответствующего ее присутствию в комплексе, можно с помощью этой переменной блокировать (разблокировать) исполнения связанных с ней правил в надмодели и в других моделях, что соответствует отсутствию данной модели в комплексе. Кроме того, подмодели могут представлять собой не реальные динамические объекты, а являться наборами различных правил трансформации состояния одного и того же объекта. В этом случае надмодель, меняя состояния подмоделей, может в разных ситуациях использовать различные наборы правил.

Определить необходимое расширение функций перехода достаточно просто. Реализовать его программным путем несколько труднее. Рассмотрим некоторые принципиальные вопросы программной реализации комплекса моделей взаимодействующих динамических систем.

5. Некоторые вопросы реализации комплекса моделей.

5.1. Структура данных модели. Реализация такого комплекса моделей может быть эффективно произведена средствами объектно-ориентированного программирования. В этом случае мы имеем дело с программным объектом, в состав которого включены объекты того же самого типа. При этом многие описания и алгоритмы создания, сохранения и обновления состояния объектов (моделей) могут быть реализованы как рекурсивные, что упрощает реализацию комплекса в целом. С учетом сказанного и в соответствии с формализмом представления процессов программный объект представления модели должен включать в свой состав:

- вектора логических (дискретных) переменных: входных, состояний и режимов, предикатов, выходных;
- логический вектор состояний подмоделей;
- вектора вещественных (непрерывных) переменных: входных, состояний, выходных;

- список правил трансформации логических состояний и режимов;
- список правил трансформации предикатов диапазонов;
- список правил трансформации непрерывных состояний;
- процедуру создания модели;
- процедуру сохранения модели;
- процедуру обновления переменных состояния модели.

Рассмотрим кратко информационное содержание приведенных составляющих модели. Если данная модель содержит подмодели, то в векторе состояния подмоделей должно иметься соответствующее количество записей. Каждая запись должна содержать:

- 1) указатель на такой же программный объект, представляющий подмодель;
- 2) логическую переменную, представляющую состояние данной подмодели; имя подмодели;
- 3) текстовую переменную для хранения справочных данных о подмодели.

Поскольку в формировании ситуаций и в процедурах вычисления транзитивных отношений для непрерывных состояний могут использоваться переменные различных векторов и различных моделей, то идентификаторы переменных в трансформационных правилах должны содержать данные о своих моделях и векторах. Такие идентификаторы должны включать в свой состав следующее:

- 1) идентификатор модели ,
- 2) идентификатор подвектора ,
- 3) индекс состояния в данном подвекторе ,
- 4) указатель на значение переменной ,
- 5) указатель на имя переменной.

Указатель на значение введен для повышения быстродействия исполняющей процедуры, а указатель на имя — для упрощения процедур редактирования векторов и правил.

5.2. Основные алгоритмы функционирования модели. Как видно из описания структуры комплекса, модель является рекурсивным объектом. Это значит, что элементами ее структуры могут быть такие же объекты, как она сама. При этом всегда создается только одна модель, которая может содержать произвольное количество подмоделей, каждая из которых, в свою очередь, также может содержать произвольное число подмоделей и т. д. Таким образом, в общем случае мы имеем дело с деревом моделей, отношения между вершинами которого являются отношениями *часть—целое*. Если необходимо создать одноуров-

новый комплекс моделей, создается двухуровневый комплекс, в котором модель верхнего уровня не используется. Она не содержит ни векторов, ни правил и является, по существу, оболочкой комплекса моделей. Функционирование подобных объектов удобно задавать с помощью рекурсивных алгоритмов и соответственно рекурсивных процедур и функций. При работе рекурсивного алгоритма он выполняется как относительно корневой модели, а также относительно всех содержащихся в ней подмоделей. При этом происходит обход дерева моделей. Возможны различные варианты обхода дерева: прямой и обратный обходы в глубину и обход дерева в ширину. В случае использования обратного обхода дерева в глубину производится переход к листовой модели дерева, т. е. к модели, не содержащей подмоделей, обрабатываются все листовые модели, после чего алгоритм переходит на вышестоящий уровень. Заканчивается такой алгоритм обработкой корневой модели. Ниже в качестве примеров приведены описания некоторых алгоритмов функционирования модели.

Для описаний использован псевдокод алгоритмических языков программирования. Примем соглашение о том, что имя модели совпадает с именем директория файловой системы, в котором хранятся вектора и правила данной модели. Это не должно создать какие-либо неудобства для пользователя, но позволит существенно упростить процедуры создания и сохранения модели.

Задача алгоритма создания модели состоит в том, чтобы считать данные из файлов векторов состояния и правил трансформации и поместить их в соответствующие массивы записей о состоянии элементов модели и в связанные списки правил. Алгоритм является рекурсивным, т. е. он производит обход дерева моделей и выполняется для каждой подмодели каждой модели.

```
алгоритм_конструктора Создать_модель (Имя_модели);
начало
    если имеются подмодели, то
        повторить с первой до последней подмодели
            Создать_модель ( Имя_подмодели );
        Считать данные из файлов директория Имя_модели
            в структуры данных программного объекта;
    конец алгоритма;
```

Задача алгоритма обновления состояния модели заключается в том, чтобы сканировать списки трансформационных правил модели, вычислять условную часть каждого правила. Для правил относительно непрерывных и логических состояний и режимов алгоритм должен анализировать результат и если он имеет значение «Истина», то выполнить действия, указанные в исполнительной части правила. Эти

действия состоят в том, чтобы присвоить специфицированные логические значения состояниям и режимам из исполнительской части продукционных правил. Исполнительская часть правила трансформации непрерывных состояний содержит имя процедуры, ее параметры, указатель на переменную состояния и указатели на аргументы. В этом случае алгоритм должен запустить процедуру, и присвоить полученное значение переменной состояния, указанной в данном правиле. Что касается правил трансформации предикатов диапазонов, то в их условной части представлены неравенства относительно непрерывных переменных и констант, поэтому алгоритм непосредственно вычисляет предикат, задаваемый неравенством, и полученное значение присваивает соответствующей логической переменной.

Алгоритм обновления состояния модели является объединением алгоритмов обработки всех трансформационных правил. Он должен быть рекурсивным, т. е. применяться ко всем подмоделям данной модели:

```
Алгоритм обновить_состояние_модели(указатель на модель)
начало
  если имеются подмодели, то
    повторить с первой до последней подмодели
      обновить_состояние_модели(указатель на подмодель);
    обновить_логические_состояния(указатель на модель);
    обновить_предикаты_диапазонов(указатель на модель);
    обновить_непрерывные_состояния(указатель на модель);
  конец_алгоритма;
```

6. Иллюстративный пример моделирования сложной динамической системы. Описанный формализм спецификации комплекса моделей был использован при разработке новой версии прототипа компьютерной среды моделирования EnviCon-5. В целях проверки формализма и тестирования прототипа среды разработан комплекс моделей для моделирования системы координации процессов в двух автоматических системах позиционирования ролика на плоскости. Данная система состоит из объекта управления (ролик на плоскости) и контроллера. Контроллер обеспечивает перемещение ролика из исходной в заданную позицию на плоскости за счет управления углом наклона плоскости. Система координации процессов может задавать целевую позицию для каждого ролика и угол наклона плоскости (управляющее воздействие) для каждого контроллера. Структурная схема трехуровневого семимодельного комплекса приведена на рис. 2. Задача системы координации состоит в том, чтобы обеспечить одновременный приход роликов в целевую позицию независимо от их исходных позиций. Возможны различные стратегии координации. Была ре-

лизована следующая. Система координации устанавливает роликам одинаковые целевые позиции. Затем она определяет, у какого ролика расстояние до цели больше и назначает ему максимально допустимый угол наклона. Далее, зная соотношение дальностей до цели и закон движения роликов, система вычисляет требуемое значение угла наклона плоскости для второго ролика и назначает его второму контроллеру.

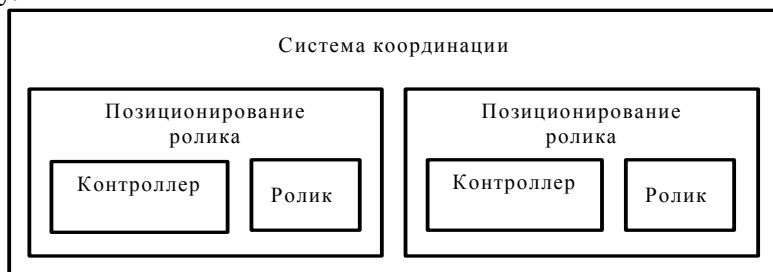


Рис. 2. Структурная схема комплекса моделей

На рис. 3 показаны положения роликов и углы наклона плоскостей в начале и в конце движения, а на рис. 4 приведены графики процессов позиционирования роликов.

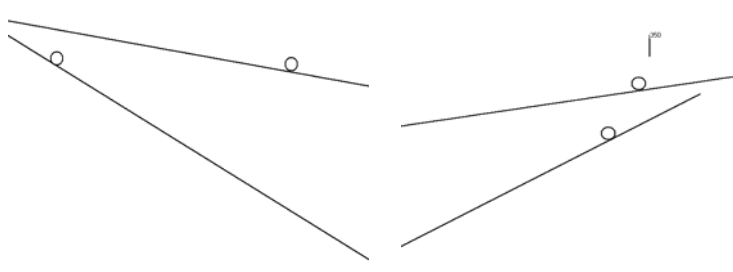


Рис. 3. Положения роликов в начале и в конце движения.

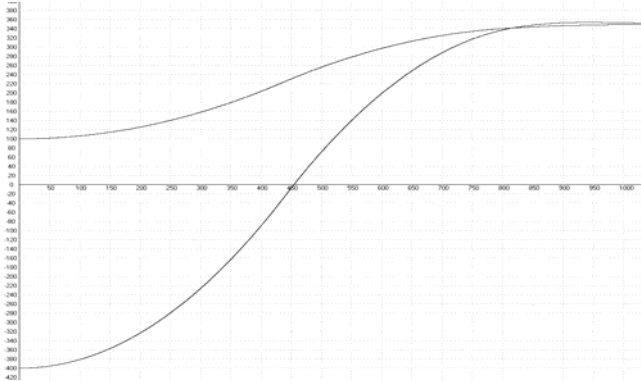


Рис. 4. Графики процессов позиционирования роликов.

Как видно из рис. 3, углы наклона плоскостей на участках разгона и торможения различны. Угол наклона плоскости меньше в той системе, где отклонение ролика от целевой позиции меньше. Углом определяется ускорение ролика, его изменение позволяет изменять время достижения роликом цели. Из графиков видно, что при различных исходных позициях ролики приходят в целевую позицию одновременно.

7. Заключение. Предложенная холоническая структура комплекса моделей является адекватной многим сложным динамическим системам. Приведен формализм спецификации процессов в автономной системе и предложен способ расширения этого формализма. Расширенный формализм позволяет обеспечить спецификацию взаимодействия процессов подсистем различных уровней. Проведенная на основе расширенного формализма реализация трехуровневого семимодельного комплекса для моделирования системы координации процессов является экспериментальным подтверждением пригодности приведенного формализма для спецификации сложных динамических систем.

Литература

1. *Henzinger T.A.* The Theory of Hybrid Automata // Proc. of the 11th An. IEEE . on Logic in Computer Sci. (LICS 96). 1996. P. 278–292.
2. *Шнаков В.М.* Ситуационные спецификации имитационных моделей гибридных реактивных систем // Тр. СПИИРАН. 2002. Вып. 1, т. 2. С. 212–222.
3. *Шнаков В.М.* Спецификация знаний динамики на основе транзитивной модели непрерывных процессов // Тр. СПИИРАН. 2006. Вып. 3, т. 1. С. 191–197.
4. *Alur R., Henzinger T.A., Lafferriere G., Pappas G. J.* Discrete Abstractions of Hybrid Systems // Proc. of the IEEE. 2000. N 88. P. 971–984.
5. *Koestler. A.* The Ghost in the Machine. London: Arkana Books, 1989.

6. *Adrien R. Presley, Donald H. Liles. A holon-based process modeling methodology // Intern. J. of Operations & Production Management. 2001. Vol. 21, N 5/6. P. 565–581.*
7. *Поспелов Д.А. Ситуационное управление: теория и практика. М.: Наука, 1986. 288 с.*

Шпаков Владимир Михайлович — канд. физ.-мат. наук, доцент; старший научный сотрудник лаборатории интегрированных систем автоматизации Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Область научных интересов: автоматическое управление, искусственный интеллект, логическое программирование, экспертные системы, поддержка принятия решений. Число научных публикаций — 54. vlad@iias.spb.su; ЛИСА СПИИРАН, 14-я линия В.О, д. 39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328-8071, факс +7(812)328-4450.

Shpakov Vladimir Michajlovich — Ph.D. (Tech.), associate professor; senior researcher, laboratory of integrated systems for automation, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: automatic control, artificial intelligence, logic programming, expert systems, decision making support. The number of publications — 54. vlad@iias.spb.su; SPIIRAS, 39, 14th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812)328-8071, fax +7(812)328-4450.

Поддержка исследований. В публикации представлены результаты исследований, поддержанные грантом РФФИ 08-07-00252-а, рук. В.М. Шпаков.

Рекомендовано лабораторией интегрированных систем автоматизации, заведующий лабораторией А.В. Смирнов, д-р техн. наук, проф.
Статья поступила в редакцию 22.12.2009.

РЕФЕРАТ

Шпаков В.М. Спецификация комплекса моделей взаимодействующих динамических систем.

Статья посвящена разработке теоретических основ создания компьютерных средств реализации совокупностей взаимодействующих процессов. Целью работы является разработка формализма, обеспечивающего возможность однозначного и непротиворечивого описания сложных динамических систем, имеющих холоническую структуру. Такие системы могут содержать в своем составе подсистемы такой же структуры. Формализм должен обеспечивать спецификацию комплекса моделей соответствующей структуры. Он должен обладать высокой выразительной способностью, т. е. быть пригодным для удобного описания широкого класса динамических систем и должен допускать эффективную программную реализацию.

Цель достигается за счет модификации известного ситуационно-событийного формализма спецификации процессов автономной системы. Данный формализм основан на использовании математической модели гибридного автомата и представлении его функций перехода с помощью трансформационных правил. Условные части этих правил представляют собой конъюнкции логических состояний процессов, которые интерпретируются как динамические ситуации. Модификация формализма состоит в следующем. Расширяется множество переменных модели. Вводятся логические переменные состояния подмоделей данной модели, которые могут использоваться для формирования ситуаций этой модели и ее подмоделей. Вводятся множества выходных логических и вещественных переменных, которые являются подмножествами соответствующих переменных состояния данной модели. Выходные переменные могут использоваться в трансформационных правилах моделей одного с данной моделью уровня и в модели более высокого уровня. Таким образом обеспечивается связь между всеми моделями комплекса.

Включение в описание модели элементов такого же типа как сама модель превращает это описание в рекурсивное описание, позволяющее описывать комплекс моделей. Программные объекты, имеющие такие описания, эффективно реализуются с помощью механизмов рекурсии, входящих в состав компиляторов большинства универсальных языков программирования. Структура реверсивных алгоритмов показана на примерах алгоритмов создания комплекса моделей и обновления состояний моделей. Экспериментальная проверка проведена путем разработки на основе предложенного формализма трехуровневого семимодельного комплекса моделирования системы координации процессов в двух автоматических системах позиционирования ролика на плоскости.

Малая трудоемкость разработки комплекса и удобство проведения модельных экспериментов подтвердили ожидаемую высокую эффективность формализма. Она определяется возможностью неоднократного использования моделей одних и тех же подсистем и удобством интерпретации результатов моделирования, а также высокой эффективностью программной реализации.

SUMMARY

Shpakov V.M. **Interacting dynamic Systems Model Complex Specification.**

The article is devoted to working out of theoretic base for development of computer tools for collection of interacting processes realization. The goal of the work is development of formalism enabling an unambiguous non-contradictory description of complex dynamic systems, which have holonic structure. Such systems can contain subsystems of the same structure. The formalism must enable specification of the same structure model complex. It must have rather high expressive ability that is it must be suitable for convenient depiction of very different dynamic systems. It must also be suitable for an effective program realization.

The goal is achieved at the expense of the known situation-event formalism for a freestanding system processes specification. The formalism is based on a use of a mathematical model of a hybrid automaton, and on the automaton transition function representation by means of transformation rules. Conditional parts of these rules are conjunctions of processes logic states. The conjunctions are naturally interpreted as dynamic situations. The formalism modification consists in the following. The set of model variables is expanded. Logic variables of the model submodels states are introduced. These states may be used for composing the model and its submodel situations. New sets of logic and real output variables are introduced. The sets are subsets of the model proper states variables. The output variables may be used in composing transformation rules of the same and upper level models. In that way connections between all complex models are provided. Introduction into the model description elements of the same type as the model itself turns the description into recursive one. Such a description enables to describe the model complex. Program objects having such descriptions may be effectively realized with help of a recursive mechanism, compilers of most program languages having such mechanisms. A recursive algorithm structure is demonstrated by the examples of a model complex construction algorithm and an algorithm of models states updating. An experimental test was done by means of development on base of the formalism proposed a three-level seven-model complex for processes coordination system simulation. The system coordinates processes in two automatic systems of a roll on a plane positioning. A small labor intensiveness of complex development and a convenience of model experiment conducting have proved the expected high effectiveness of the formalism. The effectiveness is produced with capability of repeated use of the same subsystem models and with convenience of simulating results interpreting. Then it is provided with a high effectiveness of the model program realization.