

КОМПЬЮТЕРНАЯ РЕАЛИЗАЦИЯ ПРОЦЕССОВ НА ОСНОВЕ ИХ ЛОГИКО-ДИНАМИЧЕСКОЙ СПЕЦИФИКАЦИИ

В. М. Шпаков^а, канд. техн. наук, доцент

^аСанкт-Петербургский институт информатики и автоматизации РАН, Санкт-Петербург, РФ

Постановка проблемы: потребность в реализации процессов существует при компьютерном моделировании динамических систем, при разработке систем управления. Важное значение имеют трудоемкости разработки спецификации и программной реализации модели системы на ее основе, а также эффективность этой реализации. Кроме того, важны выразительность и понятность (прозрачность) спецификации для специалистов и определяемые ею возможности по модификации модели и удобству управления процессом моделирования. Логико-динамическая спецификация процессов пока не получила широкого распространения, ее свойства не известны широким кругам разработчиков моделей. **Цель:** экспериментальное исследование технологии разработки имитационных моделей динамических систем, основанной на использовании логико-динамической спецификации процессов; оценка трудоемкости разработки и выразительных возможностей спецификаций, а также качества получаемых моделей. **Результаты:** разработаны методики логико-динамической спецификации моделей систем, исходные знания о функционировании которых представлены в виде физических моделей, структурных схем элементарных процессов, систем дифференциальных уравнений. Свойства спецификаций иллюстрируются на примерах разработки трех моделей процессов. Приведенные спецификации и проведенное компьютерное моделирование процессов позволяет сделать вывод о широких выразительных возможностях логико-динамической спецификации процессов, пригодности ее для эффективной программной реализации процессов и об удобстве валидации и модификации получаемых на ее основе моделей. **Практическая значимость:** использование логико-динамических спецификаций процессов позволит существенно упростить разработку моделей сложных динамических систем, особенно в тех случаях, когда исходная информация о динамике отдельных частей системы представлена в различных формах. Благодаря прозрачности спецификаций модели могут быть эффективно использованы в образовательных целях.

Ключевые слова — имитационное моделирование, логико-динамические системы, основанное на правилах программирование.

Введение

Функционирование динамических систем определяется происходящими в них процессами. По характеру динамики различают процессы непрерывные, дискретно-событийные и гибридные. В гибридных процессах наряду с непрерывными изменениями состояний происходят дискретно-событийные изменения как состояний, так и динамики. Очевидно, непрерывные и дискретно-событийные процессы являются частными случаями гибридных процессов. К спецификациям процессов, помимо полноты и непротиворечивости, предъявляются требования выразительности и пригодности для эффективной программной реализации. Под выразительностью понимается возможность спецификации широкого класса процессов. Кроме того, важным свойством является прозрачность спецификации, которая характеризуется легкостью понимания спецификации специалистами в проблемной области.

Развитие процессов определяется законами соответствующих динамик и логикой функционирования конкретных динамических систем. Для спецификации дискретно-событийных и гибридных процессов используются модели конечного и гибридного автоматов. Рассматриваемый подход к спецификации процессов различной динамики основан на достаточно естественном объ-

единении вычислительных моделей гибридного и конечного автоматов. При этом дискретные состояния обоих видов процессов и режимы гибридных процессов представляются функциями логических переменных.

Наибольший интерес представляют компьютерные средства, обеспечивающие создание исполняемых спецификаций [1]. Модели, полученные на основе таких спецификаций, не требуют дополнительных преобразований для своего запуска на исполнение. Они могут запускаться непосредственно в среде разработки, что удобно, так как это позволяет реализовать итеративный подход к разработке моделей: после внесения дополнений и изменений можно сразу запустить модель на исполнение и оценить эффект от сделанных изменений. При необходимости корректировки полученной модели делается очередной шаг итерации.

Для решения задач анализа дискретно-событийных и гибридных динамических систем было предложено использовать транзитивные модели протекающих в системах процессов [2]. Эти модели непосредственно определяют переход (transition) системы из текущего состояния в следующее и позволяют представить процесс непосредственно в виде последовательности или траектории состояний. Для создания исполняемых спецификаций процессов используются транзитивные модели представления машины конеч-

ных состояний и гибридного автомата, которые непосредственно задают отношение следования между входным воздействием и текущим состоянием (режимом) с одной стороны и новым следующим состоянием (режимом) — с другой [3].

Логико-динамическая спецификация взаимодействующих процессов

Текущее состояние 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 логических воздействий (команд) V и непрерывных вещественных воздействий 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 — множество непрерывных переменных состояния.

Для спецификации процессов необходимо определить 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 диапазонов от непрерывных состояний процессов.

Спецификация процессов сводится к заданию конкретных способов определения и вычисления указанных выше функций перехода. Эти способы существенным образом влияют на качество спецификации.

Функция γ является логической по определению, ее реализация связана с вычислением неравенств следующего вида:

$$(x_{j_1} \geq (a_k + x_{j_2})) \wedge (x_{j_3} \leq (b_k + x_{j_4})) \rightarrow g_k, \quad (3)$$

где $x_{j_1}, \dots, x_{j_4} \in X$, $g \in G$, a_k и b_k — границы диапазона.

Функцию перехода σ задают с помощью совокупности продукционных правил вида «Усло-

вие \rightarrow Действие» [4]. В качестве условия удобно использовать элементарные конъюнкции логических переменных, а в качестве действия — присвоение требуемых значений определенным дискретным состояниям и режимам процессов. Такие конъюнкции интуитивно понятным образом могут интерпретироваться как логико-динамические ситуации. Ситуацию можно определить с помощью следующих двух утверждений:

1) любая логическая переменная (воздействие, состояние, режим, предикат) или ее отрицание является динамической ситуацией;

2) если S_i и S_j — динамические ситуации, то $S_i \wedge S_j$ также является динамической ситуацией.

С учетом сказанного функция σ может быть представлена с помощью совокупности правил следующего вида:

$$S_j \rightarrow q'_{j_1}, \dots, q'_{j_i}, \dots, q'_{j_m}, \quad (4)$$

где S_j — текущая ситуация; $q'_{j_i} \in Q \times \{False, True\}$, $m = 1, \dots, |Q|$. Штрихи в данном случае обозначают сдвиг по времени между возникновением ситуации и изменением значений состояний процессов.

Для задания функции перехода δ необходимо определить алгоритмы вычисления транзитивных отношений между непрерывными состояниями для каждого режима. Это может быть сделано на основе использования алгоритмов и процедур реализации элементарных динамических процессов, элементарных функций и арифметических операций [5]. Основными элементарными процедурами являются интегрирующие, дифференцирующие, реализующие аperiodические и колебательные звенья, а также звенья чистого запаздывания.

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' — следующее состояние процесса. 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 задания зависимости непрерывных состояний от логики развития гибридного процесса алгоритмы 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 — произвольная переменная процесса.

Таким образом, процессы могут быть специфицированы путем задания множеств логических и вещественных состояний (1), (2) и формирования совокупности трансформационных правил (3)–(5).

О реализации логико-динамического подхода к спецификации взаимодействующих процессов различной динамики

Компьютерная реализация процессов, специфицированных трансформационными правилами (3)–(5), производится путем использования интерпретатора этих правил. Интерпретатор представляет собой программный продукт, который может быть легко реализован с помощью компьютерных средств, использующих алгоритмические языки программирования, например Delphi. Архитектура и алгоритмы функционирования интерпретатора достаточно просты. Множества переменных состояния процессов (X и W) в интерпретаторе представляются массивами записей, каждая из которых содержит имя переменной и ее значение. Все правила реализуются с помощью условных операторов «if...then...». Обе части оператора содержат записи об использованных в данном правиле переменных. Основу интерпретатора составляет исполняющая процедура, которая в цикле сканирует списки правил. Алгоритм обработки правил вычисляет значение условной части правила. Если это значение равно *True*, то запускается алгоритм выполнения исполнительной части правила. В случае правила (4) логическим переменным из исполнительной части правила присваиваются указанные значения. При этом, естественно, могут измениться значения ситуаций, в определения которых эти переменные входят. В случае правила (5) запускается указанная в исполнительной части правила процедура вычисления транзитивного отношения, и найденное новое значение состояния непрерывного процесса присваивается соответствующей переменной.

На каждом шаге цикла обновления состояний процессов производится продвижение времени на интервал Δt . Возможны два режима продвижения времени. В режиме реального времени измеряется длительность каждого цикла, и время увеличивается на измеренное значение. В этом случае быстроедействие соответствует реальным процессам, а точность зависит от производительности компьютера и размерности модели. В режиме модельного времени время продвигается после каждого цикла на заданную пользователем величину. При этом точность определяется величиной заданного приращения времени, а быстроедействие зависит от производительности компьютера

и размерности модели. Использование современных персональных компьютеров для реализации моделей, содержащих тысячу правил, позволяет обеспечить величину приращения реального времени, не превышающую нескольких десятков микросекунд, что обеспечивает точность, приемлемую для большинства приложений.

Экспериментальные исследования описываемой технологии проводятся в СПИИРАН с помощью разработанного исследовательского прототипа компьютерной среды EnviCon [6], реализующей описываемый подход. Среда имеет достаточно наглядный интерфейс представления трансформационных правил и средства визуализации процессов. Рассматриваемые ниже примеры спецификации процессов будут, как правило, представлены в форматах редакторов этой среды.

Физическая модель движения тела вращения по наклонной плоскости

Специфицируем модель процессов, происходящих при движении тела вращения по наклонной плоскости. Для определенности будем считать, что телом вращения является шарик радиуса R . Шарик движется под влиянием скатывающей силы F , приложенной к центру и являющейся тангенциальной составляющей веса P : $F = P \sin \varphi$, где φ — угол наклона плоскости. Нормальная составляющая веса $F_n = P \cos \varphi$ компенсируется реакцией опоры. Сила трения $F_{тр}$ приложена к точке на поверхности шарика, касающейся склона. Взаимодействие этих сил определяет силу, вызывающую движение центра масс:

$$F_{дв} = F - F_{тр}, \quad (6)$$

и момент

$$M = F_{тр} R, \quad (7)$$

который вызывает вращение шарика вокруг центра [7]. Таким образом, движение шарика в модели представляется суммой прямолинейного движения центра шарика вдоль плоскости и вращения шарика вокруг центра.

Рассмотрим вначале движение шарика из состояния покоя. Ускорение $a_{дв}$ и скорость $V_{дв}$ центра шарика определяются силой $F_{дв}$ и массой шарика m : $a_{дв} = F_{дв} / m$, $V_{дв} = \int a_{дв} dt$. Угловое ускорение ε и угловая скорость вращения шарика ω вокруг центра определяются вращающим моментом M и моментом инерции (МИ) шарика относительно центра J_0 : $\varepsilon = M / J_0$, $\omega = \int \varepsilon dt$. Скорость точки шарика, касающейся плоскости, во вращательном движении $V_{вр} = \omega R$. Если трение отсутствует, то отсутствует и вращающий момент. Шарик скользит по плоскости без вращения. При наличии трения шарик вращается и скорость вращения точки касания плоскости противополо-

ложна скорости движения. Если трение маленькое, то скорость движения центра шарика $V_{дв}$ по модулю больше скорости $V_{вр}$ вращательного движения. В этом случае шарик вращается и скользит со скоростью скольжения $V_{ск} = V_{дв} - |V_{вр}|$. При увеличении трения скорость движения центра уменьшается, а скорость во вращательном движении увеличивается. Если эти скорости равны, то шарик катится по плоскости без скольжения. Дальнейшее увеличение трения не вызывает увеличения вращающей силы и вращающего момента, не приводит к изменению движения шарика. Это объясняется тем, что сила трения является силой реакции, она не может совершать работу, а может только расходовать энергию. Сила трения направлена против скорости скольжения. Если бы при увеличении трения скорость во вращательном движении стала больше скорости движения центра, то это привело бы к изменению направлений скорости скольжения и, соответственно, силы трения на противоположные значения. При этом изменился бы знак вращающего момента и увеличилась бы сила, движущая центр масс. В результате скорость вращения уменьшилась бы, а скорость движения центра увеличилась, что привело бы к восстановлению их равенства. Таким образом, свободное движение шарика по наклонной плоскости без скольжения является состоянием динамического равновесия. В этом случае силу, вызывающую вращение шарика, правильнее называть не силой трения, а тангенциальной составляющей силы реакции опоры. Сила трения зависит от коэффициента трения и нормальной составляющей веса:

$$F_{тр} = k_{тр} F_n, \quad (8)$$

где $k_{тр}$ — коэффициент трения. Эта зависимость выполняется в ситуации, когда имеется скольжение шарика, т. е. когда скорость движения центра масс $V_{дв}$ больше скорости движения наружной части шарика $V_{вр}$. При увеличении трения скорость скольжения уменьшается. Когда скольжение исчезает, сила трения достигает в данных условиях своего максимального значения $F_{тр}^*$ и не будет увеличиваться при увеличении коэффициента трения или нормальной силы. Значение этой силы определяется из условия равенства ускорения центра масс $a_{дв}$ тангенциальному ускорению $a_{вр}$ движения наружной точки шарика в его вращении. Сила $F_{тр}^*$ зависит от скатывающей силы F и от МИ шарика относительно центра масс J_0 . Определим эту зависимость. Ускорение $a_{вр} = \varepsilon R$, где ε — угловое ускорение. С учетом (7) имеем $a_{вр} = F_{тр} R^2 / J_0$. Ускорение центра масс $a_{дв} = (F - F_{тр}) / m$. Приравняв ускорения, находим

$$F_{тр}^* = \frac{F}{mR^2 / J_0 + 1}. \quad (9)$$

Для обруча, МИ которого $J_0 = mR^2$, имеем $F_{тр}^* = 0,5F$. Для шарика $F_{тр}^* = F/3,5$, так как для шарика $J_0 = 0,4mR^2$. Условием скольжения при движении тела вращения из состояния покоя является истинность предиката

$$|k_{тр} F_n| < |F_{тр}^*|. \quad (10)$$

При движении из состояния покоя силы и вращающий момент не меняются во время движения. Поэтому их можно вычислить перед началом моделирования как начальные условия. Начальные условия можно вычислять любым образом, а найденные значения состояний вводить вручную. Но удобнее вычислять начальные условия также с помощью правил. Правила для вычисления начальных условий обрабатываются не в цикле, а один раз перед началом моделирования. В данном случае достаточно задать угол наклона плоскости φ , массу тела m и его радиус R , ускорение свободного падения и одну из координат центра. Остальные начальные условия можно вычислить с помощью правил перед началом моделирования. Вычисляются: вес P , скатывающая сила F , нормальная сила F_n , МИ относительно центра J_0 , сила трения при скольжении (8) и сила трения при отсутствии скольжения (9), момент вращения M и угловое ускорение ε , сила движения центра масс $F_{дв}$ и ускорение центра $a_{дв}$. Также перед началом моделирования вычисляется значение предиката (10), в зависимости от которого силе трения $F_{тр}$ присваивается одно из двух возможных значений. При движении происходит изменение угловой скорости и угла поворота тела, скорости центра и значений его координат. Состояния этих процессов вычисляются в цикле обновления состояний.

Если в начальном состоянии тело не неподвижно, а имеет скорость движения вверх или вниз вдоль плоскости или (и) вращается с некоторой угловой скоростью в одну или в другую сторону, то в начальное время скольжение будет иметь место независимо от величины коэффициента трения и нормальной составляющей веса. Скорость скольжения равна сумме скорости движения центра и скорости точки на поверхности тела во вращательном движении и может быть направлена вверх или вниз вдоль плоскости. При наличии скольжения величина силы трения определяется формулой (8), а ее направление противоположно скорости скольжения. Для представления ситуации, когда имеется скольжение, введем логическую переменную скольжение. Эта переменная должна принимать значение *True* как в случае истинности предиката (10), так и в случае отличия от нуля абсолютного значения скорости скольжения, т. е. истинности предиката

$$|V_{ск}| > 0. \quad (11)$$

Значение ситуации скольжение формируется двумя правилами (4), реализующими дизъюнкцию (логическое ИЛИ) от этих двух предикатов:

$$|k_{\text{тр}} F_{\text{н}}| < |F_{\text{тр}}^*| \rightarrow \text{скольжение};$$

$$|V_{\text{ск}}| > 0 \rightarrow \text{скольжение}.$$

Для задания знака силе трения необходимо определять знак скорости скольжения $\text{sign}(V_{\text{ск}})$. Так как в этом случае сила трения может менять величину и знак, то ее вычисление необходимо производить в цикле обновления состояний.

На части экранной формы редактора правил среды EnviCon представлены правила для обновления состояний процессов (рис. 1). В зависимости от значения ситуации скольжение первое или второе правило определяет величину силы трения. Если значение ситуации должно быть *True*, то ее идентификатор представляется в правиле на белом фоне, иначе — на сером. Третье правило присваивает силе трения знак, противоположный скорости скольжения. Следующее правило вычисляет вращающий момент в соответствии с (7). Далее находится угловое ускорение, после чего интегрированием находятся угловая скорость и угол поворота тела. С помощью восьмого правила вычисляется скорость наружной точки тела во вращательном движении. Девятое правило по формуле (6) находит силу, движущую центр масс тела. Далее последовательно находятся ускорение центра, его скорость, составляющие скорости по координатным осям и значения координат. Шестнадцатое правило вычисляет скорость скольжения как сумму скорости центра и скорости наружной точки тела во вращательном движении.

Следующее правило находит абсолютное значение этой скорости, которое необходимо для вычисления значения предиката (11). Последнее правило определяет знак скорости скольжения, используемый в третьем правиле. В представленном виде модель адекватна системе при произвольных значениях и знаках начальных условий, таких как наклон плоскости, скорость движения центра и угловая скорость вращения вокруг центра.

Нетрудно видеть, что спецификация процессов с помощью трансформационных правил очень близка языку математики, используемому для описания физической модели системы.

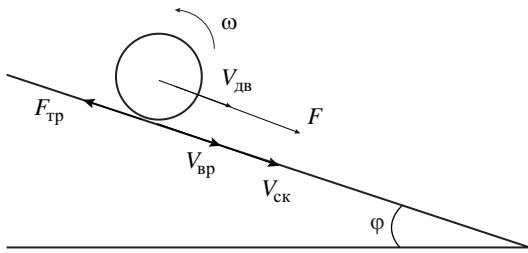
Валидация модели проводилась путем проверки выполнения закона сохранения энергии. С этой целью правила (см. рис. 1) дополнялись правилами для вычисления начальных и текущих значений потенциальной и кинетической энергий. При проведении экспериментов с моделью производились изменения знака и величины наклона плоскости, радиуса и МИ тела, а также изменения начальных условий: величин и знаков угловой скорости и скорости центра масс. Во всех экспериментах погрешность выполнения закона сохранения энергии была меньше 0,001 % при величине продвижения модельного времени $dt=2$ мкс.

Приведем результаты одного из экспериментов. Параметры системы: угол наклона $\varphi=-20^\circ$, $k_{\text{тр}}=0,3$, масса $m=1$, радиус $R=1,5$, МИ шарика $J_0=0,4mR^2$. Начальные условия: скорость движения центра масс $V_{\text{дв}}=0$, угловая скорость $\omega=20$. Схема движения шарика и отношения сил и скоростей показаны на рис. 2.

Графики процессов изменения скоростей движения шарика приведены на рис. 3. В данном

№	Переменная	Процедура	Коэффициент	Аргумент 1	Аргумент 2	Ситуация / событие	Имя параметра	значение	Имя параметра	значение	Приоритет
1	Сила трения Fтр	Пропорциональная	1.00	kтр * Fn		скольжение					1
2	Сила трения Fтр	Пропорциональная	1.00	Fтр		скольжение					1
3	Сила трения Fтр	Умножение	-1.00	Сила трения Fтр	Sign(Vскольж)	Ever True					1
4	момент вращения M	Умножение	-1.00	Сила трения Fтр	радиус R	Ever True					1
5	угловое ускорение	Деление	1.00	момент вращения M	МИ отн. центра J0	Ever True					1
6	угл. скор. Omega	Интеграл	-1.00	угловое ускорение		Ever True					1
7	ал. угол вращ.	Интеграл	-1.00	угл. скор. Omega		Ever True					1
8	Ввр вращения	Умножение	1.00	угл. скор. Omega	радиус R	Ever True					1
9	Сила Fдв ц.м.	Сумма / разность	1.00	сила танг. Fтнг	Сила трения Fтр	Ever True	2-ой коэффициент	-1.00			1
10	Ускорение ц. м.	Деление	1.00	Сила Fдв ц.м.	масса m	Ever True					1
11	скорость движ.	Интеграл	1.00	Ускорение ц. м.		Ever True					1
12	Vx	Умножение	1.00	скорость движ.	cos(phi)	Ever True					1
13	Vy	Умножение	1.00	скорость движ.	sin(phi)	Ever True					1
14	X координата	Интеграл	1.00	Vx		Ever True					1
15	Y координата	Интеграл	1.00	Vy		Ever True					1
16	Vск скольжения	Сумма / разность	1.00	скорость движ.	Ввр вращения	Ever True	2-ой коэффициент	-1.00			1
17	Abs(Vскольж)	Абсолютная	1.00	Vск скольжения		Ever True					1
18	Sign(Vскольж)	Реле	1.00	Vск скольжения		Ever True	Верх. ограничение	1.00	Нижн. ограничение	-1.00	1

■ Рис. 1. Правила трансформации состояний процессов при движении шарика по наклонной плоскости



■ Рис. 2. Отношения сил и скоростей в эксперименте



■ Рис. 3. Процессы изменения скоростей и силы трения при движении шарика

случае в начале движения скорость вращения $V_{вр}$ положительна, сила трения $F_{тр}$ — отрицательна. Коэффициент трения мал, и сила трения $F_{тр}$ меньше скатывающей силы F . Поэтому шарик движется вниз по склону в положительном направлении со скоростью $V_{дв}$, медленно возрастающей. Скорость вращения $V_{вр}$ достаточно быстро уменьшается. Скорость скольжения $V_{ск}$ равна сумме этих скоростей и тоже уменьшается. Когда скорость скольжения становится равной скорости движения, скорость вращения меняет знак и растет по величине. Когда скорость вращения сравнивается по величине со скоростью движения, скорость скольжения обращается в ноль. В этот момент происходит скачкообразное уменьшение силы трения, после чего $V_{дв}$ и $V_{вр}$ увеличивают скорости роста, оставаясь равными по величине и противоположными по знаку. Скорость скольжения остается равной нулю.

Спецификация процессов, заданных структурной схемой элементарных процессов

Одним из способов спецификации динамики систем является использование передаточных функций элементарных динамических звеньев и структурных схем их соединения для представления сложных динамических систем [8]. При этом

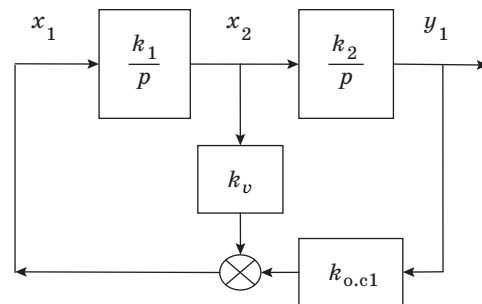
используются последовательные и параллельные, прямые и обратные соединения элементарных звеньев. Такой подход находит применение при разработке систем управления. Процессы, заданные структурными схемами соединений элементарных динамических звеньев, могут быть специфицированы и реализованы с помощью трансформационных правил (5). Особенности формирования трансформационных правил рассмотрим на примере спецификации процессов, заданных структурной схемой (рис. 4).

Схема состоит из двух интеграторов, охваченных общей отрицательной обратной связью и отрицательной обратной связью по скорости. Эта схема специфицирует колебательный процесс. Как видно из схемы, реализация специфицируемых ею процессов может быть произведена путем соединения интегрирующих, суммирующих и пропорциональных звеньев.

При спецификации процессов с помощью правил каждый элемент схемы представляется правилом, реализующим соответствующую процедуру. Соединение процедур производится за счет того, что выходная переменная одной процедуры используется в качестве аргумента другой. На рис. 5 приведены правила, соответствующие схеме рис. 4. Имена переменных в правилах совпадают с именами, использованными на схеме.

Первое правило формирует процесс обратной связи по скорости с $k_v=0,10$, второе — процесс основной обратной связи с $k_{o.c1}=1$. Третье правило осуществляет суммирование обеих обратных связей и инвертирование суммарного значения, тем самым вычисляет значение x_1 на входе первого интегратора. Четвертое правило производит интегрирование с $k_1=20$ и определяет значение x_2 на входе второго интегратора. Пятое правило осуществляет интегрирование с $k_2=20$ и вычисляет значение y_1 состояния процесса на выходе схемы.

Процессы, определяемые схемой, описываются обыкновенными линейными дифференциальными уравнениями с постоянными коэффициентами, аналитические решения которых находятся достаточно просто. Применяя формулу для замыкания системы отрицательной обратной



■ Рис. 4. Структурная схема колебательного звена

Редактор правил трансформации непрерывных состояний, Колебат. звено

Иск. ПМ вык. **Логические** Возд. Входы Состояния Предикаты ПМ сост. из модели Колебат. звено OK

№	Переменная	Процедура	Коэффициент	Аргумент 1	Аргумент 2	Ситуация / событие	имя параметра	значение	имя параметра	значение	Приоритет
1	Kv * X2	Пропорциональная	0.10	X2		Ever True					1
2	Koc * Y1	Пропорциональная	1.00	Y1		Ever True					1
3	X1	Сумма / разность	-1.00	Kv * X2	Koc * Y1	Ever True	2-ой коэффициент	1.00			1
4	X2	Интеграл	20.00	X1		Ever True					1
5	Y1	Интеграл	20.00	X2		Ever True					1

Рис. 5. Правила для реализации процессов, заданных структурной схемой

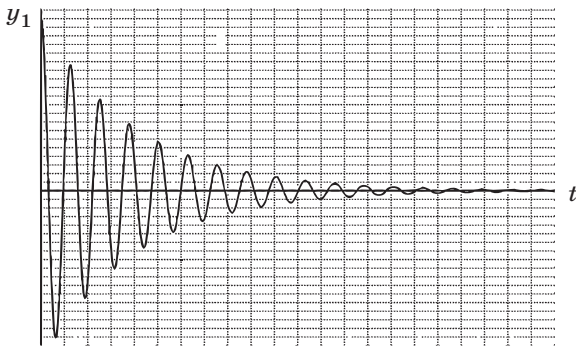


Рис. 6. График процесса $y_1(t)$ на выходе схемы рис. 4

связью к обоим контурам, находим его передаточную функцию в следующем виде:

$$\Phi_1(p) = \frac{k_1 k_2}{p^2 + k_1 k_v p + k_1 k_2 k_{o.c1}}$$

Вид переходного процесса в данном звене определяется корнями знаменателя $\lambda_{1,2}$ приведенной передаточной функции. Для выбранных значений коэффициентов $\lambda_{1,2} = -1,0 \pm j19,975$. Соответствующий найденным значениям корней переходный процесс представляет собой колебательный процесс с круговой частотой $\omega_1 = 19,975$ (период колебаний $T = 0,314$ с) и амплитудой, изменяющейся по экспоненте $A = A_0 e^\alpha$ с показателем $\alpha = -t$. На рис. 6 представлен процесс изменения y_1 , реализованный с помощью правил (см. рис. 5) для начального значения $y_1(0) = 200$ и нулевых начальных значений остальных переменных. Масштаб времени равен 100, т. е. 1 секунда равна 100 делениям. Измерение параметров этого процесса показало, что они хорошо совпадают с расчетными значениями.

Спецификация процессов, заданных обыкновенными дифференциальными уравнениями

К настоящему времени разработано и исследовано значительное количество обыкновенных дифференциальных уравнений, описывающих процессы в различных динамических системах, представляющих интерес для науки и техники.

Компьютерная реализация этих процессов также, очевидно, может представлять интерес. Она достаточно просто может быть произведена с помощью трансформационных правил.

Рассмотрим вначале самое простое дифференциальное уравнение $dy/dt = y$. Нетрудно видеть, что это уравнение удовлетворяет экспоненциальная функция $y = e^t$, производная от которой равна самой функции. Очевидно также, что в данном случае справедливо интегральное уравнение $y = \int y dt$. В соответствии с этим уравнением данный процесс может быть реализован с помощью одного трансформационного правила, содержащего транзитивную процедуру интегрирования (рис. 7).

График процесса, реализованного с помощью этого правила при начальном значении $y(0) = 1$, представлен на рис. 8. На графике масштаб по оси ординат равен 1, по оси абсцисс — 100. График соответствует функции $y = e^t$, например $e^3 = 20,086$.

Аналогичным образом с помощью трансформационных правил можно реализовать процессы, заданные системами обыкновенных дифференциальных уравнений. Покажем это на примере реализации процессов, заданных системой

Редактор правил трансформации непрерывных состояний, Простейшее диф. ур

Иск. ПМ вык. **Логические** Возд. Входы Состояния Предикаты ПМ сост.

№	Переменная	Процедура	Коэффициент	Аргумент 1	Аргумент 2	Ситуация / событие
1	Y	Интеграл	1.00	Y		Ever True

Рис. 7. Правило для реализации процесса, заданного уравнением $dy/dt = y$

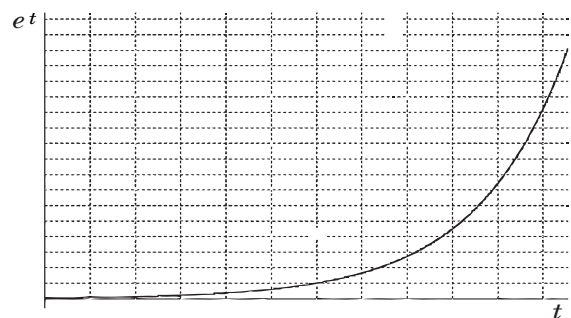


Рис. 8. Реализация процесса e^t , специфицированного правилом (см. рис. 7)

Редактор правил трансформации непрерывных состояний, Ур-ие Лоренца

Исх. ПМ вых. **Логические** Возд. Входы Состояния Предикаты ПМ сост. из модели Ур-ие Лоренца

№	Переменная	Процедура	Коэффициент	Аргумент 1	Аргумент 2	Ситуация / событие	Имя параметра	значение
1	$x * z$	Умножение	1.00	x	z	Ever True		
2	$-(y + x * z)$	Сумма / разность	-1.00	y	$x * z$	Ever True	2-ой коэффициент	1.00
3	$x * y$	Умножение	1.00	x	y	Ever True		
4	dx/dt	Сумма / разность	10.00	y	x	Ever True	2-ой коэффициент	-1.00
5	dy/dt	Сумма / разность	1.00	$-(y + x * z)$	x	Ever True	2-ой коэффициент	28.00
6	dz/dt	Сумма / разность	1.00	$x * y$	z	Ever True	2-ой коэффициент	-2.6666667
7	x	Интеграл	1.00	dx/dt		Ever True		
8	y	Интеграл	1.00	dy/dt		Ever True		
9	z	Интеграл	1.00	dz/dt		Ever True		

■ Рис. 9. Правила для реализации процессов, заданных системой Лоренца

Лоренца, представляющей собой систему трех уравнений первого порядка:

$$\frac{dx}{dt} = \sigma(y - x); \quad \frac{dy}{dt} = \rho x - y - xz; \quad \frac{dz}{dt} = xy - \beta z.$$

Здесь x , y и z — переменные системы; t — время, а σ , ρ , β — параметры системы. Эта система уравнений была разработана Эдвардом Лоренцом в 1963 г. и использовалась им в качестве математической модели процессов атмосферной конвекции. При разработке трансформационных правил, соответствующих системе уравнений, вначале формируются правила для вычисления переменных, представляющих правые части уравнений. Затем эти переменные используются в качестве подынтегральных функций для реализации переменных системы. На рис. 9 приведены правила реализации процессов системы Лоренца для следующих значений параметров: $\sigma = 10$, $\rho = 28$, $\beta = 2,666667$.

Первые три правила выполняют вспомогательные арифметические операции с переменными системы. Следующие три правила вычисляют производные от переменных системы, соответствующие трем правым частям уравнений, а три последних правила их интегрируют и вычисляют переменные системы.

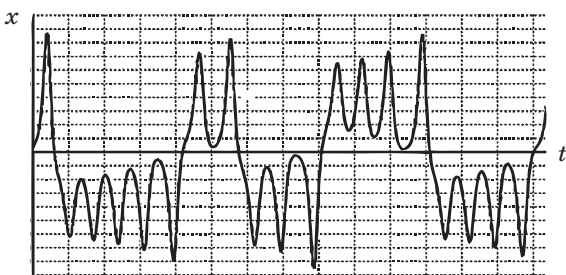
При запуске этих правил на обработку интерпретатором реализуются процессы изменения

всех переменных системы. Визуализация изменения x представлена на рис. 10. Масштабный коэффициент визуализации по оси ординат равен 10. Были установлены следующие начальные значения переменных: $x = 1$, $y = 1$, $z = 10$. Приведенный график процесса совпадает с графиком (см. Figure 4.9 в работе [9]) решения данной системы уравнений, полученного с помощью MatLab при тех же начальных условиях.

Заключение

Исходными спецификациями процессов являются физические модели динамических систем, которые в большинстве случаев представляются на языке математического анализа, элементарной математики и математической логики. Эти модели полностью определяют разработку логико-динамических спецификаций процессов, которые в свою очередь могут быть использованы для компьютерной реализации процессов. При этом трудоемкости разработок как спецификаций, так и программных средств их обработки небольшие. Как видно из приведенных примеров, язык спецификаций близок к языку математических описаний физических моделей. Это обеспечивает прозрачность спецификаций и упрощает валидацию и модификацию моделей.

Рассмотренный подход к реализации процессов был подвергнут всестороннему экспериментальному исследованию. На его основе был создан прототип компьютерной среды (EnviCon) для разработки исполняемых спецификаций взаимодействующих гибридных процессов. С помощью этой среды были разработаны модели нескольких типовых тестовых систем (стабилизация перевернутого маятника, позиционирование ролика на плоскости и др.) и модели трех больших промышленных установок [10]. Полученный опыт позволяет утверждать, что логико-динамические спецификации обладают высокими выразительными возможностями и могут быть использованы для создания эффективных программных средств реализации процессов.



■ Рис. 10. График процесса $x(t)$ системы уравнений Лоренца

Литература

1. Шпаков В. М. Исполняемые спецификации транзитивных моделей технологических процессов // Мехатроника, автоматизация, управление. 2004. № 3. С. 38–45.
2. Alur R., Henzinger T. A., Lafferriere G., Pappas G. J. Discrete Abstractions of Hybrid Systems // Proc. of the IEEE 88. 2000. P. 971–984.
3. Lacoste-Julien S. Hybrid Systems Modeling. — McGill University, School of Computer Science, Technical Report. August 2002. <http://msdl.cs.mcgill.ca/people/slacoste/research/report/SummerReport.html> (дата обращения: 30.06.2016).
4. Brownston L., Farrell R., Kant E., Martin N. Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming: Addison-Wesley Series in Artificial Intelligence. — Addison-Wesley, 1985. — 471 p.
5. Howe R. M. Analog Computer Fundamentals // IEEE Contr. Syst. Mag. 2005. Vol. 25. N 3. P. 29–36.
6. Шпаков В. М. Прототип среды моделирования структурированных совокупностей взаимодействующих процессов // Имитационное моделирование. Теория и практика: сб. докл. Второй Всерос. конф., Санкт-Петербург, 19–21 октября 2005. Т. II. С. 292–295.
7. Старжинский В. М. Теоретическая механика. — М.: Наука, 1980. — 464 с.
8. Бессекерский В. А., Попов Е. П. Теория систем автоматического регулирования. — М.: Наука, 1966. — 992 с.
9. Edward R. Scheinerman. Invitation to Dynamical Systems. — Prentice Hall, 1995. — 373 p.
10. Буткевич И. К., Рыдник Е. А., Шпаков В. М. Использование среды EnviCon для моделирования производства жидкого гелия // Тр. СПИИРАН. 2007. Вып. 5. С. 328–332.

UDC 681.5.017

doi:10.15217/issn1684-8853.2016.4.68

Computer Implementation of Processes on the Base of Their Logic-Dynamic Specification

Shpakov V. M.^a, PhD, Tech. Associate Professor, vlad@iias.spb.su^aSaint-Petersburg Institute for Informatics and Automation of RAS, 39, 14 Line, V. O., 199178, Saint-Petersburg, Russian Federation

Introduction: The need for process implementation exists in computer modeling of dynamic systems when developing control systems. Essential are the complexity of developing a specification and developing its software implementation based on the system model, as well as the effectiveness of this implementation. Besides, the expressiveness and clarity (transparency) of the specification are important, as well as specification-determined abilities to modify the model and make the simulation more convenient. A logic-dynamic specification of processes is not widely used yet, and its properties are not known to a wide circle of modelers. **Purpose:** Our goal is an experimental study of the technology for developing dynamic system simulation models, based on logic-dynamic specification of the processes, evaluation of complexity and expressive possibilities of the specifications, and the quality of the models produced. **Results:** Means of logic-dynamic specification of system models are worked out. The sources of knowledge about the functioning of the systems are presented in the form of physical models, structural diagrams of elementary processes, or systems of differential equations. The specification properties are illustrated by examples of three models developed. The given specifications and computer simulation of the processes suggest broad expressive possibilities of logic-dynamic specification, its suitability for effective software implementation of processes, and the convenience of validation and modification of the models obtained on its base. **Practical relevance:** Logic-dynamic specification of processes can greatly simplify the development of computer models for complex dynamic systems, especially when the initial information about the dynamics of individual parts of the system is presented in different formats. Due to the transparency of the specifications, the models can be efficiently used for educational purposes.

Keywords — Simulation Modelling, Logic-Dynamic Systems, Rule-Based Programming.

References

1. Shpakov V. M. Executable Specifications of Production Processes Transitive Models. *Mekhatronika, avtomatizatsiia, upravlenie*, 2004, no. 3, pp. 38–45 (In Russian).
2. Alur R., Henzinger T. A., Lafferriere G., Pappas G. J. Discrete Abstractions of Hybrid Systems. *Proc. of the IEEE 88*, 2000, pp. 971–984.
3. Lacoste-Julien S. *Hybrid Systems Modeling*. McGill University, School of Computer Science, Technical Report. August 2002. Available at: <http://msdl.cs.mcgill.ca/people/slacoste/research/report/SummerReport.html> (accessed 30 May 2016).
4. Brownston L., Farrell R., Kant E., Martin N. *Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming*. Addison-Wesley Series in Artificial Intelligence. Addison-Wesley, 1985. 471 p.
5. Howe R. M. Analog Computer Fundamentals. *IEEE Contr. Syst. Mag.*, 2005, vol. 25, no. 3, pp. 29–36.
6. Shpakov V. M. An Environment Prototype for Modeling of Structured Sets of Cooperating Processes. *Sbornik dokladov vtoroi Vcerossiiskoi konferentsii "Imitatsionnoe modelirovanie. Teoriya i praktika"* [Proc. 2nd All-Russian Conf. "Simulation. Theory and Practice"], Saint-Petersburg, 2005, vol. 2, pp. 292–295 (In Russian).
7. Starzhinskii V. M. *Teoreticheskaiia mekhanika* [Engineering Mechanics]. Moscow, Nauka Publ., 1980. 464 p. (In Russian).
8. Bessekerskiy V. A., Popov E. P. *Teoriia sistem avtomaticheskogo upravleniia* [The Theory of Automatic Control Systems]. Moscow, Nauka Publ., 1966. 992 p. (In Russian).
9. Edward R. Scheinerman. *Invitation to Dynamical Systems*. Prentice Hall, 1995. 373 p.
10. Butkevich I. K., Rydник E. A., Shpakov V. M. Using Environment EnviCon for Liquid Helium Production Simulation. *Trudy SPIIRAN*, 2007, vol. 5, pp. 328–332 (In Russian).