

Оценка и мониторинг проектов разработки высокотехнологичных изделий на примере микроэлектроники

А. А. Лебедев, А. П. Рыжов

Введение

На сегодняшний день в различных областях человеческой деятельности, будь то медицина, политология, маркетинг или менеджмент, возникают задачи оценки текущего состояния и дальнейшего развития некоторой проблемы или процесса на основе данных различной степени точности и достоверности. При решении таких задач экспертами возникает проблема субъективности их оценок, созданию автоматических средств препятствуют слабая формализованность задач и неточность входных данных.

Хорошим средством для решения подобного рода задач является технология информационного мониторинга [20]. Описываемая в настоящей работе система использует эту технологию для решения задачи оценки состояния и возможностей коллектива разработчиков электронного устройства (микрочипа).

В первой главе изложена предыстория создания системы и задачи, поставленные перед разработчиками. Вторая глава содержит общее описание технологии информационного мониторинга. Третья глава описывает основные аспекты создания системы. В четвёртой главе представлены некоторые теоретические результаты, гарантирующие устойчивость работы системы к неточностям входных данных.

1. Задача мониторинга процесса разработки электронного устройства

Одной из первых и важных проблем любого инновационного проекта является проблема оценки способности коллектива разработчиков выполнить проект в необходимое время с требуемым качеством при заданном ресурсном обеспечении [16, 17]. Примерами вопросов, которые возникают на этой начальной стадии, являются:

- Достаточно ли у коллектива разработчиков навыков и ресурсов для завершения проекта?
- Какие части проекта представляют наибольшие трудности?
- Каким образом различные вложения средств (например, обучение, закупка оборудования, программного обеспечения) повлияют на возможности коллектива?
- Как улучшить возможности коллектива разработчиков оптимальным образом при заданном ресурсном обеспечении?

В настоящее время на эти вопросы отвечают эксперты, менеджеры проекта, основываясь на своём личном опыте и интуиции. Цена ошибки на этом этапе — увеличение стоимости проекта, увеличение времени разработки изделия — вплоть до провала проекта. Поэтому любое повышение надёжности оценок и ответов на сформулированные вопросы является важным, иногда критичным, для любого серьёзного проекта.

В работе описывается прототип такой системы оценки и мониторинга проектов на примере разработки изделий микроэлектроники. В рамках разработки системы происходит формализация знаний экспертов и автоматизация процессов обработки информации о проектном коллективе для получения ответов на приведённые выше и другие вопросы. Отметим, что данная предметная область (микроэлектроника) не обладает какими-либо специфическими свойствами, удобными для разработки системы. Ее выбор — лишь удачное стечение обстоятельств. Дело в том, что компания Cadence Design Systems (<http://www.cadence.com>) — лидер рынка САПР (систем автоматизированного проектирования) микроэлектроники — имеет необходимый

опыт (модель «виртуальной САПР» Cadence VCAD) и потребность в такой системе как инструменте совершенствования услуг и сервисов VCAD. Описываемый ниже прототип разработан с привлечением экспертов компании и находится в опытной эксплуатации в компании.

Cadence VCAD

Кратко, идея Cadence VCAD [18] заключается в следующем. Так как на сегодняшний день процесс разработки электронного устройства чрезвычайно сложен и во многом слабо формализован, многие, даже крупные компании-разработчики могут не обладать достаточными ресурсами (опытными руководителями, компетентными в конкретной области разработчиками, программным и аппаратным обеспечением) для успешного выполнения проекта. В этом случае, недостающие ресурсы (как людские, так и аппаратные и программные) обеспечивает служба VCAD, «виртуально расширяя» отдел разработки компании. В связи с этим, естественно возникает вопрос оптимального, с точки зрения качества конечного результата и объёма затрат, привлечения и распределения ресурсов. Именно этот круг вопросов (оценка имеющихся ресурсов, привлечение дополнительных, мониторинг выполнения проекта) и решает описываемая система.

2. Технология информационного мониторинга

Технологической основой системы является технология информационного мониторинга [20], разработанная для анализа сложных, слабоформализованных проблем (процессов) на основе всей доступной информации, построения прогнозов их развития и выработки рекомендаций по управлению их развитием. Системы информационного мониторинга, разработанные на базе этой технологии, ориентированы на обработку информации, обладающей такими свойствами, как:

- разнородность информации: информация может иметь различный формат: числовые данные, лингвистические описания и т. д.;

- фрагментарность: информация чаще всего относится к какому-либо фрагменту проблемы, причем разные фрагменты могут быть по-разному «покрыты» информацией;
- разноуровневость: информация может относиться ко всей проблеме в целом, к некоторой ее части, к конкретному элементу проблемы;
- различная степень надежности: информация может содержать конкретные данные различной степени надежности, косвенные данные, результаты выводов на основе надежной информации или косвенные выводы;
- возможная противоречивость: данные из различных источников могут совпадать, слегка различаться или вообще противоречить друг другу.
- изменяемость во времени: проблема/процесс развивается во времени, поэтому и информация в разные моменты времени об одном и том же элементе проблемы может и должна различаться;
- возможная тенденциозность: информация отражает определенные интересы источника информации, поэтому может носить тенденциозный характер. В частном случае она может являться намеренной дезинформацией (например, для политических проблем или для проблем, связанных с конкуренцией).

Системы информационного мониторинга можно отнести к классу иерархических нечетких дискретных динамических систем. Теоретическую основу такого класса систем составляет теория автоматов, теория нечетких множеств, дискретная математика, методы анализа иерархий, которые были разработаны в работах Саати (Saaty, США) [11], Месаровича (Messarovich, США) [19], Заде (Zadeh, США) [2, 22], Яблонского С. В. [13], Кудрявцева В. Б. [3] (Россия) и других авторов. Системы, разработанные на базе этой технологии, позволяют иметь развивающуюся во времени модель проблемы на основе оценок аналитиков, подкрепленную ссылками на все информационные материалы, выбранные ими, с общими и частными оценками состояния проблемы и/или ее аспектов. Использование времени как параметра системы позволяет проводить как ретроспективные анализ, так

и строить прогнозы развития проблемы (отвечать на вопросы типа «Что будет, если...?»). В последнем случае возникает возможность выделения «критических путей» — таких элементов модели, малое изменение которых может вызвать значительные изменения в состоянии всей проблемы. Знание таких элементов имеет большое практическое значение и позволяет выявить «слабые места» в проблеме на текущий момент времени, разработать мероприятия по блокированию нежелательных ситуаций или провоцированию желательных, то есть в некоторой степени управлять развитием проблемы в интересах организации, ее отслеживающей. Эта технология зарекомендовала себя в задачах мониторинга международных договоренностей [21], экономики (например, мониторинг конкурентоспособности товаров — <http://www.intsys.msu.ru/invest/monitoring/competitiveness>), медицины (например, мониторинг развития атеросклеротических заболеваний населения России [1]) и других областях.

Первым и наиболее трудно формализуемым этапом разработки любой системы информационного мониторинга является разработка модели проблемы или процесса. Модель представляет собой ациклический граф, каждой вершине которого, соответствующей некоторому аспекту проблемы, сопоставлена лингвистическая переменная, причём значение вершины-«родителя» может как вычисляться на основе значений вершин-«потомков» и заданных операторов агрегирования информации, так и назначаться пользователем (рис. 1).

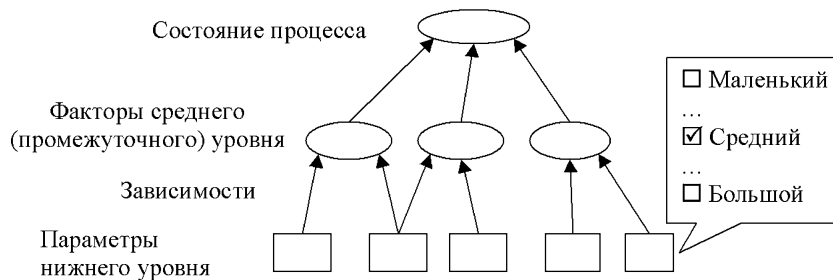


Рис. 1. Схема модели проблемы.

Предполагается, что эксперт описывает возможность реализации выделенных процессов в предметной области в виде лингвистических

значений. Субъективная степень удобства такого описания зависит от набора и состава таких лингвистических значений. Предполагается, что система отслеживает развитие проблемы, то есть изменение ее во времени. Предполагается также, что она интегрирует оценки различных экспертов. Это значит, что один объект может описываться разными экспертами. Поэтому желательно иметь гарантии того, что разные эксперты описывают один и тот же объект наиболее «единообразно». Исходя из сделанных замечаний мы можем сформулировать первую проблему следующим образом: можно ли, учитывая некоторые особенности восприятия человеком объектов реального мира и их описания, сформулировать правило выбора оптимального множества значений признаков, по которым описываются эти объекты? Возможны два критерия оптимальности: (1) под оптимальными понимаются такие множества значений, используя которые человек испытывает минимальную неопределенность при описании объектов; (2) если объект описывается некоторым количеством экспертов, то под оптимальными понимаются такие множества значений, которые обеспечивают минимальную степень рассогласования описаний. Эти вопросы подробно исследуются в [10]. Показано, что мы можем сформулировать методiku выбора оптимального множества значений качественных признаков. Более того, показано, что такая методика является устойчивой, то есть возможные при построении функций принадлежности естественные маленькие ошибки не оказывают существенного влияния на выбор оптимального множества значений. Множества, оптимальные по критериям (1) и (2), совпадают.

Технология информационного мониторинга предполагает хранение информационных материалов (или ссылок на них) и их лингвистических оценок в базе данных системы. В связи с этим возникает следующая проблема: можно ли определить показатели качества поиска информации в нечетких (лингвистических) базах данных и сформулировать правило выбора такого множества лингвистических значений, использование которого обеспечивало бы максимальные показатели качества поиска информации? Эта проблема подробно исследуется в [7]. Показано, что можно ввести показатели качества поиска информации в нечетких (лингвистических) базах данных и формализовать их. Показано, что возможно сформулировать методiku

выбора оптимального множества значений качественных признаков, которое обеспечивает максимальные показатели качества поиска информации. Более того, показано, что такая методика является устойчивой, то есть возможные при построении функций принадлежности естественные маленькие ошибки не оказывают существенного влияния на выбор оптимального множества значений.

Важной проблемой при разработке систем информационного мониторинга является задача выбора адекватных операторов агрегирования информации в модели проблемы. Эта задача возникает в силу того, что системы информационного мониторинга ориентированы на обработку разноуровневой фрагментарной информации. Это означает, что при вводе информации в систему пользователь может осуществлять привязку информационных материалов к узлам различных уровней иерархии в модели проблемы (и, соответственно, подтверждать/изменять их оценки). Это коренным образом отличает системы информационного мониторинга от аналогичных иерархических систем (например, поддержки принятия решений), в которых разрешается менять только оценки нижнего уровня иерархии. Указанное допущение позволяет использовать системы информационного мониторинга при решении значительно более широкого класса практических задач, однако платой за это является необходимость разработки соответствующей теории и создания инструментария выбора адекватных операторов агрегирования информации. Эта задача подробно исследуется в [8]. Таким образом, мы можем гарантировать оптимальность свойств систем информационного мониторинга с указанных точек зрения: описание объектов, поиск информации, агрегирование информации.

3. Применение технологии

Создание модели

Для разработанного приложения экспертами были выделены следующие факторы, определяющие способности проектного коллектива (Рис. 2):

1 уровень: способность завершить отдельные стадии процесса разработки (разработка, тестирование, отладка и т. д.)

2 уровень:

- навыки — различные для каждой стадии;
- оборудование — различное для каждой стадии (вычислительные ресурсы, программное обеспечение);
- инфраструктура — общая для всех стадий (доступ к литературе, работа сети и т. д.)

Дальнейшие уровни иерархии содержат атрибуты, более детально описывающие факторы второго уровня (например, для навыков — знакомство с конкретными программами, опыт участия в похожих проектах и т. п.) Модель содержит более 50 элементов и более 70 связей, объединенных в 5 уровней иерархии.

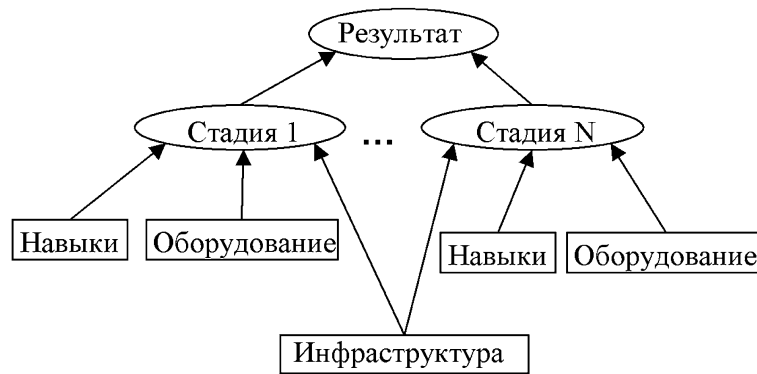


Рис. 2. Структура модели системы.

Каждой вершине графа необходимо сопоставить набор лингвистических значений — набор значений, которые может принимать соответствующий вершине атрибут. Проблема выбора оптимального набора описана выше и проанализирована в [10]. Кроме того, так как многие атрибуты не имеют численных значений (например, качество программного обеспечения, уровень навыка разработчика), необходимо разработать правила, позволяющие единообразно отображать ситуации реального мира в модели. Единообразие необходимо как для выбора операторов агрегирования экспертами, так и для введения

данных и получения результатов пользователем. Для этой же цели для всех вершин была введена виртуальная численная шкала оценок на отрезке $[0; 1]$, в которой и проводятся все вычисления. В случае наличия у атрибута собственной численной шкалы (например, объём памяти) выбиралось отображение этой шкалы на виртуальную. На виртуальной шкале лингвистические значения должны образовывать полное ортогональное семантическое пространство [10].

Выбор операторов агрегирования

Выбор операторов агрегирования происходит следующим образом. Оператор агрегирования — функция, определяющая значения вершины-«родителя» на основе значений вершин-«потомков». Так как большинство атрибутов и параметров системы допускают лишь лингвистические оценки, причём с «размытыми» границами (например, нет точной грани между высоким и средним уровнем навыка), то было решено разрабатывать операторы на основе нечёткого логического вывода [10]. В этом случае основным элементом оператора является набор правил, связывающих наборы значений параметров (вершин-«потомков») с соответствующим значением результата (вершины-«родителя») (например: если навык разработчика высокий и программное обеспечение хорошее, то качество результата высокое). Задание такого правила для каждой комбинации значений параметров эквивалентно заданию функции на n -мерном параллелепипеде:

$$F : E_{k_1} \times \dots \times E_{k_n} \rightarrow E_k,$$

где k_1, \dots, k_n, k — число лингвистических значений вершин-«потомков» и вершины-«родителя» соответственно.

Особенности предметной области вводят некоторые ограничения на допустимый вид функции. В частности:

- функция сохраняет ноль: $F(0, \dots, 0) = 0$;
- функция сохраняет максимальное значение: $F(k_1-1, \dots, k_n-1) = k - 1$;
- функция не убывает по всем аргументам;

- многие аргументы обладают свойством критичности:

$$F(a_1, \dots, a_{i-1}, 0, a_{i+1}, \dots, a_n) = 0 \text{ или даже}$$

$$F(a_1, \dots, a_{i-1}, x, a_{i+1}, \dots, a_n) \leq x \quad \forall a_i.$$

С учётом этих ограничений функция аппроксимируется линейной, после чего значения функции на некоторых наборах могут дополнительно изменяться. Опыт показал, что при правильном выборе наборов лингвистических значений эксперт может назвать результат для любой комбинации значений параметров.

Дополнительная настройка (обучение)

Используя только наборы правил, система может обрабатывать дискретные оценки. Если же возникают ситуации, которые эксперт не может охарактеризовать с помощью одного лингвистического значения и помещает их между соседними оценками (например, оборудование «среднее» со степенью 0.3 и «хорошее» со степенью 0.7), то в вычислениях должна быть учтена эта нечёткость. Кроме того, в нечётком случае возникает необходимость настройки (обучения) системы без изменения наборов правил.

Наиболее активно процедуры обучения используются при разработке нечетких контроллеров [10], то есть аппаратно реализованных систем правил нечеткого логического вывода. Нечеткие контроллеры в течение значительного времени используются во многих областях от бытовой техники до систем вооружений и показали свою простоту, надежность и эффективность. Обучение производится путем коррекции по определенным правилам функций принадлежности используемых нечетких переменных.

Непосредственное использование такого типа процедур при обучении нечетких иерархических систем не представляется естественным, так как приводит к разной формализации одних и тех же лингвистических значений в разных узлах модели. Поэтому предлагается в качестве параметра обучения использовать не функции принадлежности, а параметры t -норм и t -конорм.

Приведём схему вычисления на основе нечёткого логического вывода (в том виде, в каком она применяется в системе).

Пусть вычисляется функция $y = f(x^1, \dots, x^n)$, y принимает численные значения на виртуальной шкале $[0; 1]$ и лингвистические значения $T^Y = \{Y_i\}$, связанные с численными функциями принадлежности $\mu_i^Y(y)$ (рис. 3), аргументы принимают лингвистические значения $T^j = \{X_i^j\}$, значения аргументов определяются степенями принадлежности лингвистическим понятиям $\mu^j : T^j \rightarrow [0; 1]$.

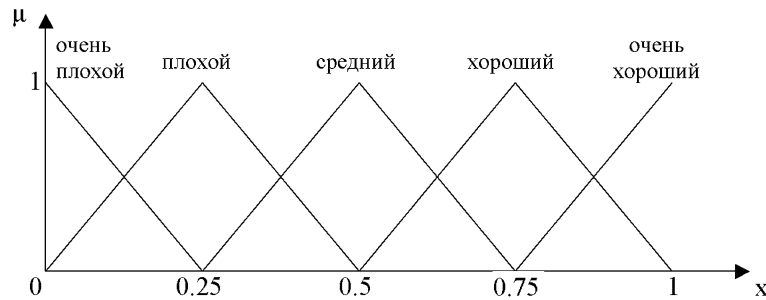


Рис. 3. Графики функций принадлежности лингвистических значений на виртуальной шкале (для 5 значений).

Задан набор правил $\{R_j\}$ вида

$$x^1 = X_j^1 \& \dots \& x^n = X_j^n \Rightarrow y = Y_j$$

для всех наборов лингвистических значений $(X^1, \dots, X^n) \in T^1 \times \dots \times T^n$.

Вычисление проводится в 3 этапа:

- 1) Вывод. Для каждого правила вычисляется функция

$$m_j(y) = \top_j(\mu^j(X_j^1), \dots, \mu^j(X_j^n), \mu_j^Y(y));$$

- 2) Композиция. Вычисляется «объединение» функций

$$M(y) = \perp_j m_j(y);$$

- 3) Дефазификация. Вычисляется значение y на виртуальной шкале

$$y = \frac{\int_0^1 yM(y) dy}{\int_0^1 M(y) dy}.$$

Функции $\top(\mu_1, \mu_2)$ и $\perp(\mu_1, \mu_2)$ (t -норма и t -конорма — обобщения операторов «И» и «ИЛИ» на нечёткий случай), определённые на квадрате $[0; 1]^2$, должны удовлетворять следующим свойствам:

- 1) ассоциативность;
- 2) коммутативность;
- 3) монотонное неубывание по обоим аргументам;
- 4) $\top(\mu_1, 0) = 0$, $\top(\mu_1, 1) = \mu_1$, $\perp(\mu_1, 0) = \mu_1$, $\perp(\mu_1, 1) = 1$.

Аксиомам t -нормы удовлетворяют, например, функции $\min(\mu_1, \mu_2)$ и $\mu_1\mu_2$, аксиомам t -конормы — функции $\max(\mu_1, \mu_2)$ и $\mu_1 + \mu_2 - \mu_1\mu_2$.

Замечание. Из аксиом следует, что \min является максимальной t -нормой (то есть $\min(\mu_1, \mu_2) \geq \top(\mu_1, \mu_2) \forall \top(\mu_1, \mu_2)$), а \max — минимальной t -конормой.

Легко проверяется следующее

Утверждение 1. Если \top_1 и \top_2 — t -нормы, то их выпуклая комбинация

$$\lambda\top_1 + (1 - \lambda)\top_2, \quad \text{где } \lambda \in [0; 1],$$

также является t -нормой. Аналогичное утверждение верно и для t -конорм.

Таким образом, мы получаем непрерывно параметризованное множество t -норм

$$\{\lambda \min(\mu_1, \mu_2) + (1 - \lambda)\mu_1\mu_2, \quad \lambda \in [0; 1]\}$$

и t -конорм

$$\{\lambda \max(\mu_1, \mu_2) + (1 - \lambda)(\mu_1 + \mu_2 - \mu_1\mu_2), \quad \lambda \in [0; 1]\}.$$

С помощью этих множеств можно проводить обучение системы — настройку операторов агрегирования путём выбора t -норм и t -конорм. Была реализована следующая процедура на основе классического генетического алгоритма [8, 12, 14, 15].

Хромосомы — бинарные строки, представляющие код Грея [12] вещественных параметров λ_i , размерности $8(R + 1)$ (по числу битов отвечающих за параметр на число правил в системе и один параметр t -конормы). Был реализован одноточечное скрещивание, оставляющее двух потомков, и мутация — бинарный код случайного вещественного числа, лежащего в области определения параметра. Функция пригодности вычислялась как среднеквадратичная ошибка, определяемая расстоянием между Y — выходом контроллера и Y^* — выходом обучающей выборки:

$$\tilde{Y}' = \sqrt{\sum_{i=1}^D (Y_i - Y_i^*)^2 / D}, \quad \text{где } D \text{ — размерность выборки.}$$

Укрупненная схема работы алгоритма настройки выглядит следующим образом:

Инициализация:

На этом этапе работы алгоритма инициализируется случайный набор хромосом. Определяется качество каждой хромосомы, в соответствии с которым хромосомы ранжируются.

Шаг:

- 1) В новом поколении с вероятностью воспроизведения, рассчитанной оператором отбора, воспроизводятся дубликаты лучших хромосом и генерируются хромосомы, полученные скрещиванием и мутацией.
- 2) Проверяется «качественность» для каждой хромосомы популяции в соответствии с функцией пригодности.
- 3) Отмирают «неприспособленные» индивидуумы популяции.
- 4) Если качество лучшей хромосомы популяции удовлетворяет запросам задачи, то лучшая хромосома и является решением задачи — гены ее составляющие являются параметрами искомых t -норм и t -конормы.

Иначе шаг алгоритма повторяется.

Автоматическое генерирование правил

Эффективное применение описанного выше алгоритма возможно при выполнении 2 условий:

- задан полный набор правил;
- имеется достаточно большая обучающая выборка.

В случае невыполнения этих условий можно применить другой алгоритм [6, 8], основанный на представлении эксперта о поведении оператора агрегирования. Такие представления могут выражаться высказываниями вида «При сильном возрастании первого аргумента значение функции слегка убывает», «При совместном возрастании аргументов x_3 и x_5 значение функции сильно возрастает» и т. п., которые, взятые вместе, определяют некоторый нечётко заданный класс функций k -значной логики.

Рассмотрим простейший пример. Пусть у нас есть одно нечеткое условие S на поведение функции от одной переменной $f(x)$.

Нечеткое отношение \hat{S} , соответствующее нечеткому условию S , описывает принадлежность функции к данному классу на основе значений функции в точках i и $i + 1$ ($1 \leq i \leq k - 1$). Значение $\mu_{\hat{S}}(p, q)$ есть степень принадлежности функции к данному классу при условии, что $f(i) = p$, $f(i + 1) = q$ ($0 \leq p, q \leq k - 2$). Приведем примеры такой формализации обычных и нечетких условий.

Пример 1. Рассмотрим обычное, не нечеткое, условие возрастания функции. Оно разбивается на следующие локальные требования:

$$\forall i (1 \leq i \leq k - 1) f(i) < f(i + 1).$$

Матрица, описывающая это условие, имеет следующий вид:

Наружные строка и столбец содержат все возможные значения функции в двух соседних точках (i и $i + 1$). На пересечении p -ой строки и q -го столбца матрицы находится число из $\{0, 1\}$, характеризующее степень принадлежности функции к описываемому классу при условии, что $f(i) = p$, $f(i + 1) = q$.

Пример 2. Рассмотрим следующее нечеткое условие «При возрастании x функция $f(x)$ слегка возрастает». Матрица, описывающая это условие, может иметь следующий вид:

$f(i) \setminus f(i+1)$	0	1	2	3	4	5
0	0	1	1	1	1	1
1	0	0	1	1	1	1
2	0	0	0	1	1	1
3	0	0	0	0	1	1
4	0	0	0	0	0	1
5	0	0	0	0	0	0

$f(i) \setminus f(i+1)$	0	1	2	3	4	5
0	0.6	1	0.6	0.2	1	1
1	0	0.6	1	0.6	0.2	1
2	0	0	0.6	1	0.6	0.2
3	0	0	0	0.6	1	0.6
4	0	0	0	0	0.6	1
5	0	0	0	0	0	0.6

Так же как и в предыдущем примере, наружные строка и столбец содержат все возможные значения функции в двух соседних точках (i и $i+1$). На пересечении p -ой строки и q -го столбца матрицы находится число из $[0, 1]$, характеризующее степень принадлежности функции к описываемому классу при условии, что $f(i) = p, f(i+1) = q$.

Функция удовлетворяет нечеткому условию, если она удовлетворяет ему для всех значений i ($1 \leq i \leq k-2$). Таким образом, по матрице нечеткого отношения степень принадлежности любой функции этому условию вычисляется однозначно. Она будет равна некоторой t -норме соответствующих степеней принадлежности из матрицы:

$$\mu_S(f) = \&_{i=1}^{k-2} \mu_{\hat{S}}(f(i), f(i+1)).$$

Пример 3. Рассмотрим две функции: возрастающую и «похожую» на возрастающую (f_1 и f_2 соответственно):

x	0	1	2	3	4	5
f_1	0	1	2	3	4	5
f_2	0	1	2	2	4	5

В качестве t -нормы возьмем умножение. Это представляется удобным, так как в этом случае при подсчете степени принадлежности функции мы будем учитывать ее поведение во всех соседних точках. Пусть S_1 — условие строгого возрастания функции (таблица из примера 1), S_2 — условие «слегка возрастает», задаваемое таблицей из примера 2. Тогда

$$\begin{aligned}\mu_{S_1}(f_1) &= \mu_{\hat{S}_1}(0,1) \cdot \mu_{\hat{S}_1}(1,2) \cdot \mu_{\hat{S}_1}(2,3) \cdot \mu_{\hat{S}_1}(3,4) \cdot \mu_{\hat{S}_1}(4,5) = \\ &= 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 1; \\ \mu_{S_1}(f_2) &= \mu_{\hat{S}_1}(0,1) \cdot \mu_{\hat{S}_1}(1,2) \cdot \mu_{\hat{S}_1}(2,2) \cdot \mu_{\hat{S}_1}(2,4) \cdot \mu_{\hat{S}_1}(4,5) = \\ &= 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 = 0; \\ \mu_{S_2}(f_1) &= \mu_{\hat{S}_2}(0,1) \cdot \mu_{\hat{S}_2}(1,2) \cdot \mu_{\hat{S}_2}(2,3) \cdot \mu_{\hat{S}_2}(3,4) \cdot \mu_{\hat{S}_2}(4,5) = \\ &= 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 1; \\ \mu_{S_2}(f_2) &= \mu_{\hat{S}_2}(0,1) \cdot \mu_{\hat{S}_2}(1,2) \cdot \mu_{\hat{S}_2}(2,2) \cdot \mu_{\hat{S}_2}(2,4) \cdot \mu_{\hat{S}_2}(4,5) = \\ &= 1 \cdot 1 \cdot 0.6 \cdot 0.6 \cdot 1 = 0.36;\end{aligned}$$

Комбинируя подобного рода условия с помощью некоторой t -нормы, получим функцию $\mu_{\hat{S}}(f)$ — степень соответствия оператора набору условий. В качестве оператора агрегирования следует выбрать функцию с максимальной степенью соответствия. Процедура поиска снова осуществляется с помощью генетических алгоритмов: функцией пригодности является функция $\mu_{\hat{S}}(f)$, хромосомы содержат закодированный набор правил и параметры t -норм и t -конормы. Схема работы алгоритма идентична описанной в предыдущем разделе.

Использование системы

Так как различные проекты имеют различную структуру (включают разные стадии, требуют разное число разработчиков и т. д.), невозможно создать универсальный шаблон или универсальную модель. Поэтому была предложена следующая схема использования системы.

- 1) *Разработка модели.* Эксперты составляют модели стадий процесса, разрабатывают стандартные наборы правил для опера-

торов агрегирования и правила выбора лингвистических значений.

- 2) *Адаптация модели.* Инженер (не обязательно являющийся экспертом в области разработки электронных устройств, но имеющий опыт в использовании технологии информационного мониторинга), используя подмодели, составляет модель для конкретного проекта.
- 3) *Использование.* Пользователь (например, менеджер проекта), используя правила выбора лингвистических значений, вводит в модель данные о команде разработчиков и получает необходимые результаты. Доступны следующие возможности:
 - оценка текущей ситуации;
 - моделирование изменений;
 - поиск оптимальных способов улучшения ситуации.

Графическое изображение описанной схемы использования системы приведено на рис. 4.

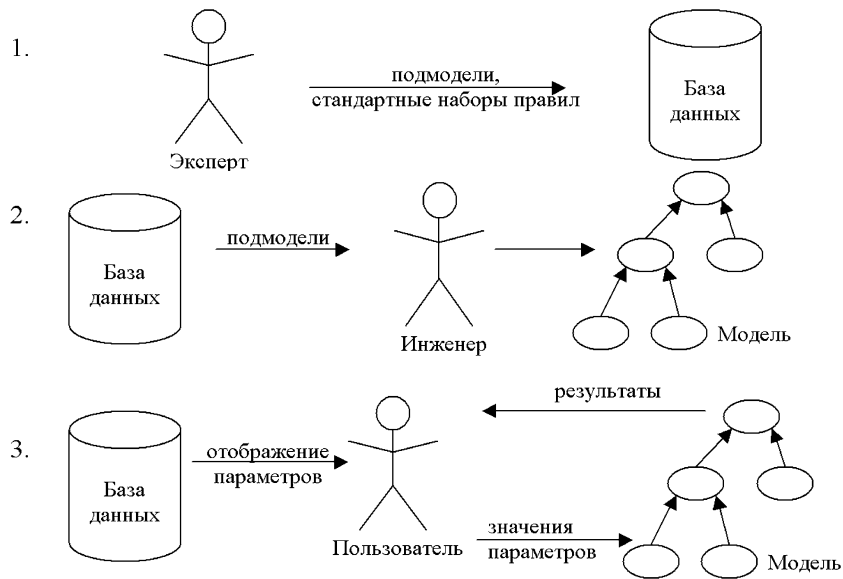


Рис. 4. Схема использования системы.

Графический интерфейс

В процессе разработки прототипа системы был реализован графический интерфейс, поддерживающий все этапы разработки и настройки модели, а именно:

Редактирование графа (рис. 5):

- добавление элемента;
- изменение элемента;
- удаление элемента и его потомков;
- добавление связи;
- удаление связи;

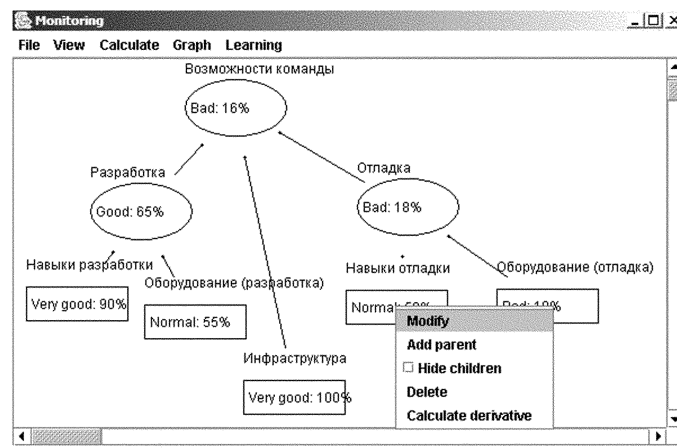


Рис. 5. Окно редактирования модели.

Выбор операторов агрегирования (рис 6):

- ручной ввод правил;
- автоматический ввод правил (на основе формул k -значной логики);
- точная настройка (рис. 7);

Функционирование:

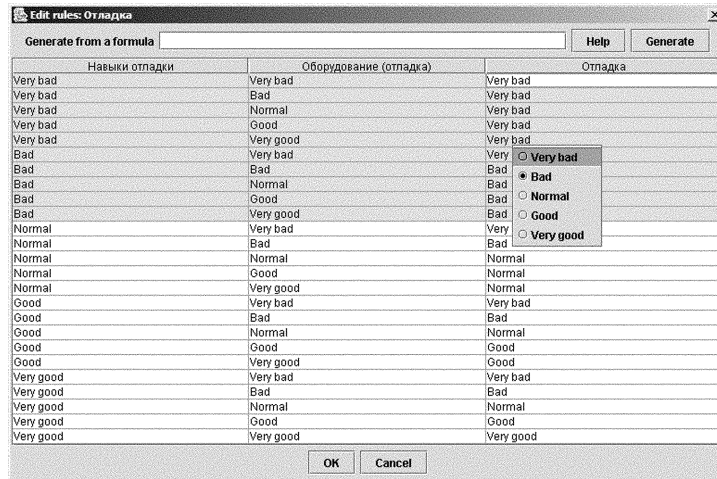


Рис. 6. Окно редактирования набора правил.

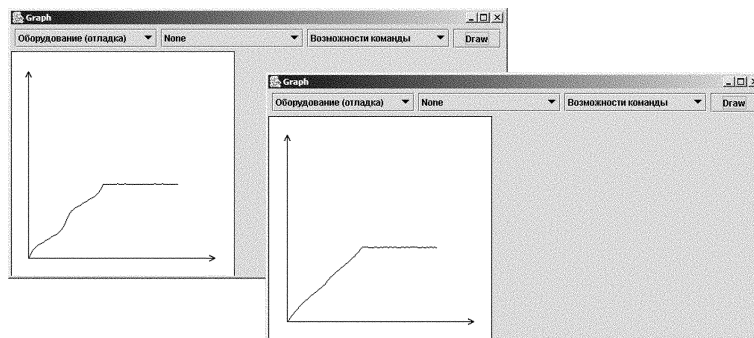


Рис. 7. Графики зависимости результата от одного из параметров до (слева) и после (справа) обучения.

- изменение значения вершины (с автоматическим вычислением результата);
- поиск «критических путей».

Такое средство позволяет достаточно быстро и эффективно как разрабатывать подобные прототипы «с нуля», так и проводить их настройку и адаптацию.

4. Оценка распространения погрешности

Так как при разработке систем информационного мониторинга изначально предполагается, что входные данные неточны, необходимо иметь оценки того, как эта погрешность будет изменяться при вычислениях.

Пусть $f(x_1, \dots, x_n) : [0; 1]^n \rightarrow [0; 1]$ — оператор агрегирования, заданный на виртуальной шкале. Будем считать, что аргументы x_i представляют собой нечёткие числа с треугольной функцией принадлежности $\mu_i(x_i)$ (рис. 8). Выходом оператора также будет нечёткое число, функцию принадлежности которого мы определим следующим образом:

$$\mu_y(y) = \max_{\vec{x}: f(\vec{x})=y} \mu(\vec{x}), \text{ где } \vec{x} = (x_1, \dots, x_n), \mu(\vec{x}) = \&_i \mu_i(x_i).$$

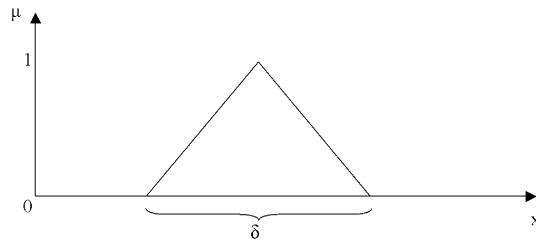


Рис. 8.

Обозначим длины отрезков, на которых $\mu_i(x_i) > 0$ и $\mu_y(y) > 0$, δ_i и δ соответственно. Тогда нашей задачей будет оценить зависимость $\delta(\delta_1, \dots, \delta_n)$.

Заметим, что стандартный подход оценки качества систем на базе нечётких множеств, основанный на понятии степени нечёткости [9], не подходит для рассматриваемой задачи. В качестве примера можно рассмотреть нечёткое множество с функцией принадлежности

$$\mu(x) = \begin{cases} 1, & x \in [x_0, x_0 + \delta], \\ 0, & \text{иначе.} \end{cases}$$

Оно имеет степень нечёткости, равную 0, но при больших δ (близких к мощности универсального множества) результат подобного ви-

да, полученный в результате работе системы, не будет представлять никакой ценности для пользователя, так как не позволяет дать оценку с малой погрешностью. С другой стороны, малое значение δ гарантирует малость степени нечёткости ξ , связанной с δ соотношением

$$\xi \leq \delta/|U|,$$

где $|U|$ — мощность универсального множества (в нашем случае $U = [0; 1]$, $|U| = 1$).

Локальный подход

Будем полагать δ_i бесконечно малыми. В этом случае требуется оценить модуль градиента функции $f(x_1, \dots, x_n)$. Рассмотрим наиболее часто встречающийся случай.

Лемма 1. Пусть функция f строится на основе нечёткого вывода с t -нормой и t -конормой \min и \max соответственно. Пусть результатом выполнения операций вывода и декомпозиции стало следующее выражение:

$$((y = \tilde{y}_1) \& \mu_1) \vee ((y = \tilde{y}_2) \& \mu_2),$$

где $0 \leq \mu_2 \leq 0.5 \leq \mu_1 \leq 1$, \tilde{y}_1 и \tilde{y}_2 — «соседние» лингвистические значения для y , имеющие треугольные функции принадлежности с шириной основания $2b$ и центрами в точках y_0 и $y_0 + b$ соответственно (то есть к точке (x_1, \dots, x_n) с ненулевой степенью применимы только правила, содержащие в качестве вывода 2 соседних значения). Тогда $-\frac{2}{5}b \leq \frac{\partial y}{\partial \mu_1} \leq 0$, $0 \leq \frac{\partial y}{\partial \mu_2} \leq 2b$.

Доказательство. Определим значение y , которое, напомним, вычисляется как центр масс фигуры на рисунке 9, в зависимости от μ_1 и μ_2 .

Для вычисления центра масс разобьём фигуру на 2 подфигуры:

- трапецию А с массой $m_1 = b(2 - \mu_1)$ и положением центра масс в точке y_0 ;

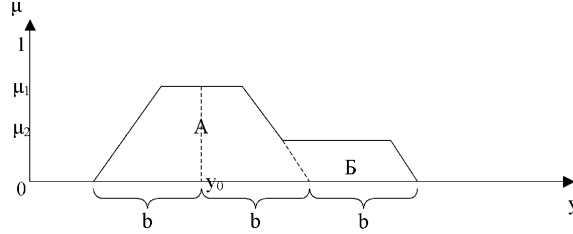


Рис. 9.

- параллелограмм Б с массой $m_2 = b\mu_2$ и центром масс в точке

$$y_0 + y_2 = y_0 + \frac{3}{2}b - \frac{\mu_2 b}{2}.$$

Тогда центр масс всей фигуры будет находиться в точке

$$y = y_0 + \frac{m_2 y_2}{m_1 + m_2} = y_0 + b\mu_2 \frac{3 - \mu_2}{2(2\mu_1 + \mu_2 - \mu_1^2)}.$$

Вычислим и оценим частные производные $\frac{\partial y}{\partial \mu_1}$ и $\frac{\partial y}{\partial \mu_2}$:

$$\begin{aligned} \frac{\partial y}{\partial \mu_1} &= -\frac{b\mu_2(3 - \mu_2)(2 - 2\mu_1)}{2(2\mu_1 + \mu_2 - \mu_1^2)^2} = -b \frac{(3\mu_2 - \mu_2^2)(1 - \mu_1)}{(2\mu_1 + \mu_2 - \mu_1^2)^2} = \\ &= -b \frac{(9/4 - (3/2 - \mu_2)^2)(1 - \mu_1)}{(1 - (1 - \mu_1)^2 + \mu_2)^2}. \end{aligned}$$

$$\begin{aligned} \frac{\partial y}{\partial \mu_2} &= \frac{b}{2} \cdot \frac{(3 - 2\mu_2)(2\mu_1 + \mu_2 - \mu_1^2) - (3\mu_2 - \mu_2^2)}{(2\mu_1 + \mu_2 - \mu_1^2)^2} = \\ &= \frac{b}{2} \cdot \frac{6\mu_1 - 3\mu_1^2 - 4\mu_1\mu_2 - \mu_2^2 + 2\mu_1^2\mu_2}{(2\mu_1 + \mu_2 - \mu_1^2)^2} = \frac{b}{2} \cdot \frac{\mu_1(2 - \mu_1)(3 - 4\mu_2) - \mu_2^2}{(2\mu_1 + \mu_2 - \mu_1^2)^2}. \end{aligned}$$

Учитывая, что $0.5 \leq \mu_1 \leq 1$, $0 \leq \mu_2 \leq 0.5$, получим $-\frac{2}{5}b \leq \frac{\partial y}{\partial \mu_1} \leq 0$, $0 \leq \frac{\partial y}{\partial \mu_2} \leq 2b$, что и требовалось доказать.

Следствие 1. f монотонна по μ_1 и μ_2 .

Следствие 2. Если набор правил монотонен (в смысле функций k -значной логики), то f монотонно не убывает по всем переменным.

Обозначим a_i расстояния между центрами соседних лингвистических значений переменной x_i , μ_1^i и μ_2^i — степени принадлежности x_i соответственно большему и меньшему лингвистическому значению (рис. 10). Тогда

$$\frac{\partial \mu_1^i}{\partial x_i} = \frac{1}{a_i}, \quad \frac{\partial \mu_2^i}{\partial x_i} = -\frac{1}{a_i}.$$

Далее, так как μ_1 и μ_2 вычисляются в виде

$$\mu_j = \max_{(j_1, \dots, j_n) \in A_j} \min_i (\mu_{j_i}^i(x_i)), \quad \text{где } j, j_i \in \{1, 2\}, \quad A_1 \cup A_2 = \{1, 2\}^n, \text{ а}$$

$$|\min(t_1 + \Delta t_1, \dots, t_n + \Delta t_n) - \min(t_1, \dots, t_n)| \leq \max |\Delta t_i|,$$

$$|\max(t_1 + \Delta t_1, \dots, t_n + \Delta t_n) - \max(t_1, \dots, t_n)| \leq \max |\Delta t_i|,$$

имеем: $|\Delta \mu_j(x_1, \dots, x_n)| \leq \max_i \frac{\delta_i}{a_i}$. Используя результаты леммы, получаем:

$$\begin{aligned} |\Delta f(x_1, \dots, x_n)| &= \left| \frac{\partial f}{\partial \mu_1} \Delta \mu_1(x_1, \dots, x_n) \right| + \left| \frac{\partial f}{\partial \mu_2} \Delta \mu_2(x_1, \dots, x_n) \right| \leq \\ &\leq \frac{12}{5} b \cdot \max_i \left| \frac{\delta_i}{a_i} \right|. \end{aligned}$$

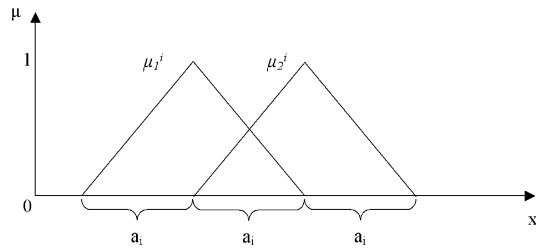


Рис. 10.

Итак, основным результатом данного раздела является следующая

Теорема 1. Пусть функция f строится на основе нечёткого вывода с t -нормой и t -конормой \min и \max соответственно. Пусть набор правил, определяющий функцию f , функция $F(X_1, \dots, X_n)$:

$E_{k_1} \times \dots \times E_{k_n} \rightarrow E_k$ обладает следующим свойством: $\max(|X_1 - Y_1|, \dots, |X_n - Y_n|) \leq 1 \Rightarrow |F(X_1, \dots, X_n) - F(Y_1, \dots, Y_n)| \leq 1$. Тогда в приведённых выше обозначениях выполняется:

$$\delta \leq \frac{12}{5} b \cdot \max_i \left| \frac{\delta_i}{a_i} \right|, \quad \text{при } \delta_i \rightarrow 0.$$

Доказательство. непосредственно следует из предыдущих результатов раздела.

Эта теорема ограничивает сверху скорость роста погрешности и, кроме того, даёт возможность «управлять» этим ростом, меняя значения b и a_i (которые в свою очередь зависят от числа лингвистических значений).

Дискретный подход

Из результатов предыдущего параграфа видно, что малые ошибки, хоть и будут сохранять порядок малости, могут экспоненциально возрасти при последовательном вычислении от нижних уровней иерархии к верхним. В этом разделе мы покажем, что в случае конечных погрешностей такого возрастания не произойдёт.

Рассмотрим простейший случай функции одной переменной. Пусть \hat{x}_1 и \hat{x}_2 — центры соседних лингвистических значений, \hat{y}_1 и \hat{y}_2 — центры лингвистических значений, соответствующих им в правилах (без ограничения общности будем считать, что $y_1 \leq y_2$). Тогда $f(\hat{x}_1) = \hat{y}_1$, $f(\hat{x}_2) = \hat{y}_2$, а так как функция f монотонна на отрезке $[\hat{x}_1; \hat{x}_2]$ (см. результаты предыдущего раздела), то при $\hat{x}_1 \leq x \leq \hat{x}_2$, $\hat{y}_1 \leq f(x) \leq \hat{y}_2$. Это означает, что если аргумент — нечёткое число, расположенное между \hat{x}_1 и \hat{x}_2 , то значением функции будет нечёткое число, расположенное между \hat{y}_1 и \hat{y}_2 (рис. 11).

Для рассмотрения случая многих переменных введём следующие обозначения:

$P^{j_1 \dots j_n}$ — n -мерный параллелепипед $\{(x_1, \dots, x_n) : \hat{x}_{j_i}^i \leq x_i \leq \hat{x}_{j_{i+1}}^i, i = 1, \dots, n\}$, где \hat{x}_j^i — центр j -ого лингвистического значения i -го аргумента;

$\hat{y}_{\min}^{j_1 \dots j_n}$ и $\hat{y}_{\max}^{j_1 \dots j_n}$ — центры соответственно наименьшего и наибольшего лингвистических значений, принимаемых набором правил F на $P^{j_1 \dots j_n}$;

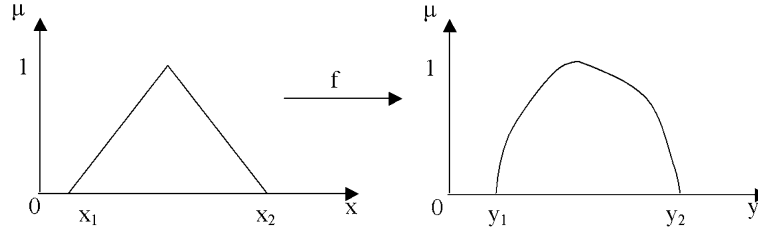


Рис. 11.

Непосредственно из схемы нечёткого вывода следует

Лемма 2 (принцип максимума). *Функция, построенная на основе нечёткого вывода, принимает минимальное и максимальное на параллелепипеде $P^{j_1 \dots j_n}$ значения (равные, соответственно, $\hat{y}_{\min}^{j_1 \dots j_n}$ и $\hat{y}_{\max}^{j_1 \dots j_n}$) на вершинах параллелепипеда.*

Используя эту лемму и проведя рассуждения, аналогичные рассмотренным для случая одной переменной, получим

Утверждение 2. *В вышеприведённых обозначениях*

$$(x_1 \pm \delta_1/2, \dots, x_n \pm \delta_n/2) \in P^{j_1 \dots j_n} \Rightarrow \delta \leq \hat{y}_{\max}^{j_1 \dots j_n} - \hat{y}_{\min}^{j_1 \dots j_n}.$$

Следствие 3. *Если $\hat{y}_{\min}^{j_1 \dots j_n}$ и $\hat{y}_{\max}^{j_1 \dots j_n}$ — центры соседних лингвистических значений, то погрешность при вычислениях не возрастает.*

Утверждение этого раздела гарантирует, что при выполнении определённых условий погрешность будет ограничена расстоянием между соседними лингвистическими значениями. Как и в предыдущем разделе, это позволяет влиять на погрешность, меняя число лингвистических значений.

Следует добавить, что более полную картину распространения погрешностей дала бы оценка изменения δ^α — ширины множества α -уровня $\{y : \mu_y(y) > \alpha\}$ (Рис. 12) — в зависимости от α . Такая оценка будет получена в дальнейшем.

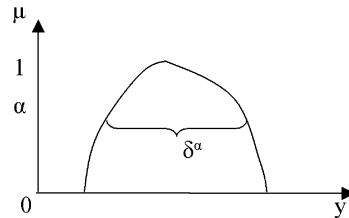


Рис. 12.

Заключение

В работе рассматривалась задача оценки состояния коллектива разработчиков и его возможности по выполнению проекта разработки электронного устройства. Было предложено решение этой задачи на основе технологии информационного мониторинга и рассмотрены основные этапы разработки системы: построение модели, выбор и настройка операторов агрегирования информации, разработка интерфейса. Также были приведены теоретические оценки качества работы системы в условиях неопределённости входных данных.

На момент написания этой работы функциональный прототип системы успешно прошёл первичное тестирование и перешёл в стадию опытной эксплуатации.

Пользуясь случаем, авторы считают своим долгом выразить признательность сотрудникам Мюнхенского офиса компании Cadence Design Systems Вольфу Мацке (Wolf-Ekkehard Matzke) и Виктору Прайсу (Viktor Preis) за поддержку этого проекта, ценные консультации, вклад в разработку и тестирование модели. Без их поддержки и участия разработка прототипа была бы просто не возможна.

Список литературы

- [1] Ахмеджанов Н. М., Жукоцкий А. В., Кудрявцев В. Б., Оганов Р. Г., Расторгуев В. В., Рыжов А. П., Строгалов А. С. Информационный мониторинг в задаче прогнозирования риска развития сердечно-сосудистых заболеваний // Интеллектуальные системы. Т. 7. Вып. 1–4. 2003. С. 5–38.

- [2] Заде Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976.
- [3] Кудрявцев В. Б., Алёшин С. В., Подколзин А. С. Введение в теорию автоматов. М.: Наука, 1985.
- [4] Нечеткие множества в моделях управления и искусственного интеллекта / под ред. Поспелова Д. А. М.: Наука, 1986.
- [5] Прикладные нечеткие системы / под ред. Т. Тэрано, К. Асаи, М. Сугэно. М.: Мир, 1993.
- [6] Рогожин С. В., Рыжов А. П. О нечетко заданных классах функций k -значной логики // V Всероссийская конференция «Нейрокомпьютеры и их применение». Сборник докладов. Москва, 17–19 февраля 1999 г. С. 460–463.
- [7] Рыжов А. П. Модели поиска информации в нечеткой среде. М.: Издательство Центра прикладных исследований при механико-математическом факультете МГУ, 2004.
- [8] Рыжов А. П. Об агрегировании информации в нечетких иерархических системах // Интеллектуальные системы. Т. 6. Вып. 1–4. 2002. С. 341–364.
- [9] Рыжов А. П. О степени нечеткости размытых характеристик // Проблемы теоретической кибернетики: Тезисы докладов VII Всесоюзной конференции 18 сентября–20 сентября 1985 г. I Часть. Иркутск, 1985. С. 53–55.
- [10] Рыжов А. П. Элементы теории нечетких множеств и измерения нечеткости. М.: Диалог-МГУ, 1998.
- [11] Саати Т. Анализ иерархических процессов. М.: Радио и связь, 1993.
- [12] Скурихин А. Н. Генетические алгоритмы // Новости искусственного интеллекта. № 4. С. 6–46.
- [13] Яблонский С. В. Основные понятия кибернетики // Проблемы кибернетики. 1959. Вып. 2. С. 7–38.
- [14] De Jong K. A. An Analysis of the behaviour of a Class of Genetic Adaptive Systems / PhD thesis. Univ. of Michigan, 1975.
- [15] Holland J. H. Adaptation in Natural and Artificial Systems // Ann Arbor: Univ. of Michigan Press, 1975.

- [16] Libecap Garry D. University entrepreneurship and technology transfer: process, design, and intellectual property. ELSEVIER, 2005.
- [17] Martin Michael J. C. Managing innovation and entrepreneurship in technology based firms. NY: John Wiley & Sons, Inc. 1994.
- [18] Matzke W. E., Strube G., Schmidt-Habich H., Drenan L. VCAD — a virtual enterprise collaboration model impacting the semiconductor industry // IASTED International Conference on Knowledge Sharing & Collaborative Engineering (KSCE 2004).
- [19] Messarovich M. D., Macko D., Takahara Y. Theory of hierarchical multilevel systems. N.Y.–London: Academic Press, 1970.
- [20] Ryjov A. Basic principles and foundations of information monitoring systems // Monitoring, Security, and Rescue Techniques in Multi-agent Systems. Springer, 2005. P. 147–160.
- [21] Ryjov A., Belenki A., Hooper R., Pouchkarev V., Fattah A., Zadeh L. A. Development of an Intelligent System for Monitoring and Evaluation of Peaceful Nuclear Activities (DISNA). IAEA, STR-310. Vienna, 1998.
- [22] Zadeh L. A. Fuzzy sets // Information and Control. 1965. V. 8. P. 338–353.