

# ФОРМИРОВАНИЕ БАЗЫ ПРАВИЛ ДЛЯ PLM СИСТЕМ

No Author Given

No Institute Given

**Аннотация** Часто возникает необходимость интеграции PLM-системы и сторонней информационной системы, не входящей в состав комплекса PLM, но участвующей в решении задач, связанных с информационной поддержкой управления производственными процессами. На данный момент задачу такой интеграции выполняет аналитик. Он должен сформировать структурную и процессную модели интегрируемой ИС, на основе которых формируются правила взаимодействия с системой. В процессе эксплуатации помимо аналитика участвует оператор, который выполняет задачу поддержания актуальности данных как интегрируемой ИС, так и всего комплекса, и лицо, принимающее решение (ЛПР), который выполняет процесс управления интегрируемой ИС, а именно поддержание её работоспособного состояния. Предлагается подход, позволяющий снизить нагрузку как на аналитика, обеспечивающего настройку взаимодействия и отображения данных, так и на оператора, участвующего в обеспечении актуальности структур данных, и ЛПР, участвующего в принятии важных решений, связанных с возможными рисками на производстве.

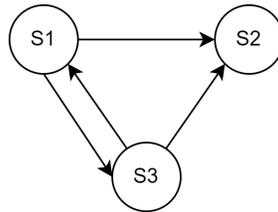
**Keywords:** База правил · PLM-системы · Управление на основе данных

## 1 Introduction

На сегодняшний день многие крупные производства используют PLM системы [1] для управления жизненным циклом изделий. PLM системы используются для осуществления контроля потоков данных на производстве, включая решение таких задач как хранение, интеграцию, поддержание в актуальном состоянии данных в каждой информационной подсистеме, входящей в комплекс. Ключевую роль в этом выполняет шина данных [2]. В каждом комплексе систем шина данных имеет свои особенности реализации [3]. Однако общей особенностью интеграции на основе шин данных является использование взаимодействия, основанного на правилах. Для настройки взаимодействия между подсистемами аналитик выполняет:

- построение модели поведения информационной системы,
- выявление ключевых признаков,
- формирование правил поведения системы.

В данной работе рассматривается не только подход к формированию взаимодействия между системами, но и подход к управлению PLM подсистемами. Часто возникает необходимость интеграции PLM-системы и сторонней информационной системы (далее в тексте будем называть ее интегрируемая ИС), не входящей в состав комплекса PLM, но участвующей в решении задач, связанных с информационной поддержкой управления производственными процессами. На данный момент задачу такой интеграции выполняет аналитик или лицо принимающее решение (ЛПР). Аналитик должен сформировать структурную и процессную модели интегрируемой ИС, на основе которых формируются правила взаимодействия с системой. Так же в процессе эксплуатации к задачам аналитика (ЛПР) добавляется процесс управления интегрируемой ИС, а именно поддержание актуальности данных и работоспособного состояния самой системы. На рисунке 1 представлен граф состояний интегрируемой ИС в общем виде.

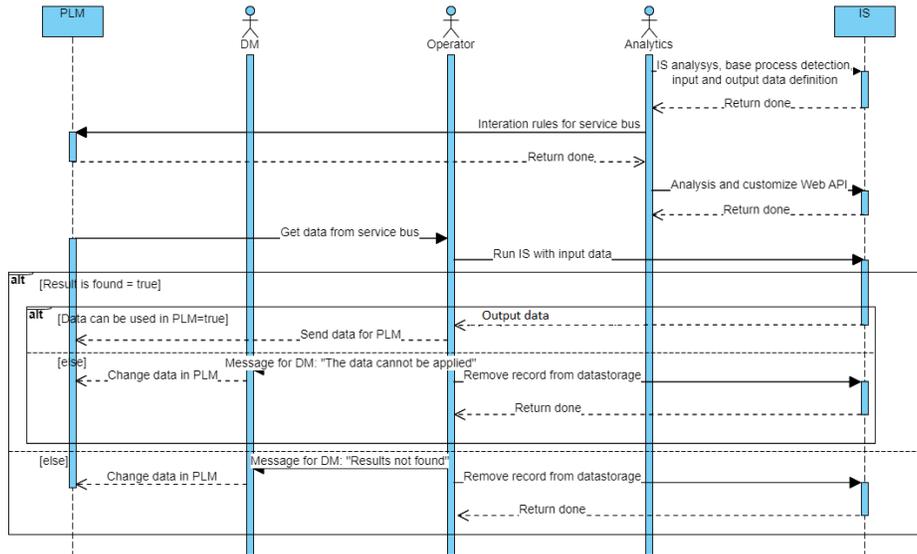


**Рис. 1.** Граф состояний интегрируемой информационной системы

На рисунке представлены основные состояния интегрируемой ИС, используемые в данной работе. Так  $S1$  — работоспособное состояние системы,  $S2$  — неработоспособное состояние системы или состояние отказа,  $S3$  — предельное состояние системы [4]. Под работоспособным состоянием системы понимается такое состояние, когда система продолжает выполнять свои основные задачи, при этом данные системы соответствуют. Под предельным состоянием системы понимается такое состояние системы, когда система продолжает выполнять свои основные задачи, при этом данные системы находятся в пограничном состоянии. Под неработоспособным состоянием системы или состоянием отказа понимается такое состояние системы, когда система не может продолжать выполнение основных задач и требуется вмешательство ЛПР. Находясь в состоянии  $S1$  и  $S3$ , система может перейти в состояние отказа, поэтому основная цель управления состоит в поддержании системы в состоянии  $S1$  и возможности возврата в это состояние без рисков, в случае, если система перешла в состояние  $S3$ .

Стоит отдельно рассмотреть процесс, когда взаимодействие интегрируемой ИС и всего комплекса PLM обеспечивается участием оператора. В этом процессе в настоящее время объем работ оператора для настройки взаимодействия, отображения данных, поддержания системы в работоспособном

состоянии достаточно велик. Роль оператора в данном процессе отображена на рисунке 2.



**Рис. 2.** Процесс организации взаимодействия и поддержания системы в состоянии работоспособности с участием человека

Аналитик должен учитывать изменения структуры данных интегрируемой ИС. Оператор отслеживает возможные риски, связанные с обеспечением соответствия данным комплекса PLM. ЛПП принимает решения относительно возможной минимизации рисков. Предлагается подход, позволяющий снизить нагрузку как на аналитика, обеспечивающего настройку взаимодействия и отображения данных, так и на оператора, участвующего в обеспечении актуальности структур данных, и ЛПП, участвующего в принятии важных решений, связанных с возможными рисками на производстве.

**1.1 Обзор существующих методов и подходов к управлению**

В настоящее время подходы к управлению производственными ИС раскрываются в следующих источниках: [5,6,7,8,9,10]. В источниках [5,6,8] предлагается использовать модель интегрируемой ИС. Данные подходы напрямую зависят от точности и полноты модели, однако формирование модели реализуется либо аналитиком [5,8], либо автоматически [6], что не всегда гарантирует адекватность и точность модели. В источнике [5] представлено классическое управление на основе модели процессов и данных информационной

системы, где роль аналитика велика. В источнике [6] предлагается использовать фреймворк для формирования модели. Не смотря на то, что процесс формирования модели происходит автоматически, управляющий компонент всё ещё зависит от точности и полноты модели. В источнике [7] предлагается использовать обучающую выборку и машинное обучение без учителя. Однако в зависимости от полноты и качества данных выборки зависит формируемая модель интегрируемой ИС, что зачастую может привести к неверному истолкованию результатов прогнозирования механизмов управления, в случае если данные имеют ошибки или сильно зашумлены. В источнике [8] предлагается подход цифровых двойников. Несмотря на преимущества данного подхода в целом, применительно к поставленной задаче, а именно интеграция информационных систем, отображение данных и снижение трудозатрат аналитика, оператора и ЛПР, подход цифровых двойников требует дополнительный качественный анализ и построение модели информационной системы, что приводит к увеличению нагрузки на аналитика.

В источнике [9] описан подход к управлению human-in-the-loop. Основная идея данного подхода заключается в полном контроле поведения системы управления со стороны человека и сводится к машинному обучению с учителем. Данный подход является наиболее безопасным с точки зрения предупреждения рисков, связанных с производством, но более комплексным и требующим участия ЛПР на всех этапах жизненного цикла управляющей системы, начиная с формирования обучающей выборки, и заканчивая контролем поведения управляющей системы. Таким образом, предлагается использовать подход управления на основе данных [10]. Предполагается снижение трудозатрат аналитика и оператора за счет выполнения задачи интеграции и частично задачи управления интегрируемой ИС разработанной управляющей системой. Подход управления на основе данных подразумевает:

1. выполнение моделирования системы, но не в классическом понимании[5], а за счет формирования метамодели данных интегрируемой ИС на основе анализа ее хранилища;
2. отображение данных при интеграции систем PLM и интегрируемой ИС предприятия за счет использования базы правил поведения интегрируемой ИС;
3. снижение рисков перехода в состояние отказа 1 и поддержание системы в работоспособном состоянии без участия в данном процессе ЛПР.

## 1.2 Обзор существующих подходов к формированию баз правил

Для решения задачи предлагается использовать продукционную модель базы правил. Классические нечеткие системы основаны на подходе Мамдани [11]. В таких системах существует 2 модуля преобразования обычных данных в нечеткие. Модуль фаззификации устанавливает соответствие между реальными значениями входных данных и нечеткими значениями, основываясь на функции принадлежности. С другой стороны, модуль дефаззифи-

кации устанавливает соответствие между нечеткими значениями и реальными значениями выходных данных предметной области. Нечеткие базы правил (системы, основанные на нечетких правилах) основаны на принципе преобразования четких значений в нечеткие. Правила в данном случае представляют собой множество лингвистических термов и выходные данные ассоциируются с ними, например, правило может иметь несколько входных данных и только определенное значение выходного параметра. Существует несколько вариантов классической системы, основанной на нечетких правилах:

1. Система, основанная на нечетких правилах Мамдани [12]. Каждая переменная правила представляет собой значение из множества лингвистических термов, например. Если  $X_i$  – переменная, представленная множеством  $\{l_1, l_2, l_3\}$ , тогда в переменной правила  $X_i$  может быть представлена как  $\{l_1, l_2\}$ . Переменная может принадлежать набору лингвистических терминов в правиле. Это помогает сократить количество правил, чтобы избежать проблемы увеличения размера самой базы. Таким образом, правило может иметь вид:  $x_1 = \{l_{11}, l_{12}\} \wedge x_2 = \{l_{23}\} \wedge \dots \wedge x_n = \{l_{n1}, l_{n2}\} \rightarrow y = Y$ .
2. Приближенные системы, основанные на нечетких правилах Мамдани [13]. Такие системы включают в себя несколько элементов наборов терминов, которые могут снизить интерпретируемость вывода. Системы способны добиться большей точности ценой потери интерпретируемости. Каждое правило имеет свой собственный нечеткий набор вместо использования лингвистических терминов. Этот подход генерирует семантически свободные правила и обладает более высокой выразительностью благодаря использованию различных нечетких множеств в каждом правиле. Он может принимать разное количество правил в зависимости от сложности проблемы. Что касается недостатков, подобные системы страдают от потери интерпретируемости, а также они могут переопределять обучающие данные и плохо работать в случае невидимых данных.
3. Системы классификации на основе нечетких правил [14]. Система классификации на основе нечетких правил — это система, которая использует нечеткие правила в качестве средства обучения. В классических системах, основанных на нечетких правилах Мамдани, входные данные сопоставляются с обычно одномерным выходом, но в данном случае входные данные сопоставляются с одной из меток класса. Структура правила выглядит следующим образом:  $x_1 = l_{1i} \wedge x_2 = l_{2i} \wedge \dots \wedge x_n = l_{ni} \rightarrow y = c$ .

Существуют варианты неклассических систем, основанных на нечетких правилах:

1. Иерархические нечеткие системы [15] состоят из нескольких низкоразмерных нечетких систем, расположенных иерархическим образом. Правила в иерархических нечетких системах сгруппированы в модули в соответствии с их ролями в системе. Каждый модуль вычисляет частичное

решение, которое далее передается на модули следующего уровня. Хотя каждый модуль представляет собой нечеткую систему, он генерирует значительно меньшее количество правил, чем плоская нечеткая система. Несмотря на широкое распространение применения иерархических баз правил при работе с big data, существует ряд недостатков подобных систем. В статье [16] приведены примеры оптимизации иерархических нечетких систем с помощью генетического алгоритма. В статье [17] приведены примеры оптимизации каскадных иерархических нечетких систем с использованием нейронных сетей, однако из-за того, что предложенный метод задействует все входные переменные, теряется преимущество сокращения количества правил.

2. Нейро-нечеткие системы [18] представляют собой слияние систем, основанных на нечетких правилах, с искусственными нейронными сетями. Основная идея систем состоит в способности принятия решений на основе заданных правил и обучения за счет использования нейронных систем. База правил является плоской (одномерной, в отличие от иерархических нечетких систем), что увеличивает размеры самой базы правил, но упрощает процесс обучения нейронной сети в процессе работы системы управления. Нейро-нечеткие системы состоят из двух модулей, первый отвечает за настройку и структурирование условия правила, а второй отвечает за формирование следствия. Нейронная сеть на основе существующих правил, принимая входные данные, настраивает условие правила и частично вычисляет функцию принадлежности, соответствующую нечеткому множеству входных данных. На втором этапе происходит вычисление следствия правила от нечеткого множества следствий до точных значений вывода.
3. Эволюционные нечеткие системы [19]. Для решения задачи формирования базы правил используются генетические алгоритмы. Отличие данного типа нечетких баз правил состоит в том, что она самообучаема и самооптимизируема. Реализация этого подхода выполняется в два этапа:
  - на первом этапе генетический алгоритм (ГА) используется для нахождения кандидатов нечетких правил в базе знаний;
  - на втором этапе с помощью ГА выполняется процедура оптимизации базы знаний с целью исключения из рассмотрения худших правил, мало влияющих на получение окончательного решения.

Для решения задачи исследования в организации данных было принято решение объединить подходы иерархической нечеткой базы правил и нейро-нечетких баз правил с выводом результата, основанном на нечетких правилах Мамдани, с использованием подхода эволюционного алгоритма для формирования правил на основе изменений метамодели. Таким образом, база правил будет построена как показано на рисунке 3.

Таким образом, согласно диаграмме база знаний состоит из нескольких уровней условий, учитывающих возможные изменения в метамодели и в базе данных интегрируемой информационной системы, и нескольких уровней

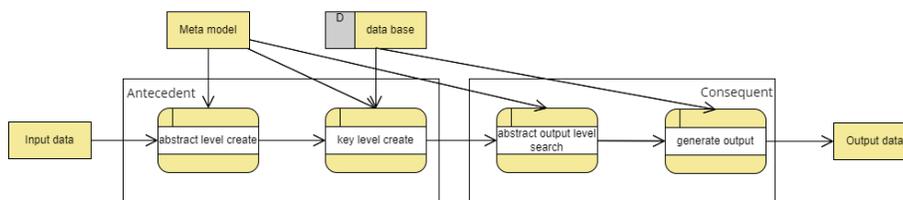


Рис. 3. Структура гибридной базы с нечетким логическим выводом

следствий, формирующих четкий логический вывод на основе лингвистического представления правил в базе правил.

## 2 Модель базы знаний поведения интегрируемой информационной системы

Согласно рисунку 3 база правил будет представлена в виде иерархической структуры с двумя уровнями правил. Ранее авторами была получена структурная модель метаданных  $M$  интегрируемой ИС [1]. Так первый уровень будет представлен в виде правил, состоящих из лингвистических термов, и зависеть от изменений метамодели. Второй уровень правил будет формироваться динамически на основе результатов, полученных на первом уровне.

Пусть  $INP = \{INP_1, INP_2, \dots, INP_z\}, z \in N$  – множество лингвистических термов, представляющих входные данные модели метаданных  $M$ , а  $OUT = \{OUT_1, OUT_2, \dots, OUT_w\}, w \in N$  – множество лингвистических термов, представляющих ключевые процессы модели метаданных  $M$ , тогда правило, описывающее первый уровень будет иметь теоретико-множественное представление в виде:

$$P(INP) \rightarrow \{INP^{OUT_s}, OUT_s, \tag{1}$$

где  $OUT_s$  – лингвистический терм, отражающий конкретный ключевой процесс модели метаданных  $M$ , а  $\{INP^{OUT_s}\}$  – множество лингвистических термов, отражающих входные данные для конкретного ключевого процесса модели метаданных  $M$ .

Представим  $\{INP^{OUT_s}\}$  как  $X$ , а  $OUT_s$  как  $Y$ . В таблице 1 представлены входные данные для правила второго уровня, которые будут использованы для формирования конечных значений поведения интегрируемой ИС.

Колонки 1-4 отображают значения входных данных для ключевых процессов информационной системы, а колонка 5 представляет значения данных, которые являются реакцией системы на значения входных данных.

Для формирования правил второго уровня управления интегрируемой ИС определим следующий функционал:

$$p(X, V) \rightarrow Y, \tag{2}$$

**Таблица 1.** Табличное представление входных данных для правила второго уровня

$X_1$	$X_2$	...	$X_m$	$Y$
$v_1^1$	$v_2^1$	...	$v_m^1$	$y^1$
$v_1^2$	$v_2^1$	...	$v_m^2$	$y^2$
...	...	...	...	...
$v_1^n$	$v_2^n$	...	$v_m^n$	$y^n$

где  $V = \{\{v_1^1, \dots, v_m^1\}, \{v_1^2, \dots, v_m^2\}, \dots, \{v_1^n, \dots, v_m^n\}\}$ , – входные значения  $m$  параметров  $X = \{x_k\}, k = [1, m], m \in N, n \in N, Y = \{y^i\}, i = [1, n], n \in N$ , –  $n$  состояний интегрируемой ИС.

Состояние системы  $y^i$  определяется вектором входных значений  $\{v_1^i, \dots, v_m^i\}$ . Таким образом, для формирования правила управления системы для перевода ее в состояние (выдачи управляющих воздействий)  $y^i$  необходимо в antecedent правила включить сравнение вектора параметров  $X$  со значениями  $\{v_1^i, \dots, v_m^i\}$ :

$$p^i(X, \{v_1^i, \dots, v_m^i\}) \rightarrow y^i. \quad (3)$$

При работе интегрируемой ИС могут возникать ситуации, когда разные значения входных параметров приводят к одному и тому же состоянию  $y^i$ . В этом случае их необходимо группировать, явно указав одно и то же выходное состояние.

Для учета неопределенности во входных значениях будем использовать нечеткие функции принадлежности треугольной формы  $\mu(y^i)(x^i)$  [11]. Данная функция значений входных параметров  $x^i$ , присущих состоянию системы  $i$  позволяет выполнять логический вывод даже в том случае, когда вектор входных значений содержит значения, не совпадающие в точности со значениями, использующимися в antecedентах правил.

### 3 Алгоритм формирования выходных данных на основе иерархической базы правил

На рисунке 4 представлен алгоритм принятия решения с использованием иерархической нечеткой базы правил с нечетким логическим выводом, основанном на подходе Мамдани [11].

Предварительно на основе метамодели формируется первый уровень базы правил (generate abstract level rule base), который не будет изменяться до тех пор, пока не произойдут изменения в самой метамодели интегрируемой ИС.

Алгоритм, представленный на рисунке 4, состоит из следующих шагов:

- Входные данные (input data), представленные в виде кортежа данных ключ-значение ( $inp1 = 7$ ) разного типа (целочисленные, строковые, дата

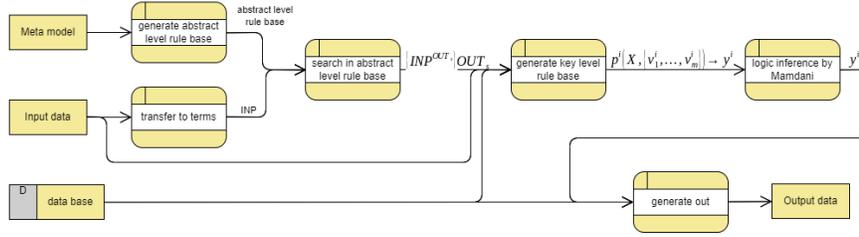


Рис. 4. Алгоритм принятия решений

и логические переменные типа boolean), преобразуются в лингвистические термины (transfer to terms), представленные в виде  $INP = \{INP_1, INP_2, \dots, INP_z\}, z \in N$ .

- Используя базу правил первого уровня (abstract level rule base) и преобразованные входные данные ( $INP$ ), осуществляется логический вывод (search in abstract level rule base), представленный в виде  $\{\{INP^{OUT_s}\}, OUT_s\}, s \in N$ .
- Результат выполнения правила первого уровня ( $\{INP^{OUT_s}\}, OUT_s$ ), исходные входные данные (input data), представленные в виде кортежа данных ключ-значение ( $inp1 = 7$ ) разного типа (целочисленные, строковые, дата и логические переменные типа boolean), и база данных интегрируемой ИС (data base) участвуют в динамическом формировании правил второго уровня (generate key level rule base), математическое представление которых представлено в формуле 3.
- В процессе нечеткого логического вывода (logic inference by Mamdani), основанного на подходе Мамдани, получается результат выполнения правила ( $y^i$ ) на основе базы правил второго уровня.
- На заключительном этапе (generate out) формируются подходящие выходные данные (output data), представленные в виде картежа данных ключ-значение ( $out1 = 7$ ) разного типа (целочисленные, строковые, дата и логические переменные типа boolean). Заключительный этап использует в качестве входных данных базу данных интегрируемой ИС (data base) и результат выполнения правила ( $y^i$ ) на основе базы правил второго уровня.

Таким образом, происходит процесс принятия решений на основе иерархической нечеткой базы правил с нечетким логическим выводом.

#### 4 Иллюстративный пример иерархической базы правил

В качестве примера рассмотрим полученную базу правил на основе информационной системы, предназначенной для обработки заявок, faveo service desk [20]. В работе [] отображена метамодель данных, представленная на

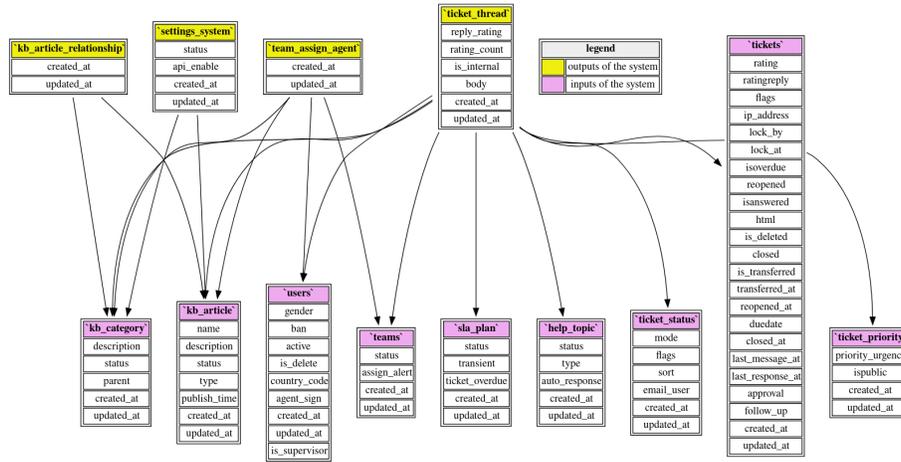


Рис. 5. Метамодел информационная система faveo

рисунке 5. На основе метамодели строится база правил первого уровня, отображенная на рисунке 6. Первый уровень содержит лингвистические термы, отображающие зависимость ключевых процессов от входных данных. Входные данные предварительно были преобразованы в лингвистическое описание.

```

IF
  INP = {'kb_article', 'kb_category'}
  THEN OUT = {'kb_article', 'kb_category'}, 'kb_article_relationship';
IF
  INP = {'timezone', 'time_format', 'date_format', 'date_time_format'}
  THEN OUT = {'timezone', 'time_format', 'date_format', 'date_time_format'}, 'settings_system';
IF
  INP = {'teams', 'users'}
  THEN OUT = {'teams', 'users'}, 'team_assign_agent';
IF
  INP = {'tickets', 'users', 'ticket_source'}
  THEN OUT = {'tickets', 'users', 'ticket_source'}, 'ticket_thread';

```

Рис. 6. База правил первого уровня информационной системы faveo

При сравнении рисунков 5 и 6 видны отличия между названиями входных сущностей для метамодели и входных данных, представленных в виде лингвистических термов в условии правила. Это связано с тем, что входные данные преобразовываются перед тем как формировать базу правил первого уровня. При переходе по ссылке одного из правил, представленных на рисунке 6, динамически формируется база правил второго уровня. На рисунке 7 представлен пример базы правил второго уровня для ключевого процесса ticket\_thread, отображенного на рисунке 5.

Условия правил содержат ключевые параметры с целочисленными значениями. В процессе нечеткого логического вывода осуществляется поиск

```

IF
ticket_id=14 AND user_id=1 AND source=NULL
THEN
id=23;
IF
ticket_id=15 AND user_id=1 AND source=NULL
THEN id=24;
IF
ticket_id=8 AND user_id=1 AND source=NULL
THEN id=32;
IF
ticket_id=15 AND user_id=1 AND source=NULL
ticket_id=14 AND user_id=1 AND source=NULL
ticket_id=13 AND user_id=1 AND source=NULL
ticket_id=12 AND user_id=1 AND source=NULL
ticket_id=11 AND user_id=1 AND source=NULL
ticket_id=10 AND user_id=1 AND source=NULL
ticket_id=9 AND user_id=1 AND source=NULL
THEN id=31;

```

**Рис. 7.** База правил второго уровня информационной системы faveo для ключевого процесса ticket\_thread

подходящего правила с использованием функции принадлежности, в результате которого формируются понятные для пользователя данные, соответствующие состоянию ключевого процесса при переданных входных данных. Пример вывода результата представлен в таблице 2.

Таким образом, при заданных входных данных формируется результат, который отражает состояние ключевого процесса информационной системы faveo service desk. Этот вывод подтверждает целесообразность применения предложенного подхода, так как сформированный результат отражает реальные данные при работе с системой заявок faveo service desk и получение результата не зависит от запуска самой интегрируемой информационной системы, что является преимуществом при отображении данных и параллельной работе интегрируемой информационной с комплексом PLM.

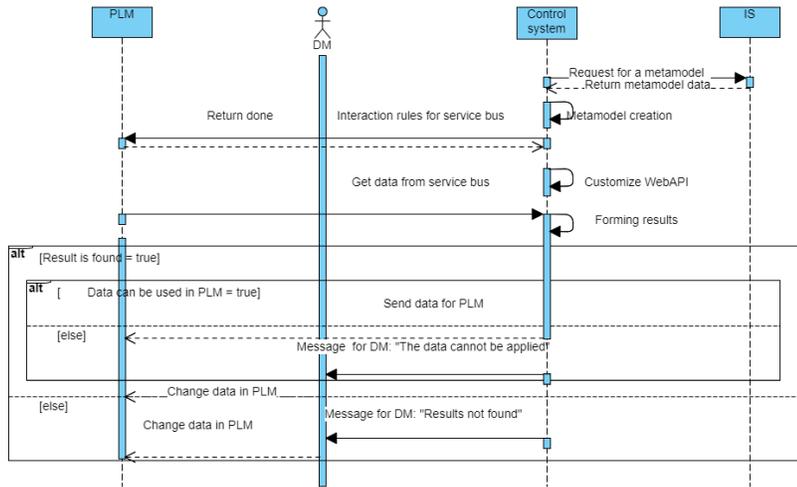
## 5 Управляющая система в общем комплексе управления

Предлагаемый подход подразумевает использование управляющей системы для сокращения временных и трудовых затрат оператора, участвующего в процессе управления. Управляющая система (УС) — это программный инструмент, реализующий предлагаемый подход управления на основе данных [1]. Управление включает в себя конфигурирование и обмен данными, в данном случае, за счет использования метамодели интегрируемой ИС и иерархической базы правил с нечетким логическим выводом. Управляющая система выступает адаптером для обеспечения взаимодействия интегрируемой ИС с программным комплексом PLM.

На рисунке 8 представлен процесс организации управления информационной системой с участием управляющей системой.

При сравнении рисунков 2 и 8 отметим, что из процесса управления исключается роль аналитика и оператора, их функции выполняет УС. Также стоит отметить, что в процессе управления и в том, и в другом случае присутствуют несколько условий:

1. Существование возможной реакции интегрируемой ИС на входные данные;
2. Влияние полученной реакции на состояние PLM системы (возможность добавления данных в комплекс без ошибок и рисков для всего производственного комплекса).



**Рис. 8.** Процесс организации взаимодействия и поддержания системы в состоянии работоспособности с участием УС и ЛПП

Таким образом, можно увидеть, что основная задача ЛПП состоит в корректировке данных общего комплекса PLM, не взаимодействуя при этом с интегрируемой информационной системой. Общий процент сценариев, где должен участвовать человек при работе с системой без УС составляет 85%, а общее количество сценариев при работе с системой с УС составляет 18%. Основной задачей является, в случае использования УС, реакция на ошибки при формировании результатов работы интегрируемой ИС. Получаем снижение роли лица, принимающего решения на 67%.

## 6 Заключение

В статье приведен подход управления на основе данных применительно к производственным процессам. Подход подразумевает использование метамодели данных информационной системы и иерархическую базу правил поведения информационной системы с нечетким логическим выводом. Приведены основные выводы, подтверждающие целесообразность использования данного подхода в задачах отображения данных информационных систем и управления информационной системой с целью снижения человеческого фактора при организации управления сторонними системами на производстве. Задачи являются актуальными на многих производствах Российской Федерации, в том числе на производственных предприятиях Ульяновской области.

## Список литературы

1. Terzi S., Bouras A., Dutta D., Garetti M., Kiritsis D. Product Lifecycle Management – from its History to its New Role. *International Journal Product Lifecycle Management*, 2010, vol. 4, no. 4, pp. 360–389.
2. Menge F. Enterprise Service Bus. Free and Open Source Software Conference, 2007. Available at: [https://programm.froscon.org/2007/attachments/15-falko\\_menge\\_-\\_enterprise\\_service\\_bus.pdf](https://programm.froscon.org/2007/attachments/15-falko_menge_-_enterprise_service_bus.pdf)
3. Shvedenko V. N., Shchekochikhin O. V. Architecture of an integrated information system that ensures behavioral properties (In russian) // *Scientific and technical bulletin of information technologies, mechanics and optics*. – 2016. – V. 16. – No. 6. – pp. 1078-1083.
4. Chekal E. G. Reliability of information systems: textbook: in 2 parts. Part 1 (In russian) / E. G. Chekal, A. A. Chichev. – Ulyanovsk: UIGU, 2012. – 118 p.
5. Feng G. A survey on analysis and design of model-based fuzzy control systems // *IEEE Transactions on Fuzzy systems*. – 2006. – T. 14. – №. 5. – С. 676-697.
6. R. Sudarsan, S.J. Fenves, R.D. Sriram, F. Wang, A product information modeling framework for product lifecycle management / *Computer-Aided Design* 37 (2005) 1399–1411
7. Alahi MEE, Sukkuea A, Tina FW, Nag A, Kurdtongmee W, Suwannarat K, Mukhopadhyay SC. Integration of IoT-Enabled Technologies and Artificial Intelligence (AI) for Smart City Scenario: Recent Advancements and Future Trends. *Sensors*. 2023; 23(11):5206. <https://doi.org/10.3390/s23115206>
8. Sacks R, Brilakis I, Pikas E, Xie HS, Girolami M. Construction with digital twin information systems. *Data-Centric Engineering*. 2020;1:e14. doi:10.1017/dce.2020.16
9. Ch. Cimini, F. Pirola, R. Pinto, S. Cavalieri, A human-in-the-loop manufacturing control architecture for the next generation of production systems // *Journal of Manufacturing Systems*. - 2020. - vol. 54. - pp. 258-271
10. Hou Z.S., Xu J.X. On Data-driven Control Theory: the State of the Art and Perspective. *Acta Automatica Sinica*, 2009, vol. 35, no. 6, pp. 650–667.
11. Mamdani, E.H.: Application of fuzzy algorithms for control of simple dynamic plant. In: *Proceedings of the Institution of Electrical Engineers*, vol. 121, pp. 1585–1588. IET (1974).
12. Gonza'lez, A., Pe'rez, R., Verdegay, J.L.: Learning the structure of a fuzzy rule: a genetic approach. *Fuzzy Syst. Artif. Intell.* 3(1), 57–70 (1994)
13. Duckstein, L., et al.: *Fuzzy Rule-Based Modeling with Applications to Geophysical, Biological, and Engineering Systems*, vol. 8. CRC Press, Boca Raton (1995)
14. Chi, Z., Yan, H., Pham, T.: *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*, vol. 10. World Scientific, Singapore (1996)
15. Razak, T.R., Fauzi, S.S.M., Gining, R.A.J., Ismail, M.H., Maskat, R.: Hierarchical fuzzy systems: interpretability and complexity. *Indones. J. Electr. Eng. Inform.* 9(2), 478–489 (2021)
16. Zhang, X., Onieva, E., Perallos, A., Osaba, E., Lee, V.C.: Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction. *Transport. Res. C: Emerg. Technol.* 43, 127–142 (2014)
17. Duan, J.-C., Chung, F.-L.: Cascaded fuzzy neural network model based on syllogistic fuzzy reasoning. *IEEE Trans. Fuzzy Syst.* 9(2), 293–306 (2001)
18. Jang, J.-S.R., et al.: Fuzzy modeling using generalized neural networks and Kalman filter algorithm. *AAAI* 91, 762–767 (1991)
19. Komartsova L. G. Evolutionary methods for forming fuzzy rule bases. – 2011.

14 No Author Given

20. Продукт службы поддержки Faveo Service Desk — URL:  
<https://www.faveohelpdesk.com/servicedesk/?lang=ru> (accessed at 1.05.2024)