

Ситуационно-ориентированные базы данных: взаимодействие иерархических ситуационных моделей с внешней средой

А. С. Гусаренко • К. В. Миронов • В. В. Миронов

Уфимский университет науки и технологий

Проект «ситуационно-ориентированные базы данных» предусматривает хранение и обработку данных с привязкой к состояниям динамической модели в виде графа переходов. В качестве динамической модели используется иерархическая ситуационная модель (ИСМ), в которой совокупность «плоских» моделей (графов переходов) организована в иерархию за счёт того, что состояния модели некоторого уровня, в свою очередь, могут содержать вложенные субмодели со своими подсостояниями. В статье предлагается концепция взаимодействия ИСМ с внешней средой, обеспечивающая обмен данными в контексте текущих состояний ИСМ. Рассматриваются элементы ИСМ, и их функциональность, обеспечивающие такое взаимодействие: интерфейс состояния; данные, ассоциированные с состояниями; данные, отправляемые и получаемые через интерфейс, внешние события, управляющие переходами текущих состояний ИСМ. Вводится понятие триггера перехода, управляющего срабатыванием перехода и сменой текущего состояния. Обсуждаются функциональные компоненты триггера перехода: прологовые действия, предикат перехода, эпилоговые действия. С учётом введенных элементов ИСМ, обеспечивающих взаимодействие с внешней средой, строятся информационные модели структуры ИСМ в виде двух составных частей: порождающей модели, представляющей неизменную часть ИСМ независимо от текущего состояния, и порождённой модели, отражающей текущее состояние ИСМ.

Информационная модель; динамическая модель; конечный автомат; граф переходов; иерархическая ситуационная модель; интерфейс.

ВВЕДЕНИЕ

Современные бизнес-процессы зачастую носят ярко выраженный динамический характер, когда протекание, поведение участников, управление и принятие решений в них меняется в зависимости от сложившейся внешней и внутренней ситуации [Al-23, Кон24, Кон24b]. В рамках традиционных методологий системного анализа и моделирования, таких как IDEF, UML, BPMN [Oli26, Kal25, Cor20], используются системные модели, ориентированные на описание отдельных аспектов бизнес-процессов: функциональных, информационных, онтологических и др. [Pet23, Vuk00, Ben94]. Динамика бизнес-процессов описывается с помощью так называемых динамических моделей, в основе которых, как правило, лежат модели с конечным числом состояний, такие как графы переходов (конечные автоматы) и сети Петри [Özs25, Rol25, Gir99, Bep22, Pet23, Bör00]. В данной работе рассматриваются вопросы моделирования динамики бизнес-процессов в рамках подхода, названного авторами «ситуационно-ориентированными базами данных» [Mip17, Mir21, Mir22]. Этот подход предусматривает интеграцию информационных моделей, описывающих используемые бизнес-процессом данные, с динамическими моделями, задающими и отслеживающими состояния (ситуации) бизнес-процесса, в интересах управления и принятия решений.

Гусаренко А. С., Миронов К. В., Миронов В. В. Ситуационно-ориентированные базы данных: взаимодействие иерархических ситуационных моделей с внешней средой // СИИТ. 2026. Т. 8, № 4(28). С. 87-100. DOI: [10.54708/SIIT-2026-no4-p87](https://doi.org/10.54708/SIIT-2026-no4-p87). EDN: YSUCQN.

Gusarenko A. S., Mironov K. V., Mironov V. V. Situation-oriented databases: interaction of hierarchical situational models with the external environment. SIIT. 2026; 8(4(28)): 87-100. DOI: [10.54708/SIIT-2026-no4-p87](https://doi.org/10.54708/SIIT-2026-no4-p87). EDN: YSUCQN. (In Russian).

В основе ситуационно-ориентированной базы данных лежит особый вид динамической модели – иерархическая ситуационная модель (ИСМ) [Мир08, Мир86, Мир14], которая представляет собой иерархию графов переходов (моделей конечных состояний – Finite State Model), где вершины соответствуют состояниям бизнес-процесса, а дуги задают переходы состояний. Иерархия образуется за счёт того, что состояния могут быть укрупненными, т. е. в свою очередь могут содержать вложенные графы переходов (субмодели).

Применение в качестве основы для веб-приложений. Веб-приложение, функционирующее под управлением ИСМ, должна предусматривать взаимодействие ИСМ с внешней средой в плане обмена данными [Мир09, Мир10b, Мир16]. В известных разработках взаимодействие с внешней средой не раскрывались непосредственно в самой модели на концептуальном уровне. В ИСМ предусматривались модули-действия (Actions) – «черные ящики», привязанные к состояниям и переходам и выполняющие нужные операции по обмену данными и по их обработке в контексте текущей ситуации. В работах по этой теме рассматривались технические детали взаимодействия с различными типами и форматами внешних данных. В данной статье ставится задача в определенной степени прояснить содержимое этих «черных ящиков», дополнив ИСМ на концептуальном уровне абстракции элементами, ответственными за взаимодействие с внешней средой.

ЗАДАЧА ВЗАИМОДЕЙСТВИЯ ИСМ С ВНЕШНЕЙ СРЕДОЙ

Рассмотрим в общих чертах схему взаимодействия ИСМ с внешней средой (рис. 1).

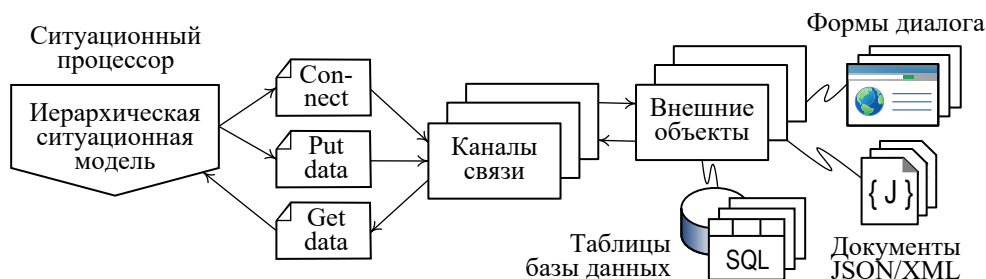


Рис. 1 Укрупнённая схема взаимодействия ИСМ с внешней средой

На этой схеме ИСМ, находящаяся под управлением *ситуационного процессора*, отслеживает текущие состояния управляемого процесса. В соответствии с текущими состояниями (в общем случае – в соответствии с историей состояний) формируются команды к *внешним объектам*, внешним по отношению к ИСМ. Команды передаются через соответствующие каналы связи и, как правило, предусматривают установление соединения с внешним объектом (Connect), отправку данных объекту (Put data), получение данных от объекта (Get data).

Внешние объекты, как правило, разнородные (гетерогенные) [Гус24, Мир21, Мир17]. На рисунке представлены примеры трёх характерных категорий таких объектов:

- реляционные базы данных (локальные или удаленные), к таблицам которых применяются операции извлечения, вставки, удаления, обновления строк данных. Доступ осуществляется с помощью запросов на языке SQL через соединение (канал), для открытия которого требуется установить ряд параметров (адрес сервера базы данных и др.) [Мир16b, Мир17b];
- веб-сервисы или внешние приложения, с которыми производится обмен документами в формате JSON или XML (API – Application Program Interface) [Мир17, Мир20, Фри17];
- диалоговые формы, предназначенные для взаимодействия с конечными пользователями (GUI – Graphic User Interface). Они позволяют как отображать пользователю ту или иную информацию, так и получать информацию от пользователя с помощью полей ввода (Inputs) и других элементов управления (Controls). Обычно такое взаимодействие происходит по веб-протоколу HTTP с помощью запросов POST [Мир10, Кан13].

Рассмотрим детали взаимодействия ИСМ с внешней средой на примере диалоговых форм. На рис. 2 приведен макет простой диалоговой формы, предназначенной для получения от пользователя определенных сведений.

Рис. 2 Макет формы на стороне клиента

Она содержит текстовые данные, отображаемые пользователю, поля для ввода пользователем текстовых данных, элементы управления (кнопки) для ввода пользователем различных команд манипуляции формой, в том числе команд, приводящих к отправке формы на сервер. Отметим, что в данном случае это пример независимой формы. В случае иерархии это может быть отдельный фрагмент формы, отвечающий за информацию, относящуюся к данному состоянию модели [Кан14, Кан14b].

Таким образом, с формой диалога связаны три вида данных:

- *метаданные*, поступающие с сервера и задающие «каркас» формы – её информационную структуру и внешний дизайн. Метаданные неизменны для всех случаев использования формы этого типа, т. е. они не зависят от текущего состояния ИСМ;
- *текущие серверные данные*, поступающие с сервера и используемые для отображения в конкретном экземпляре формы этого типа. Эти данные относятся к текущему состоянию ИСМ;
- *текущие пользовательские данные*, введенные пользователем в конкретном экземпляре формы этого типа и отправленные на сервер. Эти данные также относятся к текущему состоянию ИСМ – они предназначены для хранения или обработки на сервере.

Анализ показывает, что эти виды данных характерны не только для случая диалоговых форм, но и для других категорий объектов внешней среды, с которыми должна взаимодействовать ИСМ.

КОНЦЕПЦИЯ ВЗАИМОДЕЙСТВИЯ ИСМ С ВНЕШНЕЙ СРЕДОЙ

Порождающая и порождённая модели

Концепция взаимодействия ИСМ с внешней средой базируется на идее разделения ИСМ на порождающую и порождённую части. Это идея отделения неизменной части ИСМ от изменяющейся в результате смены текущего состояния (рис. 3).

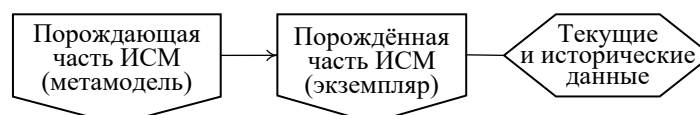


Рис. 3 Порождающая и порождённая части ИСМ

Порождённая модель содержит иерархию текущих состояний (а также историю смены состояний, если это ИСМ с сохранением предыстории). Сведения о текущих состояниях и ассоциированных данных, история состояний и данных (если это необходимо) хранятся в порождённой модели. Таким образом, порождающая и порождённая модели состоят в отношении

«класс–экземпляр», т. е. порождающая модель выступает для порождённой модели в качестве метамодели (схемы), а конкретная порождённая модель представляет собой один из возможных экземпляров порождающей модели.

Для отражения взаимодействия с внешней средой на уровне модели графы переходов в ИСМ дополнены специальными элементами, обеспечивающими обмен и обработку данных в контексте текущего состояния. На рис. 4 на абстрактном примере двухуровневой ИСМ поясняются два вида элементов взаимодействия – ассоциированные данные и интерфейсы взаимодействия.

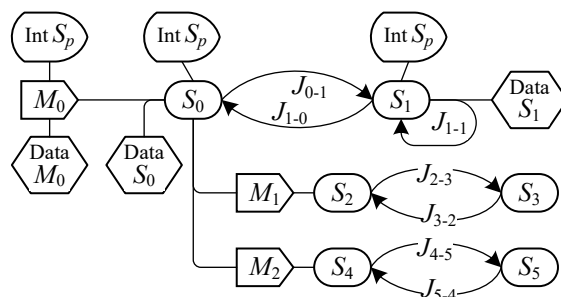


Рис. 4 Элементы ИМС для обмена данными

Структура ИСМ. Модель верхнего уровня M_0 содержит два состояния: S_0 (начальное) и S_1 . Состояние S_0 содержит исходящий переход J_{0-1} в состояние S_1 , а состояние S_1 – переход J_{1-0} в состояние S_0 и петлевой переход J_{1-1} . При этом состояние S_0 укрупненное, оно содержит две субмодели второго уровня: M_1 и M_2 , каждую с двумя подсостояниями. Подсостояния детализируют родительское состояние S_0 в плане своей субмодели.

Данные. С моделью в целом и с её отдельными состояниями могут быть связаны (ассоциированы) соответствующие данные (Data). На рис. 4 ассоциированные данные показаны только в модели M_0 : Data M_0 , Data S_0 и Data S_1 , но они могут быть и в субмоделях, и в состояниях субмоделей.

Интерфейсы. Для обмена данными с внешней средой предусмотрены интерфейсы (Interface) на уровне как субмодели в целом, так и её отдельных состояний. В модели M_0 – это Int M_0 , Int S_0 и Int S_1 . Функциональность интерфейсов рассматривается ниже.

Иерархия интерфейсов. Когда модель находится в некотором состоянии, через ее интерфейс происходит обмен данными и событиями, относящиеся к данному состоянию. Субмодели и их состояния, в свою очередь, могут содержать интерфейсы (субинтерфейсы). В результате образуется иерархия интерфейсов (см. работы [Кан13, Кан14, Кан14b] – концепция иерархических виджетов). Например, сложная форма диалога для некоторого состояния может содержать в своем составе фрагменты, соответствующие отдельным подсостояниям этого состояния.

Смена текущих состояний

Срабатывание переходов. В каждый момент времени только одно из состояний активной модели является текущим. В свою очередь, только одно из состояний каждой субмодели текущего состояния является текущим. Смена состояний происходит при срабатывании переходов, исходящих из текущего состояния, с учётом их приоритетов.

Триггер перехода. Триггер определяет механизм, обеспечивающий смену состояния. Срабатывание переходов связано с наступлением определенных *событий* – внешних или внутренних. Внешнее событие порождается внешней средой, а внутреннее событие – текущими состояниями и данными внутри ИСМ. Событие – это абстракция, для обнаружения и идентификации которого требуется обработка внешних и/или внутренних данных. Для выявления событий введено понятие триггера перехода (рис. 5).

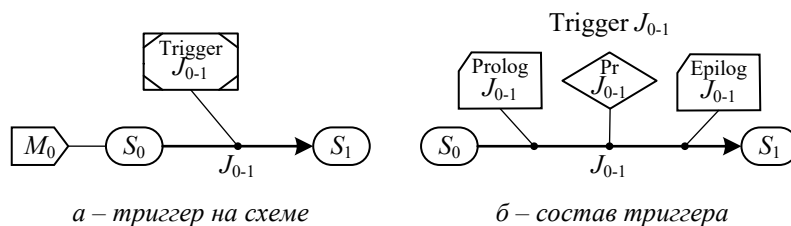


Рис. 5 К понятию триггера перехода

Триггер перехода – это модуль, привязанный к переходу (ассоциированный с переходом), который обрабатывает внешние и внутренние данные с целью обнаружения и идентификации события и реализации смены состояния. Триггер $\text{Trigger } J_{0-1}$ на рис. 5, а ассоциирован с переходом J_{0-1} , переводящим модель M_0 из состояния S_0 в состояние S_1 . Внутреннее устройство триггера (рис. 5, б) предусматривает три вида действий, выполняемых последовательно:

1) *Прологовые действия* $\text{Prolog } J_{0-1}$, выполняемые в самом начале проверки активности перехода. Они предназначены для обработки данных, полученных через интерфейс текущего состояния, а также других данных порождённой модели с целью идентификации события, вызывающего срабатывание данного перехода.

2) *Проверка предиката перехода* $\text{Pr } J_{0-1}$, т. е. проверка условия, свидетельствующего о наступлении события, вызывающего срабатывание перехода J_{0-1} .

3) *Эпилоговые действия* $\text{Epilog } J_{0-1}$. Эти действия выполняются в случае истинности предиката $\text{Pr } J_{0-1}$. Они реализуют в порождённой модели m_0 , $\text{type}(m_0) = M_0$ смену текущего состояния $S_0 \rightarrow S_1$. Для этого:

- во-первых, в порождённой модели m_0 создаётся новый экземпляр s_1 целевого состояния S_1 , $\text{type}(s_1) = S_1$. Он либо включается в голову списка истории состояний (в случае перехода с сохранением предыстории), либо заменяет список истории состояний (в случае перехода с отсечением предыстории);
- во-вторых, на основе данных, полученных через интерфейс прежнего текущего состояния S_0 , а также других данных порождённой модели, формируются данные, которые присоединяются к новому текущему состоянию S_1 .

Процедура обмена данными

Поясним взаимодействие ИСМ с внешней средой через интерфейс состояния на примере взаимодействия с пользователем на основе диалоговых форм (см. рис. 2). В этом примере предполагается, что ИСМ функционирует на серверной стороне веб-приложения, а пользователь находится на клиентской стороне и взаимодействует с сервером через веб-браузер. Для организации взаимодействия используются диалоговые формы, позволяющие на основе протокола HTTP отображать пользователю данные, загруженные с сервера, а также получать от пользователя данные и команды управления и передавать их на сервер. В ответ на запрос клиента на сервере формируется нужная форма в HTML-формате, которая отправляется клиенту.

С точки зрения сервера, функционирующего под управлением ИСМ, для организации интерфейса с внешней средой в ИСМ должно быть предусмотрено формирование запроса на организацию связи с клиентом в соответствии с протоколом сетевого обмена данными, а также: 1) хранение метаданных, задающих структуру и дизайн интерфейса, 2) извлечение серверных данных, отправляемых через интерфейс на клиентскую сторону, 3) сохранение и обработку полученных клиентских данных.

Эта функциональность поясняется на рис. 5, где приведен фрагмент модели M с двумя состояниями S_p и S_q . Состояние S_p содержит два исходящих перехода: петлевой переход $J_{p/p}$ и переход $J_{p/q}$ в состояние S_q . Предполагается, что в рассматриваемый момент состояние S_p является текущим.

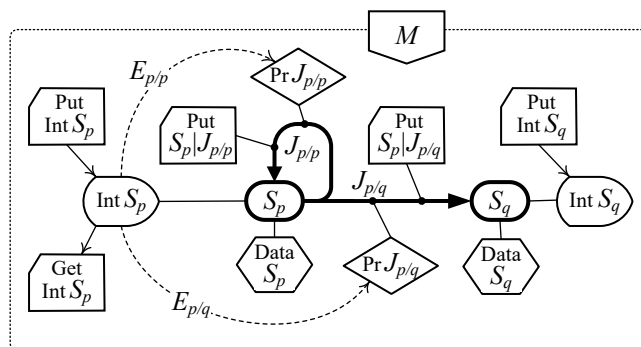


Рис. 5 К концепции интерфейсов состояний

Следующие элементы ИСМ обеспечивают взаимодействие с внешней средой.

- $\text{Int}(S_p)$ – интерфейс состояния S_p , предоставляемый ИСМ, когда это состояние является текущим. На логическом уровне эта абстракция подразумевает, во-первых, метаданные, задающие структуру и дизайн информации (например, диалоговой формы), отправляемой пользователю, во-вторых, инфраструктуру, обеспечивающую такую отправку (например, HTTP Request) и последующее получение результата (HTTP Response). Всё это должно задаваться заранее в порождающей модели и не должно зависеть от текущей порождённой модели.

- $\text{Put Int}(S_p)$ – данные, отправляемые через интерфейс состояния S_p во внешнюю среду. В случае диалоговой формы эти данные встраиваются в «каркас» формы перед ее отправкой пользователю. Эти данные относятся к текущим состояниям ИСМ, они формируются на основе данных, извлекаемых из порождённой модели в контексте текущего состояния. Например, эти данные могут формироваться на основе данных текущего состояния $\text{Data}(S_p)$, в других случаях для их формирования может потребоваться обращение к истории состояний модели M .

- $\text{Get Int}(S_p)$ – данные, получаемые через интерфейс состояния S_p из внешней среды, например данные, введенные пользователем в полях ввода диалоговой формы. Эти данные относятся к текущему состоянию, т. е. к порождённой модели – сохраняются в ней для последующей обработки, в том числе с целью идентификации внешних событий.

- $E(S_p)$ – внешние события, поступающие через интерфейс и приводящие к срабатыванию переходов и погружений, порождению новых экземпляров состояний и субмоделей. Внешнее событие – это абстракция концептуального уровня, связанная с интерпретацией тех или иных внешних сигналов. Для идентификации внешнего события, как правило, требуется обработка полученных внешних данных $\text{Get Int}(S_p)$. Например, при возврате на сервер заполненной пользователем диалоговой формы (событие физического уровня), необходимо предусмотреть обработку поступивших данных, чтобы определить, какие именно элементы управления формы были задействованы пользователем для указания своего решения (событие концептуального уровня).

Смена текущих состояний. При срабатывании переходов в результате внешних или внутренних событий происходит смена текущего состояния. Исходящие переходы $J_{p/p}$ и $J_{p/q}$ текущего состояния S_p (см. рис. 5) управляются событиями $E_{p/p}$ и $E_{p/q}$ соответственно, которые идентифицируются в результате прологовой обработки данных $\text{Get Int } S_p$, поступивших через интерфейс $\text{Int } S_p$, и определяют истинность предикатов $\text{Pr } J_{p/p}$ и $\text{Pr } J_{p/q}$.

В случае срабатывания петлевого перехода $J_{p/p}$ (истинный предикат $\text{Pr } J_{p/p}$), например при нажатии пользователем кнопки «Отмена» или при обнаружении ошибок во введенных пользователем данных при нажатии кнопки «Ввод» (см. рис. 2) происходит переход $S_p \rightarrow S_p$. В результате этого в новый экземпляр s_p , $\text{type}(s_p) = S_p$ в порождённой модели записываются данные $\text{Put}(S_p | J_{p/p})$, например сведения об обнаруженных ошибках пользовательских данных.

На основе этих сведений будет сформирован новый вариант данных $\text{Put Int}(S_p)$, встраиваемый в форму для нового экземпляра состояния S_p .

В случае срабатывания перехода $J_{p/q}$ (истинный предикат $\text{Pr } J_{p/q}$), например, при отсутствии ошибок во введенных пользователем данных при нажатии кнопки «Ввод» (см. рис. 2), происходит переход $S_p \rightarrow S_q$. В результате этого в новый экземпляр s_q , $\text{type}(s_q) = S_q$ в порождённой модели записываются данные $\text{Put}(S_q | J_{p/q})$, например данные, введенные пользователем. На основе этих сведений будет сформирован новый вариант данных $\text{Put Int}(S_q)$, встраиваемый в форму для нового экземпляра состояния S_q .

ИНФОРМАЦИОННЫЕ МОДЕЛИ ИСМ

В предыдущих разделах была рассмотрена на содержательном уровне семантика ИСМ в плане обмена данными с внешней средой. На этой основе в данном разделе строятся информационные модели ИСМ – отдельно для порождающей и порожденной частей. Информационные модели представлены на концептуальном уровне абстракции. Они представляют набор сущностей и отношений (связей) между сущностями в форме, независимой от среды их реализации. В дальнейшем эти модели служат основой для разработки моделей логического уровня абстракции, учитывающих возможности и ограничения выбранной среды реализации.

Информационная модель порождающей части ИСМ

Информационная модель порождающей части ИСМ содержит иерархию типов информационных сущностей, находящихся в отношениях «родитель–ребенок» (рис. 6). Каждый тип сущности подразумевает множество соответствующих экземпляров этого типа, состав которых зависит от конкретного воплощения ИСМ.

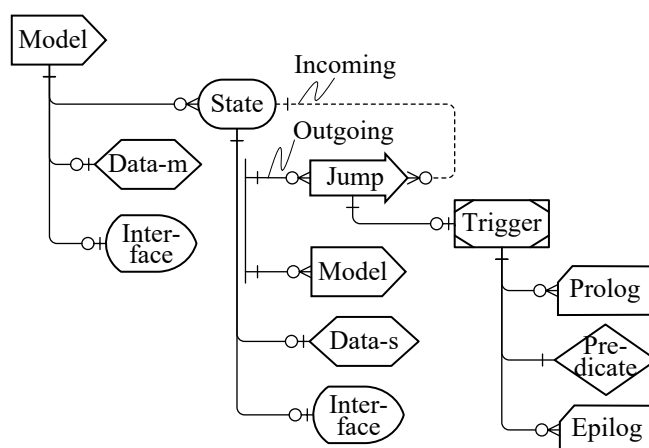


Рис. 6 Концептуальная модель порождающей части ИСМ

Дочерние сущности иерархии размещены ниже родительских, связаны с ними иерархическими ($+\circ\infty$) или условными ($+\circ+$) связями. Обозначенная пунктиром связь Incoming («входящий») представляет собой межуровневую ссылку в иерархии.

Сущности упорядочены внутри своего родителя по принципу «сверху вниз». При этом двойной коннектор сущностей типа Jump и Model означает, что их экземпляры могут располагаться вперемешку относительно друг друга.

Детальные пояснения к сущностям и связям порождающей части модели представлены в табл. 1.

Таблица 1

Сущности и их отношения в порождающей части ИСМ

Сущность	Содержание	Родитель	Отношение с родителем
Model	<i>Модель</i> (модель верхнего уровня, корневая модель). Контейнер для состояний, данных модели, интерфейса модели. Атрибуты специфицируют свойства экземпляров модели.	—	—
	<i>Модель (субмодель состояния)</i> . Структура аналогична модели верхнего уровня. Атрибуты специфицируют свойства экземпляров субмодели.	State	Субмодель всегда принадлежит одному родительскому состоянию. Состояние может иметь несколько субмоделей или вовсе их не иметь.
State	<i>Состояние</i> . Контейнер для данных состояния, интерфейса состояния, переходов, вложенных субмоделей. Атрибуты специфицируют свойства экземпляров состояния.	Model	Состояние всегда принадлежит одной родительской модели (субмодели). Модель (или субмодель) может иметь несколько состояний или вовсе их не иметь.
Jump	<i>Исходящий переход</i> . Контейнер для триггера перехода. Атрибуты задают тип перехода и другие свойства, которые определяют его функционирование. Атрибуты специфицируют свойства экземпляров перехода.	State	<i>Связь Outgoing</i> . Переход всегда принадлежит одному родительскому состоянию, для которого он исходящий (outgoing). Состояние может иметь несколько исходящих переходов или вовсе их не иметь (терминальное состояние). <i>Связь Incoming</i> . Переход всегда принадлежит одному родительскому состоянию, для которого он входящий (incoming). Состояние может иметь несколько входящих переходов или не иметь их вовсе.
Data-m	<i>Данные модели</i> . Спецификация данных, ассоциированных с моделью: имя, тип, ограничения и другие свойства.	Model	Данные модели всегда принадлежат одной родительской модели (субмодели). Модель (субмодель) может иметь или не иметь данных.
Data-s	<i>Данные состояния</i> . Спецификация данных, ассоциированных с состоянием: имя, тип, ограничения и другие свойства.	State	Данные состояния всегда принадлежат одному родительскому состоянию. Состояние может иметь или не иметь данных.
Interface	<i>Интерфейс модели</i> . Спецификация канала связи, позволяющего обмениваться данными с внешней средой на уровне конкретной модели.	Model	Интерфейс модели всегда принадлежит одной родительской модели (субмодели). Модель (субмодель) может иметь или не иметь интерфейс.
	<i>Интерфейс состояния</i> . Спецификация канала связи, позволяющего обмениваться данными с внешней средой на уровне конкретного состояния модели.	State	Интерфейс состояния всегда принадлежит одному родительскому состоянию. Состояние может иметь или не иметь интерфейс.
Trigger	<i>Триггер перехода</i> . Контейнер для действий, связанных со срабатыванием перехода.	Jump	Триггер всегда принадлежит одному родительскому переходу. Переход может иметь или не иметь триггер (переход без триггера всегда активный).
Prolog	<i>Пролог (блок прологовых действий)</i> . Спецификация действий по обработке данных, предваряющих проверку срабатывания триггера.	Trigger	Пролог всегда принадлежит одному родительскому триггеру. Триггер может иметь несколько прологов, выполняемых последовательно, или ни одного.
Predicate	<i>Предикат перехода</i> . Спецификация условия срабатывания триггера.	Trigger	Предикат всегда принадлежит одному родительскому триггеру. Триггер может иметь один предикат или ни одного (переход с триггером без предиката всегда активный).
Epilog	<i>Эпилог (блок эпилоговых действий)</i> . Спецификация действий по обработке данных, завершающих срабатывание триггера.	Trigger	Эпилог всегда принадлежит одному родительскому триггеру. Триггер может иметь несколько эпилогов, выполняемых последовательно, или ни одного.

Информационная модель порождённой части ИСМ

Информационная модель порождённой части ИСМ (рис. 7) отражает факты создания порождающей частью ИСМ экземпляров соответствующих текущих сущностей в ходе отслеживания текущего состояния. Она содержит множество типов информационных сущностей, соответствующих сущностям порождающей части, а также связей, задающих отношения между ними (внутренние связи) и с сущностями порождающей части ИСМ (внешние связи). Внешние связи определяют контекст для интерпретации порождённых сущностей. Имена элементов порождённой части начинаются со строчных букв, чтобы отличить от соответствующих элементов порождающей части модели.

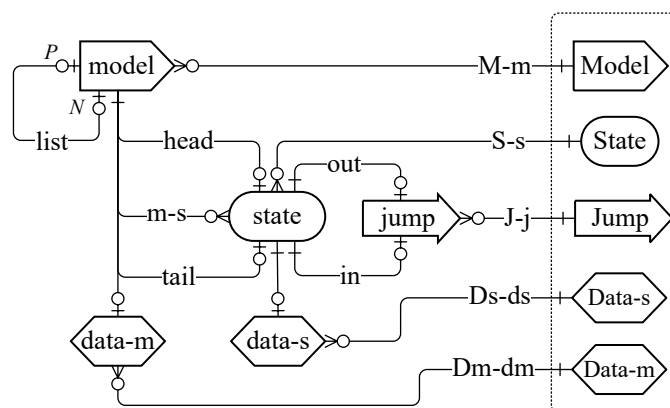


Рис. 7 Концептуальная модель порождённой части ИСМ

Детальные пояснения к сущностям и связям порождённой модели представлены в табл. 2.

Таблица 2

Сущности и отношения в порождённой части ИСМ

Сущность	Содержание	Родитель	Отношение с родителем
model	Экземпляр модели (субмодели). Отражает факт того, что соответствующая порождающая модель активна в настоящее время или была таковой ранее. Атрибуты задают моменты времени, связанные с этим фактом.	Model	<i>Внешняя связь M-m.</i> Экземпляр модели всегда соответствует одной родительской порождающей модели (субмодели). Порождающая модель (субмодель) может иметь несколько порождённых экземпляров или вовсе их не иметь. <i>Внутренняя связь list.</i> Экземпляры одной порождающей модели хронологически упорядочены в списки. Экземпляр может иметь или не иметь один <i>следующий</i> экземпляр (<i>N – next</i>) и один <i>предыдущий</i> экземпляр (<i>P – previous</i>).
state	Экземпляр состояния. Отражает факт того, что соответствующее порождающее состояние – текущее в настоящее время или было ранее. Атрибуты задают моменты времени, связанные с этим фактом.	State	<i>Внешняя связь S-s.</i> Экземпляр состояния всегда соответствует одному родительскому порождающему состоянию. Порождающее состояние может иметь несколько порождённых экземпляров или вовсе их не иметь.
		model	<i>Внутренняя связь m-s.</i> Экземпляр состояния всегда соответствует одному родительскому экземпляру модели. Экземпляр модели может иметь несколько экземпляров состояний или вовсе их не иметь. <i>Внутренняя связь head («голова»).</i> Один из множества экземпляров состояния, принадлежащих одному родительскому экземпляру модели, – самый поздний по времени порождения – является для него <i>текущим</i>. Экземпляр модели может иметь только один текущий экземпляр состояния.

Сущность	Содержание	Родитель	Отношение с родителем
jump	Экземпляр перехода. Отражает факт того, что соответствующий порождающий переход сработал. Атрибуты задают моменты времени, связанные с этим фактом.	Jump	<i>Внутренняя связь tail («хвост»)</i> . Один из множества экземпляров состояния, принадлежащих одному родительскому экземпляру модели, – самый ранний по времени порождения – является для него <i>исходным</i> . Экземпляр модели может иметь только один исходный экземпляр состояния.
		state	<i>Внешняя связь J-j</i> . Экземпляр перехода всегда соответствует одному (родительскому) порождающему переходу. Порождающий переход может иметь несколько порождённых экземпляров или вовсе их не иметь. <i>Внутренняя связь out («из»)</i> . Экземпляр перехода всегда соответствует одному (родительскому) экземпляру состояния, для которого он исходящий. Экземпляр состояния может иметь только один экземпляр исходящего перехода или ни одного (в случае текущего экземпляра состояния). <i>Внутренняя связь in («в»)</i> . Экземпляр перехода всегда соответствует одному (родительскому) экземпляру состояния, для которого он входящий. Экземпляр состояния может иметь только один экземпляр входящего перехода или ни одного (в случае исходного экземпляра состояния).
data-m	Экземпляр данных (блока данных модели). Задаёт множество значений для соответствующего порождающего блока данных модели.	Data-m	<i>Внешняя связь Dm-dm</i> . Экземпляр данных модели всегда соответствует одному (родительскому) порождающему блоку данных. Порождающий блок данных может иметь несколько порождённых экземпляров или вовсе их не иметь.
data-s	Экземпляр данных (блока данных состояния). Задаёт множество значений для соответствующего порождающего блока данных состояния.	Data-s	<i>Внешняя связь Dm-dm</i> . Экземпляр данных модели всегда соответствует одному (родительскому) порождающему блоку данных. Порождающий блок данных может иметь несколько порождённых экземпляров или вовсе их не иметь.

ЗАКЛЮЧЕНИЕ

Таким образом, в данной работе представлено развитие проекта ситуационно-ориентированных баз данных, который предусматривает хранение и обработку данных с привязкой к состояниям динамической модели. В качестве динамической модели используется иерархическая ситуационная модель (ИСМ), в которой совокупность «плоских» моделей (графов переходов) организована в иерархию за счёт того, что состояния модели некоторого уровня, в свою очередь, могут содержать вложенные субмодели со своими подсостояниями.

Предложена концепция взаимодействия ИСМ с внешней средой, обеспечивающая обмен данными в контексте текущих состояний ИСМ. Для этого в состав ИСМ введены элементы, обеспечивающие такое взаимодействие: интерфейсы состояний и субмоделей; блоки данных, ассоциированные с состояниями; блоки данных, отправляемые и получаемые через интерфейсы, внешние события, управляющие переходами текущих состояний ИСМ. Рассмотрена функциональность введенных элементов.

Введено понятие триггера перехода, управляющего срабатыванием перехода и сменой текущего состояния. Обсуждаются функциональные компоненты триггера перехода, связанные с обработкой данных, полученных через интерфейсы: прологовые действия, предикат перехода, эпилоговые действия.

С учётом введенных элементов ИСМ, обеспечивающих взаимодействие с внешней средой, строятся информационные модели структуры ИСМ в виде двух составных частей: порождающей модели, представляющей неизменную часть ИСМ, независимую от текущего состояния, и порождённой модели, отражающей текущее состояние ИСМ.

Разделение информационных моделей на порождающую и порождённую части допускает гибкое варьирование среды реализации ИСМ. Например, порождающую часть ИСМ можно реализовать в виде XML-документа для удобства задания иерархии, а порождённую – в реляционной среде для эффективного решения задач аналитики. В этом направлении предполагается проведение исследований по дальнейшему развитию данной тематики.

СПИСОК ЛИТЕРАТУРЫ | REFERENCES

- [Al-23] Al-Okaily A., Teoh A.P., Al-Okaily M. Evaluation of data analytics-oriented business intelligence technology effectiveness: an enterprise-level analysis // *Business Process Management J.* 2023; 29(3): 777–800. DOI [10.1108/BPMJ-10-2022-0546](https://doi.org/10.1108/BPMJ-10-2022-0546).
- [Ben94] Benjamin P. C. et al. *IDEF5 Method Report*. Prepared for Armstrong Laboratory AL/HRGA. Knowledge Based Systems Inc. Sept. 21, 1994. 187 pp. URL <https://online-pmo.com/wp-content/Education/idef5.pdf>.
- [Bör00] Börger E., Cavarra A., Riccobene E. Modeling the dynamics of UML state machines // *Int. Workshop on Abstract State Machines*. Berlin, Heidelberg: Springer, 2000. C. 223-241. EDN [BEKEKZ](#).
- [Cor20] Correia A., e Abreu F. B. Enhancing the correctness of BPMN models // *Sustainable Business: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2020. P. 373-394. DOI [10.4018/978-1-5225-9615-8.ch017](https://doi.org/10.4018/978-1-5225-9615-8.ch017).
- [Fri17] Frid A. I., Vulfin A. M., Berkholts V. V., et al. Architecture of the security access system for information on the state of automatic control systems of aircraft // *CSIT'2017: Proc. 19th Int. Workshop*, Baden-Baden, Germany, 08–10 Oct. 2017. 2017. Vol. 1. P. 16-19. EDN [YMGQPR](#).
- [Gir99] Girault A., Lee B., Lee E. A. Hierarchical finite state machines with multiple concurrency models // *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1999; 18(6):742-760. DOI [10.1109/43.766725](https://doi.org/10.1109/43.766725).
- [Mir14] Mironov V. V., Yusupova N. I., Gusarenko A. S. Hierarchical finite state model driven data management in situation-oriented databases // *ITIDS+RRS'2014: Proc. 2nd Int. Conf.*, Ufa, May 18–21, 2014. Vol. 3. 2014. P. 48-54. EDN [TAAPCO](#).
- [Mir17] Mironov V. V., Gusarenko A. S., Yusupova N. I. JSON documents processing using situation-oriented databases // *CSIT'2017: Proc. 19th Int. Workshop*, Baden-Baden, Germany, Oct. 8–10, 2017. Vol. 1. P. 97-103. EDN [YMGQTZ](#).
- [Mir20] Mironov V., Gusarenko A., et al. JSON documents processing using situation-oriented databases // *Acta Polytechnica Hungarica*. 2020. Vol. 17, No. 8. P. 29-40. EDN [EHDMHX](#).
- [Mir21] Mironov V., Gusarenko A., Yusupova N. Building of virtual multidocuments mapping to real sources of data in situation-oriented databases // *Communications in Computer and Information Science*. 2021. Vol. 1204. P. 167-178. EDN [RNDDGK](#).
- [Mir22] Mironov V. V., Gusarenko A. S., Yusupova N. I. Situation-oriented databases: processing office documents. *MOIT*. 2022; 10(2(37)). EDN [WXTJMP](#).
- [Oli26] Oliver M., Crocker A. Enhancing the sustainment warfighter function in constructive training // *2026 IEEE Int. Systems Conference (SysCon)*, Halifax, NS, Canada, 2026. P. 1-8. DOI [10.1109/SysCon66367.2026.11503628](https://doi.org/10.1109/SysCon66367.2026.11503628).
- [Özs25] Özsoy H. Ö. Using flowcharts and state diagrams as tools for industrial design of smart products // *Sanat ve Tasarım Dergisi*. 2025; (36): 205-233. EDN [AWSUJG](#).
- Al-Okaily A., Teoh A.P., Al-Okaily M. Evaluation of data analytics-oriented business intelligence technology effectiveness: an enterprise-level analysis. *Business Process Management J.* 2023; 29(3): 777–800. DOI [10.1108/BPMJ-10-2022-0546](https://doi.org/10.1108/BPMJ-10-2022-0546).
- Benjamin P. C. et al. *IDEF5 Method Report*. Prepared for Armstrong Laboratory AL/HRGA. Knowledge Based Systems Inc. Sept. 21, 1994. 187 pp. URL <https://online-pmo.com/wp-content/Education/idef5.pdf>.
- Börger E., Cavarra A., Riccobene E. Modeling the dynamics of UML state machines. *Int. Workshop on Abstract State Machines*. Berlin, Heidelberg: Springer, 2000. C. 223-241. EDN [BEKEKZ](#).
- Correia A., e Abreu F. B. Enhancing the correctness of BPMN models. *Sustainable Business: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2020: 373-394. DOI [10.4018/978-1-5225-9615-8.ch017](https://doi.org/10.4018/978-1-5225-9615-8.ch017).
- Frid A. I., Vulfin A. M., Berkholts V. V., et al. Architecture of the security access system for information on the state of automatic control systems of aircraft. *CSIT'2017: Proc. 19th Int. Workshop*, Baden-Baden, Germany, 08–10 Oct. 2017. 2017; 1: 16-19. EDN [YMGQPR](#).
- Girault A., Lee B., Lee E. A. Hierarchical finite state machines with multiple concurrency models. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1999; 18(6):742-760. DOI [10.1109/43.766725](https://doi.org/10.1109/43.766725).
- Mironov V. V., Yusupova N. I., Gusarenko A. S. Hierarchical finite state model driven data management in situation-oriented databases. *ITIDS+RRS'2014: Proc. 2nd Int. Conf.*, Ufa, May 18–21, 2014. 2014; 3:48-54. EDN [TAAPCO](#).
- Mironov V. V., Gusarenko A. S., Yusupova N. I. JSON documents processing using situation-oriented databases. *CSIT'2017: Proc. 19th Int. Workshop*, Baden-Baden, Germany, Oct. 8–10, 2017; 1:97-103. EDN [YMGQTZ](#).
- Mironov V., Gusarenko A., et al. JSON documents processing using situation-oriented databases. *Acta Polytechnica Hungarica*. 2020; 17(8):29-40. DOI [10.12700/APH.17.8.2020.8.3](https://doi.org/10.12700/APH.17.8.2020.8.3).
- Mironov V., Gusarenko A., Yusupova N. Building of virtual multidocuments mapping to real sources of data in situation-oriented databases. *Communications in Computer and Information Science*. 2021; 1204: 167-178. DOI [10.1007/978-3-030-78273-3_17](https://doi.org/10.1007/978-3-030-78273-3_17). (In Russ.)
- Mironov V. V., Gusarenko A. S., Yusupova N. I. Situation-oriented databases: processing office documents. *MOIT*. 2022; 10(2(37)). DOI [10.26102/2310-6018/2022.37.2.021](https://doi.org/10.26102/2310-6018/2022.37.2.021).
- Oliver M., Crocker A. Enhancing the sustainment warfighter function in constructive training. *2026 IEEE Int. Systems Conference (SysCon)*, Halifax, NS, Canada, 2026; P. 1-8. DOI [10.1109/SysCon66367.2026.11503628](https://doi.org/10.1109/SysCon66367.2026.11503628).
- Özsoy H. Ö. Using flowcharts and state diagrams as tools for industrial design of smart products. *ST Dergisi*. 2025; (36):205-233. DOI [10.18603/sanatvetasarim.1699442](https://doi.org/10.18603/sanatvetasarim.1699442).

- [Pet23] Petrosov D. A. Application of the theory of Petri nets in the development of simulation models of business processes based on the IDEF3 methodology // *Computational Nanotechnology*. 2023; 10(3): 51-63. EDN [TGIJWN](#).
- [RoI25] Rolo T., Preguiça N., Araujo F. Improving Availability in Event Sourcing Systems // 20th European Dependable Computing Conf. Companion Proc. (EDCC-C). IEEE, 2025; 18-21. DOI [10.1109/EDCC-C66476.2025.00020](#).
- [Vuk00] Vuksic V., Giaglis G., Hlupic V. IDEF diagrams and Petri nets for business process modeling: suitability, efficacy, and complementary use // *Proc. 2nd Int. Conf. Enterprise Information Systems*, Stafford, UK, July 4-7, 2000. 242-247. DOI [10.1007/978-94-017-1427-3_21](#).
- [Вер22] Веретельникова Е. Л. Теоретическая информатика. Теория сетей Петри и моделирование систем. Litres, 2022. EDN [XUFLXF](#).
- [Гус24] Гусаренко А. С., Миронов В. В. Совместная программная обработка разнородных конструкторских документов в учебном ИТ-проектировании // *СИИТ*. 2024. Т. 6, № 3(18). С. 102-118. EDN [QATAMS](#).
- [Кал25] Калянов Г. Н. Визуальные языки и методы моделирования бизнес-процессов: перспективы развития и проблематика обучения // *Открытое образование*. 2025. Т. 29, № 6. С. 58-70. EDN [YVWVQM](#).
- [Кан13] Канашин В. В., Миронов В. В. Иерархические виджеты: ввод и контроль данных пользователя в веб-приложениях на основе ситуационно-ориентированных баз данных // *Вестник УГАТУ*. 2013. Т. 17, № 5(58). С. 166-176. EDN [RPWYPV](#).
- [Кан14] Канашин В. В., Миронов В. В. Иерархические виджеты: алгоритмы контроля данных пользователя в веб-приложениях на основе ситуационно-ориентированных баз данных // *Вестник УГАТУ*. 2014. Т. 18, № 1(62). С. 204-213. EDN [SDHUMT](#).
- [Кан14b] Канашин В. В., Миронов В. В. Иерархические виджеты: опыт применения в веб-приложении на основе ситуационно-ориентированной базы данных // *Вестник УГАТУ*. 2014. Т. 18, № 2(63). С. 185-196. EDN [SEAXWH](#).
- [Кон24] Конев К. А., Антонов В. В. Разработка метода поддержки принятия решений в сфере обеспечения качества на промышленных предприятиях // *СИИТ*. 2024. Т. 6, № 4(19). С. 98-110. EDN [XIYRTJ](#).
- [Кон24b] Конев К. А., Антонов В. В. Поддержка принятия решений при обеспечении качества с использованием ситуационно-онтологического подхода. М.: Русайнс, 2024. 152 с. EDN [SHBZGQ](#).
- [Мир08] Миронов В. В., Шакирова Г. Р., Яфаев В. Э. Иерархическая модель персонализированных документов и ее XML-реализация // *Вестник УГАТУ*. 2008. Т. 11, № 1. С. 164-174. EDN [JXECDD](#).
- [Мир09] Миронов В. В., Маликова К. Э. Интернет-приложения на основе встроенных динамических моделей: идея, концепция, безопасность // *Вестник УГАТУ*. 2009. Т. 13, № 2. С. 167-179. EDN [MTEUFT](#).
- [Мир10] Миронов В. В., Маликова К. Э. Интернет-приложения на основе встроенных динамических моделей: элементы управления пользовательского интерфейса // *Вестник УГАТУ*. 2010. Т. 14, № 5(40). С. 170-175. EDN [PWTLMZ](#).
- Petrosov D. A. Application of the theory of Petri nets in the development of simulation models of business processes based on the IDEF3 methodology. *Comp. Nanotech*. 2023; 10(3): 51-63. DOI [10.33693/2313-223X-2023-10-3-51-63](#).
- Rolo T., Preguiça N., Araujo F. Improving Availability in Event Sourcing Systems. *20th European Dependable Computing Conf. Companion Proc.* (EDCC-C). IEEE, 2025; 18-21. DOI [10.1109/EDCC-C66476.2025.00020](#).
- Vuksic V., Giaglis G., Hlupic V. IDEF diagrams and Petri nets for business process modeling: suitability, efficacy, and complementary use. *Proc. 2nd Int. Conf. Enterprise Information Systems*, Stafford, UK, July 4-7, 2000. 242-247. DOI [10.1007/978-94-017-1427-3_21](#).
- Veretelnikova E. L. *Theoretical Computer Science. Petri Net Theory and System Modeling*. Litres, 2022. EDN [XUFLXF](#).
- Gusarenko A. S., Mironov V. V. Joint software processing of heterogeneous design documents in educational IT design. *SIIT*. 2024; 6(3(18)): 102-118. DOI [10.54708/2658-5014-SIIT-2024-no3-p102](#). (In Russ.)
- Kalyanov G. N. Visual languages and methods of business process modeling: development prospects and training issues. *Open Education*. 2025; 29 (6): 58-70. DOI [10.21686/1818-4243-2025-6-58-70](#). (In Russ.)
- Kanashin V. V., Mironov V. V. Hierarchical widgets: input and control of user data in web applications based on situationally-oriented databases. *Vestnik UGATU*. 2013; 17(5(58)): 166-176. EDN [RPWYPV](#). (In Russ.)
- Kanashin V. V., Mironov V. V. Hierarchical widgets: algorithms for monitoring user data in web applications based on situation-oriented databases. *Vestnik UGATU*. 2014; 18(1(62)): 204-213. (In Russ.)
- Kanashin V. V., Mironov V. V. Hierarchical widgets: experience of application in a web application based on a situation-oriented database. *Vestnik UGATU*. 2014; 18(2(63)): 185-196. EDN [SEAXWH](#). (In Russ.)
- Konev K. A., Antonov V. V. Development of a method for supporting decision-making in the field of quality assurance at industrial enterprises. *SIIT*. 2024; 6(4(19)): 98-110. DOI [10.54708/2658-5014-SIIT-2024-no4-p98](#). (In Russ.)
- Konev K. A., Antonov V. V. *Decision Support in Quality Assurance Using a Situational-Ontological Approach*. Moscow: Ruscains, 2024. 152 pp. EDN [SHBZGQ](#). (In Russ.)
- Mironov V. V., Shakirova G. R., Yafaev V. E. Hierarchical model of personalized documents and its XML implementation. *Vestnik UGATU*. 2008; 11(1):164-174. EDN [JXECDD](#). (In Russ.)
- Mironov V. V., Malikova K. E. Internet applications based on embedded dynamic models: idea, concept, security. *Vestnik UGATU*. 2009; 13(2):167-179. EDN [MTEUFT](#). (In Russ.)
- Mironov V. V., Malikova K. E. Internet applications based on embedded dynamic models: user interface controls. *Vestnik UGATU*. 2010; 14(5(40)): 170-175. EDN [PWTLMZ](#). (In Russ.)

- [Мир10b] Миронов В. В., Маликова К. Э. Интернет-приложения на основе встроенных динамических моделей: архитектура, структура данных, интерпретация // Вестник УГАТУ. 2010. Т. 14, № 1(36). С. 154-163. EDN [PWZSWR](#).
- [Мир16] Миронов В. В., Гусаренко А. С., Юсупова Н. И. Ситуационно-ориентированные базы данных как виртуальный интеграционный слой в веб-приложениях // Information Technologies for Intelligent Decision Making Support (ITIDS'2016): Proc. 4th Int. Conf., Ufa, May 17–19 2016. 2016. Vol. 1. С. 123-128. EDN [VYFCHF](#).
- [Мир16b] Миронов В. В., Гусаренко А. С., Юсупова Н. И. Ситуационно-ориентированные базы данных: интеграция XML-данных с реляционной средой MySQL // Системы управления и информационные технологии. 2016. № 3(65). С. 48-55. EDN [WITPMN](#).
- [Мир17] Миронов В. В., Гусаренко А. С., Юсупова Н. И. Инвариантность виртуальных данных в ситуационно-ориентированной базе данных при отображении на разнородные хранилища // Вестник комп. и информ. технологий. 2017. № 1(151). С. 29-36. EDN [XQQRHR](#).
- [Мир17b] Миронов В. В., Гусаренко А. С., Юсупова Н. И. Отображение виртуальных XML-документов на таблицы MySQL в ситуационно-ориентированных базах данных: "распределенный" подход // Информ. технологии и выч. системы. 2017. № 1. С. 77-89. EDN [YKJYOD](#).
- [Мир86] Миронов В. В., Головкин Ю. Б., Юсупова Н. И. Об автоматной модели динамической ситуации // Управление сложными техническими системами: Межвузовский научный сборник. Т. 9. Уфа: Уфимский авиационный институт, 1986. С. 3-10. EDN [YQLZAX](#).
- Mironov V. V., Malikova K. E. Internet applications based on embedded dynamic models: architecture, data structure, interpretation. *Vestnik UGATU*. 2010; 14(1(36)): 154-163. EDN [PWZSWR](#). (In Russ.)
- Mironov V. V., Gusarenko A. S., Yusupova N. I. Situation-oriented databases as a virtual integration layer in web applications. *Information Technologies for Intelligent Decision Making Support (ITIDS'2016): Proc. 4th Int. Conf., Ufa, May 17–19 2016*. 2016; 1:123-128. EDN [VYFCHF](#). (In Russ.)
- Mironov V. V., Gusarenko A. S., Yusupova N. I. Situation-oriented databases: integration of XML data with the MySQL relational environment. *Control Systems and Information Technologies*. 2016; (3(65)):48-55. EDN [WITPMN](#). (In Russ.)
- Mironov V. V., Gusarenko A. S., Yusupova N. I. Invariance of virtual data in a situation-oriented database when mapping to heterogeneous storages. *Bulletin of Computer and Information Technologies*. 2017; (1): 29-36. DOI [10.14489/vkit.2017.01.pp.029-036](#). (In Russ.)
- Mironov V. V., Gusarenko A. S., Yusupova N. I. Mapping virtual XML documents to MySQL tables in situationally-oriented databases: a "distributed" approach. *Information Technologies and Computing Systems*. 2017; (1):77-89. EDN [YKJYOD](#). (In Russ.)
- Mironov V. V., Golovkin Yu. B., Yusupova N. I. On the automatic model of a dynamic situation. *Control of Complex Technical Systems: Interuniversity scientific collection*. Vol. 9. Ufa: Ufa Aviation Institute, 1986. Pp. 3-10. EDN [YQLZAX](#). (In Russ.)

ОБ АВТОРАХ | ABOUT THE AUTHORS

ГУСАРЕНКО Артем Сергеевич

Уфимский университет науки и технологий, Россия.

gusarenko@ugatu.su. ORCID: [0000-0003-4132-6106](#).

Доц. каф. автоматизированных систем управления. Дипл. информатик-экономист (Уфимск. гос. авиац. техн. ун-т). Канд. техн. наук по мат. и прогр. обеспечению выч. машин, комплексов и комп. сетей (там же). Иссл. в обл. иерархич. моделей и ситуац. управления.

МИРОНОВ Константин Валерьевич

Уфимский университет науки и технологий, Россия.

mironovconst@gmail.com. ORCID: [0000-0002-4828-1345](#).

Доц. каф. вычислительной техники и защиты информации. Дипл. спец. по защите инф-и (Уфимск. гос. авиац. техн. ун-т). PhD (Техн. ун-т Вены). Иссл. в обл. робототехники, применения ИИ в техн. системах.

МИРОНОВ Валерий Викторович

Уфимский университет науки и технологий, Россия.

mironov@list.ru. ORCID: [0000-0002-0550-4676](#).

Проф. каф. автоматизированных систем управления. Дипл. радиофизик (Воронежск. гос. ун-т). Д-р техн. наук по упр. в техн. системах (Уфимск. гос. авиац. техн. ун-т). Иссл. в обл. иерархич. моделей и ситуац. управления.

GUSARENKO Artem Sergeyevich

Ufa University of Science and Technologies, Russia.

gusarenko@ugatu.su. ORCID: [0000-0003-4132-6106](#).

Docent, Dept. of Automated Control Systems. Dipl. Informatic economist (Ufa State Aviat. Techn. Univ.). Cand. of Tech. Sci. on mathematical and software support for computers, complexes, and computer networks (ibid). Research in the field of hierarchical models and situational management.

MIRONOV Konstantin Valeryevich

Ufa University of Science and Technology, Russia.

mironovconst@gmail.com. ORCID: [0000-0002-4828-1345](#).

Assoc. Prof., Department of Computer Science and Information Security. Information Security Specialist (Ufa State Aviation Tech. Univ.). PhD (Vienna University of Technology). Research in the field of robotics and intelligent control.

MIRONOV Valeriy Viktorovich

Ufa University of Science and Technologies, Russia.

mironov@list.ru. ORCID: [0000-0002-0550-4676](#).

Prof., Dept. of Automated Control Systems. Dipl. Radio physicist (Voronezh State Univ.). Dr. of Tech. Sci. (Ufa State Aviation Technical University). Research in hierarchical models and situational management.

МЕТАДАННЫЕ | METADATA

Заглавие: Ситуационно-ориентированные базы данных: взаимодействие иерархических ситуационных моделей с внешней средой

Авторы: Гусаренко А. С., Миронов К. В., Миронов В. В.

Аннотация: Проект «ситуационно-ориентированные базы данных» предусматривает хранение и обработку данных с привязкой к состояниям динамической модели в виде

Title: Situation-oriented databases: interaction of hierarchical situational models with the external environment.

Authors: Gusarenko A. S., Mironov K. V., Mironov V. V.

Abstract: The "situation-oriented databases" project provides for the storage and processing of data linked to the states of a dy-

графа переходов. В качестве динамической модели используется иерархическая ситуационная модель (ИСМ), в которой совокупность «плоских» моделей (графов переходов) организована в иерархию за счёт того, что состояния модели некоторого уровня, в свою очередь, могут содержать вложенные субмодели со своими подсостояниями. В статье предлагается концепция взаимодействия ИСМ с внешней средой, обеспечивающая обмен данными в контексте текущих состояний ИСМ. Рассматриваются элементы ИСМ, и их функциональность, обеспечивающие такое взаимодействие: интерфейс состояния; данные, ассоциированные с состояниями; данные, отправляемые и получаемые через интерфейс, внешние события, управляющие переходами текущих состояний ИСМ. Вводится понятие триггера перехода, управляющего срабатыванием перехода и сменой текущего состояния. Обсуждаются функциональные компоненты триггера перехода: прологовые действия, предикат перехода, эпилоговые действия. С учётом введенных элементов ИСМ, обеспечивающих взаимодействие с внешней средой, строятся информационные модели структуры ИСМ в виде двух составных частей: порождающей модели, представляющей неизменяемую часть ИСМ независимо от текущего состояния, и порождённой модели, отражающей текущее состояние ИСМ.

Ключевые слова: Информационная модель; динамическая модель; конечный автомат; граф переходов; иерархическая ситуационная модель; интерфейс.

Язык: Русский.

Статья поступила в редакцию 16 июня 2026 г.

dynamic model in the form of a transition graph. A hierarchical situation model (HSM) is used as a dynamic model, in which a set of "flat" models (transition graphs) is organized into a hierarchy due to the fact that the states of a model at a certain level, in turn, can contain nested submodels with their own substates. This article proposes a concept for the interaction of the HSM with the external environment, ensuring data exchange in the context of the current states of the HSM. The following HSM elements and their functionality that ensure such interaction are considered: the state interface; data associated with the states; data sent and received through the interface; external events that control transitions of the current states of the HSM. The concept of a transition trigger is introduced, which controls the firing of a transition and a change in the current state. The functional components of a transition trigger are discussed: prologue actions, transition predicate, and epilogue actions. Taking into account the introduced IMS elements that ensure interaction with the external environment, information models of the HSM structure are constructed in the form of two components: a generative model representing the constant part of the HSM regardless of the current state, and a generated model representing the current state of the HSM.

Key words: Information model; dynamic model; finite state machine; transition graph; hierarchical situational model; interface.

Language: Russian.

The article was received by the editors on 16 June 2026.