

УДК 004.7

РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ СИСТЕМЫ УПРАВЛЕНИЯ РЕСУРСАМИ ПРЕДПРИЯТИЯ

Высочкин А.В., Портнов Е.М., Слюсарь В.В.

Национальный исследовательский университет «МИЭТ», Москва, e-mail: evgen_uis@mail.ru

В настоящее время в системах управления ресурсами предприятия постоянно возникает потребность в создании и обработке триггеров, которые связаны с возникновением определенных событий в системе. Проблема, поставленная в данной работе, заключается в том, что существующие на данный момент ERP-системы не обладают достаточными возможностями для создания триггеров внутри системы для определенных задач предприятия, так как либо эти возможности ограничены определенным диапазоном объектов и действий, для которых можно создать триггер, либо такой функционал отсутствует и все триггеры должны устанавливаться при интеграции системы на предприятии и не могут быть впоследствии изменены. В связи с этим была разработана математическая модель алгоритма управления бизнес-процессами для ERP-систем, позволяющего настраивать триггеры внутри системы, а не при её интеграции. Исходя из разработанной математической модели, для создания триггеров в ERP-системах можно применять два способа: создание на уровне программного кода и создание на уровне базы данных. Использование данной модели обеспечивает уменьшение числа запросов на 52%, а временных затрат на построение и обновление ERP-триггеров на 54%.

Ключевые слова: ERP-системы, триггеры, математическая модель, бизнес-процессы, граф, быстродействие

ENTERPRISE RESOURCE CONTROLLING SYSTEM MATHEMATICAL MODEL DEVELOPMENT

Vysochkin A.V., Portnov E.M., Slyusar V.V.

National Research University of Electronic Technology, Moscow, e-mail: evgen_uis@mail.ru

Currently, enterprise resource management systems constantly need to create and process triggers that are associated with the occurrence of certain events in the system. The problem posed in this paper is that the currently existing ERP-systems do not have sufficient capacity to create triggers within the system for certain tasks of the enterprise, because either these capabilities are limited to certain the range of objects and actions for which you can create a trigger, or there is no such functionality and all triggers must be installed during the integration of the system in the enterprise and can not be subsequently changed. In this regard, a mathematical model of the business process control algorithm for ERP-systems was developed, which allows to configure triggers within the system, not during its integration. Based on the developed mathematical model, two methods can be used to create triggers in ERP systems: creation at the program code level and creation at the database level. Using this model reduces the number of requests by 52%, and the time spent on building and updating ERP triggers by 54%.

Keywords: ERP-systems, triggers, mathematical model, business processes, graph, speed

В настоящее время управление предприятием непрерывно связано с использованием систем планирования ресурсов предприятия (ERP), которые позволяют автоматизировать процессы управления бухгалтерским учетом, инвентаризацией, менеджментом и другими отделами предприятия, интегрируя все это в одну систему [1, 2].

При работе в таких системах постоянно возникает потребность в выполнении определенных действий при наступлении некоторых событий, т.е. триггеров (напри-

мер, добавление товаров в систему учета при поступлении их на склад). В большинстве ERP-систем настройка триггеров происходит отдельно для каждого предприятия при интеграции системы, связана с большими временными и финансовыми затратами и не может быть в дальнейшем использована при интеграции системы на других предприятиях.

Ниже приведен пример запроса для создания триггера на языке SQL, который добавляет в таблицу values запись об изменении таблицы example:

```
CREATE OR REPLACE TRIGGER ExampleUpdatedTrigger
AFTER UPDATE ON example
BEGIN
insert into info values ('table "example" has changed');
END;
```

При появлении новых бизнес-процессов на предприятии добавление в ERP-систему новых триггеров также требует отдельной настройки системы и связано с дополнительными затратами, а существующие

в системе триггеры, объекты триггеров или действия, исполняемые триггерами при вызове, не могут быть изменены или расширены, либо их модификация ограничена возможностями системы [3, 4].

Цель исследования: повышение быстродействия процессов построения и вызова триггеров управления бизнес-процессами предприятия в ERP-системах.

Материалы и методы исследования

Разработаем математическую модель триггеров ERP-системы и представим триггерные связи, возникающие при создании, изменении и удалении объектов ERP-системы, в виде графов – графа потока управления CFG и графа зависимостей DFG.

Триггер в рамках ERP-системы – это совокупность T множества событий системы $E = \{e_i | i = 1, \dots, n\}$ и набора действий $W = \{w_j | j = 1, \dots, k\}$. При возникновении любого события e_i из множества E триггер T запускает по очереди действия w_j из набора W . Каждое событие $e \in E$ в ERP-системе связано с объектом x системы и представляет собой изменение, создание, удаление объекта x или его атрибутов $a_x \in A_x$.

Возникновение события задается функцией $h: E \times X \rightarrow \{0, 1\}$, для которой $h(e, x) = 1$, если объект x был изменен, и 0 в остальных случаях. Действия $w \in W$ также представляют собой изменение, создание или удаление объектов ERP-системы и их атрибутов, причем множество изменяемых объектов X_c может пересекаться с множеством объектов событий X_e . Действие w может принимать одно из значений:

$w(x) = x, A_x \cap A_x \neq \emptyset$ (изменение объекта x , в результате которого часть атрибутов объекта изменяется); $w(x) = y, X' = X \cup \{y\}$ (создание нового объекта y); $w(x) = \emptyset, X' = X \setminus \{x\}$ (удаление объекта x); $w(x) = x$ (действие, не изменяющее объект x).

Таким образом, триггер можно задать с помощью формулы

$$t(E, W) = (h(e_1) \vee h(e_2) \vee \dots \vee h(e_n)) \rightarrow f(w_1 w_2 \dots w_n),$$

где e_1, e_2, \dots, e_n – события триггера, w_1, w_2, \dots, w_n – действия триггера, $f: 2^W \rightarrow \{0, 1\}$ – функция, равная 1 в случае успешного выполнения последовательности действий $w = w_1 w_2 \dots w_n = 1$, и 0 в случае неудачи.

Для отображения триггерных связей между объектами ERP-системы удобно воспользоваться графовыми структурами, такими, как граф потока управления и граф зависимостей [5, 6].

Граф потока управления представляет собой совокупность $CFG(V, T)$, где $V = \{v_i | i = 1 \dots n\}$ – множество действий над объектами ERP-системы, а $T = \{t_{ij} = (v_i, v_j) | i = 1 \dots n, j = 1 \dots n\}$ – множество триггерных связей между объектами ERP-системы (рис. 1). В графе потока управления каждый узел (вершина) графа соответствует базовому блоку – прямолинейному участку кода, не содержащему в себе ни операций передачи управления, ни точек, на которые управление передается из других частей программы.

Имеется лишь два исключения:

- точка, на которую выполняется переход, является первой инструкцией в базовом блоке;
- базовый блок завершается инструкцией перехода.

Направленные дуги используются в графе для представления инструкций перехода. Также, в большинстве реализаций добавлено два специализированных блока:

- входной блок, через который управление входит в граф;
- выходной блок, который завершает все пути в данном графе. Блок, не связанный со входным блоком, считается недостижимым («мёртвый» код).

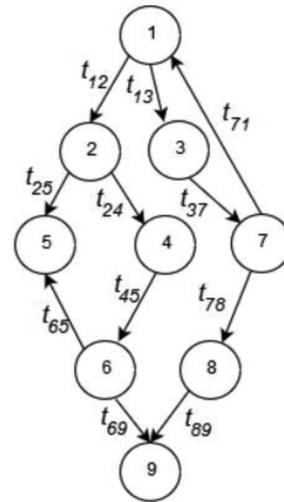


Рис. 1. Граф потока управления CFG в ERP-системе

Недостижимый блок может быть удален из программы. Блок, не связанный с выходным блоком, содержит бесконечный цикл. Полагаясь на это утверждение, удастся обнаружить не все бесконечные циклы в связи с проблемой останова.

Пусть дано множество $V = \{v_i | i = 1 \dots n\}$ объектов ERP-системы и отношение транзитивности над этим множеством $R = V \times V$, где $(a, b) \in R$ следует зависимость для вычисления a нужно сначала вычислить b .

Тогда граф зависимостей представляет собой совокупность множеств $DFG(V, T)$, где $T \subseteq R$ и R – транзитивное замыкание T (рис. 2).

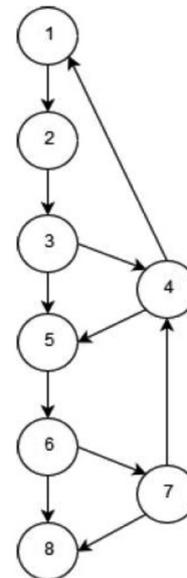


Рис. 2. Граф зависимостей DFG в ERP-системе

Формализация задачи построения триггеров в ERP-системе состоит из двух критериев [7]. Данные критерии представляют собой оценку быстродействия алгоритма построения триггеров в ERP-системе.

Пусть $T(n)$, определенная как функция над графом $CFG(V, T)$

$$T(n) = \sum_{i=1}^n \sum_{j=1}^n e(t_{ij})l(v_j), \quad (1)$$

где $e(t_{ij})$ – время на создание триггерной связи $t_{ij} = (v_i, v_j)$, $l(v_j)$ – время на выполнение действий триггера над объектом j , представляет собой временные затраты на создание триггерных связей между объектами ERP-системы при накладываемых ограничениях.

$$\sum_{i=1}^n e(t_{ij}) \leq T_{\max}. \quad (2)$$

Тогда первый критерий быстродействия триггерного алгоритма

$$\lim_{n \rightarrow \infty} \frac{T_1(n)}{T_0(n)} < 1, \quad (3)$$

где T_0 – временные затраты на создание триггерных связей до применения предложенной в данной работе методики; T_1 – временные затраты на создание триггерных связей после применения данной методики.

2. Пусть $D(n)$ – определенная как функция над графом $DFG(V, T)$

$$D(n) = \sum_{i=1}^n \sum_{j=1}^n d(t_{ij})l(v_j), \quad (4)$$

где $d(t_{ij})$ – задержка возникновения триггерной связи, возникающая при конфликте по данным (RAW, WAR). Тогда второй критерий быстродействия триггерного алгоритма – критерий быстродействия обновления графа триггеров при наличии конфликтов по данным:

$$\lim_{n \rightarrow \infty} \frac{D_1(n)}{D_0(n)} < 1, \quad (5)$$

где D_0 – временные затраты на обновление графа вызова триггеров до применения предложенной в данной работе методики; D_1 – временные затраты на обновление графа вызова триггеров после применения данной методики.

Результаты исследования и их обсуждение

Исходя из разработанной математической модели, для создания триггеров в ERP-системах можно применять два способа: создание на уровне программного кода и создание на уровне базы данных. На основе данной модели была разработана методика построения триггеров управления бизнес-процессами в ERP-системах.

На первом этапе в отношении любой ERP-системы необходимо определить набор общих стандартных моделей (например, моделей организации), объектов (например, планы и списки поставщиков) и процессов (например, управление заказами).

Если ранее для каждого предприятия нужно было учитывать существующую на ней структуру и организацию данных, то

при использовании ERP-системы необходимым и достаточным является использование одних и тех же моделей для каждой организационной единицы [7, 8]. Качество выбранных моделей оказывает огромное влияние на общий успех интеграции ERP-системы на предприятии.

В рамках языка программирования событие – это сообщение, которое возникает в различных точках исполняемого кода при выполнении определённых условий. Для решения поставленной задачи создаются обработчики событий: как только программа попадает в заданное состояние S , т.е. как только произойдет изменение, создание или удаление соответствующего объекта ERP-системы, происходит событие, посылается сообщение, а обработчик перехватывает это сообщение и выполняет действия $w_1, w_2, \dots, w_n \in W$.

В общем случае в обработчик не передается ничего, либо передается ссылка на объект, инициировавший (породивший) обрабатываемое событие. В особых случаях в обработчик передаются значения некоторых переменных или ссылки на какие-то другие объекты, чтобы обработка данного события могла учесть контекст возникновения события. Таким образом, создание триггера на уровне программного кода позволяет запускать любые действия при вызове их обработчиков.

Аналогично, как и в случае обработчиков событий, как только база данных перейдет в состояние S , при котором произойдет изменение каких-либо объектов x ERP-системы, запускается хранимая процедура, соответствующая измененным объектам, и исполняются действия $w_1, w_2, \dots, w_n \in W$. Однако в связи с тем, что внутри хранимой процедуры в БД можно указывать только действия, связанные с БД, то такой способ построения триггеров не дает возможности исполнять действия триггера вне базы данных [6].

Объединив два перечисленных способа, в работе применяется смешанный подход, состоящий в хранении триггера в сериализованном виде в базе данных, что одновременно обеспечивает и портируемость созданных ERP-триггеров, и возможность задавать любые действия, выполняемые при вызове триггера. При создании триггера все запросы к БД и другие действия в ERP-системе сериализуются и помещаются в БД. При вызове триггера данные десериализуются, оттуда извлекаются и запускаются действия триггера [9, 10].

Из построенной модели триггеров следует, что при вызове триггера могут изменяться данные объектов, использующиеся при вызове других триггеров. Такая ситуация может произойти, если в результате

действий $w \in W$ множество изменяемых объектов X_c пересекается с множеством объектов событий X_E :

$$X_c \cap X_E \neq \emptyset. \quad (6)$$

При этом могут возникать конфликты по данным [10]:

1. Read after Write (RAW). Триггер T_A записывает значение в переменную, которую использует триггер T_B .

2. Write after Read (WAR). Триггер T_A считывает значение переменной, в которую записывает новое значение триггер T_B .

3. Write after Write (WAW). Оба триггера записывают значения в одну и ту же переменную.

Чтобы избежать появления конфликтов по данным, применяется метод топологической сортировки графа триггеров с помощью обхода графа в глубину, который обеспечивает обновление графа зависимостей триггерных связей (рис. 3).

Для оценки быстродействия разработанных методики и алгоритмов были проведены испытания по построению графа ERP-триггеров в системе планирования ресурсов предприятия Greensight ERP.

В качестве критериев быстродействия применялись следующие:

1. Временные затраты на создание триггерных связей между объектами

$$T(n) = \sum_{i=1}^n \sum_{j=1}^n e(t_{ij})l(v_j), \quad (7)$$

где $e(t_{ij})$ – время на создание триггерной связи $t_{ij} = (v_i, v_j)$, $l(v_j)$ – время на выполнение действий триггера над объектом v_j .

2. Временные затраты на обновление графа вызова ERP-триггеров при наличии конфликтов по данным

$$D(n) = \sum_{i=1}^n \sum_{j=1}^n d(t_{ij})l(v_j), \quad (8)$$

где $d(t_{ij})$ – задержка возникновения триггерной связи, возникающая при конфликте по данным (RAW, WAR), $l(v_j)$ – время на выполнение действий триггера над объектом v_j .

В табл. 1 и 2 приведены результаты испытаний.

Таблица 1

Быстродействие операций построения триггеров до применения разработанных алгоритмов

Количество объектов	Число запросов	Временная задержка, мс
10	15	13
100	31	85
1000	75	400
10000	130	7600
100000	738	15700

Таблица 2

Быстродействие операций построения триггеров после применения разработанных алгоритмов

Количество объектов	Число запросов	Временная задержка, мс
10	2	7
100	7	37
1000	19	250
10000	88	570
100000	170	3600

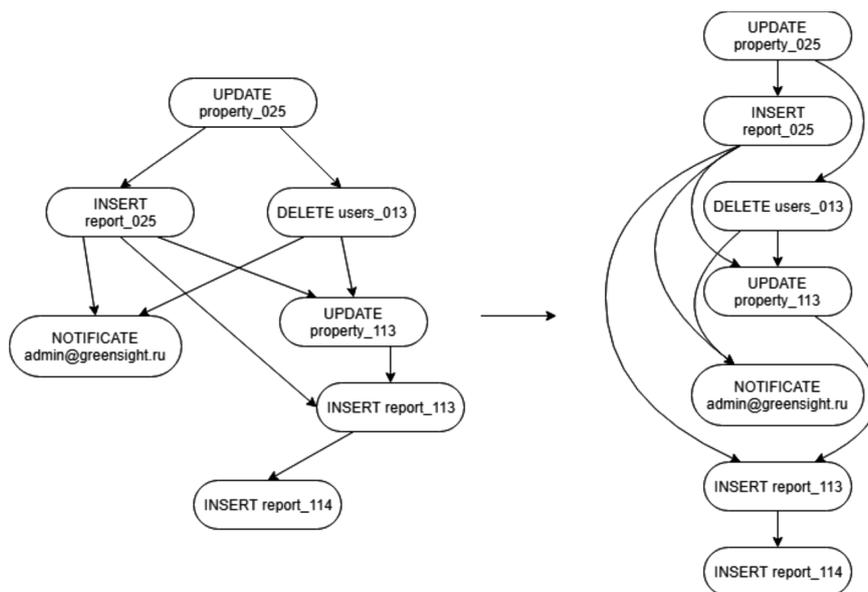


Рис. 3. Топологическая сортировка графа ERP-триггеров

Выводы

Проведено исследование современных подходов построения триггеров в ERP-системах. На основе данных исследования был выбран подход, объединяющий триггеры как хранимые процедуры особого типа в базах данных и триггеры как обработчики событий.

В работе была разработана математическая модель и методика построения триггеров управления бизнес-процессами в ERP-системах. В рамках данного подхода триггеры хранятся в сериализованном виде в базе данных, что одновременно обеспечивает и портируемость созданных ERP-триггеров, и возможность задавать любые действия, выполняемые при вызове триггера. Проведены испытания для оценки быстродействия построения и обновления графа ERP-триггеров с помощью разработанной модели, в результате которых получено уменьшение числа запросов на 52%, а временных затрат на построение и обновление ERP-триггеров – на 54%.

Работа выполнялась при финансовой поддержке РФФИ (договор № 18-07-00079\18).

Список литературы

1. Якунина А.В. Исследование теоретических подходов к управлению системами планирования ресурсов на

предприятии // Актуальные проблемы гуманитарных и естественных наук. 2016. № 1–3. С. 182–184.

2. Кузнецов Л.А. Управление ресурсами предприятия // Надежность. 2005. № 3 (26). С. 37–42.

3. Павлов Д.В. Триггеры в PL/SQL // Аллея науки. 2016. № 4. С. 759–761.

4. Sliusar V.V., Nikolaev O.V., Dorogov V.G., Gagarina L.G., Andrianov A.M. Usage of triggers for business process controlling in ERP systems // Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2018-January, P. 1567–1570. DOI: 10.1109/ElConRus.2018.8317398.

5. Михайлов А.А., Хмельнов А.Е. Метод визуализации графа потоков управления // Вестник Бурятского государственного университета. Математика, информатика. 2018. № 2. С. 50–62.

6. Шибанов С.В., Вишняков П.В., Лысенко Э.В., Смирнов Д.С., Орешкин К.А. Механизмы управления событиями в активных базах данных // Математическое и программное обеспечение систем в промышленной и социальной сферах. 2014. № 2 (5). С. 68–75.

7. Довбенко А.В. Идентификация изменений HTML-структур, приведенных к формату JSON // Наука, техника и образование. 2015. № 11 (17). С. 49–53.

8. Синдеев С.С. Исследование использования типа данных JSON в базе данных MySQL // В сборнике: Фундаментальные и прикладные исследования: от теории к практике: материалы II международной научно-практической конференции, приуроченной ко Дню Российской науки. 2018. С. 201–204.

9. Портнов Е.М., Баин А.М., Чумаченко П.Ю. Методика разработки систем автоматизации бизнес-процессов предприятий // Оборонный комплекс – научно-техническому прогрессу России. 2011. № 2. С. 69–73.

10. Портнов Е.М., Баин А.М., Чумаченко П.Ю. Методика натурального моделирования распределенных автоматизированных систем управления бизнес-процессами предприятий // Оборонный комплекс – научно-техническому прогрессу России. 2010. № 4. С. 27–30.