

Business Rules Modeling for Business Process Events: An Oracle Prototype

Rajeev Kaula

Computer Information Systems Department, Missouri State University, Springfield, MO 65897 (USA)

E-Mail: RajeevKaula@missouristate.edu

Abstract—Business rules guide business process activities and events, besides impacting associated business entity types. This paper outlines an approach to model business rules associated with business process events through traditional data modeling techniques like entity-relationship diagrams, and then transform the data model into application procedures that can effect business process events. A prototype to model a sample set of business rules pertaining to a business process event into a relational database is provided to demonstrate the application of the concept. The paper utilizes the Oracle database for illustrating the concepts. The prototype is explained through an application in Oracle's PL/SQL Server Pages.

Index Terms—Business Rules, Business Process, Data Modeling, Entity-Relationship Model, PL/SQL Server Pages, Relational model, Web Application.

I. INTRODUCTION

A business process is a collection of structured activities to complete some task. Business processes activities transform a set of inputs into outputs (product or service) for another process or event. Examining business processes helps an organization determine bottlenecks, and identify outdated, duplicate, and smooth running processes. To stay competitive organizations automate and optimize their business processes [2, 6, 14, 20].

Any automation of a business process is influenced by the business rules that guide its activities and events, besides impacting associated business entity types. Business rules are abstractions of the policies and practices of a business organization [4, 5, 7, 18]. Organizations that consolidate their business rules and automate their implementation, often derive strategic advantage [15].

In general business rules facilitate specification of some business guideline expressed declaratively in condition-action terminology as IF condition THEN action. A condition is some constraint, while the action clause reflects the action. Figure 1 shows an example of a business rule that describes a set of constraints applicable for handling a new hotel reservation event in hotel reservation business process.

```

IF    num_adults >= 2 AND
      num_adults <= 4 AND
      num_kids = 0
THEN rooms = 2
  
```

Figure 1. Sample Business Rule

Business rules influence on business processes is accomplished through their inclusion in business applications utilized by such processes to complete their tasks or activities [1, 7, 8]. Even though business rules may be outlined declaratively, from a development standpoint they are often expressed as an addendum to database development in the form of either a proprietary product or some DBMS encoded trigger and/or procedure [1, 4, 5, 7].

Often times business rules that are embedded within application logic are difficult to document and change [19]. Since any business process needs to be adaptive to remain competitive, it is necessary to develop business rules that can be expressed declaratively, yet modeled for change and implementation automatically. In this context, data modeling of business rules provides an approach that facilitates standardized representation of business rules.

Several attempts have been made to apply the standard techniques of traditional entity-relationship modeling to structure the business rules as a rule repository in a relational database management system [8, 9, 10, 11, 12, 16, 17]. However, these approaches are limited to either a singular representation of business rules semantic definition within a conceptual framework in database context, or utilization of expert system framework in database context. There is a lack of an approach that can (i) structure business rules with multiple constraints in database, and (ii) also associate their specification with database operations from a business process perspective.

This paper outlines such an approach by modeling business rules associated with business process events through traditional data modeling techniques like entity-relationship diagrams (ERD), and then transform the data model into application procedures that can effect business process events. Such a model represents a specialized logical schema for structuring business rules for storage in relational database. As shown in Figure 2, while business process events are governed through business rules, such events also impact business process entity types. Consequently, any modeling of business rules

should also explore the association between the business rules structure and the associated entity types.

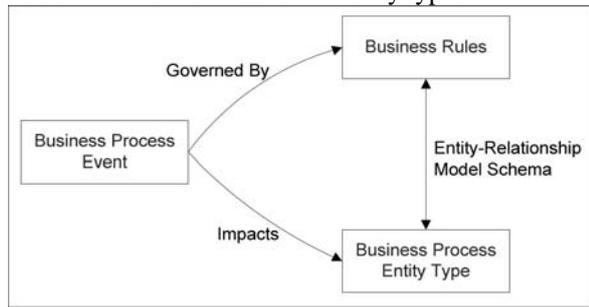


Figure 2. Business Process Event Modeling Perspective

Developing the logical schema for modeling of business rules for business process events now involves (i) ERD representation of business rules associated with business process events, (ii) transformation of the ERD into a relational model, (iii) representation of relationship between the business rules and the associated entity type, and (iv) outline the development of an application procedure that can impact business process event. The approach is illustrated through an Oracle 11g database using a prototype in PL/SQL Server Pages [3, 13]. PL/SQL server pages is a server-side scripting approach for developing database driven dynamic Web pages. The PL/SQL server page uses Oracle's primary database language PL/SQL as a scripting language along with HTML to generate database driven Web pages.

As each business rule is atomic and represents one complete thought, it cannot be decomposed further [7]. Consequently, the paper focuses on a business rule structure where the constraints are expressed using the AND operator, and there is only one action specification in the THEN clause. For this reason, a business rules statement expressed with the OR operator can be represented through separate atomic rules involving the AND operator.

The paper now briefly reviews existing research, and then outlines the modeling and transformation of business rules structure, followed by its implementation through a Web prototype.

II. REVIEW OF EXISTING RESEARCH

Utilization of business rules in the context of entity-relationship model and business process working can be categorized through four approaches.

The first approach [8, 9, 10] focuses on how entity-relationship diagrams can themselves be represented as business rules through the notion of term, fact, derivation, and constraint. However, the approach does not explore the representation of relationships that can exist among entity attributes in the formation of declarative IF/THEN business rules. Besides there is no specification on the impact of business rules on business events.

The second approach [11] expresses business rules as part of cardinality specification. In this approach business rules represent changes in the states of business entities (entities life histories) as a response to business events. However, as the approach illustration is limited to

business rules with only one constraint, it is not clear how multiple constraints are addressed.

The third approach [16] focuses on expressing business rules through expert system like knowledge base and then integrate it with database through entity-relationship diagrams. However, the paper is more of developing business rules system, and not on interaction with existing database entity types.

The fourth approach [17] outlines a meta-schema of an entity-relationship model in the form of an extended ER model that does include rules and events that can result in database triggers and procedures. However, the outline is largely conceptual and lacks the details on implementation.

III. BUSINESS RULES MODELING

The entity relationship modeling of business rules involves the following elements: (i) entity type structure of business rules, (ii) relational table representation, and (iii) business rules entity association with business process events entity type. Each element builds on one another.

A. Entity Type Structure

The business rule structure is represented through a schema of entity types. Such entity types essentially follow the abstract structure of a business rule statement as shown in Figure 3. In the figure each “constraint-i operator-i value-i” clause is some constraint, while the “entity action” clause is some action when the constraint conditions are true.

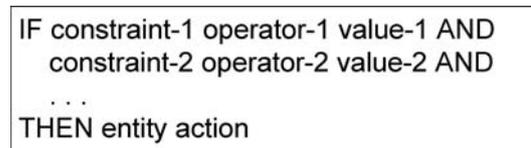


Figure 3. Abstract Structure of Business Rule

Entity action clause may specify descriptive statement like “create purchase order” or can be specific about what attribute value in the affected business process entity event needs to be changed as shown in Figure 1. In this paper, the focus is on business rules that impact individual business process entity types.

The business rule structure is represented through a collection of two entity types having an identifying relationship with each other as shown in Figure 4. The “Business Rule” entity type in the figure refers to the entity action clause of the business rule statement. The “Business Rule” entity type is the strong entity type. The attribute descriptions are as follows: the “RuleID” attribute is the primary key; the “Description” attribute provides a brief description of the business rule; the “Rule Type” attribute specifies what business process event operation will be the trigger for the business rule in the form of insert, update, or delete operations; the “Action Attribute” is the name of the business process transactional entity type attribute as expressed in the

entity action clause; the “Action Value” is the value that will be assigned to the Action Attribute in the entity action clause; the “Action Description” is a description of the nature of action that will be performed; and the “Action Unit” is the name of the program unit that will implement the business rule.

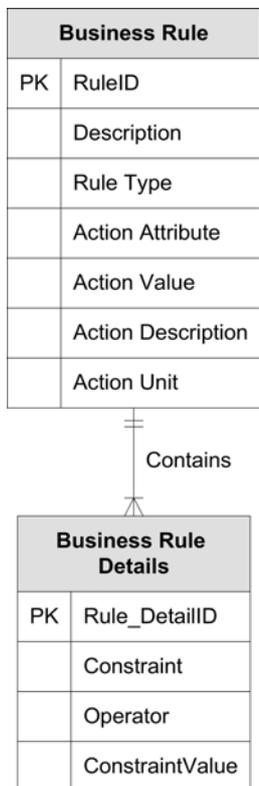


Figure 4. Business Rules Entity Schema

The “Business Rule Details” entity type in the figure is a weak entity type representing the various constraint clauses of the business rule. The “Constraint” attribute is the name of the constraint entry in the constraint clause, the “Operator” attribute is the condition operator in the constraint clause, while the “ConstraintValue” attribute is the value assigned to the constraint condition.

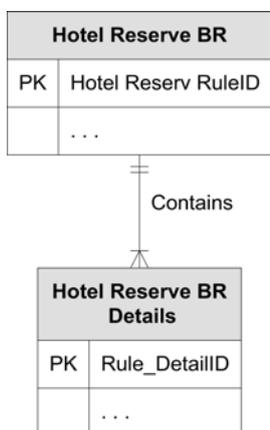


Figure 5. Business Rule Model for Hotel Reservation Event

For instance, business rules pertaining to the creation of a hotel reservation event in the hotel reservation

business process as shown in Figure 1 will be represented through the business rule entity structure as shown in Figure 5. The “Hotel Reserve BR” represents details about the various rules along with the entries in the THEN clause of the business rule. The “Hotel Reserve BR Details” contains the details of the associated IF clauses of the business rule.

Figure 6 shows the entity instances of the business rule shown in Figure 1. The business rule is represented through a single Hotel Reserve BR entity instance which contains the values associated with the THEN clause of the business rule. This entity instance is associated with three entity instances of Hotel Reserve BR Details entity type for the three clauses in the IF part of the business rule. The Hotel Reserve BR entity type is the strong entity type having an identifying relationship with the weak Hotel Reserve BR Details entity type.

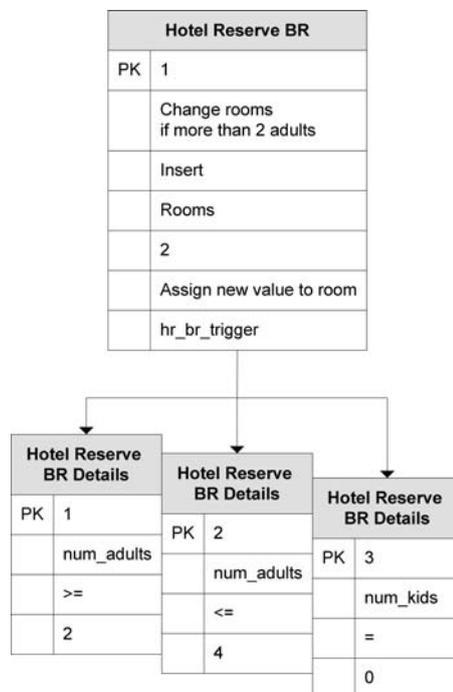


Figure 6. Entity Instances of Business Rule Specification for Hotel Reservation Event

B. Relational Table Representation

The Business Rule entity type and Business Rule Details entity type are represented as separate tables in a relational database. For example, Figure 7 shows the table structure of Figure 6 entity instances of Hotel Reserve BR and Hotel Reserve BR Details entity types. The Hotel Reserve BR Details entity type being the weak entity type has the composite primary key. The foreign key Hotel_Reserv_RuleID in Hotel Reserve BR Details table represents the 1:N relationship with the Hotel Reserve BR table. The 1:N relationship between the strong and weak entity types binds the THEN entity action clause instance with the multiple IF constraint clauses of a business rule statement.

Hotel_Reserve_BR						
Hotel_Reserv_RuleID	Description	Rule_Type	Action_Attr	Action_Value	Action_Desc	Action_Unit
1	Change rooms if more than 2 adults	Insert	rooms	2	Assign new value to room	hr_br_trigger

* Hotel_Reserv_RuleID is the primary key.

Hotel_Reserve_BR_Details				
Rule_DetailID	Hotel_Reserv_RuleID	Rule_Constraint	Rule_Operator	ConstraintValue
1	1	num_adults	>	2
2	1	num_adults	<	5
3	1	num_kids	=	0

* Rule_DetailID, Hotel_Reserv_RuleID is the primary key
 * Hotel_Reserv_RuleID is foreign key to Hotel_Reserve_BR table

Figure 7. Relational Database Table Representation for Hotel Reservation Event

C. Business Rule Entity Association with Business Process Event Entity Type

As business rules can be represented and stored in a relational database, it is possible now to establish a binary relationship between the business rule entity types and the business process event entity type. This relationship can then be utilized to maintain values on what rows in the entity type are changed due to individual business rules. The binary relationship will be many-to-many (M:N) as shown in Figure 8, as it is possible that one business rule may influence attribute values in more than one business process event entity instance, and vice-versa.

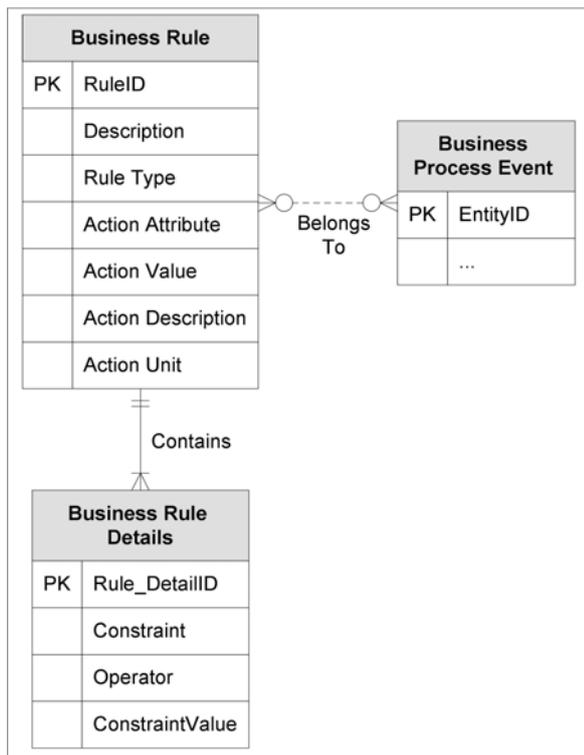


Figure 8. Database Relationship between Business Rules Entity Types and Business Process Event Entity Type

The M:N binary relationship is optional on both sides, implying that a business process event entity may not always be affected through the business rule. Also business rules may exist without having impacted any instance of the business process event entity type. The relational model representation will transform the M:N relationship between the business rule entity type and the

business process event entity type into two 1:N relationships with a new associate business process rule link entity type as shown in Figure 9.

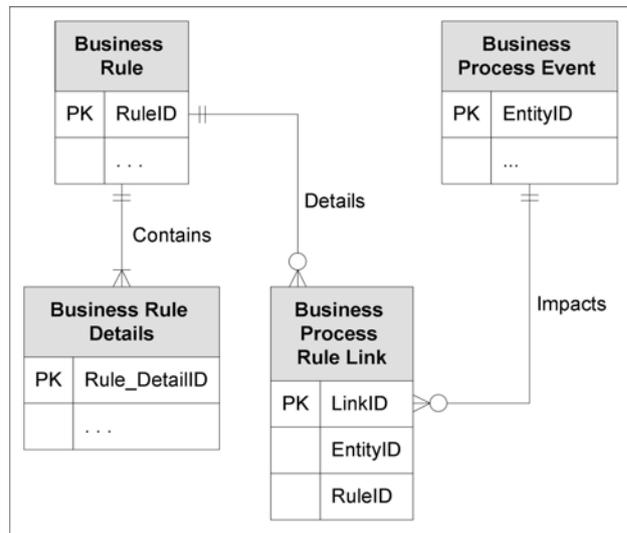


Figure 9. Transformation of Database Relationship between Business Rules Entity Types and Business Process Event Entity

For instance, the Hotel Reserve BR entity type of Figure 5 will be associated with business process entity type “Hotel Reservation” utilized during the hotel reservation event. This association as shown in Figure 10 is represented through “Hotel Reserve BR Link” associate entity type.

Once the business rules are stored in relational database tables, it is possible to query the table “Hotel Reserve BR Link” to determine the use of a business rule for individual “Hotel Reservation” table rows. Further, if the business rules change or are dropped, the link table “Hotel Reserve BR Link” can be utilized to maintain the changes.

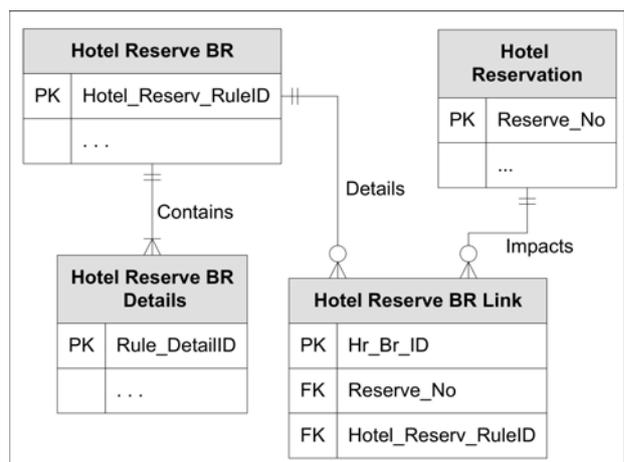


Figure 10. Database Relationship Representation of Hotel Reservation Business Rules and Hotel Reservation Event Entity

IV. BUSINESS RULES MODELING IMPLEMENTATION PROTOTYPE

A prototype that transforms business rules associated with business process event entity types into Oracle database program units is illustrated now. The prototype utilizes a small database schema associated with hotel reservation business process event. The data model of the hotel reservation schema is shown in Figure 11. Entity types are hotels, hotel_details, travel_customer, and hotel_reservation. The prototype is limited to business rules associated with only one business process event entity type – hotel_reservation. The prototype utilizes sample business rules associated with hotel reservation business process. A Web application to illustrate the business rule input in database, and its transformation into database triggers program units is shown thereafter through the Oracle's PL/SQL server pages technology. Once the business rules are implemented, their application is enforced during business process operations.

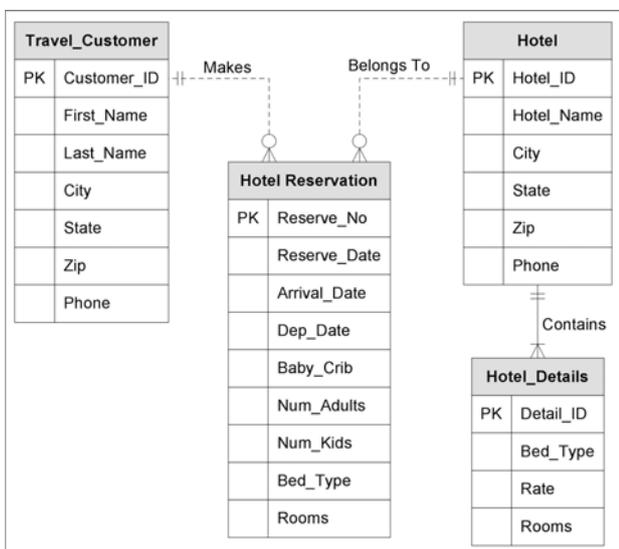


Figure 11. Hotel Reservation Database Schema

A. Business Rules Representation

As the prototype focuses on a single business process entity type hotel_reservation, the business rules are limited to new hotel reservations. Sample set of four business rules associated with hotel reservation entity type are listed below. These business rules are activated during the creation of new hotel reservation. Other combinations of attributes for business rules can be similarly developed based on business process operating guidelines. In the business rules specifications below, DQ implies “double queen.”

Business Rule 1:
 IF num_adults >= 2 AND
 num_adults <= 4 AND
 num_kids = 0
 THEN rooms = 2

Business Rule 2:
 IF num_kids > 0 AND
 num_kids <= 2 AND

num_adults <= 2
 THEN bed_type = DQ

Business Rule 3:
 IF baby_crib = Yes AND
 num_kids = 3 AND
 num_adults = 2
 THEN bed_type = DQ

Business Rule 4:
 IF num_adults <= 2 AND
 num_kids >= 2
 THEN rooms = 2

The ERD of these business rules is similar to Figure 5. Relational table representation of these business rules is listed in Figure 12 and Figure 13. Appendix A lists the SQL script for creating these tables including the Hotel_Reserve_BR_Link table.

Hotel_Reserve_BR						
Hotel_Reserv_RuleID	Description	Rule_Type	Action_Attr	Action_Value	Action_Desc	Action_Unit
1	Change rooms if more than 2 adults	Insert	rooms	2	Assign new value to room	hr_br_trigger
2	Change bed type if reserve with kids	Insert	bed_type	DQ	Assign new value to bed type	hr_dq_trigger
3	Change rooms if kids and crib	Insert	bed_type	DQ	Assign new value to bed type	hr_cr_trigger
4	Change rooms if more kids for two adults	Update	rooms	2	Change room value	hr_ur_trigger

* Hotel_Reserv_RuleID is the primary key.
 * DQ is Double Queen beds

Figure 12. Hotel Reservation Table

Hotel_Reserve_BR_Details				
Rule_DetailID	Hotel_Reserv_RuleID	Rule_Constraint	Rule_Operator	ConstraintValue
1	1	num_adults	>=	2
2	1	num_adults	<=	4
3	1	num_kids	=	0
4	2	num_kids	>	0
5	2	num_kids	<=	2
6	2	num_adults	<=	2
7	3	baby_crib	=	Y
8	3	num_kids	=	3
9	3	num_adults	=	2
10	4	num_adults	<=	2
11	4	num_kids	>=	2

* Rule_DetailID, Hotel_Reserv_RuleID is the primary key
 * Hotel_Reserv_RuleID is foreign key to Hotel_Reserve_BR table

Figure 13. Hotel Reservation Details Table

B. Web Prototype

The Web prototype illustrates the entering of the business rules in declarative format, and their eventual (i) representation in tables listed above, and (ii) generation of relevant database triggers program units. The prototype consists of two Web pages. The interaction of the two Web pages within the Web architecture is shown in Figure 14.

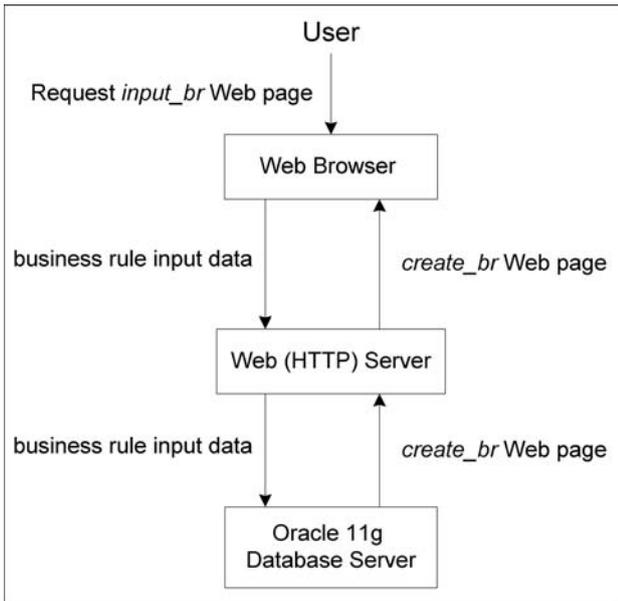


Figure 14. Prototype Web Architecture

The prototype is currently limited to business rules pertaining to insert operation (Rule Type value insert). It is possible to perform similar operations with respect to delete or update operations.

Figure 15. Input_br Web Form

PL/SQL server pages are stored as Web procedures within the Oracle database. The first page titled “input_br” displays a Web input form to enter business rules as shown in Figure 15. The entries in the Web page are similar to the entity attributes in Figure 4. The prototype currently is limited to maximum of only three constraints. A more flexible input Web form can always be developed to include more number of constraints.

The input_br Web page submits the input form data to the second Web page “create_br.” The second Web page

first stores the business rules structure in the two business rule tables – hotel_reserve_br and hotel_reserve_br_details. Thereafter, the Web page constructs two database triggers and returns a message display “Business Rule successfully created. Program Unit successfully created.”

Every business rule will have its own separate set of database triggers. The logic of the create_br Web procedure and its associated PL/SQL procedure is shown in Figure 16.

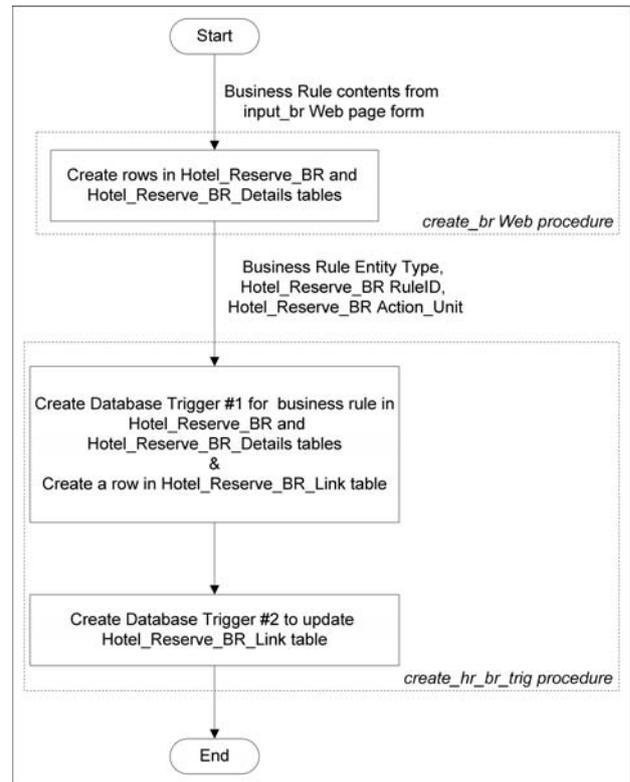


Figure 16. Create_br Web page logic

The Figure 16 logic description is as follows:

1. Insert the form input data into hotel_reserve_br and hotel_reserve_br_details table.
2. Call a procedure create_hr_br_trig with input parameters on entity table name value, hotel_reserve_ruleid value, and database trigger program unit name value. The procedure create_hr_br_trig utilizes the Oracle Dynamic SQL capability to create two database triggers as follows:
 - a. The first database trigger is named on the database trigger program unit name input parameter value, and it (i) defines the business rule as stored in hotel_reserve_br and hotel_reserve_br_details tables, and (ii) creates a row in hotel_reserve_br_link table where the hotel_reservation table primary key value is null.
 - b. The second database trigger is named similar to the same database trigger program unit name input parameter

value with a number appended, and it (i) checks for the row in hotel_reserve_br_link table where the hotel_reservation table key value is null and the hotel_reserve_ruleid is same as the applied rule, and (ii) updates the hotel_reserve_br_link table with the new hotel_reservation primary key value. The purpose of the second database trigger is to store the primary key of the business process event table (EntityID) for the associated business rule (RuleID) as shown in Figure 9.

The two database triggers generated by the prototype are listed in Appendix B. Once these business rule have been stored in the database along with their database triggers program units, whenever a new hotel reservation is created through a SQL insert statement, these rules will be executed through their associated triggers.

V. CONCLUSIONS

Entity relationship modeling of business rules for business process events in declarative format, and representing such business rules through automated database program units, provides a better structuring and management of business rules for business process. As each business rule is atomic, each such representation in entity relationship diagram is also a formal representation of a single derivation or constraint on business.

Storage of business rules as a rule repository in a relational DBMS also enables utilization of services similar to those provided for transactional database like conceptually centralized management, access optimization, recovery and concurrency controls, and so on. The rule repository schema can facilitate some additional features like:

- Any given user shall need to be aware only of those rules that are pertinent to that user (just as any given user needs to be aware only of that portion of the data in a given database that is pertinent to that user).
- Rules shall be queryable and updatable.
- Rule consistency shall be maintained.
- Rules shall be sharable and reusable across applications and users.

As the current research is limited in its generation of program units that handle attributes associated with one business process entity type or rule calculation/function, further research is ongoing to handle more complex business rules. Such complexity will be in the form of developing database program units for business rules that reference attributes in more than one business process entity type, besides developing business rules that impact more than one business process event. This could be in the form of multiple business process event entity types associated with one business rule.

APPENDIX A SQL SCRIPT FOR PROTOTYPE BUSINESS RULES TABLE

```
create table hotel_reserve_br
(hotel_reserv_ruleid integer constraint
hotel_reserve_rule1_pk primary key,
rule_desc varchar2(255),
rule_type varchar2(10),
action_attr varchar2(30),
action_value varchar2(100),
action_desc varchar2(255),
action_unit varchar2(512));
```

```
create table hotel_reserve_br_details
(hotel_reserv_ruleid integer constraint
hotel_reserv_rule1_details_fk references
hotel_reserve_br,
rule_detailid integer,
rule_constraint varchar2(255),
rule_operator varchar2(10),
constraintvalue varchar2(255),
constraint hotel_reserve_rule1_details_pk
primary key
(hotel_reserv_ruleid,rule_detailid));
```

```
create table hotel_reserve_br_link
(hr_br_id integer constraint
hotel_reserve_br_link primary key,
hotel_reserv_ruleid integer constraint
hotel_reserve_br_link_fk1 references
hotel_reserve_br,
reserve_no integer constraint
hotel_reserve_br_link_fk2 references
hotel_reservation);
```

```
create sequence hotel_reserv_br_seq
increment by 1
nocache;
```

```
create sequence hr_br_detailid_seq
increment by 1
nocache;
```

```
create sequence hr_br_seq
increment by 1
nocache;
```

APPENDIX B PROTOTYPE GENERATED DATABASE TRIGGERS

```
create or replace
trigger brule_web_trig
before insert
on HOTEL_RESERVATION
for each row
begin
if :new.NUM_ADULTS >= 2
and :new.NUM_ADULTS <= 4 and :new.NUM_KIDS
= 0 then
:new.ROOMS := 2;
insert into hotel_reserve_br_link values
(hr_br_seq.nextval, 3, null);
end if;
end;
```

```
create or replace
trigger brule_web_trig2
after insert
on HOTEL_RESERVATION
declare
```

```

new_reserve_no integer;
hr_br_ctr integer;hrbr_no integer;
hotel_ruleid_no integer;
begin
select max(reserve_no) into new_reserve_no
from hotel_reservation;
select count(*) into hr_br_ctr from
hotel_reserve_br_link
where reserve_no is null and
hotel_reserv_ruleid = 3;
if hr_br_ctr > 0 then
select hr_br_id,hotel_reserv_ruleid
into hrbr_no, hotel_ruleid_no
from hotel_reserve_br_link
where reserve_no is null
and hotel_reserv_ruleid = 3; if
hotel_ruleid_no = 3 then
update hotel_reserve_br_link set reserve_no
= new_reserve_no where hr_br_id = hrbr_no;
end if;
end if;
end;

```

REFERENCES

- [1] Y. Amghar, M. Meziane, and A. Flory, "Modeling of Business Rules For Active Database Application Specification," in *Advanced Topics in Database Research*, K. Siau, Ed., Hershey, PA: Idea Group Publishing, pp. 135-156, 2002.
- [2] M. Attaran, "Exploring the relationship between information technology and business process reengineering," *Information & Management*, Vol. 41, No. 5, pp. 585-596, 2004.
- [3] S. Boardman, M. Caffrey, S. Morse, and B. Rosenzweig, *Oracle Web Application Programming for PL/SQL Developers*, Upper Saddle River, NJ: Prentice-Hall, 2003.
- [4] C.J. Date, *What Not How: The Business Rules Approach to Application Development*, Reading, MA: Addison-Wesley, 2000.
- [5] C. J. Date and H. Darwen,, *Foundation for Future Database Systems: The Third Manifesto*, 2nd ed., Reading, MA: Addison-Wesley, 2000.
- [6] M. J. Earl, "The new and the old of business process redesign," *Journal of Strategic Information Systems*, Vol. 3, No. 1, pp. 5-22, 1994.
- [7] B. V. Halle, *Business Rules Applied*, New York, NY:John Wiley & Sons, 2002.
- [8] D.C. Hay, "A Repository Model - Business Rules - Part I (Structural assertions and derivations)," *The Data Administration Newsletter*, Issue 19, January 2002. Available: <http://www.tdan.com>
- [9] D.C. Hay, "A Repository Model - Business Rules - Part II (Action Assertions)," *The Data Administration Newsletter*, Issue 20, April 2002. Available: <http://www.tdan.com>
- [10] D.C. Hay, "Modeling Business Rules: What Data Models Do," *The Data Administration Newsletter*, Issue 27, January 2004. Available: <http://www.tdan.com>
- [11] H. Herbst, G. Knolmayer, T. Myrach, and M. Schlesinger, "The Specification of Business Rules: a Comparison of Selected Methodologies," in *Methods and Associated Tools for the Information System Life Cycle*, A.A. Verijn-Stuart and T.-W. Olle , Ed., Amsterdam: Elsevier, pp. 29-46, 1994.
- [12] P. Kardasis and P. Loucopoulos, "Expressing and organising business rules," *Information and Software Technology*, Vol. 46, No. 11, pp. 701-718, 2004.
- [13] R. Kaula, *Oracle 10g: Developing Web Applications with PL/SQL Server Pages*, New York, NY: Mc-Graw-Hill, 2006.
- [14] A. Lindsay, D. Downs, and K. Lunn, "Business processes—attempts to find a definition," *Information and Software Technology*, Vol. 45, No. 15, pp. 1015-1019, 2003.
- [15] D. Loshin, *Business Intelligence: The Savvy Manager's Guide*, San Francisco: Morgan Kaufman, 2003.
- [16] A. Maciol, "An application of rule-based tool in attributive logic for business rules modeling," *Expert Systems with Applications*, Vol. 34, No. 3, pp. 1825-1836, 2008.
- [17] S.B. Navathe, A.K. Tanaka, and S. Chakravarthy, "Active Database Modeling and Design Tools: Issues, Approach, and Architecture," *Data Engineering*, Vol. 15, No. 1-4, pp. 6-9, 1992.
- [18] S. Ram and V. Khatri, "A comprehensive framework for modeling set-based business rules during conceptual database design," *Information Systems*, Vol. 30, pp. 89-118, 2005.
- [19] F. Rosenberg and S. Dustdar, "Business rules integration in BPEL - a service-oriented approach," *Proceedings of the seventh IEEE International Conference on E-Commerce Technology*, pp. 476 - 479, 2005.
- [20] A. Scheer, *ARIS - business process modeling: Business Process Modeling*, Berlin: Springer-Verlag, 2000.