System Analysis by Navigator
for
Collaboration Task and Trading Task

Kiyoshi Itoh and Ryo Kawabata

Laboratory of Information and Systems Engineering
Sophia University
Kioi-cho 7-1, Chiyoda-ku, Tokyo 102, Japan

PHONE: +81-3-3238-3496
FAX: +81-3-3238-3311
e-mail: itohkiyo@sophia.ac.jp

Contact author: Kiyoshi Itoh
Presentation: lecture
System Analysis by Navigator for Collaboration Task and Trading Task

<Abstract>

This paper describes domain models for a family of systems in RDB-based implementations, and describes NAVI which can support the system analysis based on the domain model. This domain models are organized from the viewpoint of the combination of generic tasks such as the collaboration task and the trading task. The authors can perform domain analysis and modeling for various domains using such generic task models. The authors can apply NAVI for these domains. NAVI incorporates the domain process model, i.e., the model for development of a family of systems in the domain. NAVI also incorporates the reusable library of domain models. Therefore, users can perform the system analysis for the instance of the domain according to not a rigorous process model but a process model with short-cut path using reusable library. The domain model of the wholesale business of drinks is explained as an example in this paper.

1. Relationships between Domain Analysis and Generic Task such as Collaboration Task

Domain analysis and modeling ([Arango91][Prieto-Diaz87][Tracz92][Itoh94, 96]) is the process which obtains the domain specific model with the terminology, the problem solving / system analysis / system design strategies, the target system structures, etc. It is obtained by analyzing, identifying an organizing the properties of a family of target systems and the heuristic and reasonable knowledge for a family of past practices of system development. Its purpose is to improve the software productivity and reusability effectively.

Cooperative work domain exist in concurrent engineering (CE) fields (e.g., in [Carter92], [Bruce92], [Fowler95], [Mayer92], [Marca88], [Mayer92]), [Ottmann92], [West92], [West94]). According to a report by the Institute of Defense Analysis (IDA), CE is a comprehensive and systematic approach to the concurrent design of products and their related processes, including manufacture and support. This approach is intended to ensure developers consider the total lifecycle of a product from conception to withdrawal from the market including quality, cost, schedule and user requirements ([Carter92]).
The authors have analyzed important various domains such as the rental management domain, the sales management domain, the library management domain, the OLTP real time system domain [Itoh96], the plant control domain [Itoh94ab], the medical inspection domain [Itoh97], and the railway scheduling domain [Kawabata97], etc., and have found each domain process model. The authors recognized that the same kinds of tasks such as the collaboration task, the allocation task, the monitoring task, the operation task, and the trading task, etc. were commonly performed across these domains. The authors obtained a domain model by the domain analysis for the cooperative work domain. The applicability of the domain model of common general works such as cooperative works range over different application areas widely.

Recognition of the same kinds of tasks as generic tasks is important as shown in Fig.1. The model of each generic task can be applied across the domains. The domain can be considered as a combination of one or more generic tasks.

The authors acquire the terms used by the business / activity domain, and accumulates them. The authors clarify what the generic tasks are done by the domain. If the generic tasks are identified, the process models of the generic tasks accumulated in the past can be reused. The model of the generic tasks can be reused across two or more business / activity domains. A domain model is represented as a set of models of the generic tasks

The domain model of the wholesale business of drinks as an example in this paper can be a combination of collaboration task and trading task as generic tasks.

2. Purpose of Research

Table 1 shows Triadic Domain Model. In domain analysis and modeling, the authors obtain the domain problem model (a set of problems to be systematized) and the domain product model (a set of solutions to be systematized). The authors' objects in addition to these models are domain process models. The target of the domain process model includes the reusable development technology. This is a domain model which represents the process by which the domain product model is derived from the domain problem model [Itoh 94 and 95b]. In the rental / sales / management domains, the problem models were obtained by clarifying how to identify the problem and the components of problems. The product models were obtained by clarifying DB and transactions. The domain process models were obtained as process for deriving the product from the problem.

Keys for how to identify the problem and the components of problems are to find personnel who
work in collaboration task and to clarify the interrelationships among personnel and other entities. To find personnel can be performed from the viewpoint of analysis of the collaboration task as the generic task. To clarify interrelationships can be performed from the viewpoint of the collaboration task and the trading task. The authors have obtained the domain process model for how to identify them. In order to incorporate the process model into analysis activity, the authors developed a navigator called NAVI for analysis on collaboration and trading tasks which result in RDB-based implementations. NAVI incorporates the domain process model, i.e., the model for development of a family of systems in the domain. NAVI also incorporates the reusable library of domain models. Therefore, users can perform the system analysis for the instance of the domain according to not a rigorous process model but a process model with short-cut path using reusable library.

In this paper, the features of NAVI are explained using a collaboration task and a trading task in wholesale business domain of drinks.

3. Hierarchy of Domain Model and System Instance

3.1. Domain Model and System Instance

The domain model is the model which is peculiar to the business / activity and which is a prototype for the system development. The domain model consists of the component with logical terms called domain vocabulary. The system instance is an individual system produced by the domain model. The system instance is obtained by allocating local physical terms (called system vocabulary) to terms in domain vocabulary and modifying them, and adding new terms. One or more local or physical terms in system vocabulary can correspond to one logical term in domain vocabulary.

3.2. Domain Model Library and System Instance Library

The domain model library is the library that two or more domain models are collected for reuse. As one or more term in system vocabulary corresponds to one term in domain vocabulary for one domain model, we can make a system instance by selection of terms in system vocabulary. In addition, the typical system instance can be registered in the system instance library.

3.3. Relationships with Domain Process Model

The process for allocation or replacement of terms in system vocabulary for terms in domain vocabulary and for registration of domain models / system instances into domain model library / system instance library is captured as a domain process model. The authors’ NAVI incorporates this
domain process model.

4. Domain Model of System which can be Implemented on RDB

Domain model library of the systems with RDB-based implementation includes the wholesale business domain of drinks, the library service, the rental CD service, the rent-a-car service, the lesson registration service, the hotel business, the delivery service of newspaper, etc. The wholesale business domain of drinks is used as an example. In this domain, the personnel do dealings with the fixed form slips, and do the deliver of the drinks and the transfer of money.

The components of the domain model obtained by the analysis of the wholesale business domain of drinks are as follows: (The terms of this components are registered in the domain vocabulary. )

(1) Six kinds of PEI (Primary Entity to be Identified) classes
   (a) Goods ledger
   (b) Wholesale ledger
   (c) Store ledger
   (d) Order slip to wholesale from store
   (e) Shipping slip to store from wholesale
   (f) Shipping-instruction slip

(2) Personnel classes
   (a) Receptionist of wholesale
   (b) Warehouse keeper of wholesale
   (c) Store

(3) Association between classes

(4) Transaction by personnel

The slip is a record of the content of dealings passed among the purchasers and vendors. The ledger is a summary of information of the goods, the wholesale, or the store.

From the viewpoints of the collaboration and trading tasks as generic tasks, this domain process model is analyzed as follows:
(i) Identification of personnel as PEIs
(ii) Identification of slips as PEIs
(iii) Identification of ledgers as PEIs
(iv) Identification of slips that personnel issue or receive as associations
(v) Identification of ledgers that personnel update as associations
(vi) Identification of operation flows as transactions that personnel update ledgers according to slips or that personnel issue slips according to ledgers

5. Development Support Navigator

NAVI can support analysis of systems with RDB-based implementations. The effectiveness of the system analysis and development can be improved by using this NAVI. NAVI incorporates the domain process model, i.e., the model for development of a family of systems in the domain (as mentioned above). NAVI also incorporates the reusable library of domain models. Therefore, users can perform the system analysis for the instance of the domain according to not a rigorous process model but a process model with short-cut path using reusable library.

5.1 Function of Navigator

Main functions of NAVI are as follows:

(1) Structure description of class (personnel, slips, and ledgers): The attributes of the slips, the ledgers, and the personnel can be described.

(1-1) Registration of new classes (personnel/ledger and slip)
(1-2) Use and non-use of the classes are specified
      : The necessary one is selected from the registered class.
(1-3) Registration of new attributes
(1-4) Use and non-use of the attributes are specified
      : The necessary one is selected from the registered attribute.

(2) Association description between classes: Relationships among slips, ledgers, and personnel
(3) Registration and allocation of terms: Registration and allocation of domain vocabulary and system vocabulary

(3-1) Domain vocabulary and system vocabulary are associated in all of the class names, the attribute names, the association verbs, and the transaction names.
(3-2) Domain vocabulary / system vocabulary is modified with registration.

(4) Definition of transaction: The content of the operation to ledgers and slips by the personnel can be defined.
The domain models and the system instances are registered in the library.

The changes in the system vocabulary can be reflected in all classes, the associations, and the transactions in the same domain. This function is provided for keeping the consistency of the system instances. This change can be reflected in analysis specification in RSL and SQL.

In addition, NAVI has the following functions to manage domain models and system instances.

(7) The domain model can be selected from the domain model library.
(8) The system instance can be selected from the system instance library.
(9) The domain model and the system instance which have been modified can be registered as a new domain model and a system instance.

5.2 Process of Navigator

Fig.2 shows the navigation process by NAVI.
(a) the selection of the domain model (left of Fig.2)
  : A suitable one for the business / activity to be developed is selected from among the domain model library which has already been registered. This domain model consists of logical terms in domain vocabulary.
(b) to (e) show a main process of NAVI.

(b) Selection of classes: Classes of PEIs and personnel are selected.
(c) Selection of domain vocabulary and system vocabulary
  : The system vocabulary according to the class selected by (b) is shown and selected for the object system. Or, new terms as system vocabulary can be registered. Moreover, the data types of terms in the system vocabulary are specified.
(d) New addition terms in domain vocabulary
(e) Presentation and selection of the associations between classes
(f) Definition of transactions where personnel originates

Fig.3 shows the selection process of the domain models of above-mentioned (a). The domain model for the wholesale business domain of drinks was selected. Fig.4 shows the definition of terms in the system vocabulary of PEIs corresponding to (b) (c). Fig.5 shows the definition of the personnel classes. Fig.6 shows the definition of the associations between classes corresponding to above-mentioned (e). Fig.7 shows a description of the transaction where the personnel originates by SQL.
5.3 Output of Navigator

There are two kinds of registration of the output obtained by NAVI. As the first type of registration, the instance of domain model is registered (feedback loop of the inside in Fig.2). As the second type of registration, the domain model is registered newly into the library (feedback loop the outside in Fig.2). The latter shows that a new domain model is acquired when the classes are added or deleted to / from the domain vocabulary.

There are RSL and SQL forms in this navigator NAVI output. NAVI outputs the analysis specification in the RSL (Requirements Specification Language) form based on object-oriented analysis. SQL expression is an input to RDB. RSL specification and SQL expression are shown in Fig. 8 and 9 for the system instance of the wholesale business domain of drinks.

6. Concluding Remarks

This paper described domain models for systems in RDB-based implementations, and NAVI which supports system analysis based on the domain model. This domain models are organized from the viewpoint of generic tasks such as the collaboration task and the trading task. The authors can perform domain analysis and modeling for various domains using such generic task models. The authors can apply NAVI for these domains.

References


<table>
<thead>
<tr>
<th>Factor characterizing each model</th>
<th>Domain problem model</th>
<th>Domain process model</th>
<th>Domain product model</th>
</tr>
</thead>
<tbody>
<tr>
<td>What to be modeled</td>
<td>Problem domain</td>
<td>Development process domain</td>
<td>Application domain</td>
</tr>
<tr>
<td>Way for model elicitation</td>
<td>Analysis on a family of problems, goals or missions in system development</td>
<td>Analysis on a family of practices in system development</td>
<td>Analysis on a family of system products (specifications or implementations) in system development</td>
</tr>
<tr>
<td>Objects to be identified</td>
<td>Typical problem components</td>
<td>Typical activities in domain specific system development</td>
<td>Typical specification / program components</td>
</tr>
<tr>
<td>Attributes to be identified</td>
<td>Terminology, vocabulary, data type, data value range, interrelationships, dynamics</td>
<td>Reasonable ordering for activities</td>
<td>Libraries, behavior, parameters, algorithms, data structures, resources, feasibility constraints</td>
</tr>
<tr>
<td>Heuristics to be used</td>
<td>Identification of domain boundary, analogy for domain and domain instance</td>
<td>Heuristic ordering for activities</td>
<td>Criteria for trade-off / recommendation / selection based on feasibility and cost</td>
</tr>
<tr>
<td>Representation style</td>
<td>Natural / specification / programming language, diagram, form, template, scenario, fill-in-blank documents</td>
<td>Verbal / informal/ formal procedure</td>
<td>Natural / specification / programming language, diagram, form, template, scenario, fill-in-blank documents</td>
</tr>
</tbody>
</table>
Fig. 1 Relationships between Domains and Generic Tasks

Fig. 2 Navigation Process by NAVI
Fig. 3  List of Domain Models

Fig. 4  Definition of PEI Classes
Fig. 5  Definition of Personnel Classes

Fig. 6  Definition of Associations between Classes
Fig. 7  Definition of Transactions

```sql
INSERT INTO
store_lodger (store_code, store_name, address, telephone_number)
VALUES (store_code, store_name, address, telephone_number);
```
create table order_ledger( order_code number(8,0), order_day date, store_code number(8,0), store_name char(50), goods_code number(8,0), goods_name char(50), unit_price number(8,0), amount number(8,0), total number(8,0), schedule_of_shpping date);
create table store_ledger( store_code number(8,0), store_name char(50), address char(50), telephone_number char(50));
create table goods_ledger( goods_code number(8,0), goods_name char(50), unit_price number(8,0), amount number(8,0));
create table order_slip( slip_code number(8,0), making_day char(50), store_code number(8,0), store_name char(50), desirable_shipping_schedule date, goods_code number(8,0), goods_name char(50), unit_price number(8,0), amount number(8,0), total number(8,0));
create table shipping_instruction_slip( slip_code number(8,0), making_day char(50), store_code number(8,0), store_name char(50), shipping_day date, goods_code number(8,0), goods_name char(50), unit_price number(8,0), amount number(8,0), total number(8,0));
create table shipping_slip( slip_code number(8,0), making_day char(50), store_code number(8,0), store_name char(50), shipping_day date, goods_code number(8,0), goods_name char(50), unit_price number(8,0), amount number(8,0), total number(8,0));
create table store( store_code number(8,0), store_name char(50), address char(50), telephone_number char(50));
create table receptionist_of_wholesale();
create table warehouse keeper_of_wholesale();

INSERT INTO store_ledger( store_code, store_name, address, telephone_number)
VALUES( store_code, store_name, address, telephone_number);

INSERT INTO goods_ledger( goods_code, goods_name, unit_price)
VALUES( goods_code, goods_name, unit_price);

DELETE FROM store_ledger WHERE store_code = X;

DELETE FROM goods_ledger WHERE goods_code = X;

SELECT stock FROM goods_ledger WHERE goods_code = X;

BEGIN
SELECT store_code, store_name, goods_code, goods_name, unit_price, amount, total_price INTO
V_STORECODE, V_STORENAME, V_ITEMCODE, V_ITEMNAME, V_UNITPRICE, V_ITEMQUANTITY, V_TOTALPRICE FROM account book WHERE order_code = x;
END;

INSERT INTO shipping_instruction_slip
VALUES(slip_code, SYSDATE, V_STORECODE, V_STORENAME, V_ITEMCODE, V_ITEMNAME, V_UNITPRICE, V_ITEMQUANTITY, V_TOTALPRICE);

BEGIN
SELECT store_code, store_name, goods_code, goods_name, unit_price, amount, total_price INTO
V_STORECODE, V_STORENAME, V_ITEMCODE, V_ITEMNAME, V_UNITPRICE, V_ITEMQUANTITY, V_TOTALPRICE FROM account book WHERE slip_code = x;
END;

BEGIN
SELECT order_code, slip_code, making_day, store_code, store_name, goods_code, goods_name, unit_price, amount, total_price INTO
V_STORECODE, V_STORENAME, V_ITEMCODE, V_ITEMNAME, V_UNITPRICE, V_ITEMQUANTITY, V_TOTALPRICE FROM shipping_instruction_slip WHERE slip_code = x;
END;

BEGIN
SELECT order_code, slip_code, making_day, store_code, store_name, goods_code, goods_name, unit_price, amount, total_price INTO
V_STORECODE, V_STORENAME, V_ITEMCODE, V_ITEMNAME, V_UNITPRICE, V_ITEMQUANTITY, V_TOTALPRICE FROM shipping_slip WHERE slip_code = x;
END;

Fig.8 RSL Specification for Wholesale Business of Drinks
store_name, shipping_day, goods_code, goods_name, unit_price, amount, total)
VALUES (slip_code, SYSDATE, V_STORECODE, V_STORENAME, V_ITEMCODE, V_ITEMNAME, V_UNITPRICE, V_ITEMQUANTITY, V_TOTALPRICE);

UPDATE goodsLedger
SET stock = stock - V_ITEMQUANTITY
WHERE goods_code = V_ITEMCODE;

End;

Fig. 9 SQL Expression for Wholesale Business of Drinks