# DEVELOPMENT OF ACTIVITI-BASED DOMAIN ANALYSIS (ADA) BY DESIGNING SPATIAL DATA INFRASTRUCTURE (SDI)

**Tsuneki SAKAKIBARA*, Ryosuke SHIBASAKI****
**University of Tokyo, Tokyo**
**Center for Spatial Information Science (CSIS)**
[illegible]
****shiba@skl.iis.u-tokyo.ac.jp**

Working Group C#/G#

**KEY WORDS:** Database Designing, Spatial Data Infrastructure (SDI), Object Oriented Analysis and Design (OOAD), ACTIVITY, Application Schema (AS)

## ABSTRACT

Last decade, there increase various digital information, especially spatial information in our life. And we usually also refer those spatial information. However, we cannot provide those spatial information services easily. This is because development and maintenance of data require huge costs. Another reason is the duty related with data maintenance is expected quite heavy.

Consequently, in the early step of use of spatial information, development of a Spatial Data Infrastructure (SDI) is required to promote currency of SI and enlargement of spatial information industry.

Therefore, in this study, we finally aim to develop a methodology to design a data architecture of SDI, effectively and efficiently. To archive this goal, we divide it into three steps. The first is "development of ACTIVITY-based Domain Analysis (ADA)". We consider ADA as a methodology to analyze and to clarify needs for (spatial) information. In this paper, we will discuss ADA. The second is "development of an information use model". Using this model, we simulate the change of activities by the changing of provided information. The last is "development of cost-effectiveness model". By this model, we aim to estimate the effectiveness of each data item or attribute.

In this study, we developed a structure and a procedure of ADA. But now those are not well defined at present. Since we design the system architecture of ITS or Spatial Data Infrastructure, we, now, formulate the structure and the procedure of ADA. In addition, as example of analyzing by ADA, we will clarify the architecture of ITS and Spatial Data Infrastructure.

## 1 OVERVIEW

### 1.1 Background

According to the progress of mobile and network computer environment, we frequently use various and enormous digital data. That is the same situation for the spatial data. When the needs grow, the suppliers also increase. So now there are a lot of vendors who develop digital spatial data with map projection in Japan; map publishers, surveying companies, car navigation system vendors and so on.

At present vendors develop their own product independently. And of course every product has some characteristics until in detail. However, there is not so quite difference, especially the basic features (primitives) such as buildings, land lots or governmental boundaries. We can almost say that much multiple investment has been done to develop basic digital spatial data.

By the way, now we discuss the nature of spatial information. In the ISO/TC211 standardization drafts, the conceptual information model, which is held by each GIS user, is called "Universe of Discourse (UoD)". Naturally, all of UoDs are different, but it can be said that the basic or primitive features in almost all of UoDs are mutual. Next, we also discuss the technology of developing digital spatial data. The technologies used by each vendor are also not so difference. So, we can also say, the products' quality is almost the same level in the basic features. (The main discrimination factor among vendors is the data that are corrected mainly by field survey.) Therefore we should develop the basic or primitive data and share them as an infrastructure.

### 1.2 Existing approach & discussion

Now we review the existing approaches related designing GIS database. When we install GIS in our business, we often do not analyze the needs in detail. But in an organization, when we plan to install GIS in various divisions, we try to share a database. At that time, we have to comprehend the needs for data. In Japan this kind of sharing is also done in some local governments. We call this sharing GIS database "integrated-GIS".

We briefly explain the most popular approach to design it. At first, we decide the divisions sharing database and list up the data used in each division. Then we just find out the data that have the same or close meanings and make a kind of thesaurus. And we select the data expected relatively many users. The result of this is main container of the sharing database.

This approach is not only systematical but also methodological rational. Especially, by this way, we never comprehend potential users of spatial data. And another program is the result has very close relationship between the existing GIS usages. Almost all of existing GIS usages are just routine works. In the usages, the data structure should be quite simple. So if we design database based of this structure, it is difficult to expand the database in future.

In the next, we refer the ISO standardization. There is some organization work for standardization of spatial information. In ISO, there are two technical committees (TC); TC211 and TC204. TC211 is the TC for discussing GIS. They discuss mainly spatial data model, cataloging and quality of data. They tend to deal with conceptual and exchanging standardization. TC204 is originally the TC for discussing Intelligent Transportation Systems (ITS). In ITS, navigation systems are exactly geographic information systems. So they also deal with spatial data. In TC204, they do not discuss spatial data model or quality, but try to standardize the analysis method to expose the ITS related information systems and data structure.

### 1.3 Purpose

As we discuss above, at least the primitives should be developed as an infrastructure. And when we develop it as an infrastructure, we are required to satisfy the followings. First is data needs are reflected to the structure. Second is the structure should have some temporal stability, at least decade unit. Third is it should be developed effectively and efficiently. And the last is we have accountability of the data structure. But now we do not have designing methodology. Therefore we aim to develop "ACTIVITY-based Domain Analysis (ADA)" as the methodology.

## 2 PROPOSAL OF ACTIVITY-BASED DOMAIN ANALYSIS (ADA)

ACTIVITY-based Domain Analysis (ADA) is developed as the methodology for designing the conceptual and logical architecture of database. However, the existing approaches are based on the existing database, ADA is based on the data needs. By ADA we clarify the needs of spatial data users and decide the data architecture, following the clarified needs. So the ADA is the tool for the first step of designing database.

### 2.1 Terminology in ADA

Before the explanation of ADA procedure, we need to share the terms in ADA.

(1) **DOMAIN**: A Domain is the basic unit of society or organization. So we consider a domain is suited to a business community. In ADA, we deal with "daily life" as one of domains.

(2) **ACTIVITY (ACT)**: An ACTIVITY, ACT is a set of activities that have the same objective or aim, not the flow. And it can be briefly defined as the minimum unit of a series of activities. An ACT can hold a hierarchical structure, a sub-ACT and a super-ACT. A sub-ACT is a branch of a parent-ACT and has the sub-objective or sub-aim of the parent-ACT. Oppositely a super-ACT is a root of ACTs, which belong to different domains and have similar objectives or aims. We could obtain sub-ACTs by dividing an ACT and a super-ACT by aggregating some ACTs.

(3) **ACTOR**: An ACTOR is the subject of each ACT. It is arrowed an ACTOR plays as the subject of several ACTs. By defining an ACTOR for every ACT, we can also define the viewpoint to the target world.

(4) **APPLICATION SCHEMA (AS)**: An Application Schema, AS is a set of information handled in an ACT. In this definition,

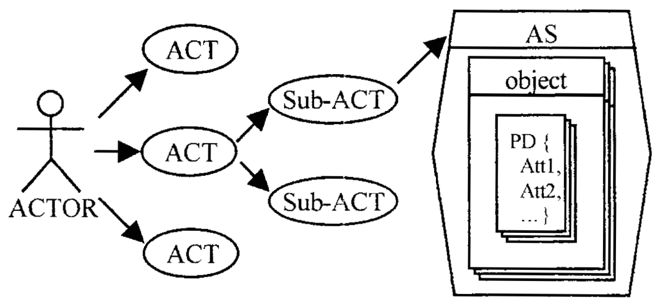


Fig. 1 Relationship from ACTOR to primitive data (PD)

"information" means "input data, output data and constrains (mainly related with resources)". An AS composed of a set of objects and the object-composing data. An object is a set of primitive data. For example, "weather" is an object and "place, date, weather, temperature, and etc" are the set of primitive data. In this context, it can consider an object is just the name of dataset holding social meanings. And finally every primitive data have attributes. So attributes are the lowest component of ADA.

## 2.2 Procedure of ADA

Now we explain the procedure of ADA. ADA is composed of 5 steps.

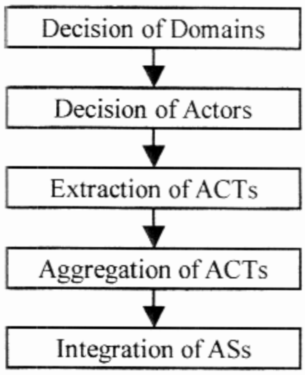

Fig. 2 Procedure of ADA

### 2.2.1 Decision of Domains

At the beginning of ADA, we have to decide Domains we analyze after this. In this step what we have to pay attention is the chosen Domains imply both Actors and ACTs. This means by deciding Domains it is automatically decided the Actors and ACTs analyzed by the following steps. In another word, deciding Domains is almost equal to deciding the core users of the database. So we should recognize the nature of developing database in choosing.

### 2.2.2 Decision of Actors

As the Domains are fixed, now we extract the Actors in every Domain. In this step, we should also confirm the nature of the database. If we expect the users are limited and the database is required to contain the data in detail, we might deal with the people who dose exception handling as an Actor. When we set a business community as a Domain, we consider divisions as Actors. Anyway in this step we extract only the main Actor in each Domain.

### 2.2.3 Extraction of ACTs

In this step we extract the main activities as ACTs for every Actor. As the same in "Decision of Actors", we tend to obtain just the main activities. And firstly we extract all of general ACTs. After that for each ACT we divide the general objective into sub-objectives as sub-ACTs. However, this division can be repeated endlessly, it had better to stop by getting 4 - 6 generations. In this paper we develop the designing methodology for spatial information database. So in this extraction, we have to extract without missing the ACTs dealing with spatial information.

In this step, we also extract ASs; objects and the composing primitive data. This extraction is mainly done for the lowest generation. Here, we need not to extract attributes of primitive data. After the following steps, a set of a lowest generation and the AS is the unit of analysis.

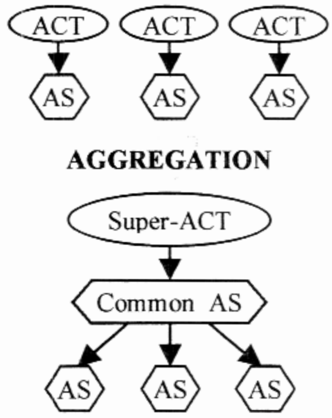

Fig. 3 Aggregation of ACTs

### 2.2.4 Aggregation of ACTs

As the ACTs and ASs extracted, we aggregate ACTs. "Aggregation" means to aggregate the ACTs that have similar objective. But, perhaps, enormous numbers of ACTs are extracted. So we cannot check the similarity of ACTs one by one. Therefore firstly we find out the ACTs that have relatively similar AS. Then reviewing the similarity of ASs, we aggregate them into a super-ACT. When some ACTs are aggregated into a super-ACT, the ASs held by the ACTs belong to the super-ACT. At this time, the common parts of AS within all of the ASs are also aggregated as a common AS.

Through this step, the total number of ACT decreases. So we obtain super-ACTs and non-aggregated ACTs as the result. Here we must discuss the weight of each ACTs, especially between super-ACTs and non-aggregated ACT. Fundamentally we, even, have to consider all of extracted ACTs. But now we are under investigation of the ACT-evaluating criteria.

### 2.2.5 Integration of ASs

At the last step, we integrate all of ASs into an only super-AS. In the integration, we merge the same or similar objects into an object and all of these objects belong to a super-AS. If the object, whichever belonging to a super-ACT or a non-aggregated ACT, it is integrated an only super-AS.

As the result of ADA, we obtain a super-AS. But this super-AS is not equal to the final data architecture of developing database.

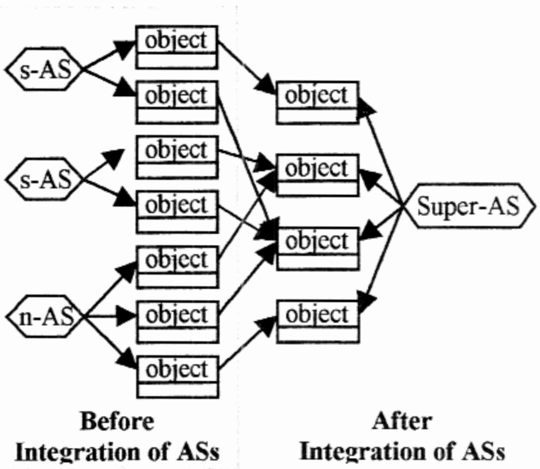

Fig. 4 Integration of ASs

By the super-AS, we just obtain the maximum data architecture.

To get the final data architecture, we need to evaluate the result on the point of view from efficacy, frequency and etc. using the weight attended to each ACTs.

## 3 PROTOTYPING OF ADATOOL

Now we know the procedures of ADA. But, as ADA composed of many complex steps, it has some difficulties to execute the analysis. Then we are also developing the software prototype to support the analysis. We call this ADATool.

### 3.1 Concept of ADAtool

We develop ADA based on structured analysis and other object-oriented analysis and design (OOAD). Therefore each analysis process are drawn by UML or written by natural languages. So we finally aim to ADATool should be a kind of case tool. To make it case tool, we can get total and seamless designing environment from analysis to evaluation.

The 6 steps in ADA can be divided into two types of execution process. One is the process under distributed environment and another is that under concentrated environment. The former is the step, "decision of Actors" and "Extraction of ACTs" and the latter is the rest. These two types of process are quite different. So we divide the ADAtool into two. The first one is called ClientTool and the second, CenterTool.

If the developing database has much various Domains, the users of ClientTool increase. On the other hand, the number of analyzer in concentrated environment is not influenced by the variety of Domains. The group should be always composed of about 5 people.

Now we are under developing the ClientTool. So in this paper we only show it.

### 3.2 Data Structure in ADATool

In ADA, we have to deal with many different kind of information. And some of them have hierarchical structure. Therefore, before developing ClientTool, we decide the data structure. It is shown in Fig.4.

In this structure, we divide Category and Domain. Category is the set of objects that can be unified by a theme. For example, "Traffic Accident" or "Traffic Jam" is Object and they are labeled by the Category "Traffic". It will provide some useful information on the step "Integration of ASs".

Object is a component of category and must have more than one property. Property is exactly equal to primitive data. In this prototype, Object exists in Domain, but it might exist out of Domain like Category or exist in Category. Actor is in Domain and has more than one ACT and also more than one ObjRef. Object is the definition of all of Object referred in

Table. 1 Data Structure in ADATool

| | | | |
|---|---|---|---|
| Category | | | |
| | CatId | | |
| Domain | | | |
| | Name | | |
| | Object | | |
| | | Id | |
| | | CatId | |
| | | Name | |
| | | Property | |
| | | | id |
| | | | Name |
| | Actor | | |
| | | Name | |
| | | ACTIVITY | |
| | | | weight |
| | | | Name |
| | | | ACTIVITY |
| | | ObjRef | |
| | | | id |
| | | | infoType |
| | | | Name |

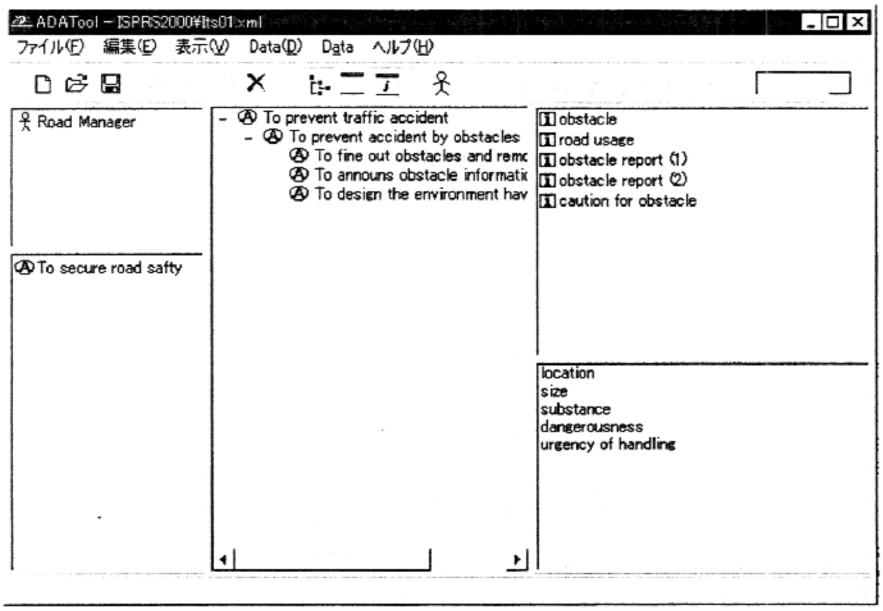


Fig. 5 Main window of ClientTool Prototype

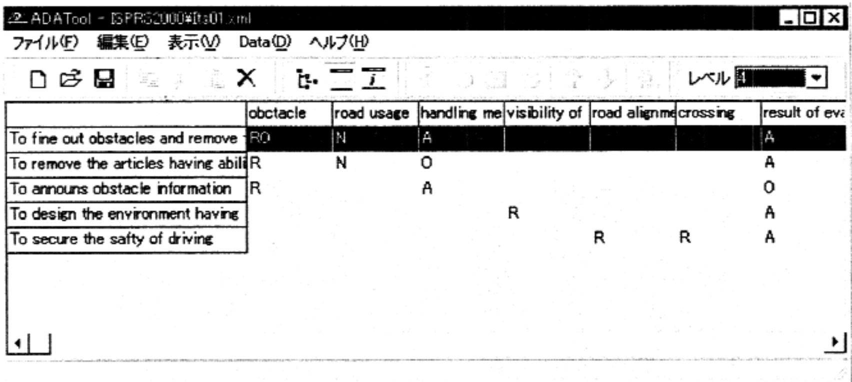


Fig. 6 Matrix view of ACT-Object tree

this Domain. And ObjRef is the Object which is referred by the including Actor.

### 3.3 ClientTool Prototype

Fig.5 is the main window of ClientTool prototype. As we mention before, we design this ClientTool as the UI for the steps "decision of Actors" and "Extraction of ACTs". This version is a standalone application, but in future we revise this to a web application to enable to analyze under distributed environment. In this prototype we use xml as the file type. Using xml, it can be easy to describe complex data architecture of ADA. So we make the DTD following data architecture in Table.1.

In Fig.5, the left-upper window shows Actors, left-lower, the first level ACTs, middle, ACTs lower than the first level, right-upper, object and right-lower, primitive data. For ACTs, we can set the relative weight. When we set the weight, we just consider the relative ratio within the brother-ACTs. To set the weight by this rule, it is automatically calculated the absolute ratio in the whole ACT tree. In addition, however, there is a description about Domain, there is no place to setting Domain.

One of the most important characteristics in this application is to set an alias to an Object. To do this, we make a kind of thesaurus from technical words to general words, and from technical words in a Domain to that in other Domains. In this sense, by setting aliases, we also communize concept.

This prototype has another view; matrix view of ACT-Object tree in Fig.6. In this view, ACTs are shown in the first column and Objects, the first row. The letter "N" in the matrix means "an ACT **needs** an Object". Others have the same relationship "R", **require**, "A", **available**, "O", **output**. By this view, we can get the overview of the relationship between ACTs and Objects. By this view, we can get the overview of the reference relationship among ACTs and

Objects. For the first column, we can set the ACT generation. So an ACT refers an Object repeatedly, it is shown many letters in the reference field.

This prototype is just the first version of ClientTool. By the final version, we will contain the input wizard of ACTs and Objects to support analysis. And dividing Category and Object part to an independent file, we plant to enable to share them among all of analyzer. To share Category and Object, it helps the step "aggregation of ACTs" and "integration of ASs".

## 4 CONCLUSION AND FUTURE WORKS

In this paper, we briefly explain the procedure of ADA and developed the ClientTool. ADA is the analysis methodology for designing conceptual and logical data architecture. However, the result cannot be used as the actual data structure, because it has a lot of unsubstantial information, which is not adequate to share. Therefore, in nearly future, we have to add the evaluation procedure to ADA. And we also have some inexplicit parts in the present procedures. Originally this analysis methodology must be a user-oriented metrology. So we refine the procedure, develop the application and apply to the actual example simultaneously. As the example, we are applying to designing the database for national road management.

## REFERENCES

[1] Ivan S., Richard B., Tom R., 1994. Cooperative Systems Design. The Computer Journal, 37, 5, 357-366
[2] Toyoaki N., Tetsuo T., Takashi K., Hideaki T., 1998. Management of Engineering Knowledge. Asakura Syuppan (in Japanese)
[3] Jones P., 1988. Practical Guide to Structured System Design Second Edition. Prenetice-Hall Inc.
[4] Sully P., 1993. Modeling the World with Objects. Prentice Hall International Ltd.
[5] Yourdon E., 1994. Object-Oriented System Design An Integrated Approach. Prentice-Hall Inc.
[6] Mitsuru I., 1997. Knowledge Representation and High Speed Reasoning. Maruzen Syuppan (in Japanese)
[7] Tom D., 1997. Structured Analysis and System Specification. Prenetice-Hall Inc.
[8] Ivar J., Maria E., Agneta J., 1994. The Object Advantage Business Process Reengineering with Object Technology. Addison-Wesley.
[9] Ray B., Ron C., 1996. Use Case Maps for Object-Oriented Systems. Prenetice-Hall Inc.
[10] Ryoichi M., Masanobu K., 1999. An Ontology of Faults – Articulation and Organization. Journal of Japanese Society for Artificial Intelligence, 14, 5
[11] Mitsuru I., Satoshi K., Ryoichi M., Masanobu K., 1999 Fault Diagnosis Based on Ontological Consideration of Faults – Exhaustive Fault Hypotheses Generation. Journal of Japanese Society for Artificial Intelligence, 14, 5
[12] Percles L., Vassilios K., 1995. System Requirements Engineering. McGraw-Hall
[13] Mitsuru I., Haruki U., 1997 Knowledge Representation and Use. Ohmsha
[14] Kaname I., 1997. Approach for System Thinking. Seibundo Printing
[15] Haruo S., 1995. Fuzzy Theory. Kyoritsu Syuppan