1. Introduction
For the purpose of getting the latest map information of desired themes and regions at desired or proper scale whenever and wherever, we proposed a Multi-level / Multi-theme (M²) map information model[1] to maintain maps in consistency with original source datasets under distributed environment. For the perfect information sharability and consistency, there is no overlap among the datasets. So, the modification referring to several datasets results in a complex process. To solve the problem, we propose an index structure MC-R-tree to manage datasets of different levels effectively.

2. Framework
M² model gives a division method for countrywide map information, and is defined as two kinds of hierarchies: one is a directory tree, which is obtained by recursively decomposing map regions into a sequence of increasingly finer tessellations; and another is a theme tree, which consists of multi-level theme map datasets of regions corresponding to the directory tree.

As map objects are assigned to corresponding datasets without repetition, there is no overlap among these datasets: for a particular requirement, map information can be prepared dynamically by using generation functions. For example, to generate a city-level road map, we can combine the information of country and prefecture roads inside the region managed in city-level datasets. Though map information is managed distributedly, there are relationships among them: the prefecture road and city roads are managed in different level dataset; the information of an intersection between a prefecture road and a city road is managed in the city level. Thus, the datasets in the lower level is a composition of the parent node object in the lower level: e.g., the object is a road segment; the first composition of it is one of the end points; and the second composition is the least important level of child-node.

The internal nodes may contain the first two kinds of entries; and the leaf nodes contain object-entries. So, the modification referring to several datasets results in a complex process. To solve the problem, we propose an index structure MC-R-tree to manage datasets of different levels effectively.

3. Index structure for integrated maintenance
As there are relations among datasets, modification referring to several levels should be done integratedly: not only the spatial information of levels but also the relations among them need to be maintained. We propose MC-R-tree index structure for the integrated maintenance.

3.1 Definition of MC-R-tree index structure
To differentiate the levels of objects, a logical important value of a natural number is assigned to each object, in agreement with the map level: e.g., 0 for objects in country-level datasets, 1 for that in prefecture-level, and so on. We proposes a hierarchy, called main hierarchy, to record the objects with the higher important values in the higher levels of the hierarchy; and we propose another kind of hierarchy, called a composition hierarchy, to keep the relations among levels, which are pointed by leaf nodes of main hierarchy (Fig. 1). The main hierarchy is based on R-tree [2], which assures the effective spatial operations on the map information.

Each node in the hierarchy contains a number of entries, and there are three kinds of entries with the following forms:
1. Object-entry has the form (MBR, flag, composition-pointer), where MBR is the minimal bounding rectangle of the composition hierarchy, flag is a natural number that indicates the important level, and composition-pointer contains a reference to a composition hierarchy;
2. Tree-entry has the form (MBR, flag, child-pointer), where child-pointer contains a reference to a sub-tree. In this case MBR is the minimal bounding rectangle of the whole sub-tree and flag is the least important level of child-node.
3. Composition-entry has the form (composition-id, next-pointer, next-level-pointer), where composition-id is the identifier of object's composition; next-pointer contains a reference to the next composition of the parent node object: e.g., the object is a road segment; the first composition of it is one of the end points; and the next-pointer points to the next point on the same object. Next-level-pointer contains a reference to the composition of the parent node object in the lower level: e.g., the intersection between the upper-level road and lower-level road.

The internal nodes may contain the first two kinds of entries; the leaf nodes contain object-entries.

3.2 Creation of MC-R-tree
When system-user wants to update roads of a specific region, the index is created based on the referring datasets. Here, we give the algorithm for two levels:
(1) Create bottom nodes of main hierarchy based on the objects in the lower level;
(2) Create composition hierarchy for objects of upper level: next-level-pointer points to the intersection node, which is the intersection between upper-level and lower-level roads;
(3) Create main hierarchy based on the nodes generated by 1) and 2).

An example for two-level road network is given in Figure 2.

3.3 Example of update to datasets

There are important flags in object-entries and tree-entries: the index structure provides an integrated way to access datasets in different levels; the relations among levels are represented in the composition hierarchies; so, the simultaneous modification to multiple levels is possible. Here we only outline the main steps:
(1) Descend the main hierarchy to find the object-entries referred by the modification. As in the hierarchy the objects with higher importance are located on the upper level, the descending step first stops at the nodes of upper-level objects, which point to composition hierarchy if there is one;
(2) Modify datasets in multiple levels based on the relations in the composition hierarchy, and at the same time adjust the composition hierarchy;
(3) Adjust the main hierarchy.

We give an example of deleting the road segment between nb3 and na4. The operation refers to datasets in two levels. Before the update, nb3 belongs to the level 2, and na3 and na4 belong to the level 1. The road segments na3-nb3 and nb3-na4 are generated in the level 2; however, after the update, as the node nb3 becomes an end-point of the upper-level road, the road segment managed originally in the upper-level a3 (na3-na4) should be replaced by na3-nb3, and the information about nb3 should copy from the lower level to the upper level.

4. Conclusion

We have developed prototype systems in Java, which manages road and architecture themes of country, prefecture and city levels, based on the maps of Japan, Aichi Prefecture and Ichinomiya City. Comparing the road theme in our model with other multi-level models [3][4], our model outperforms them in keeping distributed multi-levels datasets sharability and consistency, supporting the effective storage and efficient spatial query on road networks; especially, outperforms them in the integrated maintenance by using MC-R-tree.

For our future work, the more themes with complex spatial information and complex relations among levels will be considered.

Reference