# The Static Model of Latvian Forest Management Planning and Capital Value Estimation

Salvis Dagis[1] and Sergejs Arhipovs[2]

[1] Faculty of Information Technologies, Latvia University of Agriculture, Liela Street 2, Jelgava, Latvia

[2] Faculty of Information Technologies, Latvia University of Agriculture, Liela Street 2, Jelgava, Latvia

**Abstract.** Latvia, where forests cover up to 45% of territory, might be proud of its forests. Forestry is the most significant export sector in Latvia and in total forestry provides up to 14% of GDP. Regardless of the significant felling areas, there can be observed more increment than there are trees cut. The main aim of forestry policy is to ensure the sustainable management of forests and forest lands, therefore it is necessary to evaluate the current situation and think ahead in order to plan the cutting of forests

## 1 Introduction

The entity of object-oriented programming is shown as a cyclic movement between architectural and micro-developmental levels. Without any assistance, iteratively correcting the system, the designer goes from the micro-developmental level to the architectural level, as well as returns back to the system requirements, then again moves forward [1]. The system grows from the set of variant usage to the full package of system classes. The usage variants become more precise, when the understanding is deepened on what is happening within the precedents and out of what points the system should consist from. The development of system is an iterative process between the processes of precedents (Use Case) and class modelling, within which the conception on the transactions and classes necessary for the system emerges. The class models form the architectural basis of system.

## 2 The Static Model of Forest Capital Value

One of the aspects of the project "The determination of forest capital value" is the model of data and transactions. This model consists of other sub-system models that do not exist separately, but are shown together as one system.
The model "Forest capital value" includes such sub-systems as: "Forest inventory", "Inventory activation", "Capital evaluation", as well as other sub-systems. The speci-

Dagis S. and Arhipovs S. (2006).
The Static Model of Latvian Forest Management Planning and Capital Value Estimation.
In *Proceedings of the 4th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems*, pages 185-188
DOI: 10.5220/0002471101850188

fication of system and sub-system consists of usage variants, relations and operations between them and their interfaces. The system realization is a set of classes and other sub-systems, which ensures the characterization and behaviour of defined specification. These three sub-systems form the basis of system designing management. Every sub-system is the hierarchy of some classes that models the main points of subject domain.

### 2.1 Class Hierarchy of Forest Territories

The territory division point is the main component of sub-system "Forest inventory". There are main components of property territory are identified. The main aim of system is to plan and evaluate the property territory from the view of forest inventory and capital value.
The result of the territorial division process is modelled by means of `Collection` samples. Every unit includes the collection of elements. Thus the diagram of classes shows the relation properties, compartments and pieces as the totality of collections: there are many compartments in every property, there are many pieces in every compartment. Every collection of model can be implemented programmatically with a class, which implements the table, or interface collection, or dynamic list, or dynamic array [2].
There is one common attribute, characteristic to all territorial units. All units are abstract regions with their data, for example, border crossings, and methods, for example, the function that estimates the area of region. It means that the territorial division of forest is a hierarchical structure with the base class `Region` and derived classes: `Cadastre`, `Compartment`, `Piece`, the objects of which are grouped by collections. Figure 1 shows the fragment of the territorial division system of forest.

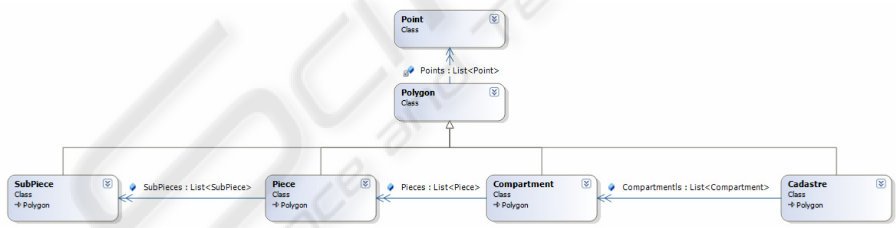


**Fig. 1.** Class hierarchy of forest territorial division.

Territorial hierarchy of forest models the place, where the forest inventory process takes place. The object of inventory process is a tree and the accounting of stock volume. Thus the class hierarchy of stock volume takes up a separate place in the diagram of forest inventory classes.

## 2.2 Class hierarchy of stock volume

The class `Tree` of the sub-system “Forest inventory” describes the real tree, which should be modelled in the system. The tree is the object that is characterized by such data as: species, age, height, and diameter. Two functions `VOLUME()` and `SQUARE()` define the functionality of tree – the area depends on the diameter of tree, but the volume is estimated depending on the species, diameter and height.
There exist two types of inventory: continuous inventory, where every tree is measured, and group inventory, where the trees are measured by grouping them according to species in one sub-piece. Using continuous inventory the cruiser examines every tree in the sub-piece and makes up a list of the sub-piece trees with the initialized `Tree` type objects.
Forestry science offers scientifically proved evaluation methods of trees and tree groups. Thus the cruiser has a possibility to choose not only continuous, but also group inventory. It is measured in absolute values as the area sum of every separate tree. The data structure filled by the cruiser is a class, which is inherited from the class `Tree`. Fields species, age, height and diameter are inherited and used without any changes, but with another interpretation: they are interpreted as average values. The previous quality `SQUARE` of the derived class `AverageTree` is re-defined not as the square value to be estimated, but as the area sum of tree group, which is evaluated by the cruiser. There is a quality `COUNT` added to the derived class, which only returns the number of trees of the given tree group.

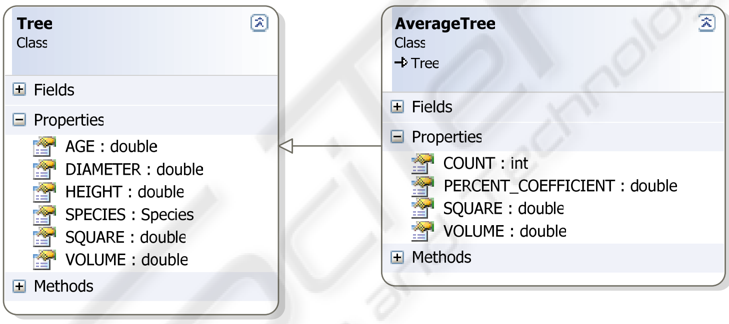


**Fig. 2.** Single and group trees classes.

Thus there are two tree inventory indicators depending on the forest inventory scenario: average indicators of a tree and actual indicators. In figure 2 both cases are related to the inheritance relation.

## 2.3 The class model of forest territories and stock volume

Since there are many trees, but not one tree growing in the forest territories, this base unit of theory consists of tree collections or the lists of typified trees shown also in figure 3.

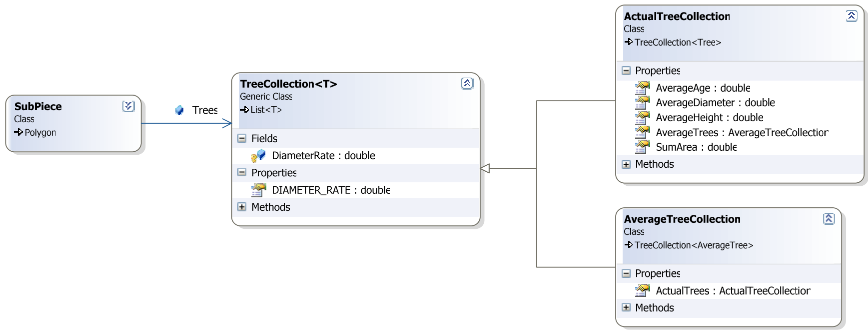


**Fig. 3.** Class `SubPiece` and its two possible collections.

The system does not know what type of inventory the cruiser is going to use in the real time. The cruiser will make this decision dynamically in the process of inventory. To this effect there was developed `TreeCollection<T>` the parametric class that models the tree collection and is the base class of two derived classes. Without specifying, which derived class will be used in the real time, the base tree collection `TreeCollection<T>` is added to the unit `SubPiece`. In order to evaluate the particular sub-piece in the real time it is necessary to choose the inventory scenario and to inform the system about it.

## 3 Conclusions

Within the project there had been developed class model and tested in the prototype of application program. There were structured data and algorithms of forestry in the diagram of classes. Algorithms and data structures were implemented and tested with real and virtual data. The updating methods of forest inventory indicators were tested with the obtained data structures that were enlarged with the attributes of capital values.

It is easier to acquaint oneself with the architecture of the system to be explored, its interaction with other systems or sub-systems by modelling static class diagrams of the system. Fully developed model of a class diagram effectively shows the peculiarities functioning within system, their characteristics and methods. Since the software code is classically generated only from the class diagrams, other diagrams are used only for the description during the modelling process.

## References

1. Robert J Muller, 1999. Database Design for Smarties: Using UML for Data Modeling. Morgan Kaufmann publishers, 1st edition.
2. Leszek A. Maciaszek, 2001. Requirements analysis and system design: developing information systems with UML, Addison-Wesley Longman Ltd. Essex, UK.