UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

A CONCEPTUAL MODEL OF CARDIAC ELECTROPHYSIOLOGY BASED ON UML

Carlos Bilich
www.carlosbilich.com.ar

July 11th , 2005

Technical Report # DIT-06-055

A Conceptual Model of Cardiac Electrophysiology based on UML

Carlos Bilich

University of Trento, Faculty of Science, Department of Information and Communication Technology, Via Sommarive, 10 - I - 38050 Trento, Italy
carlos.bilich@dit.unitn.it

Abstract. This paper constitutes the project of the Conceptual Modeling course taken in May 2005 at the Department of Information Technology of the University of Trento, taught by Dr. John Mylopoulos. This work is based on a previous survey paper submitted for the same course and present an evolution of the models described there. In this case, the model is created using a standard modeling tool such as the OMG's Unified Modeling Language (UML). UML presents characteristics that incorporate those of the previously surveyed models and includes others that facilitate the representation of concepts that the other languages could not explicitly model.

1 Introduction

As a part of the Conceptual Modeling course taught by Dr. John Mylopoulos, before this work, the author surveyed three of the most important models of the cardiac electrophysiology to date [4], that is, the KARDIO model developed by Bratko *et al.* [1] at the MIT; the Spatio-Temporal model by Weinberg *et al.* [2] from Vanderbilt University; and the Ticker Models of Kirby *et al.* [3] from the department of computer science of the King's College in Scotland. Even though the human heart is mainly composed of two subsystems the models tackled only the electrical system which is responsible for the generation of stimulation impulses that rhythmically cause the contraction of the heart muscle. All the models, in some way or another, incorporate some of the concepts that can be found all together implemented in UML, that is why the author thinks that UML is a good tool to synthesize all the ideas described in the previous attempts.

For example in Kardio, the model consists of a network composed of nodes; constraints; local rules; and global rules. An example of such a network can be seen in Fig. 1. It is not difficult to associate the diagram of Fig. 1 to an UML class diagram where the nodes are the classes and the network constitutes the association between them. UML classes and associations also support constraints fulfilling also this requirement. The state of the nodes can be associated to attributes of UML classes. For example the impulse generators and conductions pathways can be in normal or abnormal functional states. These conditions can perfectly become attributes where their condition can be set and/or got by a corresponding `GetImpGenState()` or `SetImpGenState()` method.

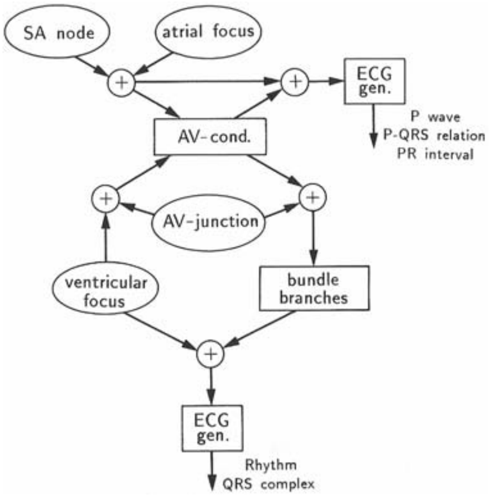

Fig. 1. Model of the heart as a network composed of impulse generators, conduction paths, impulse summators and ECG generators. [1]

With the rules of KARDIO the analogy with UML becomes even stronger because they are basically Prolog procedures than can analogously being seen as methods attached to the pertinent classes in UML.

In the case of the Spatio-Temporal Models of Weinberg *et al.* [2], one can also find similar analogies and equivalents in UML. Moreover, the authors introduce a terminology to designate the different components of their models that is very similar to that used in Object Oriented Programming (OOP). For example, in the case of the Temporal Model, it is derived from eight primitive objects that fall into three functional categories: pacemakers, conduction pathways, and heart muscle. This is a concept that can be interpreted as a kind of inheritance, something that is very well contemplated in UML. They also mention that components are organized into classes, i.e. the sino-atrial node and the atrial-ventricular node are of class *pacemaker*; the right and left bundle branch belong to the *conduction pathway* class, and so forth. They call the instantiation of a primitive component an “object” which is compatible with the same concept in UML. Finally, this model considers that the object has an associated set of parameters and functionality defined in terms of a set of fundamental processes, something that resemble the methods that UML classes can posses.

If the previous two models can be equated to UML static structures where components and architecture are fundamentally represented by classes and associations; the concepts laid out in the Ticker Models can be found in the state charts and in the activity diagrams of UML. The models of Kirby *et al.*, developed a formalism for describing the electrical state and derived a set of rules which govern transitions between these states. Then constraints were used to limit the number of states explored. State transitions are defined by rules like:

```
if <region> is in <state1>
then <region> can make a <state1> → <state2> transition
```

This kind of rules can be very well modeled with state charts. Also in [3], the inclusion of temporal constraints is done by using relations of the type:

```
A < B - A must make a transition before B
A > B - A must make a transition after B
A u B - either may make a transition first
```

which are suitable for modeling with UML activity diagrams.

The previous examples briefly tried to demonstrate how the different techniques utilized in the different models can find their corresponding analogy in UML. Moreover, given that UML provides several kinds of models, the modeler can choose which method best suits his/her needs. In the conclusion of [4], it was said that the models do not have to be seen as being in competition, but rather as different complementary levels of possible representations. With the utilization of UML, the characteristics and the benefits of each one of the models described in [4] can all be represented under one unified standard, given an overall solution that combine the advantages of otherwise separated and disjoint techniques. The project is organized as follows: The second part describes the UML model. Section 2.1 and 2.2 briefly describe the anatomy and the electrical conduction system of a human heart. In section 2.3 it is showed an UML class diagram derived from the analogies previously mentioned. Finally, the conclusion summarized the main findings of the work.

2 The UML model

Based on the analogies that have been found between the models reviewed in [4] and the UML standard, this section try to show how the former can be implemented using the latter. In order to better understand the model, the section starts with a description of the heart anatomy and electrophysiology.

2.1 Anatomy of the Human Heart

Fig. 2 shows a picture of the anatomy of the human heart. The heart is located between the lungs in the middle of the chest, behind and slightly to the left of the breastbone (sternum). A double-layered membrane called the pericardium surrounds the heart like a sac. The outer layer of the pericardium surrounds the roots of the heart's major blood vessels and is attached by ligaments to the spinal column, diaphragm, and other parts of the body. The inner layer of the pericardium is attached to the heart muscle. A coating of fluid separates the two layers of membrane, letting the heart move as it beats, yet still be attached to the body.

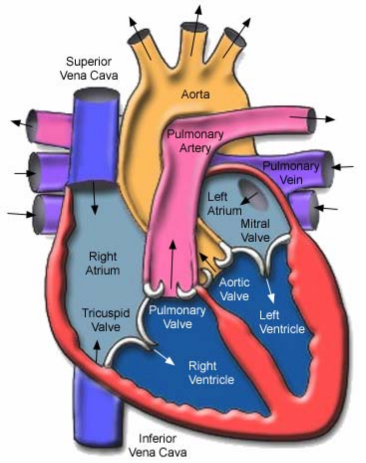

Fig. 2. Anatomy of the Human Heart: The heart weighs between 200 to 425 grams and is a little larger than the size of a fist. By the end of a long life, a person's heart may have beaten more than 3.5 billion times. In fact, each day, the average heart beats 100,000 times, pumping about 7,571 liters of blood. [5]

The heart has 4 chambers. The upper chambers are called the left and right atria, and the lower chambers are called the left and right ventricles. A wall of muscle called the septum separates the left and right atria and the left and right ventricles. The left ventricle is the largest and strongest chamber in the heart. The left ventricle's chamber walls are only about one centimeter thick, but they have enough force to push blood through the aortic valve and into the body.

Four types of valves regulate blood flow through the heart: The tricuspid valve regulates blood flow between the right atrium and right ventricle. The pulmonary valve controls blood flow from the right ventricle into the pulmonary arteries, which carry blood to the lungs to pick up oxygen. The mitral valve lets oxygen-rich blood from the lungs pass from the left atrium into the left ventricle. The aortic valve opens the way for oxygen-rich blood to pass from the left ventricle into the aorta, the body's largest artery, where it is delivered to the rest of the body.

2.2 Electrical conduction system of the heart

Fig. 3 shows a simplified picture of the electrical conduction system of a human heart. The normal electrical conduction in the heart allows the impulse that is generated by the *SA node* of the heart to be propagated to (and stimulate) the myocardium (muscle of the heart). When the myocardium is stimulated, it contracts. It is the ordered stimulation of the myocardium that allows efficient contraction of the heart, thereby allowing blood to be pumped to the body. Under normal conditions, electrical activity is spontaneously generated by the SA node. This electrical impulse is propagated throughout the *right and left atria*, stimulating the myocardium of the atria to contract.

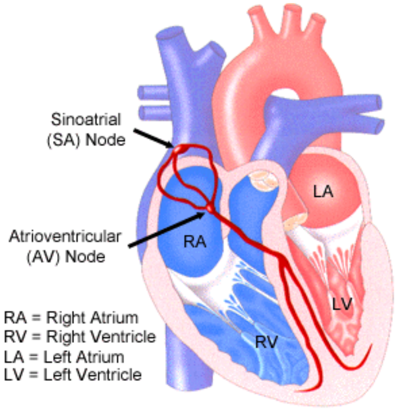

Fig. 3. The Conduction System. Electrical impulses from the heart muscle (the myocardium) cause the heart to beat (contract). This electrical signal begins in the sinoatrial (SA) node, located at the top of the right atrium. The SA node is sometimes called the heart's "natural pacemaker." When an electrical impulse is released from this natural pacemaker, it causes the atria to contract. The signal then passes through the atrioventricular (AV) node. The AV node checks the signal and sends it through the muscle fibers of the ventricles, causing them to contract. The SA node sends electrical impulses at a certain rate, but the heart rate may still change depending on physical demands, stress, or hormonal factors. [5]

Electrical activity that originates from the SA node at a rate of between 50 beats/minute and 100 beats/minute is known as **normal sinus rhythm**. As the electrical activity is spreading throughout the atria, it travels via specialized pathways from the SA node to the AV node, known as *internodal tracts*. The *AV node* functions as a critical delay in the conduction system. Without this delay, the atria and ventricles will contract at the same time, and blood won't flow effectively from the atria to the ventricles.

The distal portion of the AV node is known as the Bundle of His. The Bundle of His splits into two branches in the interventricular septum, the *left bundle branch* and the *right bundle branch*. The left bundle branch activates the *left ventricle*, while the right bundle branch activates the *right ventricle*.

2.3 Static structure and object model

This section describes an example of how the structure described previously can be putted together using a class diagram in UML. The idea behind the model is to show how the UML standard can be successfully used to model the different components of the human heart and their structure. This example, is however not comprehensive, since the detailed modeling of the cardiovascular electrophysiology is much more complex of what has been described here and clearly out of the scope of this project.

Form the previous descriptions, together with the analogies found in the introduction one can identify the following basic components in the conduction system of the heart:

- Sinoatrial node (SA-node).
- Right Atrium.
- Left Atrium.
- Internodal Tract.
- Atrioventricular (AV-node).
- Left Bundle Branch
- Right Bundle Branch
- Left Ventricle
- Right Ventricle

The components listed above can be thought as classes in UML. For example, regarding the SA-node, its main parameter is the `rate` or the sinus rhythm. In UML terminology this can be thought as an attribute of the class SA-node. Ref. [2] proposed a similar object with many more parameters (`thresholdPotential, restPotential, actionPotencial`, etc.) They are not clearly explained hence they are not included here to keep with the simplicity and clearness of the example, since the objective of this model is to remark how UML can be used to model, rather that exactingness of the model per se.

It was said that the SA-node is a pacemaker that naturally beats at a given rate, which for a healthy heart is the normal sinus rhythm. Therefore the `SA-node` class must incorporate a method that resembles this characteristic similar to the processes of [2] and the procedures of [1]. A `sendImpulse()` method of the class SA-node can satisfy this condition. Therefore, when the SA-node is instantiated, the object will use the method `sendImpulse()` to generate impulses at a frequency `rate`.

In UML associations represent relationships between instances of classes (a person works for a company; a company has a number of offices) [6]. Associations would play an important role in reassembling the architecture analogized with the anatomy of the heart and the navigability of the eletrophysiological signals.

Following the description of the conduction system of the heart, in UML the `SA-node` class must be associated to the `InternodalTract` class. This association has also a cardinality or multiplicity that indicates that there can be only one SA-node but many internodal tracts that transport the signal to the atria. The UML says that one can treat associations without arrows to mean either, the navigability is unknown or the association is bidirectional. In this case the navigability is perfectly known since the signals travel from the SA-node through the internodal tracts and not otherwise, therefore a navigability arrow is added to show that fact. The internodal tracts are conductions pathways that can conduct normally, be blocked or partially blocked in various ways (e.g. it may just cause a delay of an impulse, or it can suppress every second or third impulse, etc.). This fact is reflected through the utilization of an attribute called `status` that can be set and queried with the methods `SetStatus()` and `GetStatus()` respectively. By manipulating the status of the internodal tracts one can simulate disordes in the conduction pathways of the heart, for example some types of arrhythmias.

The internodal tracts are then associated to the atria, which is divided in two classes the `RightAtrium` and the `LeftAtrium`. In [3] it is mentioned that the left/right division is not very important when considering the electrical processes of the heart, however as it will be shown later there are some pathologies that can affect one side and not the other, such as blocks in the ventricular bundle branches; therefore here the division is implemented and hence a greater level of control and detail was gained. The contraction function of the atria is represented by the `contract()` method. In a normal heartbeat, as already noted, depolarization (i.e. contraction) begins in the sinus node, which contains cells that are particularly quick to depolarize spontaneously. Depolarization then spreads through the atrial myocardium to the AV-node. The attributes `polarizationState` and `conductionSpeed` are meant to keep track of this behavior.

The depolarization wave then travels through the AV-node indicating so with navigational associations. The AV-node slower the conduction, this delay ensures (when the heart is working correctly) that the contraction of the atria comes before the main ventricular pump, that is, it ensures that the ventricles are full of blood when they begin to contract. An either too short or too long delay in the AV-node can reduce the efficiency of the heart. To model this part it is created an `AV-Node` class with a `delay` attribute and `CheckSignal()` and `SendSignal()` methods.

Reaching the bundle of His, the depolarization wave speeds up again, and splits into two, exiting the right and the left ventricles throughout the left and right bundle branches respectively. In order to model this situation four more classes are created: the `LeftBundleBranch` class, the `RightBundleBranch` class, the `LeftVentricle` class, and the `RightVentricle` class. The bundle branches have a `status` attribute to represent the fact that being conduction pathways their conduction can be blocked as described previously in the text. The `Activate()` and `Deactivate()` methods are used to block totally or partially part of the bundles in order to simulate different kind of conduction disorders. Finally the right and ventricles will be exited by the signal transported by their respective bundles causing them to `Contract()`. Among all the possible attributes that can parameterize the function of the ventricle the ejection fraction is probably the most important because it gives the caudal of blood that is being pumped to the body. This parameter is represented as the attribute `ejectionFraction%` of the `LeftVentricle` and `RightVentricle` class.

All the behavior, the classes, attributes, methods and structure described before is represented in the UML class diagram showed in Fig. 4. UML permitted that almost all the main elements that previously appeared in separated models appear together in only one representation. To analyze the goodness of the model of Fig. 4, it is useful to pick up some typical cardiac pathologies and see how they are reflected on it.

Arrhythmias are changes in the sequence of electrical propagation in the heart where their typical causes are ectopic excitations (excitations that originate on some location other than the SA-node) and conduction disorders. As for the conduction disorders it has already been explained before how they can be simulated through the usage of the `status` attribute of the internodal tracts. The ectopic excitations due to other possible “SA-node like” pacemakers can be simple simulated increasing the multiplicity of the `SA-Node` class, thereby instantiating as many ectopic points as necessary to reproduce the pathology, where each ectopic excitation can pace at a pace given by the attribute `rate`.

2:1 Block is another well known pathology in which the AV node does conduct, but takes longer than normal to recover. The result is that after a previous contraction (i.e. depolarization), the next wave of depolarization arriving from the atria finds the AV-node still in refractory[1] state, and that atrial beat fails to be conducted to the ventricles. By the time the third beat arrives, the AV node has recovered and the beat is conducted normally. This behavior can be manipulated and reproduced through the utilization of the `delay` attribute of the `AV-Node` class.

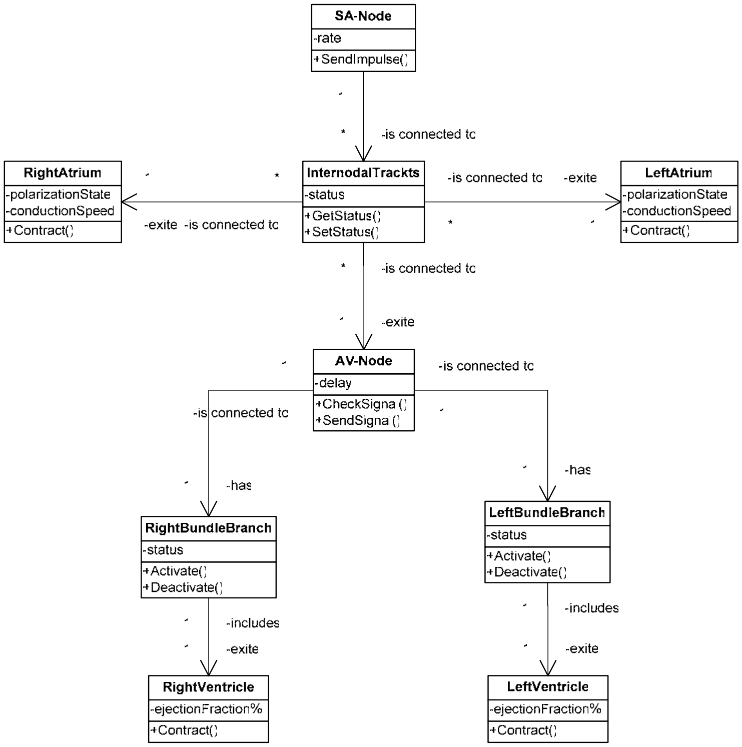

Fig. 4. UML class diagram of the electrical conduction system of a human heart. Using the power of UML structures this model blends all the characteristics of the models described in [4] in only one representation.

Ventricular ectopics can also be depicted in the diagram by just increasing the multiplicity on the association ends that link the ventricles to the bundle branches. Moreover, because of the attributes that will parameterize each of the instances is possible to know the ejection fraction percentage due to the occasional ectopic beats something that none of the models described in [4] previously did.

[1] Length of time necessary before a region can transmit another impulse.

The introduction in section 1 also mentioned inheritance as one of the properties that was perceived as a common characteristic in models reviewed in [4]. For the sake of clarity in the explanation, inheritance was not included in the model of Fig. 4, however Fig. 5. shows how it can be modeled.

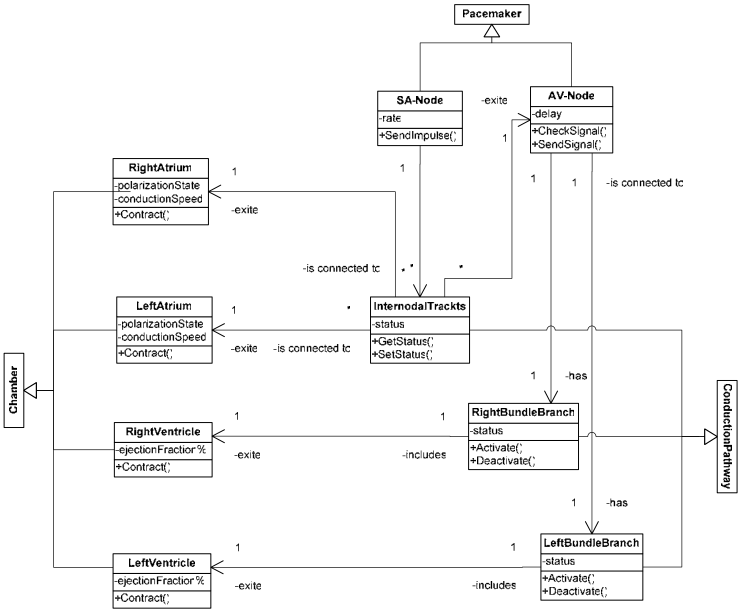

Fig. 5. UML class diagram of the electrical conduction system of a human heart showing generalizations. The model is derived from nine primitive classes that fall into three functional categories: pacemakers, conduction pathways and chambers.

Fig. 5 uses UML generalization relationships. The specializations are known as subclasses. The generalized class is known as the superclass. Generalization allows the inheritance of the attributes and operations of a superclass by its subclasses. The example of Fig. 5 shows one possible representation of the generalizations that will enable inheritance, closely following the classification given in [2], although not exactly because it includes more classes. The classes are a specialization of three general classes, named `Chamber`, `Pacemaker`, and `ConductionPathway`. But, since a model is at the end an abstraction of the reality, it can vary depending on the approach taken by the modeler which usually responds to certain objectives. For example, another way to generalize the classes of Fig. 4 is given in Fig. 6. Even if for clearness this model does not contain all the details showed in Fig. 4; it can be quickly noticed that such an interpretation provides a finer grain of generalization/specialization compared to the previous one. Such a model will enable the modeler to obtain a greater control over the characteristics of the components of the system. Generalization relationship structures are sometimes referred to as hierarchies, reflecting their tree-like structure. There is no reason why a generalization hierarchy cannot have several layers. In such a case a subclass could also be a superclass, something that is depicted in Fig. 6 where, for example, the `ConductionPathway` class, is a subclass of type `ExitableTissue` that however constitutes a superclass for the `BundleBranch` class. As Fig. 6 demonstrates, the subclass to superclass relationship could extend over several generalization relationships.

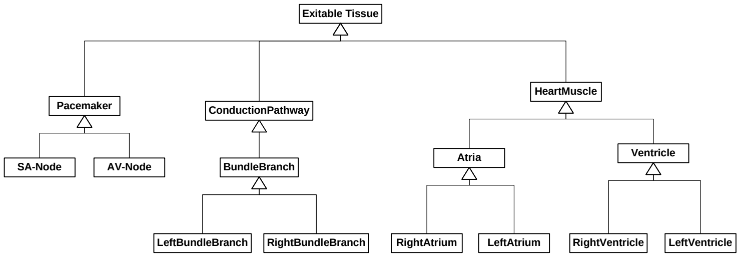

Fig. 6. UML class diagram of the electrical conduction system of a human heart showing generalizations/specifications. The finer grained generalization exhibited in this approach permits a greater control of the functional components of the model.

3 Conclusion

The object oriented programming community has been proving for more than a decade the power of UML in software engineering. However, despite the many information dissemination tools available in research and interdisciplinary exchanges between researchers of different communities, it is well known that even today it happens that the research done in one community that can have an application in another is not always known; and most of the time when it happen to be known, it occurs by chance. This work showed that UML tools which are widely used in the software engineering community can be also successfully applied to model medical and/or physiological behavior.

This project was started with a review of best known conceptual models of the human heart. As they were reviewed, it was detected that there were noticeable similarities between the physiological models of the electrical conduction system of the heart and the UML paradigm. Concepts like primitive objects sounded similar to classes; component-process sounded similar to the methods; preconditions sounded similar to attributes; logical connections similar to associations; etc. After that, one question arose: Is the heart object oriented? the answer is that may be not, but even thought it is not OO, still OO modeling techniques could be used to provide a global view of its physiology unifying those previously developed using several unconnected models like those reviewed in [4]. The examples showed throughout this project showed that it is possible and could be even desirable to use UML to successfully model some behaviors of the human heart. It is the hope of the author that this interdisciplinary work that tried to bring together the knowledge of the medical community with that of software engineering could serve as an inspiration to others in the former community to use the tools developed for the latter.

References

1. I. Bratko, I. Mozetic, and N. Lavrac: KARDIO: A Study in Deep and Qualitative Knowledge for Expert Systems, MIT Press, 1989.
2. J. B. Weinberg, G. Biswas, S. Uckun: Continuing adventures in qualitative modeling: a qualitative heart model. Proceedings of the third international conference on Industrial and engineering applications of artificial intelligence and expert systems, ACM, vol. 1, June 1990.
3. Kirby IK, JRW Hunter: Further Progress in Qualitative Modeling of Cardiac Electrophysiology. Working Papers for QR-91: Fifth International Workshop on Qualitative Reasoning about Physical Systems, pp. 245-262 (1991).
4. Bilich, C.: Qualitative Models of the Human Hear: A Survey. Survey paper for the Conceptual Modeling course, Department of Information Technology, University of Trento, Italy, May 2005.
5. Texas Heart Institute: The Heart of Discovery. Available on-line: http://www.tmc.edu/thi/anatomy2.html. Last accessed Nov. 4, 2005.
6. Fowler, M.; Scott, K.: UML Distilled Second Edition. A brief guide to the standard Object Modeling Language. Addison-Wesley, NJ, U.S.A, July 2000.
7. Bennett, S.; Skelton, J.; Lunn, K.: UML. McGraw-Hill, Jan. 2001.