Journal of Neuroscience Methods 193 (2010) 307–320

Contents lists available at ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

Modeling the vagus nerve system with the Unified Modeling Language

Bert-Jan F. van Beijnum*, Ing A. Widya, Enrico Marani

Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, Netherlands

ARTICLE INFO

Article history:
Received 1 February 2010
Received in revised form 11 August 2010
Accepted 20 August 2010

Keywords:
Vagus nerve
Neuron models
Cardiac pathways
Baroreceptor reflex
Unified modeling language
Meta-modeling

ABSTRACT

Traditionally, the means of describing anatomical and physiological structures of the autonomic nervous system is natural language, drawings and images as represented in the scientific literature. In behavioral studies of this system, mathematical and electrical models and computer simulation tools are in use. In this article, we propose the use of the Unified Modeling Language to describe and specify the anatomical and physiological structures and indicate how these can be enriched to capture the behavioral view as well. Using the metamodel facilities of the language, we propose a domain specific language that captures the domain concepts, their relationships and constraints. Application of the language is demonstrated by modeling the vagus nerve in part.

1. Introduction

For centuries the anatomy of man has been studied, first at a macroscopic level and later, following technical advances, enriched with microscopic and ultrastructural details. Considerable attention has been given to the nervous system of man, particularly the autonomic nervous system. An enormous volume of knowledge has been accumulated and made accessible mainly by traditional scientific methods. Studying the (autonomic) nervous system has predominantly been the field of expertise of the clinical field, neuroanatomy and neurophysiology. However, lasting recent decades, a new field has emerged, namely neural engineering. Here profound knowledge of the nervous system is to be acquired by the engineering scientist and used to analyze, design and test new ways of neurostimulation (for an overview see Prodanov et al., 2003), e.g., vagus nerve stimulation (Matheny and Shaar, 1997; Rozman and Bunc, 2004; Groves and Brown, 2005). The need for a framework to unify experimental and theoretical results in neuroscience has been advocated by, for instance, Eliasmith and Anderson (2003) in which a computational framework for modeling neural behavior and dynamics is presented. Two research efforts in this direction are neuroScholar (Burns, 2001), and the Neural Open Mark-up Language (NeuroML, Goddard et al., 2001). The neuroScholar system is a "computational knowledge management system with the objective of providing non-computational neuroscientist with a method to manipulate the published literature". The scope of the system is "to fully delineate the neural circuitry involved in a particular behavior" Burns (2001). The NeuroML is an open standard that consists of a set of XML based specifications for describing neural system models (neurons and neuronal networks). The purpose of NeuroML is to define a common format of these models so that these can be communicated between neuroscientific application components, such as databases, simulators and visualization applications. Simulators that use this open standard are NEURON (Carnevale and Hines, 2006), GENESIS (Bower and Beeman, 1998), NeuroSim (Revest, 1995) and JSim (JSim, 2009). Hence, NeuroML is not a neuroscientific application or application component itself. Also, because NeuroML is based on XML, it is intended for computer systems and not for communication between neuroscientists.

In this article, we propose a new framework for the modeling of the autonomic nervous system, using the Unified Modeling Language (UML). The UML is an object oriented language intended to be used to model complex systems. It supports a graphical notation, but has a firm semantic underpinning. The UML is not a programming language, but resides at a higher conceptual level. UML modeling environments usually give support for software development and for automatic and guided translations to programming languages such as Java, C or C++. The UML (version 2.0) supports thirteen diagram styles, roughly clustered in three classifications: structural diagrams (including class diagrams), behavioral diagrams and interaction diagrams. There are three main reasons for adopting UML instead of XML:

- With UML we can define neuroscientific languages (metamodels). That is, generic neuroscientific rules and domain knowledge can

* Corresponding author. Tel.: +31 53 4892834; fax: +31 53 4892287.
E-mail address: beijnum@ewi.utwente.nl (B.-J.F. van Beijnum).

0165-0270/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.jneumeth.2010.08.015

be captured and enforce application of these rules and knowledge to neuroscientific models. This is not possible in XML.
- With UML conceptual neuroscientific models are created as apposed to the logical and data models in XML based approaches (Wand and Weber, 2002; Parsons and Cole, 2005).
- XML schemata are easily generated from UML using model transformations (Routledge et al., 2002; Bruhn, 2006).
- In software development environments there is a broader support for application component design using UML than there is using XML. This support constitutes model transformations, such transformation exist for UML to generate databases (web-based) user interfaces (including model navigation) and skeleton code for application logic. A good example of such a software development environment is IBM's Rational Systems Developer (RSD, 2008).

Finally, UML models are graphical and better suited as means of communication between neuroscientists. Compared to other conceptual modeling languages, such as Entity-Relationship Diagrams, the UML is the better choice because it is semantically richer.

A disadvantage is however, that yet another language is to be learned by those that are to use it. However, although a deep understanding of the semantics of the UML may cause some problems, the language is quite intuitive and relatively easy to learn. Another point of concern is that the UML supports many different types of models, however as will be demonstrated the class diagram is a type of UML model with which the most essential conceptualizations can already be captured.

In this article, the focus is entirely on conceptual modeling. The objectives are to extend the UML to a language for the modeling of the autonomic nervous system in man and to demonstrate its applicability. The scope of the language extension is twofold as it specifies the concepts, relationships and constraints of:

- the macroscopic anatomical view: in this view the autonomic nervous system comprises the nerves that connect the various nuclei in the brainstem and the organs such as smooth muscles, glands and viscera, and
- the microscopic physiological view: in this view neurons structures and their synaptic interconnections are modeled, with suggestions made regarding the physiological aspects.

In addition, the relationship between these two views is specified, thus allowing the construction of integrated neuroanatomic-physiological models. The application of this approach is demonstrated by modeling (parts of) the vagus nerve.

Models built in this way can be checked for conformity with the metamodel automatically when a proper UML development tool is used, e.g., the IBM Rational Systems Developer (RSD, 2008).

So, in this article we do not consider the design of neuroscientific application components, such as databases (web-based) model construction interface design, or simulator design. Nor do we discuss how existing application components can be integrated. However, with reference to IBM's Rational Systems Developer, it is evident that the languages proposed in this article can be realized as a UML profile and thus made available and used in modeling. In addition, integrated software design environments also support UML based application component design through model transformations. Today's state of the art in software engineering makes it possible to generate databases (using, e.g., the Java Persistence API), web-based interfaces (e.g., using Java Server Faces) and application logic skeletons (Session Beans) based on UML models.

In neuroscientific collaborations, developed models and also their corresponding metamodels, for example as proposed in this article, may need to be exchanged between the collaborating sites. For this purpose the OMG has specified the XML Metadata Interchange (OMG XMI) standard. Using XMI, UML profiles and UML models can be serialized to sequences of characters or bit strings such that they can be transferred over network connections and reconstructed in a meaning preserving way at the receiving sites. When those models have been implemented in a particular neuroscientific database, the export and import facilities of the database may also be used to exchange those models. Remote export and import facilities typically use a serialization technique similar to the one mentioned previously, for example on Java platforms using Java objects and Remote Method Invocations (RMI), JavaScript Object Notation (JSON) or on SQL oriented databases using the available SQL – XML mappings.

With a UML metamodel for the autonomic nervous system, the domain knowledge is made explicit and is therefore easier to share. In addition to the traditional approach, a modeling language aids in more precise and accurate models. A common language promotes communication among experts in the field, but also among different disciplines. In this way, it can facilitate multidisciplinary research. The models developed may serve as schemata for the development of a database that captures the knowledge developed in neuroscience and can be maintained and shared among researchers in the field. With a metamodel and models proposed in this article a basis is defined that can be further extended as required, for instance to support morphometric analysis studies, or neuronal network simulation studies. Models developed using a common metamodel promotes and enables the exchange between existing analysis and simulation tools and using a metamodel to check models for correctness is easily automated and thus assists in the modeling process. Because integrative neuroanatomic-physiological models can be built, automated methods to analyze the potential effects and side-effects of nerve stimulation (as function of the location of stimulation) become possible.

In this article, we propose UML metamodels that capture concepts in neuroanatomy and neurophysiology at nerve level and neuronal level. This way, we are able to construct models that are correct by construction, that is only syntactically and semantically meaningful models can be built from this language. Moreover, these models are conceptual and thus technology independent (unlike, e.g., NeuroML). Technology specific models are easily generated using existing transformation tools. The models discussed consider parts of the autonomic nervous system, as apposed to models considered by Goddard et al. (2001) and Burns (2001) who focus on the central nervous system.

2. The Unified Modeling Language

We apply the Unified Modeling Language (UML) (Booch et al., 1999) to model the autonomic nervous system. The visual representation of the UML notation makes the models more accessible for both neuroscientists and ICT engineers. The modeling elements are sufficiently clear to enable both groups to work with the specified models and to provide modeling input in their domains of expertise. The basic UML modeling constructs are illustrated in Fig. 1 and explained in the following.

A class is a set of objects that has features in common, such as attributes, operations and relationships. Fig. 1(a) shows how a UML class is represented; in this case the class is named "A" and all members of the set represented by it have the attribute "name" and "description" and have the operations "getName", "setDescription" and "create". An association defines a relationship between classes; hence it defines a subset of the Cartesian product of these classes. A UML association can relate two or more classes. In Fig. 1(b), a binary association is shown. An association has a name ("BinaryAssociationAB" in this case), at the association ends (i.e., near the classes

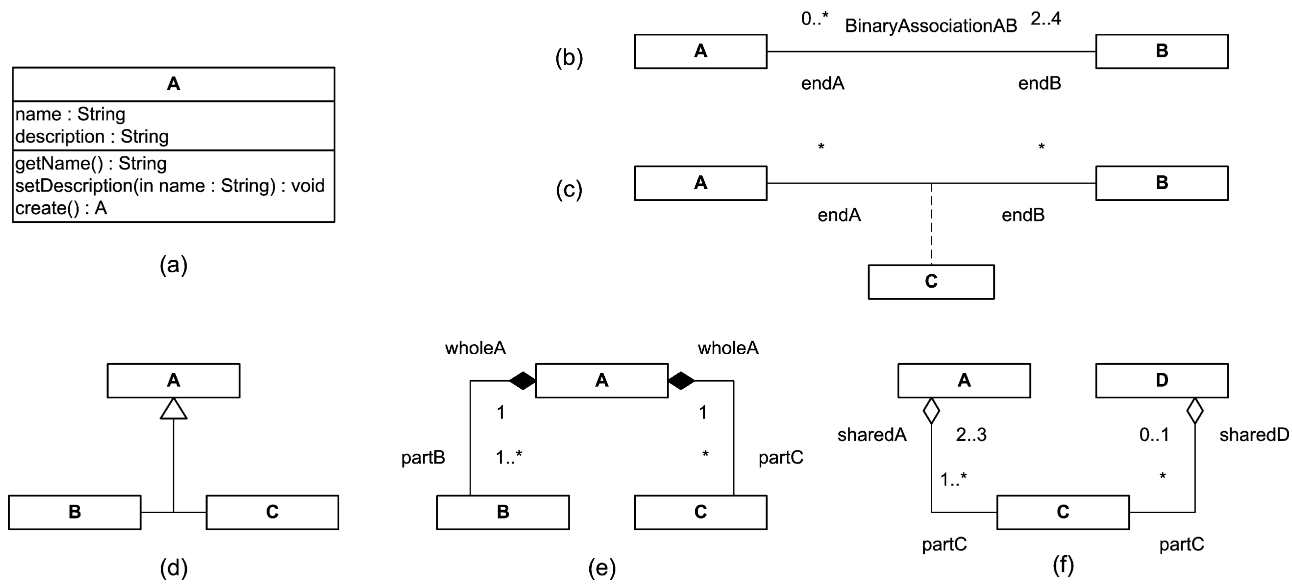

Fig. 1. Basic UML constructs: (a) class; (b) binary association; (c) association class; (d) generalization; (e) composite aggregation; (f) (shared) aggregation.

associated) a name may be specified so as to describe the role of the class in the association (hence "endA" and "endB" in this case respectively). Finally, at each association end, the allowed multiplicities are specified ("0..*" and "2..4" respectively). This means that for each object in set "B" zero or more objects (denoted as "0..*") in set "A" may be linked. Similarly, for each object in set "A", two, three or four objects (denoted as "2..4") from set "B" must be linked.

An association class has the features of an association and a class, its graphical notation is given in Fig. 1(c). Using an association class, properties and methods can be defined for the association.

In Fig. 1(d) generalization is illustrated (the line with the open arrow head). The class "A" is (relative to the generalization) the general class, and the classes "B" and "C" are (relative to the generalization) the specific classes. A generalization is a taxonomic relationship, meaning that instances of a specific class (i.e., an object) are also an instance of the general class. The specific class ("B", "C" respectively) also has the features defined for the general class (this is often called inheritance).

Fig. 1(e) and (f) shows examples of special kinds of associations; these are called composite aggregation and (shared) aggregation respectively. In the composite aggregation example, class "A" is the whole and the classes "B" and "C" are the parts. At the composite aggregation ends, names may be included to describe the roles of the classes in the association (e.g., "wholeA" and "partB" in Fig. 1(e)). The composite aggregation is to be understood in the following terms: the parts cannot exist without the whole; furthermore the parts can only be part of one whole. Hence, the existence of a part is bound to the existence of the whole. The (shared) aggregation is a weaker form, where parts may exist independent of the whole; a part may also be part of more than one whole. For example, an instance of class "C" is a shared part of two or three instances of class "A" and it may be a shared part of at most one instance of class "D". At the (shared) aggregation ends, names describing the roles of the classes in the association may be included (e.g., "sharedA" and "partC" in Fig. 1(f)).

In the above review, the basic language constructs to design UML models are discussed by example UML models. However, the question arises as to how these language constructs are defined. The specification of the UML language is at the metamodel level. The full specification of the UML version 2 can be found in OMG UML Superstructures (2007) and OMG UML Infrastructure (2007). As part of the UML language specification several language extension mechanisms are included, this allows new modeling concepts to be introduced that are specific for the problem or application to which the UML is applied. One extension mechanism consists of specializing existing language concepts and extending these with additional constraints and relationships.

The UML metamodel is the specification of the UML language, a UML model is a model expressed in this language. The UML metamodel is important for this article because the approach followed is to define a dedicated language for the modeling of nerves (and nerve networks) and neurons (and neuronal networks). However, the formal specification of these extensions is beyond the scope of this article. Instead, we will turn to an intuitive presentation. All concepts illustrated in Fig. 1, and in fact many more, are defined in terms of metaclasses and their relationships and constraints. Thus, there are the metaclasses "Class", "Association", "AssociationClass", "Generalization" and so on. We may extend the UML language by specifying new concepts (i.e., new metaclasses) as specializations of existing UML metaclasses. An example of defining new concepts from the metaclass "Class" is given in Fig. 2(a), it defines the concepts "MyConceptA" and "MyConceptB". Such language extensions are useful because they allow the specification of additional or new constraints that do not hold for the generalized concepts. As an example, when specifying the metaclass Neuron as a specialization of "Class", we can establish the constraint that it must contain several neuron parts, one of which must be an input part (via which inhibition or excitation can occur) and one of which must be an output part (which can cause inhibition or excitation). Input parts and output parts form junctions or synapses; this is yet another concept specific for the problem domain. Junction will be specified as a specialization of the metaclass "AssociationClass". In the nerve and nerve network language that is specified in the next section, a similar approach is followed. Specializations of the metaclass "Association" may be specified too. As shown in Fig. 1, an association (possibly with some form of aggregation) involves specific notations and modeling options (such as the multiplicities and role names at the ends). In order to overcome complexities of the UML metamodel, we introduce a simple notational convention to define new types of association. The new associations that will be defined in the next section are all binary associations; hence, they constrain the number of ends of an association to two. A new binary association is defined by specializing the "BinaryAssocia-

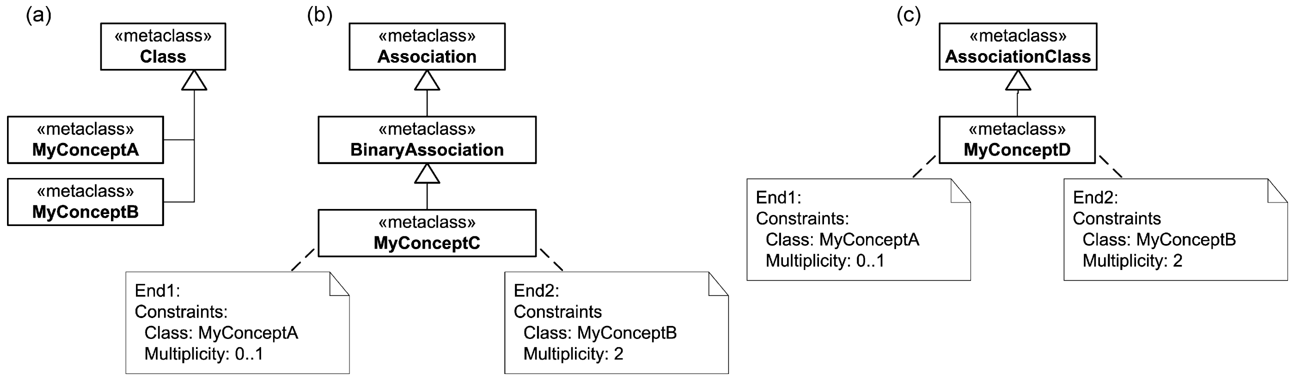

Fig. 2. Defining new language constructs by specializing: (a) Class; (b) Association; and (c) AssociationClass.

tion" as shown in Fig. 2(b). The definition of a new association, such as "MyConceptC", may further constrain the ends of the association. The constraints may include the metaclasses to which the association can be applied, and the multiplicities at the ends of the association. Examples of such constraints are shown in Fig. 2(b). The same holds for specializations of association classes as shown in Fig. 2(c).

3. Modeling language for the autonomic nervous system

The autonomic nervous system is in many aspects a complex system. At a course grained level, the physical level, it is complex due to the nerve structures connecting the various end points such as the numerous nuclei of the brain stem and the viscera. It is complex also due to the huge number of neurons involved and the huge number of fibers carried by the nerves of the autonomic nervous system. For the vagus nerve alone 40,000 myelinated fibers have been estimated (Kobus, 2008, see also Pick, 1970), as well as many unmyelinated ones. Another dimension of complexity is found when considering the signals propagated and "processed" by individual neurons and neuronal networks as to exert autonomic control of the viscera. In the context of this article, we will often use the generic term *behavior* to refer to this aspect of neurons and neuronal networks.

Inspired by ICT system development and working practices (e.g., the OSI and ITU-T Reference Model of Open Distributed Processing: X.901-X.904, 1996), two distinct modeling languages have been developed: one is suited for the modeling of nerves and nerve networks, the other is suited for the modeling of neurons and neuronal networks. These two languages focus on different aspects of the autonomic nervous system (thus, providing different viewpoints of the same system). The modeling language for nerves and nerve networks enables us to express the nervous system in terms of nerves, nerve branches, plexuses, nuclei and the viscera to which these connect. On the other hand, the modeling language for neurons and neuronal networks takes the viewpoint of the neurons and neuronal junctions that are responsible for the visceral, efferent and afferent behavior of the autonomic nervous system. Consequently, models can be constructed for each of these viewpoints. The relationship between models in these different viewpoints and application of the languages proposed for the vagus nerve are presented and discussed below.

3.1. Nerves and nerve networks

In this section, a language to model the nerves of the autonomic nervous system in man is presented. This language must enable us to model precisely the anatomy of nerves in the autonomic nerve system. Devising such a modeling language is not trivial, as illustrated by the following. Consider the vagus nerve (either the left or right vagus nerve) which arises from nuclei in the medulla oblongata and "wanders" through the body eventually reaching the abdominal cavity. Along its course, the jugular and nodose ganglia are encountered. Furthermore, the number of axons varies along the path of the vagus nerve as nerves branch from the vagus to innervate various viscera. In conclusion, it must be possible to model the vagus nerve at macroscopic level as a network of nerve parts so that ganglia that are part of a nerve and the path followed by axons can be modeled explicitly.

For this reason a language is proposed that allows nerves to be modeled in a way that is independent of the anatomical naming conventions, although the relationship with these conventions is easily included. Furthermore, the language is based on the intrinsic characteristics of the nerves themselves and uses a more fine grained set of concepts that is rich enough to model the relationships with the neuron and neuronal network level (see next section), and to model the relation with the traditional anatomical naming conventions. The language proposed is also relevant for clinical applications and neurostimulation research. For instance, when considering course grained neural stimulation, such as vagus nerve stimulation, the location of stimulation can be precisely pinpointed. In addition, in case a sufficiently detailed model of the vagus nerve is available, the effects and potential side-effects can be analyzed and understood by both clinical and non-clinical experts.

The basic idea for modeling nerves and nerve networks is to consider these as a system of tube-like components that are used as a carrier for neurons and their parts (e.g., axons). Hence, a nerve network connects functional bodily parts (such as nuclei of the brain stem and viscera). The language proposed considers nerve networks in a constructive way being composed of various types of tube-like components. The language defines the various nerve network component types and the rules for modeling a nerve network.

Note that in the following, we are concerned with a metamodel (i.e., a language) with which models of nerves and nerve networks can be constructed. The different types of components needed in the modeling of a nerve or nerve network include the following: segment, bifurcation, plexus, ganglion, nucleus, organ, viscus and muscle. These component types (shown in Fig. 3) and their relationship (shown in Fig. 4) are defined as follows:

- Segment («segment»): a *segment* is a tube-like part of a nerve network that can carry axon fibers. A segment must have two connections to different *segment endpoints*. Relative to the segment, one segment endpoint may be designated a direction, such as cranial (i.e., superior) or caudal (i.e., inferior), lateral or medial.
- Segment Endpoint («endpoint»): a *segment endpoint* is an abstract metaclass used to specify entities that may be connected to a seg-

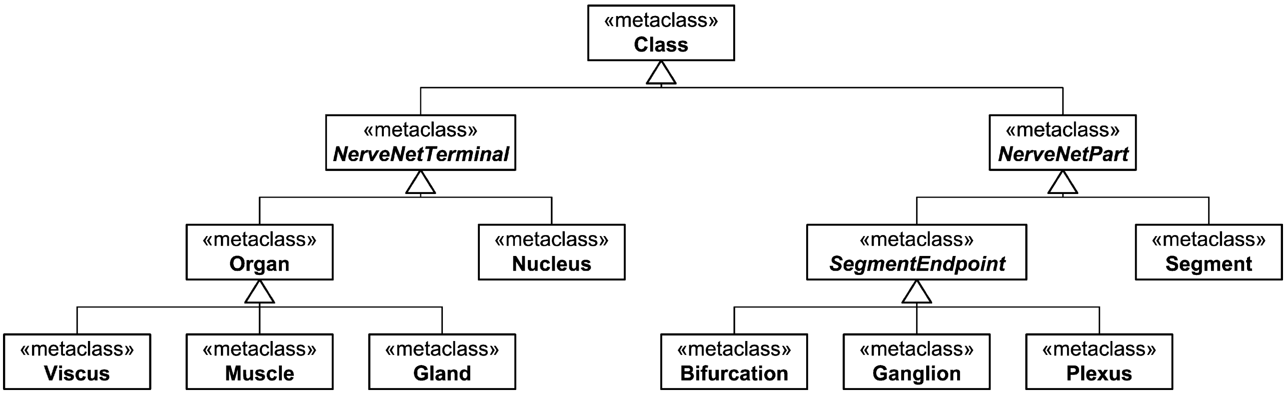

Fig. 3. The metaclass hierarchy for modeling nerve networks.

ment and are part of a nerve network. Concrete sub-metaclasses are Bifurcation, Plexus and Ganglion.

- Bifurcation («bifurcation»): a *bifurcation* is a nerve network part where the network splits into two different directions, axons carried may go either direction. A Bifurcation is connected to exactly three different segments. One of these segments carries the full bundle of axons and this bundle bifurcates, i.e., splits, towards the other two segments. It must be noted that a single axon fiber may split in collaterals and then bifurcate at other segments.
- Plexus («plexus»): a *plexus* is a part of a nerve network with multiple segments connected to it. For example, two could be designated as incoming, and two as outgoing. A plexus provides a means for axons to be routed flexibly. Regarding the naming of the nerve segments connected to the plexus, these are usually all given different anatomical names. In some cases, a plexus is a composite; as an example, the cardiac plexus consists of the superficial part and the deep part. In other cases, a plexus may contain a ganglion, for example the pelvic plexus. For reasons of brevity, this detailing of *plexus* will not be considered further in this article.
- Ganglion («ganglion»): a *ganglion* is part of a nerve network with one or more incoming segments and one or more outgoing segments. A ganglion hosts somas, and it may host synaptic junctions. Different subtypes may be distinguished: ganglion that hosts the soma's of pseudounipolar neurons and that carries axons (a dorsal root ganglion); and a (visceral) ganglion where axon fibers terminate (i.e., it hosts axon terminals) and synapse with dendrites of neurons whose somata are hosted by the ganglion. In the following, for brevity we will not explicitly model different subtypes.
- Nerve network terminal («terminal"): a *nerve network terminal* is a metaclass representing an endpoint of the nerve network, it is a point (one out of possibly many) from which the nerve network arises or terminates. Concrete nerve network terminals are Nucleus, Muscle and Viscus.
- Nucleus («nucleus»): a *nucleus* is an anatomically delineated structure in the central nervous system, usually functioning as a hub. In the context of this article, we deal with several nuclei in the brainstem as these are afferent or efferent terminal points of the vagus nerve. A nucleus may be the terminal point of one or more segments.
- Viscus («viscus»): a *viscus* is an organ in the body to which segments are connected. A viscus may be the terminal point of one or more segments.
- Gland («gland»): a gland synthesizes a substance, e.g., a hormone, for release.
- Muscle («muscle»): a muscle is a body tissue that can contract; in the context of the autonomic nervous system only smooth muscles are controlled. Multiple segments may be connected to a muscle.
- Connection («connection»): a *connection* specifies the relations between NerveNetParts and NerveNetTerminals to build nerve networks. A connection is a binary association. A connection always connects a segment and segment end point or nerve network terminal. Furthermore, the metamodel specifies (though not shown in the figures) constraints on the number of connections for each nerve network part and nerve network terminal. These are: a nerve network terminal may have arbitrary number of connections (hence multiplicity *); a bifurcation has exactly three connections (hence multiplicity 3); a ganglion has at least 2 connections (hence multiplicity 2..*); a plexus has at least four connections (hence multiplicity 4..*) and a segment has exactly two connections (hence multiplicity 2).
- Nerve («nerve»): a *nerve* is a subnet of a nerve network consisting of segments, bifurcations and ganglia and to which an anatomical name is given. A nerve comprises at least one segment, and may comprise multiple bifurcations and ganglia (cranial or dorsal root ganglia). Furthermore, any segment and any bifurcation must be part of a nerve (assuming that anatomical nomenclature takes care of giving names to all sub structures of a nerve network). A ganglion may be part of a nerve. This "part of" relationship is modeled using the standard UML composite aggregation.

3.2. Neurons and neuronal networks

The human autonomic nervous system contains many different types of neurons. The differences exist in terms of the structure of the neuron (efferent and afferent neurons), the way in which neu-

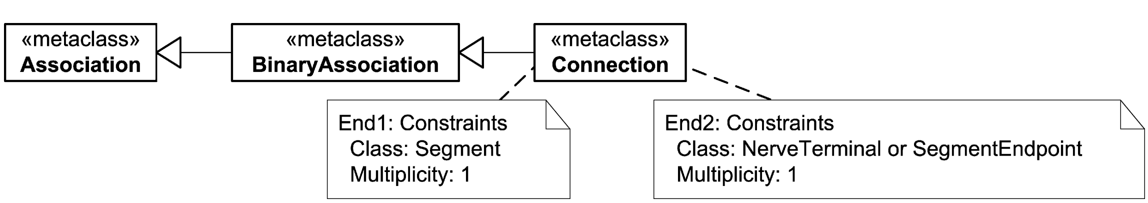

Fig. 4. Nerve network connections.

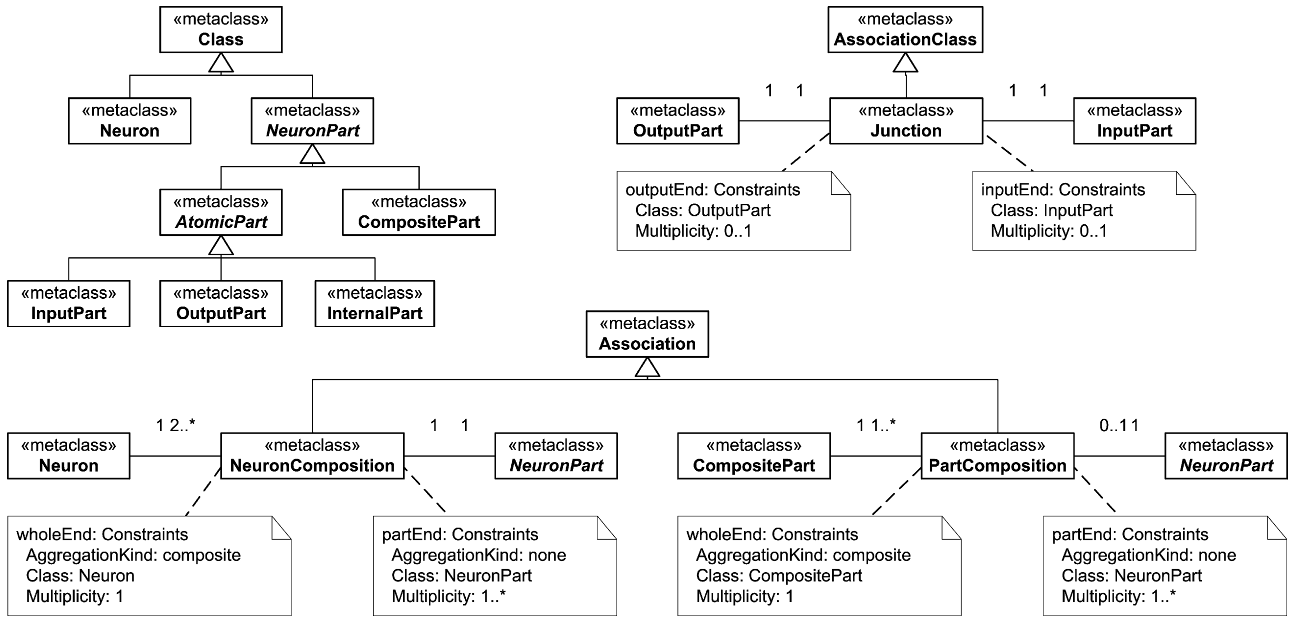

Fig. 5. The neuron metamodel.

rons interact (chemical synapse and electrical synapse), the context in which they perform their function (pre- and postganglionic neurons), and the role played by a neuron in the nervous system (sympathetic and parasympathetic neuron). The neuron metamodel defines the concepts and their relationships on the basis of which different neurons and neuronal networks can be modeled. In general, neurons have a specific structure. Specific parts of a neuron are involved in the interactions. Furthermore, some neuron parts function as neuron input and other parts function as outputs of the neuron. When modeling neurons or a neuronal network, it will depend on the purpose of these models what details are to be included. For instance, in more detailed studies it may be necessary to model the full dendritic tree structure (Heldoorn et al., 2003), while in other cases a dendrite is considered to be comprised of a collection of dendritic terminals functioning as neuron input (Heldoorn et al., 2001). The intention of the metamodel shown in Fig. 5 is to accommodate this modeling flexibility. In the following, the concepts and their relationship are explained further in terms of various neuron and neuronal network models that use these concepts as discussed in Section 4.

- Neuron («neuron»): Neuron is a metaclass that represents a neuron as a whole; a neuron may contain neuron parts, using the neuron composition association. Neurons have at least two parts for neuron interactions, which are named input part and output parts respectively. Neuron parts may themselves be composites of parts, modeled using the neuron part composition, and may be used to further refine the internal structure of a neuron part such as the dendritic tree structure. The Neuron concept we introduce here supports the neuron compartment modeling as reported by Rall (1962, 1964) and those encountered in, e.g., NEURON (Carnevale and Hines, 2006), GENESIS (Bower and Beeman, 1998). Although not considered in the sequel, membrane specification can be included as well.
- Neuron part («part»): a neuron part is a part of a neuron modeled using neuron composition; the neuron is the whole and the neuron part is the part. A neuron part may be a composite of neuron parts, modeled using the neuron part composition (specialized from composite aggregation) and may be used to further refine the internal structure of a neuron part such as the dendritic tree structure.
- Atomic part («atomic»): an atomic part is an indivisible neuron part. This means that the model under consideration does not decompose the neuron part.
- Composite part («composite»): a composite part is a neuron part that is a (aggregate) composition of neuron parts (hence it is not an atomic part).
- Input part («input»): an *input part* is a neuronal atomic part through which the neuron interacts with (parts of) other neurons (via a junction). In this interaction the neuron part takes on the role of input (hence the part that is triggered externally).
- Output part («output»): an *output part* is a neuronal atomic part through which the neuron interacts with (parts of) other neurons (via a junction). In this interaction the neuron part takes on the role of input (hence the part that triggers externally).
- Internal part («internal»): an internal part is a atomic part, it is a part used to build a neuron or neuron part and does not have the ability to interact with other neurons.
- Junction («junction»): a junction is a specialization of AssociationClass; it models the synaptic relation between input parts and output parts. A junction may have properties that play a role in characterizing general excitation or inhibition and modulation.
- Neuron composition («n»): A neuron may be composed of neuron parts. Neuron composition is a specialization of composite aggregation.
- Part composition («p»): A neuron part may be composed of other neuron parts. Part composition is a specialization of composite aggregation.

A neuron must have at least two neuron compositions; this is to enable neuron models to have interaction capabilities. Though not shown in Fig. 5, one neuron composition must have an input part, and one must have an output part. As shown in Fig. 5, a junction must have an input part and an output part.

The metamodel does not specify specific constructs for the modeling of relationships between neuron parts belonging to the same neuron (e.g., domain specific associations). Because a neuron part is a specialization of the metaclass Class, these relationships can simply be modeled using the standard concepts of the UML. Examples of neuron models are discussed in Section 4.2.

3.3. Language relationships

In the previous sections, a modeling language has been defined for the modeling of nerves and nerve networks, and for neurons and neuronal networks respectively. The nerve–nerve network language enables the modeling of the macroscopic anatomy of the autonomic nervous system, while the neuron–neuronal network language allows the detailed physiological modeling of the autonomic nervous system. The obvious issue now is to further detail the relationship between these two. With this relationship we are able to model neurons, or parts of a neuron that are spread out over a nerve network. This spreading out satisfies a number of specific rules captured by this relationship.

At least two different approaches can be thought of to define this relationship: a direct relationship and an indirect relationship. In a direct relationship, relations between the constructs of the two languages are defined (e.g., a hosting relationship defining that a segment can host axons). In an indirect relationship, intermediary concepts are used to relate the concepts of the two languages. An example of such an intermediary concept is NeuronBundle; a NeuronBundle is a collection of neurons with the same types of neurons, with their respective components at the same location in the nerve network or following the same path through the nerve network (such as the axons of these neurons). In the following the focus is on a direct relationship. An overview of the language relationship is shown in Fig. 6.

In the autonomic nervous system, a neuron may have parts that are hosted by a nerve part (for instance a soma can be hosted by a ganglion). Other parts of a neuron may be carried by multiple nerve parts (for instance an axon may be carried by many segments, bifurcations, plexus and ganglia). Some neurons may be fully hosted by a nerve part (for instance inter-neurons in a sympathetic ganglion). It is exactly this kind of relationship between nerve models and neuron models that can be modeled using the HostAssociation and CarryAssociation.

4. The vagus nerve and neurons

The autonomic nervous system includes the left and right vagus nerves (or left and right cranial nerve X). The vagus nerve carries fibers that innervate various structures in the head, neck, thorax and abdomen. In reality, each of the two vagus nerves is part of a nerve network in the sense as discussed in Section 3.1. An extensive description of the vagus nerves and the vagus nerve networks is found in von Lanz and Wachsmuth (1955, Man) and Kobus (2008, quantification pig). In Section 4.1 the left and right vagus nerve networks will be modeled using the nerve network language introduced and discussed above. The model proposed is detailed and because we have adopted the UML, it serves as a schema for the design of a database supporting many areas such as qualitative and quantitative anatomical research, neural engineering and knowledge sharing. The model built here is based on the information sources previously referenced and is accurate with regard to these.

The left and right vagus nerve networks have somatic, parasympathetic and sensory pathways. In Section 4.2 we model different types of neurons using the language specified in Section 3.2. Pathways, comprised of neurons, and their relationship to the nerve network are considered in Section 4.3. The latter is far from complete, although it demonstrates how the languages and their relationships allow these to be modeled.

4.1. Model of the vagus nerve

Using the language introduced, we can apply it to model the vagus nerves (i.e., the left and right vagus nerves). We start out with modeling that part of the autonomic nervous system in which the left and right vagus nerves are involved and, in effect, we model two nerve networks of the autonomic nervous system. The scope of the model presented includes the nuclei, viscera and muscles as they appear as endpoints of the nerve network under study. The model developed is based on the extensive and detailed anatomic description and supported by von Lanz and Wachsmuth (1955) and Kobus (2008).

Recall from Section 3.1 that a nerve, such as the left or right vagus nerve, is modeled in terms of nerve segments, bifurcations and ganglia. Often, but not always, a ganglion has been given an anatomical name. However, a nerve segment and a bifurcation are part of the proposed modeling language and do not have a unique name in the anatomic nomenclature. Therefore, we introduce one here. The UML supports so called namespaces. Informally, a namespace can be seen as a folder where a class. Assume that we have two folders X and Y, and assume that we have a class “A” defined in folder X, and a class “A” defined in folder Y, then these two classes “A” are different. The way in which a class is identified, is by its “fully qualified name”, which consists of, in this simple case, the folder name and the class name: X::A and Y::A respectively. In the case where we define a class for a specific nerve segment or bifurcation, we introduce a class name comprised of the first letter (S, B respectively) followed by a number (for instance S3, B2). A tree of folders is introduced for the two vagus nerve networks. A nerve network is named after its main nerve, so at the highest level we have the folder named CN.X referring to the tenth cranial nerve. This name is used instead of vagus nerve for brevity. This folder has two sub folders L and R respectively referring to the left and right vagus nerve network. Because the classes appear in the left and right vagus nerve, they can be distinguished by their fully qualified names. So, S3 of the left vagus nerve is denoted : CN.X::L::S3, and S3 of the right vagus nerve is denoted CN.X::R::S3.

A nerve network, although named after its main nerve, also includes other nerves that are usually given (sometimes) unique names. Acronyms for these names will be used for naming the various components from which it is comprised. Nerve networks have connections to nuclei, ganglia, muscles and viscera. Recall from Section 3.1 that their original names will be used (preferably following the anatomic nomenclature).

The model of the vagus nerve is relatively large. Therefore, we present a part of it only, namely down to the level where branches to the cardiac plexus exist. The model is shown in Fig. 7. The model captures the vagus nerves (i.e., left and right vagus nerve) of an individual and their branching structures to the viscera.

According to Hopkins and Armour (1982, 1998), Hopkins et al. (1984, 1996) and Hopkins and Ellenberger (1994), the vagus nerves arise from three nuclei, the nucleus ambiguus, the dorsal motor nucleus of the vagus nerve (both efferent) and the nucleus of the tractus solitarius (afferent). At the jugular ganglion two branches split off, one ultimately connecting to the auricular area, the other connecting to the dura mater. Note that the model does not yet include a nerve segment between the jugular ganglion and the petrosal ganglion (which is part of cranial nerve IX). The vagus nerve continues to the nodose ganglion, where a branch splits off the pharyngeal nerve. Further caudally the superior laryngeal nerve branches off and splits into internal and external branches, both which innervate the larynx. On the left superior laryngeal nerve the aortic depressor nerve, which innervates the aortic arch, splits off. The aortic arch is modeled as a part of the aorta using the UML’s composite aggregation. Regarding the aortic depressor nerve, which equals the cranial cardiac nerves in Man (De Ribet, 1955), the scientific literature is not conclusive about the presence of this nerve in man (von Lanz and Wachsmuth, 1955). However, it has been observed in other vertebrates (Kobus, 2008).

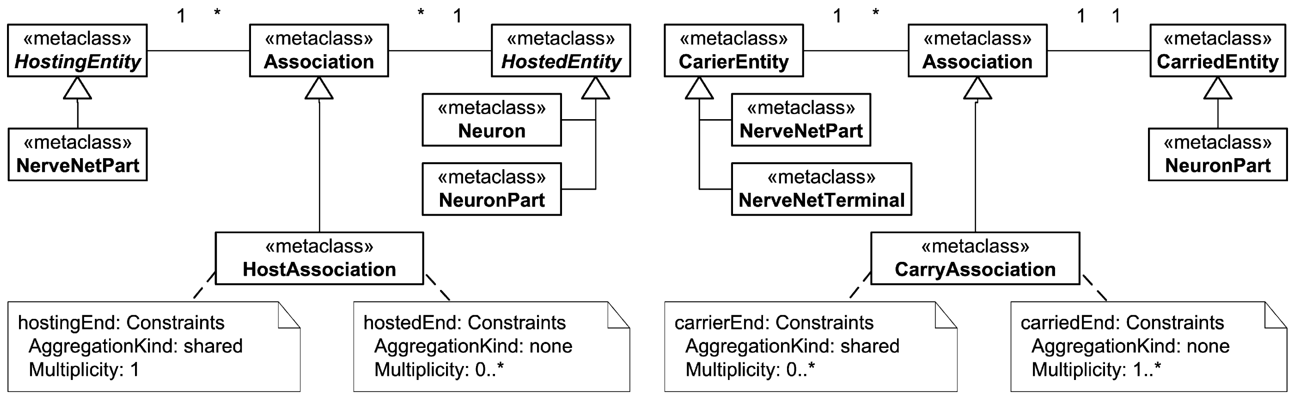

Fig. 6. Relationship between the nerve network language and the neuron network language.

Further caudally, cardiac branches split off the vagus nerve. According to Kobus (2008), there can be multiple branches (and presumably their number may vary per individual, hence per model instance). According to Gray (2000) there are two or three superior cardiac branches, and there is one inferior cardiac branch. In the model shown, Gray's description is followed, with two superior cardiac branches (named upper and lower superior cardiac branches). It must be noted, however, that according to Pick (1970, p. 305): "... a classification of separate superior, middle and inferior cervical vagal and sympathetic nerves appeared to be entirely arbitrary and meaningless because of the admixture of vagal and sympathetic fibers in the same nerve bundle and their variability in origin, disposition and anastomoses with adjacent nerves".

Continuing caudally, the recurrent laryngeal nerve branches off and gives branches to the cardiac plexuses, followed by branches that innervate the trachea and esophagus. Note that in the model no further detailing of the cardiac plexus is shown, such as its decomposition into superficial and deep parts, nor have the segments, the plexuses (for instance the left anterior pulmonary plexus, anterior pulmonary plexus, anterior coronary plexus, posterior coronary plexus) and viscera (i.e., left and right lungs, and left and right atria) that are innervated, been included.

Note that the model shown does not include communicating branches such as between the vagus nerve and the sympathetic nerves. These more accurate details can be modeled using the proposed nerve–nerve network language. Another challenge is to construct a model that accommodates individual variability.

4.2. Neuron models

The peripheral nervous system can be divided into afferent (sensory) and efferent (motor) parts. The peripheral nervous system brings sensations and processed data in the form of electrical pulses, called action potentials or spikes to the central nervous system. With respect to the autonomic nervous system, afferent neurons sense, for example, the state of blood vessels and viscera. The efferent neurons, on the other hand, transfer electrical pulses from the brainstem or spinal cord to the smooth muscles, heart and glands. The efferent limb of the autonomic nervous system can also be divided into the sympathetic and parasympathetic parts, which to some extent are antagonistic in function or in balance. That is, the action of the one limits or reduces the action of the other. A balanced coordination of these antagonistic excitatory and inhibitory actions ensures a bodily homeostasis.

The afferent part of the vagus nerve typically consists of pseudounipolar neurons. These neurons have or connect to sensory receptors at the periphery, but unlike other neurons have a "peripheral" axon starting from the periphery conveying the electrical pulses to the spinal cord directly but also reach the soma. The axon ends at axon terminals located in a specific area (nucleus) of the brain stem where they synapse with local neurons.

The sensory receptors in the viscera are generally silent mechano (presso) or chemo nociceptors, similar to the type of receptors in the skin. However, these receptors normally are not activated by noxious stimuli. Besides chemical stimuli, they respond to mechanical distortion, pressure or stretching. These receptors can be free nerve endings, i.e., have simple structure and are unmyelinated, and typically are small diameter C-fibers. A generic, detailed model of the pseudounipolar neuron is given in Fig. 8.

An efferent path from a central nervous system nucleus to a viscus typically consists of two neurons: a preganglionic neuron and a postganglionic neuron. The preganglionic nerve fibers are myelinated and the postganglionic nerve fibers are, as a rule, unmyelinated. Furthermore, sympathetic preganglionic nerve fibers are short compared to the sympathetic postganglionic fibers, while parasympathetic preganglionic nerve fibers are long and the postganglionic nerve fiber are short with branches terminating in a small area of the viscera to be innervated. The type of neurons in efferent paths are typically multipolar neurons, hence neurons with multiple dendrites, one soma, one axon and multiple axon terminals. A dendrite usually has a tree like structure, and may have many points at which axon terminals of other neurons synapse. Complex dendritic tree structures can be modeled with the proposed metamodel using PartComposition and InputPart. A synapse not only can occur at some location in a dendritic tree, but also at the boundary of the soma, and even at the axon (usually close to the axon hillock). All these are chemical synapses. Electrical synapses can also exist in those cases where somata are directly adjacent. All these different forms of synapses can be modeled using the metamodel. However, in the following a simple model of a multipolar neuron will be used. In this model, as shown in Fig. 9, the multipolar neuron consists of one soma, one axon and one or more dendrites. The axon is composed of one fiber and one or more axon terminals. In this model, the fiber is the part of the axon that may be myelinated. The axon terminals are the neuron's output and therefore presynaptic. Each dendrite contains one or more dendrite terminals, which are the neuron's inputs and are post-synaptic. Note that the soma and fiber are internal and therefore do not synapse.

Observe that the dendrites and soma are associated; which reflects that these are structurally and behaviorally connected. Similarly, Soma and Fiber are associated, and Fiber and AxonTerminal. Hence, the neuron is considered as a system consisting of interrelated components, these relationships are important in further detailing the behavior of the multipolar neuron.

Neurons interact via chemical or electrical synapses. For brevity, only chemical synapses will be addressed in the context of motor paths in the autonomic nervous system. In Fig. 10, the three basic

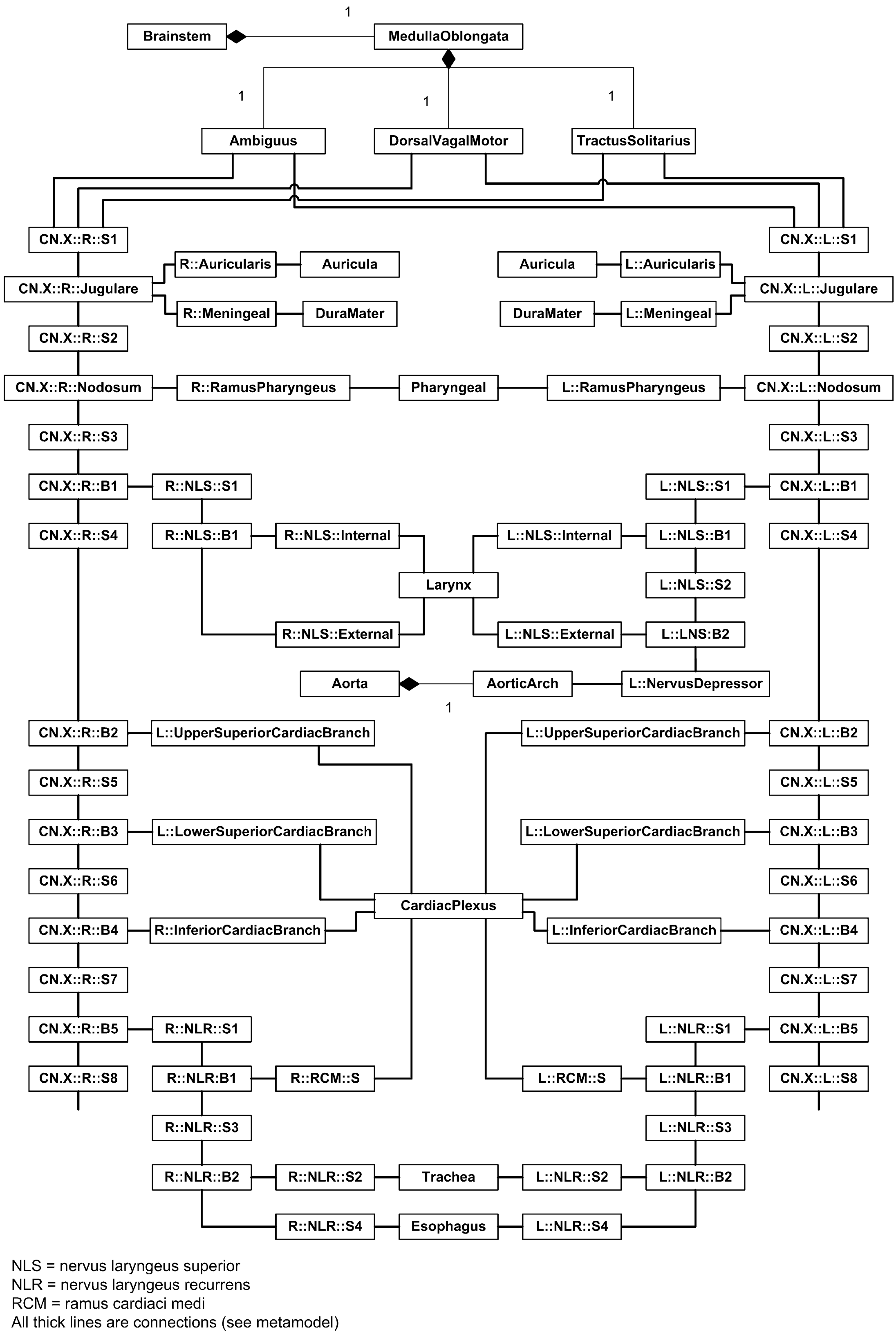

Fig. 7. The upper part of the vagus nerve network.

synapses are shown. The motor path: (a) between nucleus neuron and premultipolarneuron; (b) between premultipolarneuron and postmultipolarneuron; and (c) between postmultipolarneuron and effector cell.

A fiber conveys electrical pulses generally unidirectionally to axon terminals. Propagation speed of these electrical pulses corresponds to the diameter of the fiber. As a rule: the larger the diameter, the thicker the myelin sheath and the faster the propaga-

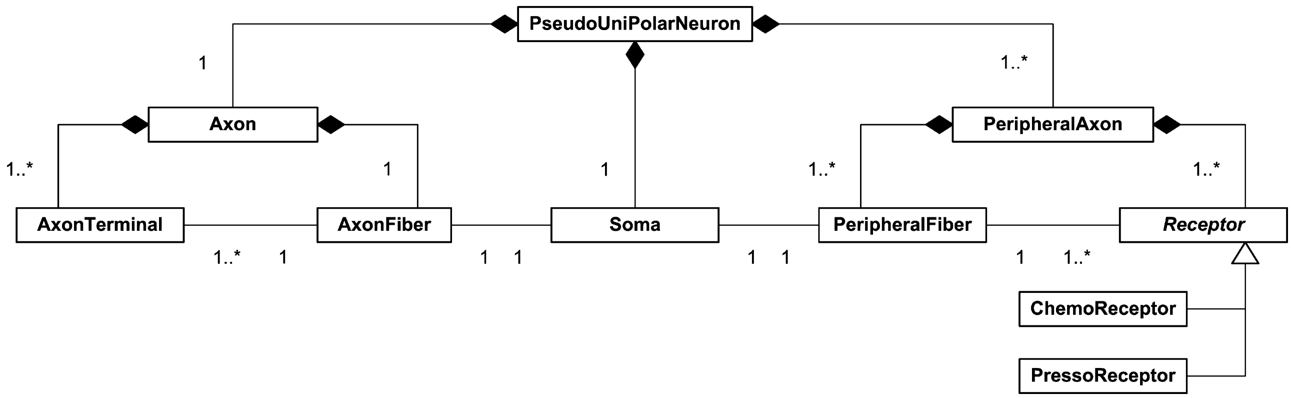

Fig. 8. A model of the pseudounipolar neuron.

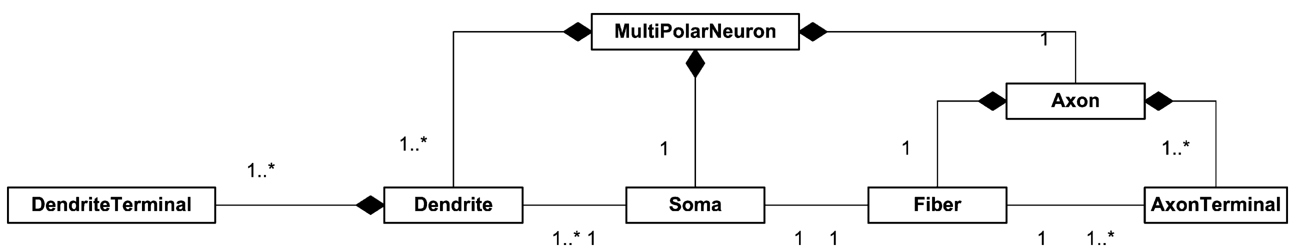

Fig. 9. A model of a multipolar neuron.

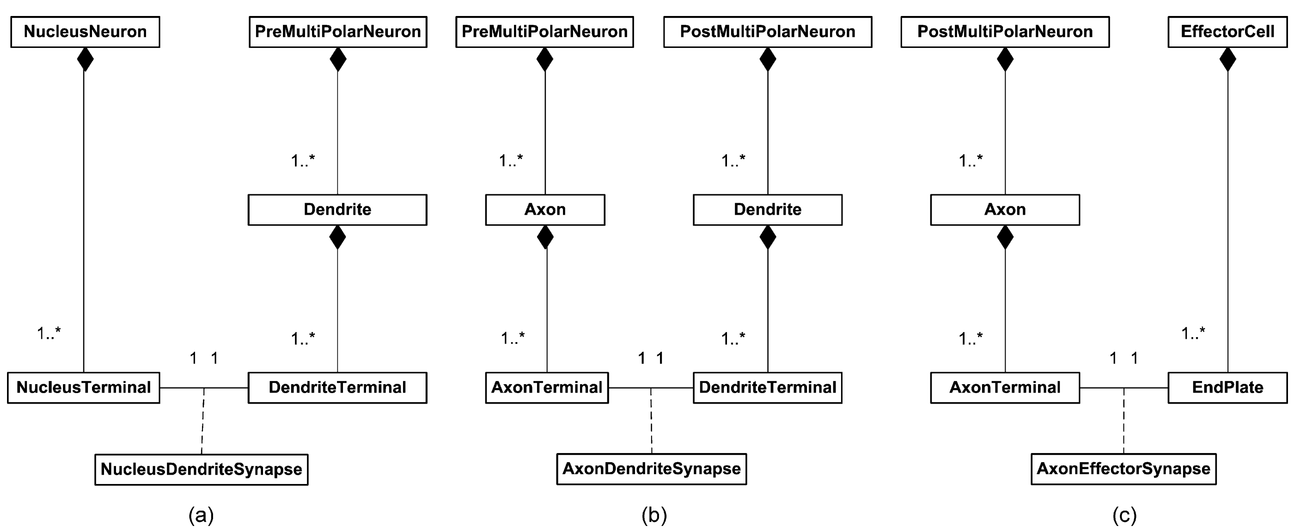

Fig. 10. Three basic synapses.

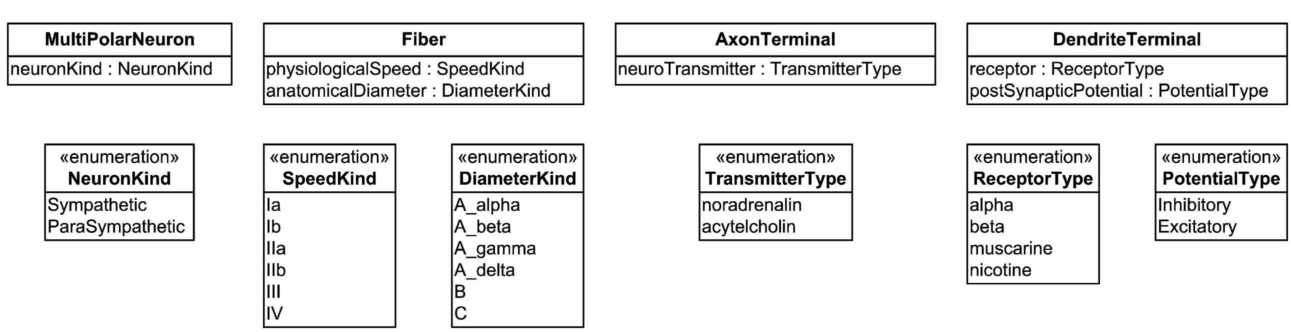

Fig. 11. Various class attributes and their type.

tion. Therefore, electrical pulses not only propagate faster in fibers with large diameters, but also travel with less distortion. Fibers are characterized by physiological speed, with values "Ia", "Ib", "IIa", "IIb" "III", and "IV" (Kandel et al., 2000, Table 36-1, p. 720) and by diameter, using the values "A-alpha", "A-beta", "A-gamma", "A-delta", "B" and "C" (Kandel et al., 2000, p. 474). An "A-alpha" fiber is thickly myelinated and found in somatic motor neurons. The fiber attributes and their value types are shown in Fig. 11.

When considered necessary, one may extend the class Axon with methods and attributes to specify the conversion or the relation between the two attributes discussed earlier. One may further include the date and time of the conversion and also the actor responsible for the conversion. For example, if the diameter of a specific axon of a specific subject is known to be "A-alpha", the value "Ia" can be assigned to the speed attribute. If this conversion uses a conversion table, a reference to this table may also be specified, for example, for auditing purposes. In this way, a computational neural model which requires the axon's speed as input value instead of the diameter can benefit from the proposed UML model.

An AxonTerminal is characterized by the neurotransmitter (represented by the attribute neuroTransmitter in Fig. 11). With few exceptions, for sympathetic multipolar post-neurons the value of this attribute is "noradrenaline", and for parasympathetic multipolar neurons the value of this attribute is "acetylcholine" as is for preganglionic sympathetic neurons.

The post-synaptic end of a synapse, such as a dendrite terminal, contains "alpha" or "beta" receptors for the sympathetic neurons and "nicotine" or "muscarine" receptors for the parasympathetic neurons. These receptors determine in part the excitatory or inhibitory effect of the neurotransmitter. Therefore, DendriteTerminal has the attributes receptor and postSynapticPotential as shown in Fig. 11.

Note that the discussion on the structure of neurons and their attributes is limited. Other neuron models can be constructed including models that take into account more detail such as the axon hillock, or a more detailed tree model of a dendrite. For the latter one can introduce attributes to model the capacitive step response constant and the propagation decay of the membrane potential along dendrite branches.

4.3. Efferent and afferent pathways

In the previous sections a (partial) model of the vagus nerve network is discussed and the main parts to model the efferent and afferent pathways are described. In this section these models are combined into an anatomical-physiological model. Such a model uses the (meta-) relationships as specified in Section 3.3.

As an example, an afferent portion of the arterial baroreceptor reflex is modeled in further detail. This, of course, does not give a complete model of the baroreceptor reflex, but it demonstrates the modeling principles applied. The arterial baroreceptor reflex controls the cardiovascular system by regulating, for example, the heart rate and the strength of cardiac muscle contractions and, thereby, also the arterial blood pressure.

The control involves afferent and efferent sympathetic and parasympathetic pathways of neurons and processing at the level of the nucleus of the tractus solitarius in the medulla. The afferent part consists of the following six parts.

- Baroreceptor path from the left and right carotid bodies to the nucleus of the tractus solitarius.
- Chemoreceptor path from the carotid sinus (left and right) to the nucleus of the tractus solitarius.
- Baroreceptor path from the aortic body to the nucleus of the tractus solitarius.
- Chemoreceptor path from the aortic body to the nucleus of the tractus solitarius.

In contrast to Scher (1977) we do not consider a further subdivision of the nucleus of the tractus solitarius. In case that it would be desired, such a refined model can be included easily.

The pseudounipolar neurons of these paths have their somata located in the nodose ganglion. On the left, the peripheral fibers are carried by the aortic depressor nerve and from there on there is only one path to reach ultimately the nucleus of the tractus solitarius. Because this path is realized by a number of pseudounipolar neurons, multiple axons are carried by the various nerve parts. The model that integrates the vagus nerve network and the neuronal network is shown in Fig. 12 (note that only the left vagus nerve is considered). The model demonstrates the use of the host- and carry-associations (Fig. 6). In a similar way, the other afferent pathways can be modeled. As well, the efferent parasympathetic pathway (whose neuron parts are also carried and hosted by the vagus nerve network) and sympathetic pathways (whose neuron parts are carried and hosted by the sympathetic trunks) can be modeled.

4.4. Computational models and simulation

So far structural aspects of nerve network and neuronal network models have been addressed. To put these models into effect, neuroscientific applications are needed. Examples of such applications include visualization of nerve networks and neuronal networks, graphical interfaces for the design of these models, simulation of neuronal networks, and databases for storing, retrieving and analyzing experimental results. A high-level plug-in based architecture for such a system has already been proposed for NeuroML (Goddard et al., 2001).

The models discussed so far capture the structural aspects of neuronal networks, which make anatomic visualization and modeling possible. Behavioral aspects have not been considered yet. Two main areas of interest here are results for neuronal measurement and computational models. In the following we give directions on how computational models can be included for simulations. To keep the discussion within limits, we restrict it to the pseudounipolar neuron (for anatomy see Matsuda and Uehara, 1981; Devor, 1999; Feirabend and Marani, 2003; for physiology see Wang et al., 1992, 1994, 1995, 1997).

For the behavior of a particular type of neuron, such as the pseudounipolar neuron, different simulation models exist, roughly classified into single compartment models and multi-compartment models (Dayan and Abbott, 2001). The way in which the behavior is represented is a design decision choice out of the following two. Firstly, the behavior can be modeled using one of the UML's diagramming techniques especially designed for this purpose; good candidates are sequence diagrams and state diagrams (Booch et al., 1999). The second approach is a descriptive approach in which knowledge about a computational model is captured. The interfaces of the model are made available, the internals of the implementation of the model are hidden; and the description of the behavioral model is captured using a system of mathematical equations. Given the fact that real simulations are to be executed using a general purpose or dedicated simulation engine, and that many simulation model implementations already exist, it is obvious that this second descriptive approach is most suitable to promote reusability of the considerable effort and results that have already been accomplished and produced in neuroscience. In neuroML (Goddard et al., 2001), the same approach is followed. In conclusion and relative to the pseudounipolar neuron, we model its (potentially multiple) computational model

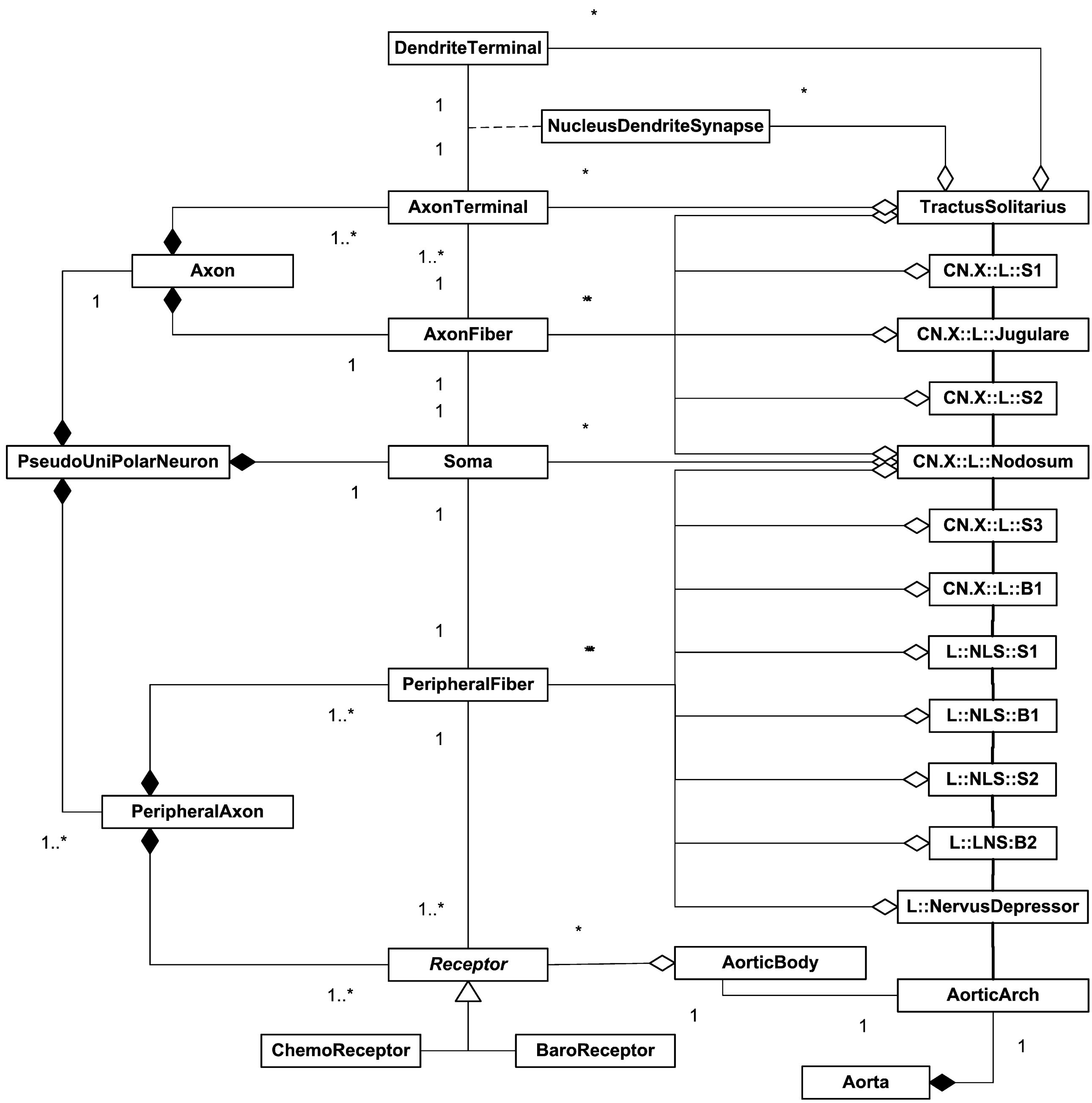

Fig. 12. The aortic body chemoreceptor and baroreceptor afferent pathways (only relative to the left vagus nerve is shown).

as a separate class, named ComputationalModel, associated with this neuron (see Fig. 13). The attributes of the computational model typically include the following: name, for the name of the model; description, for a textual description of the model; references, for the scientific publication on which this model is based (best using the document object identifier, DOI); location, giving the URL where the model is permanently stored; simulator, for the simulator to be used for the model, and modelParameters that specify the parameters of the computational model that need to be set for simulation. As an example, NEURON has a database with computational neuroscience models. This database includes the DRG neuronal model proposed by Amir and Devor

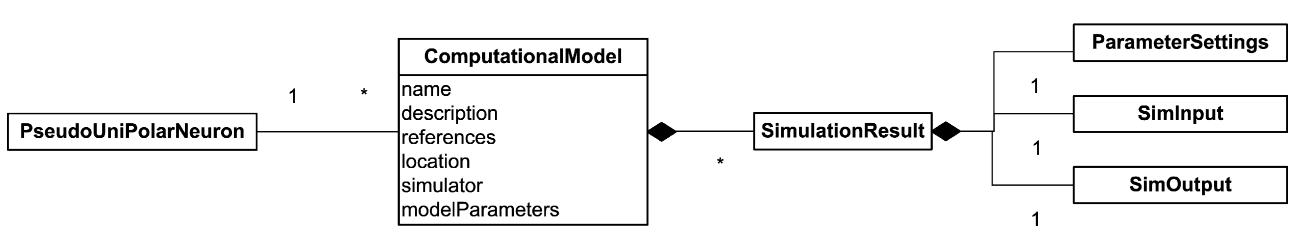

Fig. 13. Extension of the pseudounipolar neuron with computational models.

(2003), i.e., including the code to run the model in the NEURON simulator.[1]

In addition, results from simulation runs may be associated with a computational model. These results must a least capture the parameter settings used in the simulation run(s) and the simulation input data and output data resulting from the simulation run(s).

5. Discussion

In Section 1 it was stated that the proposed metamodel and models are useful for a number of reasons. In the following, these are reiterated in the context of the metamodel and the models previously proposed.

The metamodel and models based on it, may be drawn on article and discussed among researchers. These may also be designed using a UML design environment. In the latter case, publishing these models is easy as these environments usually have automated support for generating web-pages. In a more integrated approach, sub-models and model extensions can be designed, submitted, validated and published to a shared knowledge base. Furthermore, a virtual community can be created and whose members collectively build up, extend and share this knowledge.

The metamodel and models presented are claimed to promote the communication among experts in the field. Discussion among the authors of this article in the course of the modeling effort gives an early indication on the validity of this claim, further larger scale experimentation where domain experts apply the proposed language and communicate models are required for further qualitative and quantitative underpinnings.

The nerve network models may serve as schemata for database design. Although not detailed further in Section 4.1, it is obvious that the various parts may have various attributes as to characterize the part's macroscopic anatomical appearance, such as length, diameter and the like. Also, to characterize the relative location of a part, associations to other body parts can be included (such as relative position and distance to these body part). The way in which nerve networks are modeled may also prove relevant in (large scale) morphometric studies in which many researchers are involved from possibly different laboratories. First of all the schemata may help in specifying the part(s) of interest in the study, and in specifying adjunct classes for these part(s) that capture the attributes of interest to be analyzed with respect to each subject. These attributes include such things as fiber counts (myelinated, unmyelinated) and fiber diameter distributions. Data collected could then be further analyzed to generate relevant statistics.

In this article, the details of neuron behavior have been considered to some elementary level and it has been shown how computational models and simulation results can be associated with a neuron. Further rising to computational models at the neuronal network level is to be considered in the future, important work in this area has already been done by Burns (2001).

In the field of neural engineering and neural stimulation, potential desired impact and possible side-effects can be analyzed qualitatively simply by model navigation techniques. Given the nerve segment to be stimulated, the nerve networks and the fibers carried and affected by the stimulation can be analyzed as to determine the affected nerve network terminals. Quantitative analysis requires behavioral models of neurons and a sufficiently complete anatomical model, the model of the vagus nerve discussed in this article does not satisfy this latter requirement. Relative to the nerve network language, an important observation to be made is that the models built from it are structurally static. As an example, for the vagus nerve it has been assumed that there are three cardiac branches that split of the vagus nerve. However, there may be variations at the individual level: one individual could have two cardiac branches, another may have three. Inclusion of these variations into our model requires further investigation.

In this article a new direction to capture knowledge about the autonomic nervous system has been set out. The separation between metamodel and model is important, as the former specifies the concepts, relationships and constraints specific for this knowledge domain. The latter specifies concrete neuroanatomic-physiological parts of the autonomic nervous system. The focus in this article has been on the conceptual modeling of the autonomic nervous system. These conceptual models serve as a starting point for software tool design. Therefore, the proposed modeling language will be made operational as UML Profiles, designed models like the vagus nerve model can then be combined with implementation profiles such as for Java Persistence API to create the database and to generate the database logic to create, retrieve, modify and delete database content. The logic of the software tools to be developed can be designed using the same UML model and combine it with technology profiles such as Java Beans. Design of web-based user interfaces can also be generated automatically in part when combining the UML model with the Java Server Faces profile.

Transfer of the proposed UML metamodels and models between UML supporting tools and applications is supported using XMI (OMG XMI). Hence, these metamodels and models can be made available at some central or distributed repository for reuse by others. However, many ways to transfer these models or their representations in databases can be considered, for example via serialization of the Java objects or XML schema and data, which are easily generated (automatically) by the same approach mentioned earlier. Preferences of the users or designers, the need for reuse of results of past efforts in the field and the availability of tools at the collaborating laboratories determine the way to transfer these models. Development of neuroscientific application components applying the proposed metamodels and integration of existing tools (for instance simulators) is the next step we need to make.

Acknowledgements

We thank David Hopkins for his valuable and constructive comments on earlier versions of this article.

References

Amir R, Devor M. Electrical excitability of the soma of sensory neurons is required for spike invasion of the soma, but not for through-conduction. Biophys J 2003;84:2181–91.

Booch G, Rumbaugh J, Jacobson I. The Unified Modeling Language user guide. Boston, MA: Addison-Wesley; 1999.

Bower JM, Beeman D. The book of GENESIS: exploring realistic neural models with the GEneral NEural SImulation system. Springer; 1998.

Bruhn R. Designing XML schema for bioinformatics using UML. J Comput Sci Coll 2006;21(5):13–9.

Burns GAPC. Knowledge management of the neuroscientific literature: the data model and underlying strategy of the neuroscholar system. Philos Trans R Soc B 2001;356(1412):1187–208.

Carnevale NT, Hines ML. The NEURON book. Cambridge University Press; 2006.

Dayan P, Abbott LF. Theoretical neuroscience – computational and mathematical modeling of neural systems. Cambridge, MA: The MIT Press; 2001.

Devor M. Unexplained peculiarities of the dorsal root ganglion. Pain 1999;82(S1):S27–35.

De Ribet RM. Le system nerveux de la vie végétative. Paris, France: G Dion et Cie; 1955.

Eliasmith C, Anderson CH. Neural engineering: computation, representation, and dynamics in neurobiological systems. Cambridge. MA: The MIT Press; 2003.

Feirabend HKP, Marani E.The dorsal root ganglion. Encyclopedia Neurol Sci 2003;4(1):28–33.

Gray H. Anatomy of the human body. Philadelphia, PA: Lea & Febiger; 2000 [1918], bartleby.com.

[1] For details see http://senselab.med.yale.edu/ModelDB/ShowModel.asp?model=51022.

Goddard NH, Hucka M, Howell F, Cornelis H, Shankar K, Beeman D, et al. Model description methods for collaborative modelling in neuroscience. Philos Trans R Soc B 2001;356(1412):1209–28.

Groves DA, Brown VJ. Vagal nerve stimulation: a review of its applications and potential mechanisms that mediate its clinical effects. Neurosci Biobehav Rev 2005;29(3):493–500.

Heldoorn M, Van Leeuwen JL, Vanderschoot J, Marani E. Electronic coupling in a network of compartmental external urethral sphincter motoneurons of Onuf's nucleus. Neurocomputing 2001;38–40:647–58.

Heldoorn M, Marani E, Van Leeuwen JL, Vanderschoot J. A compartmental model of an external urethral sphincter motoneuron of Onuf's nucleus. Arch Phys Biochem 2003;111:193–201.

Hopkins DA, Armour JA. Medullary cells of origin of physiologically identified cardiac nerves in the dog. Brain Res Bull 1982;8:359–65.

Hopkins DA, Armour JA. Brainstem cells of origin of physiologically identified cardiopulmonary nerves in the rhesus monkey (Macaca mulatta). J Auton Nerv Syst 1998;68:21–32.

Hopkins DA, Bieger D, deVente J, Steinbusch WM. Vagal efferent projections: viscerotopy, neurochemistry and effects of vagotomy. Prog Brain Res 1996;107: 79–96.

Hopkins DA, Ellenberger HH. Cardiorespiratory neurons in the medulla oblongata: input and output relationships. In: Armour JA, Ardell JL, editors. Neurocardiology. New York, NY: Oxford University Press; 1994. p. 277–307.

Hopkins DA, Gootman PM, Gootman N, Di Russo SM, Zeballos ME. Brainstem cells of origin of the cervical vagus and cardiopulmonary nerves in the neonatal pig (Sus scrofa). Brain Res 1984;306:63–72.

ITU Recommendation X.901-X.904. Open Distributed Processing – Reference Model, Geneva, Switzerland; 1996.

JSim. The Physiome project; 2009, http://physiome.org/jsim/index.html [Last visited on December 16, 2009].

Kandel ER, Schwartz JH, Jessell TM. Principles of neural science. New York, NY: McGraw-Hill; 2000.

Kobus T. Selective stimulation and recording of the nervus vagus for control and monitoring of the heart. Master Thesis. Department of Electrical Engineering (BSS 08-14), University of Twente, the Netherlands; 2008.

Matheny RG, Shaar CJ. Vagus nerve stimulation as a method to temporarily slow or arrest the heart. Ann Thorac Surg 1997;63(6 S):S28–9.

Matsuda S, Uehara Y. Cytoarchitecture of the rat dorsal root ganglia as revealed by scanning electron microscopy. J Electron Microsc 1981;30:136–40.

OMG Unified Modeling Language (OMG UML). Superstructure V2.1.2; November 2007, http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF/ [Last visited on December 16, 2009].

OMG Unified Modeling Language (OMG UML). Infrastructure, V2.1.2; November 2007, http://www.omg.org/spec/UML/2.1.2/Infrastructure/PDF/ [Last visited on December 16, 2009].

OMG XML Metadata Interchange (OMG XMI). MOF 2.0/XMI Mapping, Version 2.1.1; December 2007, http://www.omg.org/spec/XMI/2.1.1/PDF/ [Last visited on August 11, 2010].

Parsons J, Cole L. What do the pictures means? Guidelines for experimental evaluation of representation fidelity in diagrammatical conceptual modeling techniques. Data Knowl Eng 2005;55:327–42.

Pick J. The autonomic nervous system: morphological, comparative, clinical and surgical aspects. Philadelphia, PA: JB Lippincott; 1970.

Prodanov D, Marani E, Holsheimer J. Functional electric stimulation for sensory and motor functions: progress and problems. Biomed Rev 2003;14:23–50.

Rall W. Electrophysiology of a dendritic neuron model. Biophys J 1962;2:145–67.

Rall W. Theoretical significance of dendritic tree for input–output relations. In: Reis RF, editor. Neural theory and modeling. Stanford, CA: Stanford Univ Press; 1964. p. 73–97.

Revest P. NeuroSim for Windows. Trends Neurosci 1995;18(12):556.

Routledge N, Bird L, Goodchild A. UML and XML schema. In: Proc. 13th Australasian database conference; 2002. p. 157–66.

Rozman J, Bunc M. Modulation of visceral function by selective stimulation of the left vagus nerve in dogs. Exp Physiol 2004;89(6):717–25.

RSD. Rational Systems Developer; 2008, http://www.ibm.com/developerworks/rational/products/rsd/ [Last visited on December 16, 2009].

Scher AM. Carotid and aortic regulation of arterial blood pressure. Circulation 1977;56(4):521–8.

von Lanz T, Wachsmuth W. Hals I,2, Kopf Übergeordnete Systeme I,1a, Praktische Anatomie. Springer; 1955.

Wand Y, Weber R. Research commentary: information systems and conceptual modeling – a research agenda. Inform Syst Res 2002;13(4):363–76.

Wang Z, Van den Berg RJ, Ypey DL. Resting membrane potentials and excitability at different regions of rat dorsal root ganglion neurons. Neurosci 1994;60:245–54.

Wang Z, Van den Berg RJ, Ypey DL. Do hyperpolarization-activated currents Ih and IKi play an role in neurite sprouting in sensory neurons in culture. Neurosci Res Commun 1995;17:53–5.

Wang Z, Van den Berg RJ, Ypey DL. Hyperpolarization-activated currents in the growth cone and soma of neonatal rat dorsal root ganglion neurons in culture. J Neurophysiol 1997;78:177–86.

Wang Z, Ypey DL, Van Duijn B. Inositol triphosphate-induced hyperpolarization in rat dorsal root ganglion neurons. FEBS Lett 1992;304:124–8.