Software engineering tools and approaches for neuroinformatics: the design and implementation of the NeuroScholar Knowledge Base Management system. •

Gully A. P. C. Burns [1]

[1] The Knowledge Mechanics Research Group, University of Southern California, Los Angeles, CA 90089-2520

Abstract

I describe the data model and implementation strategy of a knowledge base management system (called 'NeuroScholar') designed to allow users to construct a personalized account of their understanding of the neural connectivity underlying a specific behavior. The ontology of the system is presented in a detailed software engineering model expressed in the Universal Modeling Language (the 'UML') as a statement of the neuroinformatics strategy underlying the system. Additionally, I present software engineering scripting tools to generate abstract graph-based model of the ontology in order to accelerate the process of code generation and permit interoperability with other systems.

• Poster Presentation

Software engineering tools and approaches for neuroinformatics: the design and implementation of the NeuroScholar Knowledge Base Management system.•

Gully A. P. C. Burns [1]

[1] The Knowledge Mechanics Research Group, University of Southern California, Los Angeles, CA 90089-2520

Introduction

This presentation is concerned with the design and implementation of the NeuroScholar system, expressed as a highly detailed software engineering specification that forms a neuroscientific strategy for representing data concerning ethology, physiology and neuroanatomy. The system permits inferences to be recorded in order to represent theoretical arguments within the system and for annotations to be attached to these inferences so that human users may easily understand their underlying rationale.

This specification may be manipulated and interpreted with a scripting toolset to accelerate the process of software development by automatic code generation in order to provide the basis for a user interface design, a method of data-encapsulation and analysis techniques. This permits seamless updates from the conceptual design of the conceptual data model to the implementation of the system within code.

NeuroScholar is therefore, a neuroinformatics project in itself that addresses an important issue: knowledge management of the neuroscientific literature. It also is being developed using a structured methodology that may have wider impact for the development of similar projects, and of techniques that may promote interoperability between them.

The NeuroScholar project

The NeuroScholar project seeks to answer the following question: "What is the neural circuitry underlying a specific behavior?" based on published (and hence peer-reviewed) information. It does so by means of a simple, three-level strategy: (1) by defining a representation of a given behavioral phenonmenon; (2) by identifying which populations of neurons are involved in producing the behavior and then (3) by comprehensively delineating the neural circuits between the desired neuronal populations.

NeuroScholar addresses the task of collating and interpreting information from published experiments with subsystems that may be grouped into three categories, illustrated in Figure 1. The first set of subsystems are fundamental data structures that support the system as a whole. They consist of constructs concerning different types of measurement; a high-level framework (referred to as 'the knowledge mechanics core') concerned with document fragments, their classifications ('objects'), inferences ('relations') and annotations, and two specialized systems. One concerned with the representation of space using neuroanatomical atlases and the other concerned with the representation of networks (as mathematical graphs).

Subsystems at the second level are concerned with accurately classifying the contents of the papers themselves, including a general approach to the representation of scientific protocols (shown in Figure 1 as 'flowcharts'), and specialized data constructs for ethological, physiological and histological data.

• Poster presentation

The third and final subsection is concerned with the construction of the so-called ‘knowledge model’ to form the ultimate output of the system. These knowledge models are designed to represent behavior as a set of linked ethograms, which would then be associated to representations of populations of neurons.

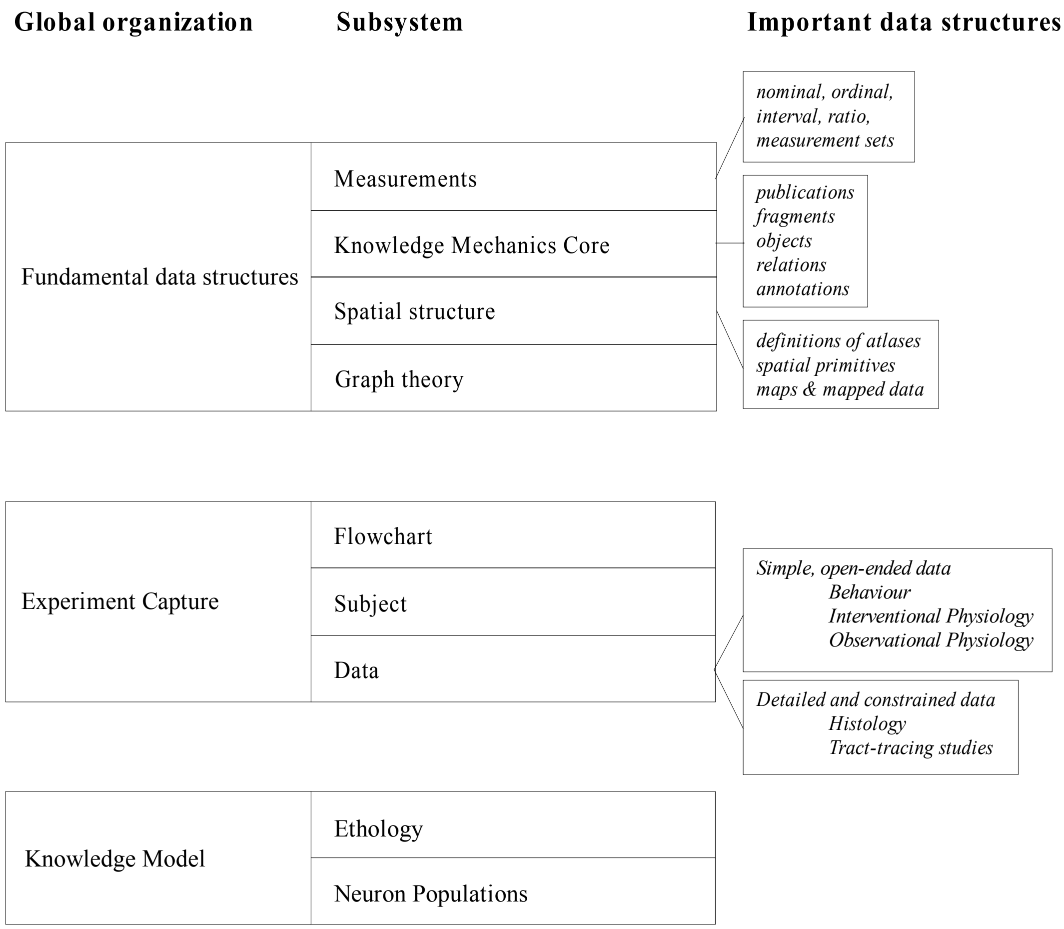

Figure 1: Large-scale organization of subsystems within NeuroScholar

The data model of the system as a whole appears as a series of class diagrams within the Universal Modeling Language. The representation of each level is sophisticated and involves a detailed explanation of the representation. For example, the representation of neuronal populations is illustrated in Figure 2. The spatial extent of the cellular components of the neuronal populations are included (as specializations of the ‘map’ class which is based on a representation of a neuroanatomical atlas), coupled with descriptions of various cellular properties (such as the varicosities of axons, the packing density of cells, *etc.*). This representation may be computationally manipulated.

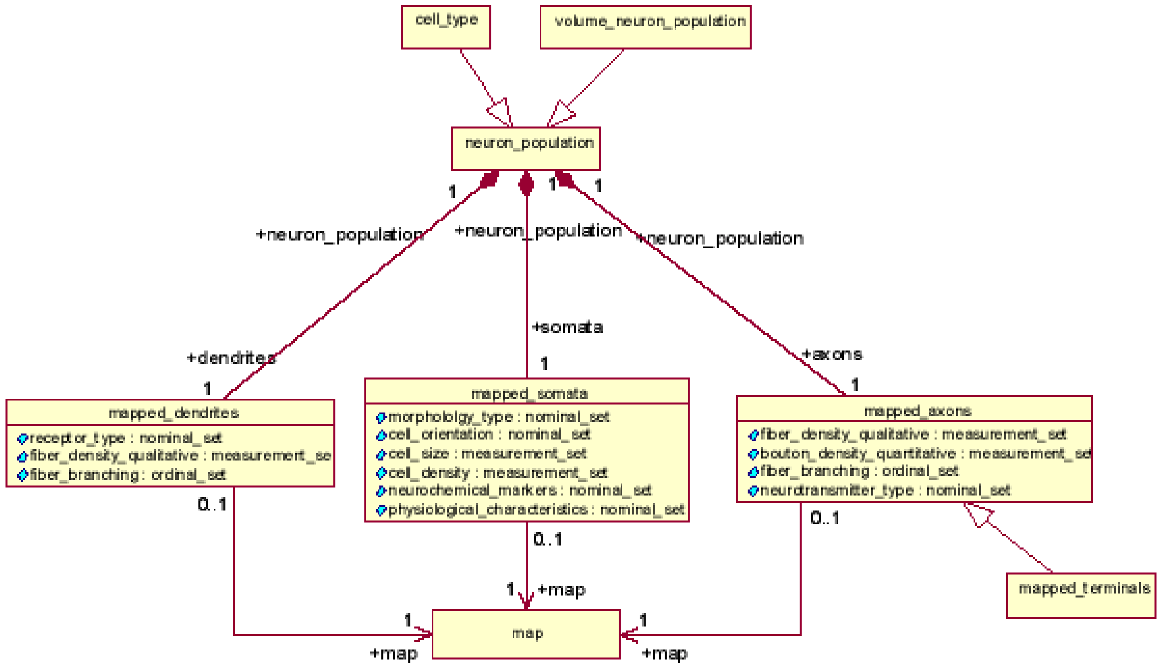

Figure 2: UML class diagram depicting NeuroScholar's representation of populations of neurons.

Software engineering tools and approaches

The software engineering environment under development is based on set of Perl scripts that use the Rational Rose's object linking and embedding (OLE) extensibility interface to build a graph-like representation of the data model of database systems. Rose may reverse-engineer the data models of a range of programming languages and forms of data representations (Java, C++, Oracle, SQLServer, XML), allowing the construction of these graphs for any of these UML-based representations. The use of methods to forward-engineer code from this representation allows designers to automate aspects of software development so that changes to the underlying design of a data model may propagate through the implementation of the system rapidly and seamlessly. Additionally, comparisons of graphs derived from different systems may allow users to compare and link data models from different systems.