Chapter 1

Introduction

1.1 Computational modelling in neurobiological research

The number of neurobiologists applying computational modelling in parallel with experimental studies has been continuously growing during the last years. Computer simulations are utilized to investigate the connection between structures of neurobiological systems and their functioning, ranging from particular neurobiological elements to complex neural systems. In experimental work computational modelling is used to determine the most important features and parameters of the systems. In some cases it can even provide information about the structure and parameters of real systems missing in experimental data.

The growth of interest in computational modelling has been driven mainly by two factors. The first is the development of equipment allowing researchers to perform experiments which provide very detailed anatomical and physiological descriptions of real neurobiological systems. These experimental efforts have led to an increasing amount of data providing the base for the construction of realistic simulation models.

The second is the rapid development of computer technologies which has made possible the implementation of complex computer models to describe real neural systems based on detailed microscopical information. This has stimulated development of computational neuroscience investigating the behavior of neurobiological models based on suggested structures.

Today, the neurobiological research essentially consists of two parts: experimental work and computational modelling. It is commonly accepted that a combination of experimental research and theoretical modelling is necessary to get insight in the functioning of neurobiological systems. Modelling allows one to study the correlation between the functioning of neural elements and their parameters, systematize collected data and determine directions of further experimental studies.

1.2 Features of existing simulation packages

As modelling becomes important in neuroscience research, the simulation software appears to be a necessary part of the laboratory. Consequently, the user-friendliness of the software application has become essential. Easiness in use includes such characteristics as simple installation, platform independence, no requirement for extensive computer knowledge, and an intuitively clear and user-friendly interface.

The currently available packages, such as Genesis [41] and Neuron [30], were developed in an attempt to create a "general" simulation system. The basic feature of a "general" simulation software is that it is capable of solving problems at any level of detail, from parts of neurons to large neural systems. However, the general domain of nerve simulation is often too large for one program to optimally deal with all problem levels.

The existing simulation systems are often Unix-based, since they require rather large computational power for solving differential equations arising from the models. In the hands of experienced users with access to an appropriate computer system, such modelling packages are powerful research tools.

However, these packages have several drawbacks. For novice users, it is quite time consuming to learn the formal structure of the scripting languages, in which users must define components and parameters for the simulations. At least initial knowledge of Unix systems and basic skills in working with them are necessary. Even for experienced users, a lot of time is necessary to set up and debug the simulation process. These packages do not provide a graphical user interface (GUI) or do so only for common model experiments. The difficulty of constructing a comprehensive GUI has its origins in the wide application domain of the packages. The large spectrum of possible models requires flexible tools for specifying model elements and their parameters, which is difficult to implement. This requirement essentially limits the packages' visualization capabilities.

Despite the object-oriented structure of most existing packages, it is not easy to modify, extend, and integrate them into other software. The open-ended implementation of simulated elements is difficult to understand, since the functions of numerical integration, object fields and commands for the interpreter are placed in the same modules. Thus, there are certain difficulties for the user to understand the model's logical structure. The extensibility is only possible on the basis of the predefined objects.

1.3 Contributions of this thesis

The central contribution of this thesis is the development of a neural simulation system based on a client-server architecture. This architecture provides the most efficient use of computer resources for modelling of neural systems.

Biologically detailed simulations of neurons and neural networks are based on compartmental modelling: Each cell is divided into many isopotential compartments described by equivalent electrical schemes. The compartments are joined by conductances and activated via simulated ionic channels and current injectors. In this case, each small segment of a neuron is described by a set of ionic equations, aiming at a faithful description of the processes of information transmission.

The computational resources of high-power computers are necessary for solving the system of differential equations describing the spread and interaction of electrical and chemical signals of real neurobiological systems. While setting up the models, control and visual presentation of the results can be left to a personal computer.

All of these aspects have stimulated the development of software with the following distinctive features:

- platform independence;
- user-friendly and intuitive GUI;
- a comprehensive library of standard models, on which basis the definition of complicated neuron systems and their modifications are possible without programming skills;
- the opportunity for users to extend this library; a standard and extensible format of the model descriptions, which ensures compatibility with different software packages;
- active utilization of visual methods for presenting neural networks and their functioning on the basis of simulation results
- the possibility of integration into educational systems.

NeuroSim was developed as a prototype of a simulation system incorporating all these features. Java was chosen for its realization. Thus, NeuroSim can run on virtually any computer and on most operating systems. The client-server architecture allows to prepare the simulation and control the results on a personal computer, as long as a high-powered server is available for performing extensive calculations. The standard XML file format used for the model description and the results of the simulations allows data exchange with a wide spectrum of applications. All aspects of setting up and controlling the simulation process are handled by a user-friendly GUI. Therefore, no programming skills are required. This feature can facilitate the application of the computational approach by experimentalists in neurobiological research. NeuroSim, integrated into the E-Chalk electronic blackboard via a handwriting interface, can be used in a modern educational process.

This thesis is organized as follows. Chapter 2 describes the main principles of computational modelling of neural systems and presents an overview of the modelling at various abstraction levels. Subsequently, the theoretical basis of the detailed modelling approach is presented in Chapter 3. It covers the functional characteristics of the components of a neural system and the principles of signal transmission. Chapter 4 contains a survey of compartmental modelling packages, including a comparison of their most important characteristics.

The main aspects of the client-server architecture, on which the new simulation system is based, are described in Chapter 5. Chapter 6 gives a comprehensive description of all aspects of working with NeuroSim. Chapter 7 presents our own C++ server, which consists of C++ classes of neural elements. An illustration of NeuroSim applied to modelling of real biological systems is given in Chapter 8, presenting the simulation results of the sensory neuron of the mollusk *Aplysia*. Chapter 9 suggests how NeuroSim could be used for educational purposes, showing that NeuroSim, integrated into an e-teaching system, offers an effective tool to be used during neuroscience lectures. A summary of the results and an outlook on further work are presented in Chapter 10.