15   Distributed Approach to Neuroinformatic Data Interchange

Šimon Řeřucha, Václav Přenosil
Masarykova Univerzita, Brno

15.1   INTRODUCTION

Despite the swift development on the field of engineering and artificial intelligence the machine control remains solely a human task. Unlike the machine, the human is much more influenced by the environment – for instance he or she is strongly liable to weariness (fatigue) or accidental loss of concentration. The fact the human operator still remains the weakest link in the human-machine interaction is caused by such factors.

Typical representative of an operator in a human-machine system is a driver of a ground vehicle. The statistics has proven that significant amount of traffic accidents has been caused by insufficient attention paid to the traffic situation by the driver. One of the most common causes of decreased attention is fatigue and the probably most serious consequence is a micro sleep.

The vigilance, the attention and the sleep has been already a focus of interest for neurologists for several decades. One of the most promising ways is searching for relations between humans' biological state and his vigilance. Rather considerable amount of outputs from various research projects on this field has been collected so far, but the research is confronting some practical issues.

The main data sources for such research are the outputs from experiments with human subjects that are often time-consuming and require complex technical equipment. Then the procedure to acquire an amount of data that is enough to perform relevant statistical analysis is often very difficult in comparison with the contribution of the experiment. The trivial solution is to define appropriate technique to store input data for further use.

In this contribution we want to introduce a concept of new neuro-informatics database (called NIDB from now on), that will provide such functionality. This contribution is focused mainly on the specification of functional requirements posed on such system, but a preliminary model is also presented.

The rest of this contribution is organized as follows: Section 'Requirements' describes several main aspects and specify the functional requirements from several points of view. Section 'Preliminary model' briefly introduces a proposed architecture of NIDB. Final chapter mentions other interesting contributions of NIDB and outlines future work on NIDB development.

15.2   REQUIREMENTS

The NIDB is supposed to become a set of tools (also referred to as a platform) useful for collecting, management and interchange of a neuroinformatic data . The basic properties of this data are that it is heterogeneous, in large amounts and sensitive. Individual data items (records) are "bulky" in comparison with sizes of current storages and with transfer speeds of contemporary computer networks. The sensitivity means that the data is treated as personal, therefore is is protected by the law and a special care must be taken while manipulation with

it. The reason to collect and organize the data is to share it among several research groups and enable a further research over it.

15.3   General requirements

From the functional point of view the core of the NIDB will be a specialized Data Base Management System (DBMS). DBMS is an application layer between the physical storage of data and the client application that need to work with that data. For our purposes, a generic DBMS can be decomposed to following three functional layers:

- methods of data storage and data management,
- the DBMS functionality performed on the top of the stored data (called core functionality in this section),
- an interface that allows clients to utilize the DBMS functionality.

There are additional functionality the system must incorporate, especially due to the specific nature of the data (volume, sensitivity):

- ability to work in a distributed environment,
- access control.

These individual components and features are discussed in following separate sections.

Finally, NIDB must comprehend a set of accompanying applications. This could include:

- the user interfaces, suited to the needs of the particular users,
- the conversion tools to match the „standards" used by the hardware manufacturers,
- the interconnections with applications and software environments used for the research over managed data.

15.4   Core functionality

The core functionality is a defined set of operations that the NIDB core is capable to perform. The requests are received from a client through the defined communication interface using defined protocol. The core executes a sequence of operations on the managed data and finally passes a response back to client.

Required set of operation is similar to that performed by traditional (R)DBMSs. The basic tasks remain the same – we need the functionality for:

- data definition and manipulation,
- data retrieval.

Since the managed data of same domain are expected to be stored with different internal structure (e.g. the HW vendors of EEG measuring facilities uses different techniques and formats to store data), such formats must be recognized by the core and the core must be able to provide data item in format that a client application requests.

The most straight-forward and most convenient way is to design a modular architecture with an standardized interface to additional modules that will provide same functionality using different underlying protocols.

Regarding the search operations, the main requirement is to provide a scale of search criteria as wide as possible. The real performance and efficiency is not a primary issue. However, the specific requirements for query operations are dependent on deeper analysis of data items and the relations among them. This goes beyond the scope of this text.

There are another classes of operations beside these required for application processing, usually called „maintenance procedures". Within NIDB core we need these:
- configuration routines,
- data integrity and consistency checks,
- operational statistics and logging.

These procedures are not crucial for the operation of NIDB core, but they are useful during operation. Finally, there are the aspects of access control and distributed working environments that are analyzed later in separate subsections.

15.5 Data management

The data that NIDB will manage fall into following categories:
- descriptive data (meta-data),
- primary (measured) data,
- secondary (derived) data.

The set of „primary data" consists of the data acquired from the experiments (e.g. EEG record, reaction time). The secondary data are the results of analyses over primary data (e.g. correlation of the reaction time and the EEG spectra). The meta-data are data about data within the meaning of DBMS terminology. This terminology has been established in order to avoid ambiguity in the documents regarding the NIDB and neurological experiments.

There are two basic problems that need to be solved within the data management problematic:
- the data definition and manipulation (e.g. storage, importing, describing),
- the data extraction (searching, querying).

The fundamental difference is that the first is needed to be robust and reliable so the database contain relevant data, the latter is needed to be fast and accurate in order to be of use.

Regarding the data manipulation, NIDB will take care of two distinctive planes:
- the basic data (both primary and secondary),
- the relations among them.

So far the basic data are usually stored within the data storage of individual workplaces. Considering the difficult manipulation with a large amount of data we want NIDB to respect present (file-based) structure of the users' data storages – in fact, we want to minimize the data transfers and movements. Another method of primary data storage is an export upon request from already existing information systems aimed at neurology and similar fields. Consequently, both these approaches require a methods to track the changes in the data and to check it's integrity. The demand on modularity and a possibility to choose proper method depending on local conditions is straightforward. It also assume a design of suitable interface between the „storage" modules and the block body of NIDB core.

Individual data items need to be supplied with some piece of information that is not included within the data item. Each data item must have some supplied "envelope", that will hold the information like the origin of the item, the data-format used, etc. It must be ensured so that the "envelopes" are not easily counter-changeable.

From the inner point of view, the items are individual records of some quantities in time (e.g. several tracks of EEG record). For simpler manipulation we need a possibility to append an arbitrary piece of information to individual track, time marker or a period of time. The piece

of information could be represented e.g. by a textual note, reference to another item in NIDB. To achieve this, the information about the internal structure of the item must be present in the „envelope".

As a consequence of such approach we can omit the basic data from the discussion about data handling, because we expect them to be left in current state. We need just „describe" them, so the data-handling problem reduces for meta-data.

A relation can exist between two data items, between a data item and a particular piece of information within another or between two particular pieces information. We need to define a generic method that could handle all these types of relation. Such "relation" can be related to another item, piece of information or a relation as well. A typical example is a relation among primary data set and derived secondary set.

The organization of the meta-data for efficient searching is a superstructure over the meta-data structure established for data manipulation. It will probably bring a demand for a data redundancy and the structure depends on the chosen search algorithm. This problematics is subject to further research.

15.6 Interfaces

There are several classes of interfaces within NIDB mentioned so far:
- user interfaces,
- interfaces between modules within the NIDB core,
- interface between NIDB core and client applications.

The NIDB core will not provide any user-interface, the users will operate a client application that will communicate with NIDB core. The client applications are mentioned later in section „Client applications".

The design of interfaces between modules within the NIDB core is a concern of functional analysis. The only requirement so far is could be called „platform independence" or „interoperability" – we require to allow to develop modules in different environment than the operational part of NIDB core.

The interface between NIDB core and client applications is required to be implemented using any of current „standard" (e.g. XML, SQL) for data exchange. The reason is evident: Since the „standard" facilities are supported in many SW development environments, it will simplify the implementation of client application.

15.7 NIDB in distributed environment

The previous sections indicate that the NIDB is expected to provide a functionality for several (geographically distant) workplaces. The first aspect to resolve is to decide whether to design NIDB as a centralized system or as a specialized network of stand-alone nodes (distributed system, overlay network).

The first approach is simpler to design and implement, but it is apparently inconvenient due to large volume of managed data. It would probably bring the technical problems (e.g. to establish and manage large data storage and connectivity) during implementation and operation and also limit the scalability.

The latter approach yields more problems during the design and implementation process, however it has significant advantages. The most important aspect is that the data will be stored within node of their origin and will be transferred only if another node requests for them. There will be no need of central storage and the data will not be stored more than once unless requested. Since the amount of the data is supposed to be large, the minimization of data transfers will be a substantive benefit.

There are another consequent convenient factors:
- the distributed approach doesn't limit the scalability (as much as the centralized model),
- it allows customization to local condition (on individual nodes),
- it allows to avoid a single point of failure.

We have already specified some functionality that we require from NIDB core. If we assume an instance of NIDB core to act as node within a collaborative network, we need to extend the functionality of NIDB core appropriately.

It means to solve following problems concerning the communication among individual nodes:
- the topology,
- the communication protocol,
- the functional requirements posed on the interconnection.

15.8   Access control

It is necessary so that NIDB will incorporate a support for following three essential aspects of access control:
- Authentication (and Identification),
- Authorization,
- Accounting.

The user authentication is required for the identification and identity verification of the user for the purposes of authorization and accounting. Because of supposed use in different environment NIDB is expected to provide an user-based identification and authentication mechanism.

The aim of authorization mechanism is to define which users are permitted to what with which data.

Finally the accounting have to ensure that every modification in the managed data are clearly linked with an user who performed the operation.

15.9   Client applications

The fundamental requirement posed on the user interfaces is to allow user to effectively utilize the functionality provided by system. The functional and behavioral requirements depend on the purposes of particular tool. At this moment we can just define few classes of applications that will be of use within NIDB:
- the user interface that allow manage, import and modify the data,
- the visualization tools,
- the conversion tools,
- the libraries that provide an interface for another development environments.

15.10   PRELIMINARY MODEL AND CONCEPTS

This section briefly describe a preliminary model of NIDB that comply with the analysis and the requirements summarized in previous sections.

Whole NIDB system will consist of separate collaborating nodes. These nodes will create a virtual network with nodes interconnected in peer-to-peer topology. Each particular node (called NIDB node) will implement the core functionality specified in section Requirements.
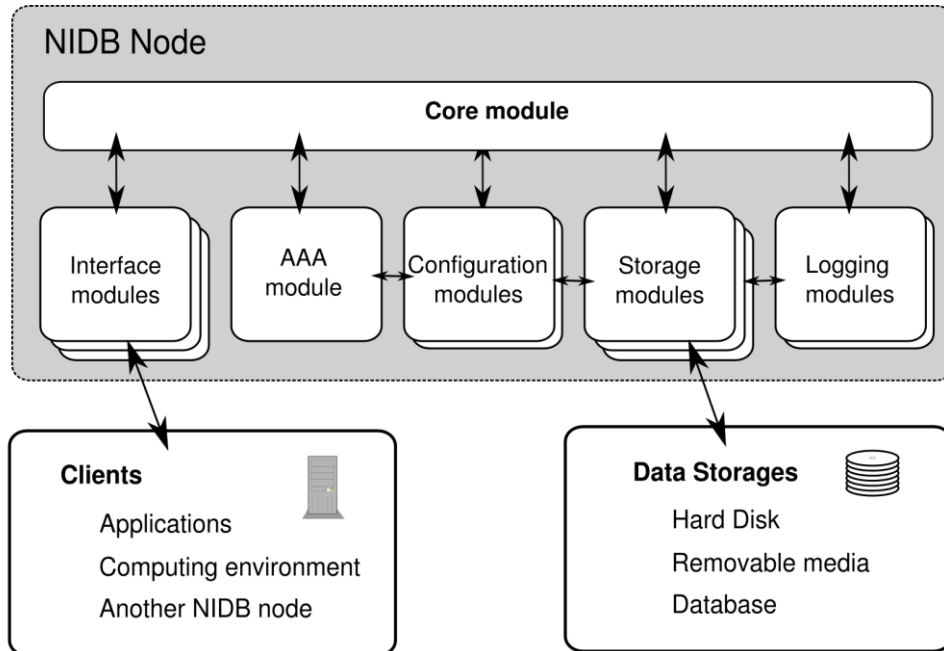

Fig 1.: The preliminary NIDB node model overview

The NIDB core will consist of several modules, where each module will provide a part of functionality – the proposed structure is shown on Figure 1. The central point will be the core module that will control the operation of entire NIDB node. Its main tasks will be launching and managing other modules and routing messages among them.

The AAA module will take care of authentication, authorization and accounting. It will load necessary initial data from selected configuration module and log via one of logging modules.

The set of interface modules will provide the standardized communication interfaces for client applications and another NIDB nodes. Such module will ask AAA module for user verification and route clients' requests to core, that will invoke proper storage modules.

The set of storage modules will perform the operations over physically stored data using various back-ends (e.g. flat files, XML database, relational DBMS).

The configuration and logging modules will enable the NIDB node operators to set up and manage the node itself. Both types will be able to load and/or store configuration parameters and log records using several ways (they can possibly use simplified storage modules).

The nodes are supposed to communicate over IP network using a specialized protocol that will be based on interchange of XML messages (for example XMPP). The particular data items will be available either on-line or off-line (library-like approach).

NIDB will use several cryptographic method for several purposes. A cryptographic hash function will be used to track the changes in the primary and secondary data and to preserve integrity and some digital signature scheme will be used to identify the origin of the data. The user authentication will utilize some public-key cryptosystem to verify the client identification.

15.11   CONCLUSION

The contribution presents a preliminary model of new tool useful for research on the field of neuroinformatics – NIDB. The text is focused on discussion about fundamental aspects of such system and the conclusions are used to draft the architecture of NIDB system. The model of NIDB architecture is described on the plane of functional blocks and basic data flows.

The development of NIDB will continue within the scope of CNNN research activities, currently supported by project ME 949 "The analysis of negative influences on driver drowsiness" in cooperation with other CNNN participants.

References

[1]... Bouchner P.: Driving simulators for HMI research (PhD. Thesis). Faculty of Transportation Sciences, CTU, Prague, 2007.

[2]... Novak M., Faber J., Votruba Z.: Problem of Reliability in Interactions between Human Subject and Artificial Systems (First Book on Micro-Sleeps). Neural Network World – monographs edition. CTU & ICS AS CR, Prague, 2004. ISBN 80-903298-1-0.

[3]... Novak M. - editor.: Neurodynamic and Neuroinformatics Studies (Second Book on Micro-Sleeps). Neural Network World – monographs edition. CTU & ICS AS CR, Prague, 2005. ISBN 80-903298-3-7.

[4]... Novak M., Faber J., Tichy T., Kolda T.: Project of Micro-Sleep Base. Research Report No. LSS 112/01, CTU, Prague, 2001.