An Agent Framework for the Analysis of Streaming Physiological Data

Ruairi D. O'Reilly[1], Philip D. Healy[1], John P. Morrison[1], and Geraldine B. Boylan[2]

[1] Computer Science Department, University College Cork, Ireland
{r.oreilly,p.healy,j.morrison}@cs.ucc.ie
[2] Department of Paediatrics and Child Health, University College Cork, Ireland
g.boylan@ucc.ie

Abstract. An agent framework that assists with the interpretation of streaming physiological data is presented. The framework operates as a module within an existing remote monitoring system for streaming physiological data. Agents complement the remote monitoring system by enhancing the users' view of the signals through the addition of annotations and the creation of derived views of the signals.

1 Introduction

The effective interpretation of physiological signals often requires expertise in a specialised area. For instance, a cardiologist may be required to interpret irregularities in an electrocardiogram (ECG) in order to diagnose a heart condition, while a neurophysiologist may be required to interpret an electroencephalogram (EEG) in order to diagnose a brain condition. A thorough analysis of a physiological signal may therefore require a significant investment in time from a highly trained specialist. Furthermore, the required expertise may not be available on-site or a second opinion may be required.

An ICT system that could reduce time to diagnosis by assisting specialists in the interpretation of a biosignal, or improve the quality of diagnosis through the involvement of offsite experts, would therefore be beneficial. Analysis by offsite experts can be achieved by streaming the biosignal as it is acquired and making it available for interpretation in near-real-time at remote locations. A reduction in the amount of time required by each expert to arrive at a diagnosis is then possible by enhancing the experts' view of the signal with additional information that assists with analysis. For example, EEG data could be enhanced through the display of a list of suspected seizures as determined by a seizure detection algorithm. The result is a reduction in cost (both monetary and in terms of specialists' time) and improvement in patient care through reduction in time to diagnosis and quality of diagnosis.

The live streaming aspect of the ideal system as described above has been implemented through the remote monitoring system described in [1]. Although developed specifically for EEG, the system is general-purpose and supports the

M. Szomszor and P. Kostkova (Eds.): E-Health 2010, LNICST 69, pp. 311–318, 2011.

remote monitoring of a variety of additional biosignal types (e.g., ECG, EOG) in near-real-time. In this paper, an agent framework is presented that complements the remote monitoring system through the enhancement of the signals that are streamed through it. Enhancement is achieved through the addition of annotations or the creation of derived views of the signals.

The framework implements some aspects of a multi-agent system as described in [2], in that a collection of loosely-coupled problem solver entities cooperate to achieve desired objectives that are beyond the individual capabilities or knowledge of each entity. An agent-oriented design was chosen because the intentional stance and the ability to reason about the problem domain that characterise the agent approach closely mimic the process of human interpretation. Also, the independent nature of the various components envisioned in the framework requirements lend themselves to implementation as agents.

The current implementation supports only a subset of the capabilities expected in a full agent implementation. Higher-level agent capabilities, such as social ability, pro-activeness and autonomy in relation to internal state, are absent. This limitation stems from the restriction of the design scope to support only the initial use cases envisioned for the system (signal annotation and derivation). Furthermore, the requirements for the system called for the ability of the agents to rapidly process large volumes of streaming clinical data in a fault-tolerant fashion, ruling out the adoption of an off-the-shelf high-level framework. As a result, a high-performance implementation of a simplified subset of agent behaviour that is tightly-coupled with the existing streaming platform was implemented. However, there are plans to introduce more advanced concepts into the framework in future, allowing higher-level agents with more human-like characteristics to be implemented.

The remainder of this paper is organised as follows: Section 2 provides an overview of related work in the application of agent concepts to health informatics, focusing in particular on the analysis of streaming physiological data. Section 3 gives a brief description of the remote monitoring system. Section 4 describes the objectives, design and operation of the framework along with a description of the agent development process and consideration of the issues of security and data integrity. The implementation of an example agent that derives the aEEG of streaming EEG signals is presented in Section 5. Finally, our conclusions and future work are discussed in Section 6.

2 Related Work

A significant body of work exists on the application of Multi Agent Systems (MASs) to medical problem domains. Medical MASs have been developed in areas such as decision support, clinical knowledge tools, patient history tracking and monitoring solutions. One of the earlier examples of the intended use of intelligent agents for intensive care monitoring is presented in [3]. The agent operates on two levels; the lower level performing data reduction and abstraction tasks and the higher level applies various reasoning rules and reactive diagnosis skills against the data.

Multi-agent decision support systems were also pioneered for specific conditions. One such system, which diagnoses cardiac disorders, is presented in [4]. Agents that incorporate new domain knowledge and methodologies can be incorporated without structural changes to the MAS. One of the agents in the system, denoted CATS (chaotic analysis of time series), uses a methodology based on non-linear dynamics that performs offline analysis of ECG data. An ontology-based intelligent healthcare agent for respiratory waveform recognition was presented in [5]. The agent uses fuzzy matching against an ontology to recognise and classify respiratory waveforms.

The BRIAN remote EEG monitoring system implements limited analysis of streamed EEG data such as blink detection [6]. A proposed extension of the system presented in [7] would allow the system support generic analysis, archiving and dissemination of brain and physiological data.

An agent server for the analysis of physiological datastreams from NICUs is presented in [8]. The agent-based intelligent support system examines streaming physiological data as well as previously acquired physiological data and clinical history data in order to attempt to detect trends and patterns that may indicate the onset of clinical conditions.

The Artemis project [9] involves the acquisition of large volumes of streaming physiological data from the ICU and the application of clinical rules to these datastreams via a stream computing platform. Clinical rules are defined as a specification as detailed within an existing clinical guideline, defined anecdotally by a clinician as part of clinical research or proposed through a set of datamining conditions in the physiological datastreams, laboratory results, and observations of a patient. A framework that supports the acquisition, real-time processing storage and datamining of physiological data streams is presented in [10]. This system analyses streaming physiological data acquired in the ICU to detect changes that might be onset predictors for selected conditions.

3 Remote Monitoring

A web-based remote monitoring system was presented in [1] that allows live recordings to be viewed in near-real-time[1]: while acquisition is ongoing. Users are presented with a continually updating view of the entire recording, allowing them to view new data as they arrive or review data transferred earlier. The system is comprised of three components: an upload application, a data server and a viewing application. The upload application is used to transfer data from the acquisition location to the data server. The data server acts as a repository of recorded data, and provides the interfaces necessary for both humans and software to interact with the data. Although the system as a whole is implemented using a combination of Java, Actionscript and C++, the data server (which is the focus of this paper) is implemented exclusively using Java.

[1] If sufficient bandwidth is available, the maximum delay between acquisition and remote display is 60s.

A typical use case of the system is as follows: A patient is identified as having a suspected neurological condition. Acquisition of EEG data commences, the resulting EEG data is saved to a file server. The file server uploads the EEG data file. Any data present in the file are immediately uploaded to the data server. The upload application then periodically determines whether additional data have been added to the file. If so, the new data are uploaded. Once the transfer of data has begun, the neurophysiologist can log into the data server and view the data streaming from the recording. The neurophysiologist can analyse the EEG data and issue a clinical report containing his/her findings.

4 Agent Framework

The objective of the agent framework is to enhance the remote monitoring system described above by allowing a user-configurable set of agents to perform analysis on physiological signals as they are streamed through the system. The output of the agents would assist with the analysts' interpretation of the data being streamed and hence reduce time to diagnosis. Two agent types were considered: agents that annotate recordings and agents that derive one signal type from another.

The following design requirements were identified:

1. **Configurability.** The set of available agents should be user configurable. The framework should be capable of determining which of the available agents are applicable to a particular recording. For example, a blink detection agent should not be applied to recordings that do not contain EOG signals.
2. **Stream processing.** The framework should process data as they arrive at the server. The framework should detect when a new data segment has been appended to a recording and notify any agents attached to the recording.
3. **Persistence.** Agent state will be recoverable in the event of a server shutdown/restart, possibly due to a fault.
4. **Error handling.** Errors generated by agents should be identified, gracefully extracted from the system, and reported to the user.
5. **Security.** Although access control to the recordings available in the remote monitoring system is outside the scope of the agent framework, agents should not be permitted to perform actions that could compromise system security.
6. **Data integrity.** Ensuring data integrity requires the enforcement of global constraints. The actions of individual agents may not violate these constraints.
7. **Resource management.** Agents may be created dynamically depending on the intended application. However, for reasons of efficiency and speed agent processing will be implemented using a specific, predefined, number of threads. Memory will be managed by storing only active agents in memory; inactive agents will have their state written to persistent storage to be reconstructed when required.

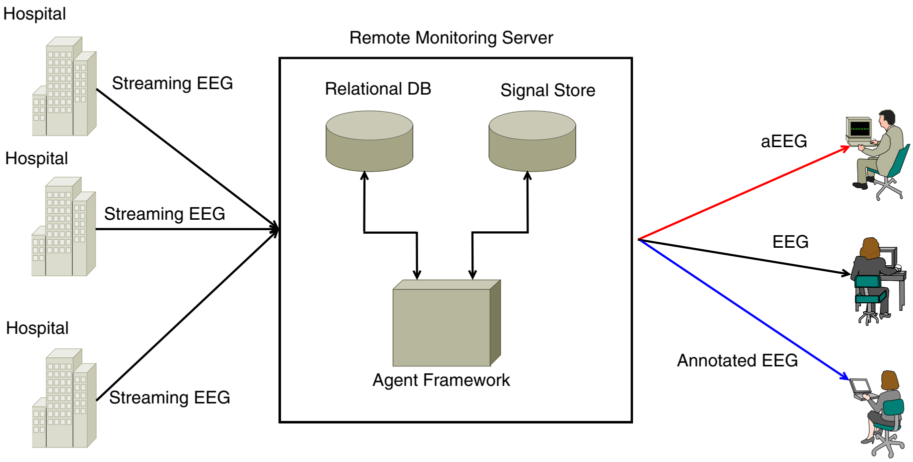

Fig. 1. Conceptual view of the agent framework applied to a teleneurophysiology application

8. **Annotation.** The framework will facilitate the creation of agents that annotate interesting segments of recordings, e.g., suspected seizures.
9. **Signal derivation.** The framework should facilitate the creation of agents that create new signals based on existing ones, e.g., aEEG from EEG.
10. **Code reuse.** Code duplication in agent implementations should be avoided by providing well-tested libraries to facilitate tasks that agents are expected to commonly perform, such as signal filtering and annotation based on threshold values.

4.1 Architecture

The agent framework operates as a module within the remote monitoring server application. Data is stored by the system in two locations: a relational database and a signal store comprised of a collection of files containing signal data. Related signals are grouped together into recordings. Any modifications to the database (such as the arrival of new recording data) result in events that can be subscribed to by any of the modules within the system.

Agents within the agent framework perceive their environment through this events system, and effect changes to their environment through modifications to the database and the addition of data to the signal store. These database changes in turn trigger further events which can be acted upon by modules other than the agent framework. For example, the creation of a new recording annotation by an agent would result in an event that would be detected by the Comet (server push) module, causing the new annotation to appear on the screens of any analysts viewing the relevant recording. Furthermore, agents are capable of cooperating in pursuit of a common goal. For example, the output of the aEEG agent described in Section 5 could in turn be processed by another agent that implements aEEG seizure detection.

The agent manager is the module responsible for providing the agents' execution environment. This involves: managing the set of configured agent factories; creating agent instances as needed; managing agent persistence; subscribing to system-wide events and forwarding events of interest to the appropriate agents; and executing the processing tasks generated by agents. Agent factories are used to instantiate agent instances on behalf of the agent manager. Agent instances are persisted and reconstructed as required using the Java Persistence API. Event forwarding is performed by determining the set of agents, if any, that a system event applies to, then reconstructing those agents from the persistent store and forwarding the event to them. A task executor with a configurable number of threads is used to perform signal processing on behalf of agents. Two possible solutions to the issues of security and data integrity that were considered are sandboxing and code review/JAR signing. The sandboxing solution would execute agents in a restricted environment using a dedicated Java security manager [11] to mediate access to system resources. The JAR signing restriction is only enabled on production builds of the remote monitoring server in order to facilitate the development and testing of agent implementations before they are submitted for review.

5 Example Agent: aEEG

Amplitude-integrated electroencephalogram (aEEG) is a derived signal from a reduced EEG and is used as a method for the continuous monitoring of a patient's brain. The method is based on filtered and compressed EEG that enables evaluation of long-term changes and trends in electrocortical background activity by relatively simple pattern recognition. It is an extension of the cerebral function monitor which was developed in the late 1960s to monitor adults in the intensive care unit [12].

In term infants aEEG is an excellent method for evaluating cerebral function and cerebral recovery after hypoxic-ischemic insults such as perinatal asphyxia and apparent life-threatening events [12]. aEEG background activity within six hours of birth in term newborns with hypoxic ischaemic encephalopathy has been shown to be predictive of later neurological outcome [13]. Given the remote monitoring system's primary purpose as a neonatal neurophysiological monitoring tool, an agent for derivation of aEEG was considered the most useful first application of the framewok.

The aEEG approximation implemented by the agent uses a 9^{th} order digital IIR least P-norm bandpass filter designed to pass energy in frequencies from 2 to 15Hz. The nonlinear mapping function is defined as follows:

$$F(\text{eeg}(t)) = \begin{cases} |\text{eeg}(t)| & |\text{eeg}(t)| \le t_h \\ t_h \log\left(|\text{eeg}(t)| - [\log(t_h) - 1]\right) & |\text{eeg}(t)| > t_h \end{cases} \tag{1}$$

where log denotes the natural logarithm and the threshold, t_h, is chosen as 20μV. The nonlinear mapping is then averaged across 3s and scaled:

$$\text{aEEG}(t) = \frac{1}{3} \int_{t-1.5}^{t+1.5} 2F(\text{eeg}(\tau))d\tau. \tag{2}$$

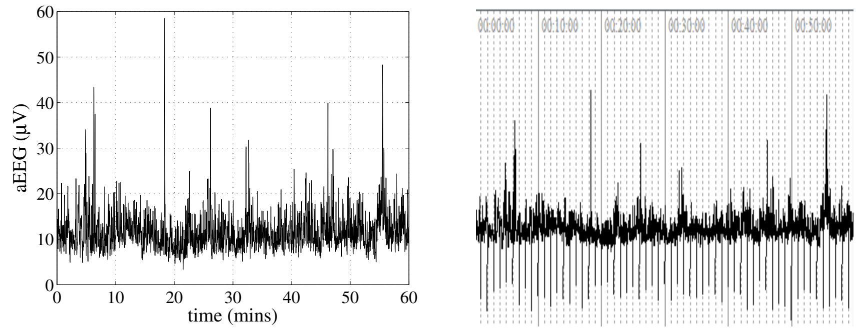

Fig. 2. aEEG visualisation of two channels F4-C4 generated by MATLAB and the aEEG Agent

The algorithm as described above was initially implemented and tested using MATLAB. Tasks produced by the agent apply the algorithm to segments of recording data. The result is an incremental application of the algorithm to streaming signal data. The MATLAB implementation was used to verify the agent's output. The persistent agent state preserved between task invocations consists of the buffered filter inputs and outputs, and the previous 2s of output from the nonlinear mapping function. Unit testing was used to ensure that the agent state was persisted and reconstructed correctly.

6 Conclusions and Future Work

The agent framework presented here complements the remote monitoring system by aiding analysts with the task of interpreting streaming physiological data. By directing the attention of the analyst to specific features in the signal, time is saved. This may lead to faster time to diagnosis and hence an improvement in patient care.

Two new agents are currently being designed along these lines. The first would decide on the most informative set of derived signals for a given stream based on the previous behaviour of human agents in manually requesting derived signals. The other would be responsible for approving or declining an agent request for human consultation of data, with the answer inferred in an ensemble-like manner by querying a number of seizure detection agents.

Acknowledgements The authors wish to gratefully acknowledge the support of Science Foundation Ireland (through grant 08/RFP/CMS1733), The Neonatal Brain Research Group (in particular Dr Nathan Stevenson for his assistance with the implementation of the aEEG agent) and The Boole Centre for Research in Informatics.

References

1. Healy, P.D., O'Reilly, R.D., Boylan, G.B., Morrison, J.P.: Web-based remote monitoring of live EEG. In: Proceeding of the 12th International Conference on E-Health Networking, Applications and Services, Lyon, France (July 2010)
2. Flores-Mendez, R.A.: Towards a standardization of multi-agent system framework. ACM Crossroads 5, 18–24 (1999)
3. Hayes-Roth, B., Uckun, S., Larsson, J.E., Gaba, D., Barr, J., Chien, J.: Guardian: A prototype intelligent agent for intensive-care monitoring. Artificial Intelligence in Medicine 4, 165–185 (1992)
4. Hudson, D., Cohen, M.: Use of intelligent agents in the diagnosis of cardiac disorders. Computers in Cardiology 29, 633–636 (2002)
5. Lee, C.-S., Wang, M.-H.: Ontology-based intelligent healthcare agent and its application to respiratory waveform recognition. Expert Systems with Applications 33, 606–619 (2007)
6. Fritschy, J., Lucia, M.D., Holder, D.: Web EEG reader for remote reporting and automatic detection of normal and abnormal patterns in EEG. Clinical Neurophysiology 117 (2006)
7. Holder, D., Bayford, R., Fritschy, J., Gilad, O., Kaube, H., Binnie, C.: Development of generic software for analysis, archiving & internet dissemination of brain and systems physiological data. University College London, Tech. Rep. (2004)
8. Foster, D., McGregor, C.: Design of an agent server for neonatal analysis and trend detection. International Transactions on Systems Science and Applications 1, 27–34 (2006)
9. Blount, M., Mcgregor, C., Ebling, M.R., Eklund, J.M., James, A.G., Percival, N., Smith, K.P., Sow, D.: Real-time analysis for intensive care. IEEE Engineering in Medicine and Biology Magazine (2010)
10. Mcgregor, C., Sow, D., James, A., Blount, M., Ebling, M., Eklund, J.M., Smith, K.: Collaborative research on an intensive care decision support system utilizing physiological data streams. In: American Medical Informatics Association 2009 Symposium, pp. 1124–1126 (2009)
11. Herzog, A., Shahmehri, N.: Using the Java sandbox for resource control. In: Proceedings of the 7th Nordic Workshop on Secure IT Systems (NordSec), pp. 135–147 (November 2002)
12. Hellstrom-Westas, L., Rosen, I., de Vries, L., Greisen, G.: Amplitude-integrated EEG classification and interpretation in preterm and term infants. NeoReviews 7(2), 76–87 (2006)
13. Boylan, G., Burgoyne, L., Moore, C., O'Flaherty, B., Rennie, J.: An international survey of EEG use in the neonatal intensive care unit. Acta Paediatrica 99, 1150–1155 (2010)