# OpenAdap.net: Evolvable Information Processing Environment

Alessandro E.P. Villa[1,2] and Javier Iglesias[1,2]

[1] Grenoble Institute of Neuroscience GIN, Centre de Recherche Inserm U 836-UJF-CEA-CHU, Neuroheuristic Research Group, University Joseph Fourier, Grenoble, France
[2] Information Systems Department, University of Lausanne, Switzerland
{Alessandro.Villa,Javier.Iglesias}@ujf-grenoble.fr
http://www.neuroheuristic.org/

**Abstract.** OpenAdap.net is an Open Source project aimed at breaking the barriers existing in the flow of information access and information processing. The infrastructure makes it possible the dissemination of resources like knowledge, tools or data, their exposure to evaluation in ways that might be unanticipated and hence support the evolution of communities of users around a specific domain. The architecture is designed by analogy with a virtual distributed operating system in which the dynamic resources are presented as files in a structured virtual file system featuring ownership and access permissions.

## 1 Introduction

The Cyberspace has become the main site of information exchange and its extension in terms of bandwidth is opening the way equally to all media of information. However, most of this information flow still remains associated to a 'declarative' approach, i.e. associated to the knowledge of 'what', as illustrated by the access of web portals built around search engines of any kind. Internet has still a huge unexploited potential in applying the 'procedural' approach, complementary to the declarative one, based on the knowledge of 'how'. The procedural approach necessarily relies on a *trial-and-error* paradigm that may be achieved more efficiently when the results are shared within a community of agents, independently of their human or non-human nature.

The benefits in terms of education, business developments, market diversification and creation of employment generated by the fast circulation and ease of access to the results of trial-and-error procedures clearly emphasizes the importance of the society, beyond the community of agents. Knowledge sharing leads to share the methods of information processing and the socialware [Hattori et al., 1999] becomes the vehicle for user adaptive collaboration support. Despite the immediate worldwide availability of new information published at Internet, the tools of information processing (e.g., software programs and methods of data analysis developed and provided by research laboratories

F. Masulli, S. Mitra, and G. Pasi (Eds.): WILF 2007, LNAI 4578, pp. 227–236, 2007.
© Springer-Verlag Berlin Heidelberg 2007

and businesses) are generally released in small niches delimited by the readers of specialized journals as well as by specific software environments and computer platforms.

Potential meaningful contributions may remain unexploited, or even forgotten, within the same field of competence and their availability is severely restricted in particular for cross-fertilized applications to other fields. This situation often leads to the delayed re-invention of similar (if not the same) methods of information processing and, to some extent, the re-discovery of the same knowledge. Despite the fact that the development of event-delivery middleware based on publish-subscribe communication allows high-performance computing [Eisenhauser et al., 2006], the development of added-value activities created by knowledge sharing is not yet widespread. The current situation tends to create an artificial 'distance' between the authors and developers of original information processing methods (i.e., the providers of the 'how' knowledge) and the users (i.e., the testers of *trial-and-error*) with the risk of erroneous applications of the methods and false evaluation of its value.

In the OpenAdap.net structure we have designed brokers playing a pivotal role, being responsible for dynamically decomposing and routing end-user tasks to appropriate resource sharers for execution. The negotiation between brokers (and workers) is inspired by the way how the brain processes information. In the present paper we present an event timing driven rule for adapting the connections between brokers belonging to the same network of community-related applications.

## 2 Functional Structure

OpenAdap.net is a distributed system composed by three types of interconnected components: brokers, workers and OAN-aware applications (see Figure 1). A broker is a process permanently running on a server in charge of managing a community of users and dispatching tasks and results on their behalf. Workers are processes shared by community members in charge of giving secured distant access to contributed resources like programs or data. OAN-aware applications are pieces of software (standalone applications, web portals, command line tools, etc.) providing access for an end-user to the community shared resources through identified connections to a broker.

The components are running on a network of computers, each of them defined by their specific CPU architecture, operating system (OS), amount of memory and available programs. Their resources are partially shared in a dynamic way. OpenAdap.net is designed by analogy with a virtual distributed OS in which all the resources are presented in a structured virtual file system. Using this high-level paradigm, resources are assigned to files in the file system tree. Security is enforced through ownership and access permissions defined on a per-file basis. OpenAdap.net goes beyond a traditional OS as the configuration is highly volatile. The file system structure and contents are the result of the constant runtime aggregation of several sources of information: the user database, the

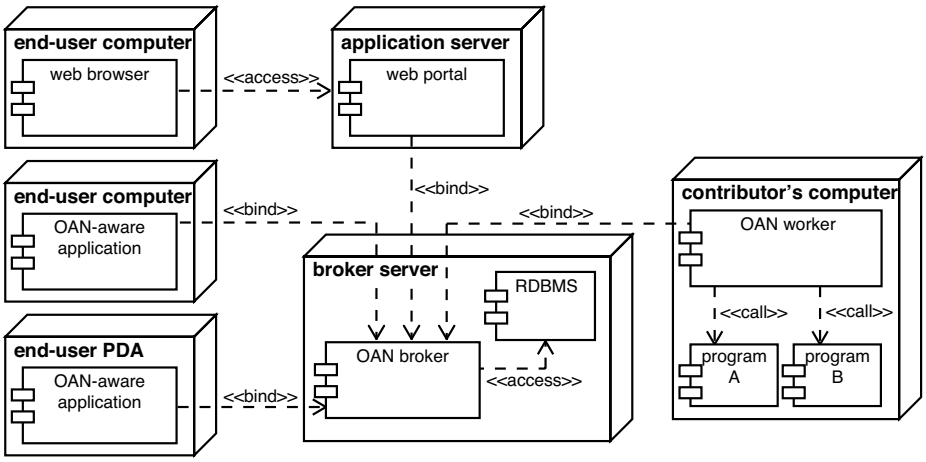


**Fig. 1.** Deployment diagram for the OpenAdap.net components. Boxes represent different computers, rectangles represent processes, and arrows represent inter-process communications over Internet.

inter-process message queues status, the worker status, etc. A dedicated URL name scheme (oan://) is proposed to identify each file in a transparent and interoperable way.

The Java 2 Platform was chosen for the implementation of the project, based on portability and platform neutrality requirements. Brokers, workers and OAN-aware applications are loosely coupled, distributed components that asynchronously communicate through a message-oriented middleware (MOM) as defined by the Java Message Service (JMS) API. The use of established industrial standards such as JMS allowed reusing existing Open Source software as a starting point for the project. It can also be expected that JMS implementations available for other platforms will allow applications written by third parties to connect to OpenAdap.net brokers and interoperate seamlessly [Laufer, 2005].

For OpenAdap.net the security is addressed transversally by defining rules on how to handle data at any point in the system, providing the required profiles for identification and authentication of the entities, and offering strongly encrypted connections when communicating through the Internet. Standards and models of the field were applied from the start of the project [Zhang et al., 2005].

## 3 Internet Application

### 3.1 Software Architecture

OpenAdap.net stands in the area of complexity and aims at providing a distributed environment where tools of all kinds (applications, data, knowledge, ...) can be accessed transparently via the Internet. At this time, three architectures are used in this field: Grid, Web-services (WS), and Peer-to-Peer (P2P). The

peculiar strongholds of these architectures are briefly described here and synthesized in Table 1.

**Grid:** *"Each user has a large dataset to manipulate with one application distributed on a set of computers."* The problem addressed by the Grid is to distribute the processing of large data sets. The required application is copied to distinct computers over a network with each treating a portion of the data. The results are recomposed from all the partial results at the end of the treatment. End-users have control on both data and applications, but little information on remote execution hosts.

**Web-services:** *"Many users exploit the same services permanently provided through a centralized authority."* Web Services provide a secured and controlled access to applications, usually to collaborators in the same company or institution. The goal is to provide distributed access to the same references, databases or applications. The system is articulated around a repository where the service interfaces are published in a self-contained form. The architecture is rather static, as services seldom change and are expected to be permanently available. End-users have no control over the applications and little information on remote execution hosts.

**Peer-to-Peer:** *"Many users exchanging pieces of data in an unsupervised way."* P2P (peer-to-peer) systems address the data-sharing problem. Copies of the applications installed on end-users computers keep open connections from one computer to peers, forwarding queries and results back and forth until a match is found somewhere on the network. The architecture is open and competing implementations coexist on a self-organized network. End-users have control over their data and information on the peer hosts. It is interesting to note that end-users tolerate incredibly poor service quality and that this architecture raises social as well as technical issues.

OpenAdap.net (OAN) falls somewhere between these three architectures exploiting several of their interesting aspects, but with the intention to address a two-way problem: To provide to a community of users in the same domain a means to interchange their resources in an open, dynamic and secured way and to provide to a community of users the access to the exploitation of information processing solutions contributed by users belonging to other communities. End-users have control over their data, but do not need to manage the resources, nor do they have complete information on remote execution hosts. Collaboration within the OpenAdap.net network allows the dynamic integration of these resources, possibly yielding new or previously unexpected composite resources. This can be summarized as follows: *Many users interchanging resources (data, applications, knowledge in general) dynamically provided by interconnected domain-oriented brokers.*

Individual users of OpenAdap.net can be classified as either contributors of shared resources, or end-users of such. People who develop and provide new methods of data analyses are able to share their contribution and people interested in processing their own data or access data stored elsewhere (e.g. in a community database) can extract the results of their analysis. In addition to

**Table 1.** Comparison of OpenAdap.net with the other approaches over Internet

| | Data treatment istribution | Hardware resource allocation | Hidden execution hosts | Application sharing | Published application interface | Data sharing | Highly dynamic system | Transparent user/resource connection |
|---|---|---|---|---|---|---|---|---|
| Grid | × | × | × | | | | | |
| WS | | × | × | × | × | | | |
| P2P | | | | | × | × | × | |
| OAN | × | × | × | × | × | × | × | × |

individual use, OpenAdap.net is open to exploitation by networked organisations and alliances, by providing an infrastructure with which they can share and integrate resources and develop new business solutions and opportunities.

It is important to emphasize that users' privacy is as important as contributors traceability. The information sent by an end-user for manipulation by a contributors resource is anonymized, despite the fact that all the transactions are identified and that the activity is tracked like in any computer system. In contrast, meta-information concerning the contributed resource like program authorship and version number should be made available to the end-user, as a mark of diligence to the contributor, but also to point the responsibilities in the information processing chain and enhance quality assessment and reproducibility. Each task submitted to the system is returned to the submitter with a record attesting where the information travelled, which processes were applied and who was responsible for the computers and the programs.

Contributors will keep complete control over their authorship as well as the source and binary codes for the software. At the same time, they will be responsible for maintaining and checking the quality of the results that the community will use and validate.

### 3.2 Demonstrator Testbed

In the last few years, we have been developing, running and testing a prototype for OpenAdap.net. The concept proof was checked in a preliminary study aimed at the analysis of multivariate time series corresponding to spike trains, i.e. time series of neuronal discharges [Villa et al., 2001]. In 2006, the middleware was reimplemented from scratch (Figure 2) based on the experience and remarks we have acquired along the prototyping phase [Villa et al., 2006]. Several applications related to the preliminary study have raised interest in the Neural Coding community and are in production via the website `http://www.neuralcoding.org`. This community is composed by academicians, students and researchers who investigate the neural coding schemes within the brain.

OpenAdap.net being an Open Source project released under the GPL and LGPL licences, contributions are welcomed from developers and interested professionals. This is encouraged by the availability of publicly accessible development tools like a version control system, mailing lists and a bug tracker. There is enough experience on the impact of such coordinated and distributed development scheme on the quality of the resulting software to embrace it.

During the prototyping phase, the need appeared to have a portable and user-friendly tool that could provide a fast glimpse on the numerical output of unrelated programs. We searched for a generic front-end for plotting a wide range of graphics obtained through the web, and none could be found that was able to run on multiple platforms without requiring complicated installation procedures, and capable of producing journal-quality graphics output for free. XY-Viewer is a Java application that is being developed for that purpose as a by-product of the main project, featuring a dedicated XML data format. This standalone application can be launched directly from the OpenAdap.net portal to visualize files in the appropriate format that can be produced with the help of libraries made available to interested contributors in C and Java under the LGPL licence.

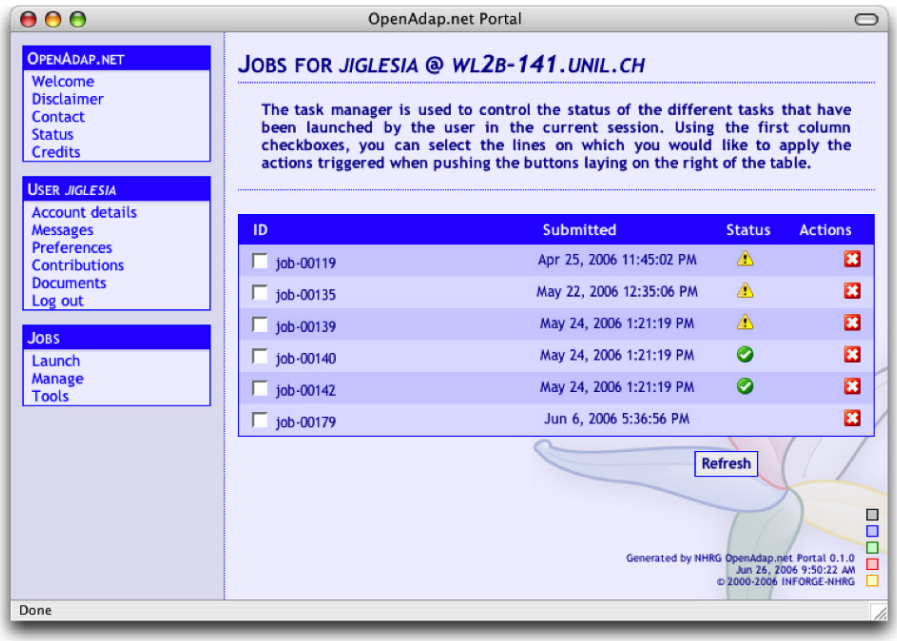


**Fig. 2.** Screenshot of the OpenAdap.net web portal using Mozilla Firefox on Apple MacOS X (as of October 2006 - beta version)

## 4 Inter-Broker Connection Dynamics

We consider that brokers handling the same data formats belong to the same network of community-related applications. A key element consists in making brokers adaptive and dynamically interconnected into an OpenAdap.net network (like a neuronal network) [Iglesias et al., 2005]. These brokers form a dynamic network of inter-brokers connections characterized by a connection strength. Internally, brokers are responsible for decomposing and routing end-user tasks to

appropriate workers for execution. The requests for resources will be processed and dispatched among the components of the system following a set of learning rules dynamically modifying the routing according to, for example, the computing load generated by specific tasks, availability of the resources, or the number of accesses. We introduce a particular feature based on a critical window for modification of the strength of inter-brokers connections incorporating a simple event timing driven model.

Let us assume that any connection between brokers $j$ and $i$ is characterized by a characteristic value of its strength $w_{ij}$ that may depend only on the type of interconnected brokers. A state variable $X$ is introduced to model the modification rule governing the strength of the connection. It is assumed *a priori* that $X$ has $N$ equilibrium configurations (attractor states) $X_1 < X_2 < \cdots < X_N$ to which it will decay with time constant $1/\tau$ in the absence of any requests received by brokers $j$ and $i$. It appears that $X(t)$ is governed by the family of equations

$$\tau \frac{dX}{dt} = -(X - X_k)\,, \qquad B_{k-1} < X \leq B_k\,, \quad k = 1, \cdots N \tag{1}$$

in which $-\infty = B_0 < B_1 < B_2 < \cdots < B_{N-1} < B_N = \infty$ are user-defined boundaries of attraction satisfying $B_{k-1} < X_k < B_k$ for $k = 1, \cdots, N$. For example one can set $B_k = (X_k + X_{k+1})/2$. At any time $t$, the connection strength $w_{ij}$ is determined by the state to which $X$ is attracted.

An event timing driven rule defines how the current value of $X$ is changed by the arrival of requests to broker $i$, by the arrival of requests to broker $j$ and, in particular, through the correlation existing between these events. On the generation of requests on broker $i$, the state $X$ is *incremented* which is a decreasing function of the elapsed time from the previous requests received by the interconnected brokers. The rule used here is

$$\Big[\, X \,\Big] = X_{\mathrm{i-j}}\, e^{-\delta t/\tau_{\mathrm{i-j}}} \tag{2}$$

where $[\,X\,]$ indicates the jump in $X$, $X_{\mathrm{i-j}}$ is the magnitude of the jump in $X$ occurring when broker $i$ receives a request after broker $j$ received the same request, $\delta t$ is the interval between the last requests received by the two interconnected brokers, and $\tau_{\mathrm{i-j}}$ is the decay constant of the reinforcement of the link.

Similarly, when the order of the requests received by the brokers is reversed in time, the state $X$ is *decremented* which is likewise a decreasing function of the elapsed time of the last requests received by the two interconnected brokers. The rule used in this case is

$$\Big[\, X \,\Big] = -X_{\mathrm{j-i}}\, e^{-\delta t/\tau_{\mathrm{j-i}}} \tag{3}$$

where $\tau_{\mathrm{j-i}}$ is the decay constant of the weakening of the link. In the example illustrated by Fig 3 all connections are initially set at level $[X_3]$. This figure illustrates also the fact that after a jump of $[X]$ occurred at time $t$, the real-valued variable $B_i j$ is reset to $B_i j = (X_k + X_{k+1})/2$.

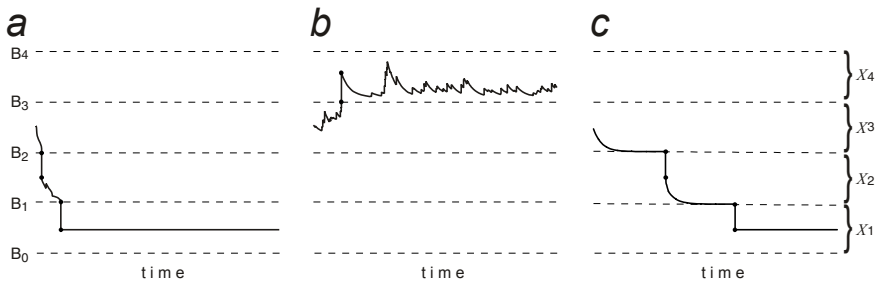


**Fig. 3.** Dynamic change of inter-brokers connectivity strength. ***(a):*** fast weakening of inter-btokers connections down to its minimal level; ***(b):*** reinforcement of inter-brokers connection with an increase in strength that is stabilized on the long term; ***(c):*** example of a link that is neither strengthened nor weakened, but whose strength decays down to its minimal level $[X_1] = 0$ according to the time constant $\tau$.

The rules themselves will evolve and optimize in an unsupervised manner, thus allowing the emergence of unexpected dynamics. In that sense, the required negotiation between brokers (and workers) may be compared to agent interaction. The OpenAdap.net network will also be able to self-adapt via learning processes derived from neural and artificial life learning. Such learning might result in new broker-broker connections, reassessments of the value of such connections, specialisation or generalisation of broker behaviour, etc.

## 5 Discussion

This paper has presented the main features of OpenAdap.net, which is an intelligent network infrastructure supporting the use of shared resources, such as data, knowledge, tools, expertise, etc. aimed at providing the latest published versions by the original authors and developers to a broad audience over Internet. OpenAdap.net is an Open Source project designed as an open architecture. Anyone is invited to contribute their own enhancements to the system, via a set of libraries and tools provided by the consortium. Such an initiative is aimed at increasing the impact of the project with all the contributions that competent contributors will imagine and realize within their specific domains.

Individual users can be classified as either contributors of shared resources, or end-users of such resources. End-users are provided with the ability to browse and apply shared resources, and dynamically compose and integrate existing resources to leverage new research insights. From the viewpoint of contributors, OpenAdap.net makes possible the dissemination of resources, and their exposure to application and evaluation by a broader user community. The support for broader evaluation of programs and data sets is particularly important in the

research arena, as is the ability of other researchers to reproduce computational results [Villa et al., 2006].

The ability to tackle a scientific problem from a new perspective relies on both the past experience and new skills adopted by an individual. This is the feature of the *trial-and-error* paradigm and characterizes the 'procedural' approach (to know 'how') vs. the 'declarative' approach (to know 'what'). The OpenAdap.net project is based on the collaboration between information scientists, electronic engineers, computer scientists and neuroscientists having diverse scientific interests and very specialized backgrounds. We feel that such a transdisciplinary approach is a necessary way for the achievement of real advances in producing impacts in the Information Society Technologies. A key point for the success of OpenAdap.net dissemination is the build-up of Communities who share common data formats tailored to their interest. In the prototype that we implemented the multivariate time series format is well known and accepted by leading institutions and researchers in the field.

The adaptive and behavioural models for the broker implementation represent major innovations of the OpenAdap.net project. This is definitely a novelty and a plus to the existing architectures for distributed environments like grids and web services that points out the project expected income to the networked computing field. Pushing existing paradigms like neuronal network inspired learning rules for the adaptable information processing or the operating system paradigm for the overall communication layout, and the lessons learned for 10 years on the self, dynamically, openly organized content on the web are key aspects of the OpenAdap.net philosophy and architecture for resource sharing.

When completed, the OpenAdap.net project network will be able to self-adapt via learning processes that could give rise to modifiable broker-broker connections, specialisation or generalisation of broker behaviour, etc. Thus, the nonlinear dynamics that will emerge from our approach makes OpenAdap.net closer to the complexity of a living organism and the socialware will benefit of the full range of advances achieved in the fast-growing fields of cognitive computation and computational intelligence.

## References

[Eisenhauser et al., 2006] Eisenhauser, G., Bustamante, F.E., Schwan, K.: Publish-subscribe for high-performance computing. IEEE Internet computing, 10(1), 407 (2006)

[Hattori et al., 1999] Hattori, F., Ohguro, T., Yokoo, M., Matsubara, S., Yoshida, S.: Socialware: Multiagent systems for supporting network communities. Comm. ACM 42(3), 55–61 (1999)

[Iglesias et al., 2005] Iglesias, J., Eriksson, J.L., Pardo, B., Tomassini, M., Villa, A.E.P.: Emergence of oriented cell assemblies associated with spike-timing-dependent plasticity. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3696, pp. 127–132. Springer, Heidelberg (2005)

[Laufer, 2005] Laufer, K.: A hike through post-ejb j2ee web application architecture. Computing in Science and Engineering, 7(5), 808 (2005)

[Villa et al., 2006] Villa, A.E.P., Iglesias, J., Ghernaouti-Helie, S.: openadap.net: A collaborative sharing environment. In: Exploiting the Knowledge Economy: Issues, Applications, Case Studies, pp. 753–760. IOS Press, Amsterdam (2006)

[Villa et al., 2001] Villa, A.E.P., Tetko, I.V., Iglesias, J.: Computer assisted neurophysiological analysis of cell assemblies. Neurocomputing 38-40, 1025–1030 (2001)

[Zhang et al., 2005] Zhang, X., Chen, S., Sandhu, R.: Enhancing data authenticity and integrity in p2p systems. IEEE Internet computing 9(6), 429 (2005)