

Loosely-Tied Distributed Architecture for Highly Scalable E-Learning System

Gierłowski and Nowicki
Gdansk University of Technology
Poland

1. Introduction

Vast majority of modern e-learning products are based on client-server architecture and utilization of web-based technologies (WBT). Such approach permits easy creation of e-learning systems that do not require a complex, operating system dependant client software. All possible mechanisms are moved to a system's server, while client side software is severely reduced and implemented by use of operating system independent WBTs (thin-client architecture). That way any user with a decent version of any popular operating system is able to access learning resources and services, providing operator of the system with a widest possible user group. Overall costs of system deployment are also significantly reduced by eliminating a need for a client software installation.

Unfortunately there are also drawbacks of such solution. Because of the majority of mechanisms are located on the server, its usage levels trend to build up quickly leading to limited system scalability. Moreover, WBTs utilized to create thin-client part of e-learning system, depend on constant, reliable network connectivity to system's server. Any disruption of such connectivity can lead to lack of service, client instability or even worse - presentation of malformed results, which is unacceptable in didactic tasks. To make things worse, thin-client element of the system cannot include advanced error handling and recovery mechanisms.

While web-based thin-client architecture works well in case of uncomplicated e-learning content distribution, its limitations will negatively impact functionality of more sophisticated systems (such as systems utilizing didactic simulation or providing advanced knowledge assessment capabilities) and are not prepared to handle limited/unstable network connectivity scenarios.

In the following chapter we describe a novel, distributed e-learning system architecture. It still utilizes web-based technologies, but its design differs radically from currently popular e-learning solutions which rely almost exclusively on thin-client architecture. In our design we employed loosely-tied distributed system architecture, full-client approach, strict modularity, and our original communications package called Communication Abstraction Layer (ComAL) - specifically designed to support communication functions of e-learning systems in diverse network conditions (including fully offline environment).

Instead of reducing client software sophistication and moving all possible mechanisms to a server part of a system, we propose to divide its functionality into self-sufficient blocks (called modules) and define methods of communications between them. Such approach gives us a number of unique advantages: system can be deployed gradually and not all of its elements need to be deployed (partial deployment); modules (which provide server or client functions) can be connected and disconnected from the system at will without disrupting its operation; the system is also resistant to network environment changes and can function as a managed entity even when there is no network connectivity whatsoever (offline scenario).

Furthermore, system scales very well and can support setups starting from a single client (without a server part), to big corporations consisting of many independent organizational units, hundreds of system's servers (each running a chosen set of modules) and thousands various clients. Integration with third party products is also easy, and third party applications can be used as system's clients or even integrated in its internal data paths.

We present our solution using our original knowledge assessment system, created according to this new architecture, as an example. The system was deployed and tested in production environment on Faculty of Electronics, Telecommunications and Informatics, Technical University of Gdansk with great success, reducing staff workload and increasing efficiency of didactic process. Included tests results also show system's versatility as the system was deployed in environments of classroom, remote and blended learning.

In the following section of the chapter we describe our motivation for creating a dedicated system for handling knowledge assessment related tasks. A basic system design considerations are included in the next section, followed by presentation of our original Communication Abstraction Layer (ComAL), designed specifically to handle communication tasks in eLearning systems. As the system is created in modular fashion a description of its architecture and existing modules is provided next. Modules are divided into server and client modules, with a special section devoted to a simulation-based ones. Next, we present description of a few possible configuration scenarios and present results of a practical deployment conducted on Gdańsk University of Technology. The chapter closes with conclusions gathered both from theoretical analysis of the new architecture and practical deployment of our distributed, ComAL-supported knowledge assessment system.

2. System motivation

The task of knowledge assessment is one of the fundamental elements of didactic process. It was also one of the first didactic tasks to be conducted by various electronic learning devices employed to support didactic process. Currently there are many e-learning solutions supporting knowledge assessment both as their main functionality or as an additional module ("Sakai...", 2008), ("Moodle...", 2008). Almost any advanced e-learning tool offers this functionality. In light of those facts we could conclude that this area of e-learning is a well explored one and suitably supported by practical e-learning products.

Our experience with e-learning systems both as their users and designers, leads us to conclusion that the above statement is far from correct. Vast majority of currently available electronic knowledge assessment tools are extremely similar and offer strictly limited functionality. Such products offer almost exclusively knowledge assessment based on various choice tests and their automatic grading mechanisms most often are not very comprehensive and fit to support different grading scenarios.

In complex e-learning systems knowledge assessment functionality is treated as mandatory element, but also receives no special consideration, which often results in a simple implementation of choice test. Specialized knowledge testing solutions (employed for example by Microsoft during their computer proficiency exams) include more advanced mechanisms, like adaptive question selection, but they are few and still do not go beyond the basic scenario of choice test (Bersin & Associates, 2004), (Jesukiewicz, P. et al., 2006).

Apart from these weaknesses, one of the most serious problems with currently available products and especially the most popular ones based on web-based thin-client architecture, is their strict dependence on network connectivity. Majority of such products require constantly active network connection during e-learning session and few are fit to function under other circumstances, such as periodic or no network connectivity, and still remain a part of managed e-learning system. The quality of network service is also a factor in case of many of such products (Gierłowski K. & Gierszewski T., 2004).

Apart from characteristics directly connected with didactic process, there is also a deployment phase of the system to consider. Designers of e-learning do not pay sufficient attention to this subject, and such approach often results in lack of user interest in good e-learning products or deployments unsuccessful despite otherwise correct choice of the system's functionality.

From our experience, the most common reasons for e-learning system deployment problems fall into one of these categories:

1. lack of required or expected functionality of the e-learning solution; user interface design,
2. incompatibility or lack of integration with other systems being used in organization,
3. inability to provide a smooth transition from previously used product.

In case of large organizations we can also add:

4. performance/availability problems,
5. extensive unification of didactic process due to deployment of a single, organization-wide system.

The first point is self-explaining and well known – choosing the correct functionality of the system is a basic step in deployment process and any lacking functions will lower users' opinion. In the event where any advantageous function is missing from the product it can be provided by external applications if the system has been designed with ease of integration in mind (see point 3).

User interface is the element of e-learning system which has direct contact with its user and, as such, directly influences the user's opinion about the product – in case of knowledge assessment systems that includes both teachers and students. Efficient and friendly user interface has a paramount impact on system deployment. Moreover, if we are deploying a new system in place of previous solutions, serious differences in menu layout will most likely be received unfavorably by users.

An e-learning knowledge assessment system most often functions in parallel with other computerized applications – grade archives, resource planners, payment/finance systems, content repositories, library indexes etc. In many cases our system can benefit from information from these sources or we need to make results of knowledge-assessment available to such outside systems. A special care should be taken to provide means of integration between such systems, as lack of this functionality will drastically reduce

usefulness of e-learning system and create resentment of users each time an additional work to transfer the data will be required. A system able to integrate with its environment can also be easily extended with new functionality, when a need arises (see point 1), which makes this characteristic even more important.

Very rarely an e-learning knowledge assessment system will be deployed as a first such solution in a given organization – more often it would be deployed in place of some previous solutions. From our experience, providing a smooth transition from previously used products is the most important requirement necessary for a successful deployment. To fulfill this goal administrators need to consider its technical and organizational aspect.

Technical aspects of a successful deployment include such critical elements as uninterrupted service and good system performance. The service provided by e-learning system should be available to all interested users continuously, even during the period when an old solution is deactivated and new service commences its operation. Moreover, care should be taken to maintain a good system performance despite a heightened load of the system, which is expected shortly post-deployment.

Organizational aspects of smooth transition are even more important and difficult – users are accustomed to their familiar ways of conducting e-learning tasks. If the deployment of our new system is to be successful, the users must have necessary information about its operation (for example: courses, manuals etc.).

It is also very probable, that teachers organized their work in a way, which allowed them to efficiently utilize the previous system – if the new proposition will require drastic changes in work organization the chances of successful deployment will be low. There is also a possibility, that users augmented the basic system's functionality with third party applications, which they loosely integrated with the system (for example they use MS Excel to analyze results exported from the system).

Moreover, users would probably have a considerable amount of data collected in the previous system (for example: test content, grades history etc.), which must be possible to transfer to a new one.

If we are plan to replace many independent solutions used through an organization with a single system, we face the most difficult scenario for successful deployment: there will be different data sets and formats to migrate to a new system and hard to evaluate functionality and performance requirements. Users from different organizational units and with different needs will be required to unify they work organization to match functionality provided by a new system and depend on centrally managed and maintained solution, where any modification of system's operation is much more complicated, time consuming and thus unlikely. Moreover, while consolidating data from different organization units can be difficult, the opposite operation (separation a from shared data store) can prove even more work consuming.

The above analysis shows that, apart from the obvious requirement of providing necessary functionality, interoperability of an advanced e-learning system is its key characteristic. Also, all methods which would enable a gradual deployment, differentiation of system operation between organizational units in large organizations and provide high performance/reliability are highly recommended.

Having analyzed above limitations of currently available knowledge assessment products and difficulties of deploying a new solution in already functioning environment, we designed and created our own dedicated knowledge assessment system. It was designed to

provide highly modifiable platform for various knowledge testing tools, able to provide its functions in any network connectivity conditions (including no connectivity scenario). The system can scale from very simple setup (adequate for servicing a single exercise) to a large, distributed solution fit to support an enterprise. Strictly modular architecture allows users to employ only a selected set of its mechanisms and extremely easily integrate it with third-party solutions. The selection of employed modules depends completely on user needs – there is no mandatory control module or management platform which must be present. We created a number of client modules with full support for low/no-connectivity scenarios, for example:

1. the classic, but highly configurable and versatile, multiple choice knowledge testing solution,
2. a simple simulation-based module, addressing security of Ethernet switches,
3. an unique simulation-based knowledge and skill assessment module, dedicated to exercises concerning Asynchronous Transfer Mode (ATM) and Frame Relay networks,
4. a number of modules allowing a real-time grading of students performance during exercises.

Our system also addresses security aspects of remote, computer based knowledge testing, in both test distribution and results gathering preserving user anonymity to unauthorized parties.

As an key element of the system, we have created an innovative Communication Abstraction Layer (ComAL) - a set of mechanisms designed to provide e-learning system designers with API containing a comprehensive set of communication functions which can make an e-learning system independent of underlying network connectivity conditions. ComAL completely isolates e-learning solution programmer from the details of network communication and can be employed to easily create networked e-learning solutions, allowing creation of an integrated, managed e-learning system even in environment without network connectivity.

3. Overall system design

During design and creation of our system we aimed to provide a solution fit to accommodate needs to assess students knowledge in the widest possible set of scenarios. To fulfill this task we considered its following aspects:

1. compatibility with a widest possible set of hardware and operating systems,
2. ability to function in variety of network connectivity environments (including lack of such connectivity) while still retaining capability to function as globally managed solution,
3. security and reliability of the system, including safety of the system itself, test content, students' solutions and personal data,
4. information storage and manipulation capabilities, to allow creation of central database of results and grades, complete with easy access methods,
5. knowledge assessment functionality including: multiple choice tests with highly customizable automatic grading and real time grading by a teacher,
6. comprehensive management interfaces for administrators and teachers,

7. ease of deployment, customization, modification and integration with third-party solutions.

To fulfill these requirements, we have chosen a client-server architecture for our system, which is a pretty standard solution today, but in contrast to the most common practice we decided to abandon thin-client technology in favor of full-client approach.

From our experience, web-based thin-client architecture despite its undisputed compatibility and ease of deployment, is not especially well suited for knowledge assessment systems, as it requires a constant network connectivity for operation and lacks a sufficient degree of control over user environment, which impairs system reliability and allows unauthorized actions on part of the users. Operating system and web browser security mechanisms are also an important issue here, as their incorrect configuration can lead to abnormal client software behavior (Nowicki K. & Gierłowski K., 2004).

Full-client approach allows client to conduct much wider range of operations compared to thin-client. This allows inclusion of more advanced internal mechanisms providing improved functionality, much better reliability and security of client operation. With proper design full-client utilizing web-based technologies can also operate independently of server which gives our system versatility, necessary to handle limited network connectivity scenario. It also facilitates creation of a strictly modular system architecture and providing of high level of scalability (as many tasks can be conducted client-side and data transfers minimized).

The most serious limitations of full-client approach, deployment and system compatibility, are also possible to over-come by employing easily deployable, platform in-dependent clients (for example Java-based). Such solution allows for all advantages of full-client and web-based technologies, while still retaining high level of hardware and system compatibility and easy (even web-based) deployment.

The second of our fundamental design decisions was maintaining a strict modularity of our product. All basic elements are constructed as modules capable of operating independently, that's why we call the architecture of our system - a loosely-tied distributed architecture.

Moreover, we are employing only standardized, self-describing data format for inter-module data transfer - Extensible Markup Language (XML).

Modular system structure complicates design and implementation, as it requires the use of additional inter-module communication mechanisms, but these difficulties are easily compensated by our ComAL API, described in later section.

On the other hand modular structure brings enormous advantages, as it is possible to substitute customized solutions in place of some standard modules or include additional elements into standard system data paths to provide additional data analysis/translation functionality (Figure 1). Advantages of these possibilities are clear, as they allow easy modification and customization of the system, including creation of dedicated interfaces for third-party systems and applications. Furthermore, there are already many solutions accepting XML input and providing XML output (for example MS Office, OpenOffice etc.).

There is also a possibility which had proven even more useful than these mentioned above during test deployments of our system - it is possible to deploy only selected elements and/or integrate it directly with third-party solutions supporting XML language.

An ability to deploy only a chosen set of modules allows for deployment precisely tailored to individual needs. If system user is interested only in simple multiple choice solution for a small number of students there is no need for a system server - it is enough to deploy only a

testing/grading module and read resulting offline data files directly with MS or Open Office. In an opposite situation, where the user is interested only in system's data storage and access functions, he can easily deploy the system's server part, substituting its clients with his own, as long as they support XML output or can provide appropriate translating interface.



Fig. 1. Integration with external applications.

This partial-deployment ability also makes transition to new system much easier, as it can be conducted in phases, by gradually exchanging existing infrastructure with modules of the system.

The most common usage scenarios include:

1. No server-side / third-party scenario - client modules are operating independently or export results to a third-party application/system.
2. Single server scenario - client modules are managed by client communication module and access central database, all data storage and control functions are available.
3. Multi-server distributed architecture - able to support large number of clients (performance) and allows different organizational units to operate independent (but integrated) servers.

As you can see, a loosely-tied distributed architecture, where all modules are able of independent operation and communications are handled with a well-known, versatile data format, satisfies many of critical development requirements, difficult to fulfill with, currently the most popular, centralized thin-client approach. Gradual (or even partial) deployment is possible, as well as easy integration with third party applications and modification of system's functionality.

A detailed description of system architecture, found in the later section, will also show that the system's operation in separate organizational units can be differentiated in a significant degree, and that high performance/reliability level is relatively easy to maintain.

4. Communication Abstraction Layer

One of key elements of modular system are inter-module communication mechanisms. The task of providing local communication (between modules on the same machine) is relatively simple, because we can precisely predict environment characteristics.

Remote connectivity (communication between modules on different machines) is another matter. It is dependent on various characteristics of available network infrastructure. Providing reliable communication and satisfying quality of service requirements of an e-learning system in wide range of network scenarios and conditions is a difficult and work intensive task (Gierłowski K. & Gierszewski T., 2004). Its complication and cost most often lead

to abandoning such attempts and creation of products which require constant and stable network connectivity lacking mechanisms for handling other scenarios (for example: the popular thin-client architecture) or employ no advanced communication functions at all.

While such approach may be sufficient for didactic content distribution systems, knowledge assessment requires a higher degree of communication between client (which interacts with user) and server part of the system (usually responsible for control, management, task assignment and results gathering).

To help developers in building a robust, networked e-learning systems we have created a set of mechanisms called Communication Abstraction Layer (ComAL) specifically designed to provide network communication functions required by e-learning environment (Figure 2). This set of mechanisms can employ a variety of communication methods, automatically choosing the one most appropriate for current working conditions, and is responsible for all communication tasks - both local and remote. It isolates e-learning system developer from particulars of implementing a network communication mechanisms by providing him with high level API.

From our experience in developing networked e-learning systems, we divided most often encountered network conditions into four scenarios:

1. Local Area Network - efficient and reliable, permanent network connectivity.
2. Internet - an environment where we have a permanent network connectivity at our disposal, but there are no Quality of Service (QoS) or reliability guarantees.
3. Periodic connectivity - most commonly encountered in case of dialup connections.
4. Offline - there is no network connectivity, but there is still a possibility of communication by offline methods (floppy, CD/DVD, USB-storage...).

ComAL provides dedicated means for maintaining a stable communication in all of these environments, and is able to detect the correct scenario automatically and keeps monitoring the situation to detect if the scenario changes.

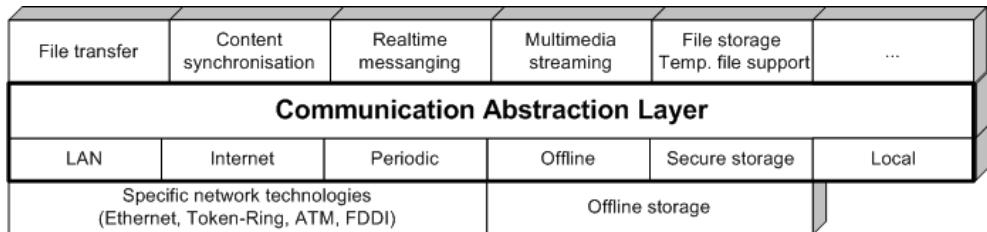


Fig. 2. Communication Abstraction layer - overall architecture.

Communication functions provided by ComAL to e-learning system creator can accommodate a wide variety of application types, ranging from sending a simple messages, through high volume file transfers and content synchronization, to reliable, real-time interactive message exchanges and multimedia transmissions. Of course, not all of these functions can be made available in all of the above scenarios. To deal with such limitations, ComAL provides feedback mechanisms informing higher application layers about functionality available under current network conditions, state of currently conducted communication activities, overall status of network connectivity and its changes.

For transport of data ComAL currently employs (Figure 3): direct TCP and UDP connectivity, SOAP over HTTPS, encrypted SOAP over SMTP and advanced, automatic,

secure file export/import functions. Some of these methods (SMTP and file-based) allow communication between system modules behind NAT. Moreover, we are currently developing a media proxy module functionality, allowing destinations behind NAT to communicate with TCP/UDP and indirect (but still secure) SOAP over HTTPS.

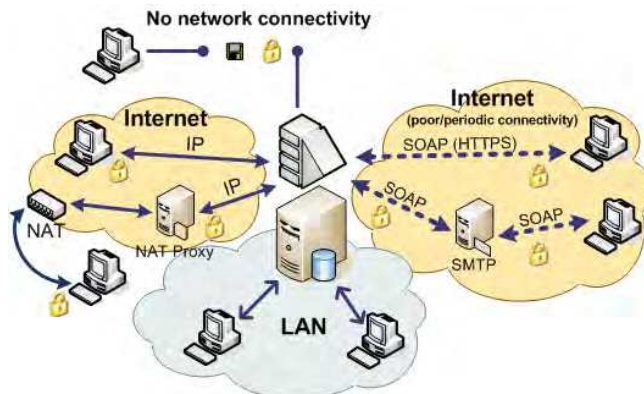


Fig. 3. Communication scenarios supported by e-learning systems with use of ComAL.

All communications can be protected with use of strong security mechanisms, ensuring their confidentiality, integrity and mutual authentication of communicating parties. The communication can also be digitally signed to ensure non-repudiation of submitted data (for example test solutions). The ComAL utilizes both symmetric and public-key cryptography and supports automatic key/certificate generation for clients.

Due to similarities between mechanisms required in case of the Offline communication scenario and committing data to a high security storage, ComAL can also be easily employed for that purpose. To provide user with such functionality, we have defined a fifth (special) communication scenario - Secure Storage. It provides a considerable range of cryptographic functions and tamper protection mechanisms:

- In-memory file storage - files are stored exclusively in memory, optionally in encrypted and signed form. It minimizes the chances of recovering protected data from disk.
- Automatic secure delete - mechanisms to securely delete files from disk - both on demand (in case of regular files) and automatically in case of temporary files. The delete procedures are somewhat relaxed in comparison with ones advised for high security solutions, in order to prevent an excessive hardware degradation while still retaining a decent level of security.
- Symmetric/asymmetric encryption and data signing - well known and widely used cryptographic mechanisms, based on OpenSSL libraries.
- File attributes verification - all file attributes (read only, archived, system, creation data, modification date etc.) can be employed in encryption/signing process, to prevent user from modifying them.
- File location verification - low level information from filesystem directory and allocation table can be employed in encryption/signing process, to prevent users from modifying/backing up/coping/overwriting protected files. For example: file

needs to be placed in the same exact disk sectors as it was initially written to, if it is to be correctly decrypted.

- Access counters – information about all operations (reads/writes) can be included with the file in secure manner.
- Date/time monitoring – dedicated mechanisms detect and report any suspicious changes of system time/date.

These advanced secure storage functions were added to ComAL for use in e-learning products deployed in remote learning /remote knowledge assessment scenario. An example of their use can be an upgraded version of our choice-test module, described in a later section.

We believe that creation of such abstraction layer, able to free e-learning system developers from difficult, specialized, costly and work consuming design and implementation of network communication functions can encourage creation of advanced solutions, taking full advantage of potential provided by a networked environment. It has been utilized in a number of our e-learning products (Nowicki K. & Gierłowski K., 2004), (“QTI...”, 2008), (Gierłowski K. et al, 2003), greatly reducing design complexity and implementation work required. It was also successfully employed to extend functionality of strictly local e-learning solution, to allow network based management.

ComAL has been designed to isolate high level e-learning system designer and programmer from specifics of network connectivity. Our experiments with various new e-learning technologies and functionalities confirmed our belief that such isolation is an advantage in most cases, but they also suggest that the layer should not block programmer/designer from accessing low level functions and statistics of network interfaces if he wishes to. Such functions should be kept as network technology independent as possible, but should be available. An excellent example of e-learning product which requires such access is the mobile version of our real-time grading module (described in later section) – it requires access to low level parameters such as radio signal strength or hardware address. To support this requirement, we decided to extend ComAL with ability of a very simple SNMP (Simple Network Management Protocol) agent, which provides such information to higher layers in standardized form.

ComAL is a basis of all inter-module communication in our knowledge assessment system, enabling our system to function as manageable entity in most diverse communication scenarios.

5. System architecture

As our knowledge assessment system follows a client-server design, its modules can be divided into two basic groups: client and server modules. Communication between system elements is conducted with use of ComAL to support various network environments.

Client modules interact directly with student or teacher during didactic process and are responsible for providing majority of system’s functionality in accordance with configuration information obtained from servers and under their control. Results of knowledge assessment conducted by client modules are returned to servers for processing and storage.

There can be many client modules providing different types of knowledge assessment or supporting functions – their design and functionality is described in the following sections.

All of these modules are able to function as independent applications and support full ComAL capabilities, including strictly offline scenario - in such case configuration and test packages are provided to them as cryptographically protected files (automatically generated by server), and results are returned to server in the same way. Also they are able to monitor presence of network connectivity and initiate automatic upload of results gathered in offline mode.

Server part of the system is a distributed database containing both didactic content and system's complete configuration information. A single server consists of a database (most often an SQL server) and at least one of two modules: system maintenance module and/or client communication module.

A system maintenance module is responsible for creating and maintaining distributed information base. It also provides a web-based administrative interface allowing administrator to create and control the system.

An administrator is responsible for creating a distributed system architecture by defining communication links between system servers, and deciding which of ComAL transport mechanisms are permitted for each link. If a server cannot maintain current transport mechanism it will switch do less demanding one if such is permitted, otherwise it will mark link as down. A simple link-state path selection protocol is then used to ensure communication between all nodes, utilizing as link metrics information from ComAL network monitoring mechanisms.

Over such communication structure works a data indexing mechanism, allowing full access to distributed information base from any system node. This distributed database includes complete information about system-wide configuration (system structure, global users and access rights, link states, distributed database state) which is replicated to all servers and didactic content (test content, grading rules, student lists, test results and grades etc.) which is kept locally on specific server and is not replicated (however, creating mirror servers is possible), but can be searched and accessed from any server if there is connectivity present, and access rights are sufficient (Figure 4).

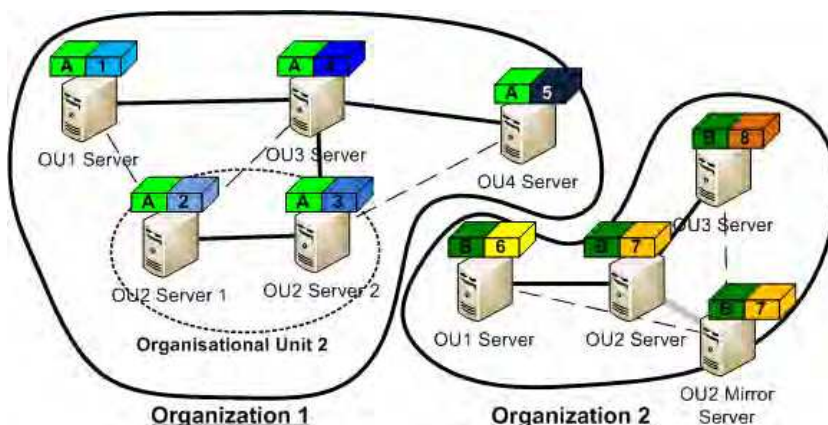


Fig. 4. Server side of the system: A,B - configuration content (replicated) in organization; 1,2,... - didactic content (indexed or mirrored).

Such architecture allows largely independent operation of a particular server (supporting for example a single course or organization department), while still allowing administration of the system as a whole and easy access to information stored on different servers. It will give various teachers or departments the freedom of independent operation and still provide means of global data access. As a result servers can be connected and disconnected from the distributed system at will with no negative effect for them or the system (other than loss of access to disconnected resources).

If a given server will communicate with client modules, it must include a client communication module. It is responsible for communication with other modules using ComAL, supplying client modules with configuration and didactic content, gathering and processing incoming results and providing teacher's interface to the system.

Teacher's interface allows its user to create test packages and assign it to specific combination of students, network workstations or time frames - it is possible to provide student with a choice of available tests. Such test package contains all information to conduct a test, namely: test content adequate for specific client module, grading rules and additional information concerning test execution (time limit, randomization of content, means of results upload etc.).

Teacher's interface allows a full read access to gathered results and ability to modify or add data concerning teacher's own tests, as well as generation of basic reports and statistics. There is also an option of importing external data in XML format into the system and exporting system data in the same format.

Described structure of the distributed database addresses important requirements necessary for easy and efficient system deployment in large organizations. It can be deployed gradually and its operation can be significantly modified in different servers while overall system's integrity remains intact. Performance problems can be easily solved by deploying more servers. Reliability level also remains very high, as server failures simply cause system to reconfigure its data paths and the only resources that are unavailable are these maintained exclusively (not mirrored elsewhere) on the malfunctioning server.

Furthermore, despite all these advanced functions and possible configurations, deployment in simple scenarios remains uncomplicated.

6. Client modules

Client modules are critical for the system as they are its point of contact with the student, responsible for many essential tasks, such as presentation of test content, enforcing configured test conditions (time, test randomization, etc.), performing it and sending all necessary information back to the system for processing and storage, etc. The grading of tests can be conducted client-side, by client modules, or server-side, by client communication module. Server side grading is more secure but less scalable and versatile solution.

6.1. Test-based knowledge assessment module

This first and most often employed client module of our system allows knowledge assessment by means of diverse choice tests. While the method itself is very popular in case of e-learning products, we designed this module to provide functionality unique among similar products.



Fig. 5. Choice-test knowledge assessment module – user interface.

Module's user interface (Figure 5) allows presentation of wide range of multimedia content including formatted text, bitmap and vector graphics, sound and movies which provides teacher with great versatility when preparing test content. Test questions can be randomly selected from a larger set, and order of both questions and answers can be randomized.

Automatic grading mechanisms support both single and multiple choice questions in single test, and use simple scripting language which allows to utilize any of popular methods of assigning points for test answers. The total test result can be normalized to a provided value and/or mapped to a grade. The module returns to server or external application a complete information about student's solution (which can be later used by, for example, server side grading mechanisms), such as: personal data, test id, timeframe of test, client-side grading results of all questions and total grade, all answers, operating system computer name and user name, IP address etc.

Module configuration allows teacher to enforce various additional test characteristics, such as necessity of solving questions in order, lack of ability correct already answered questions, time limit for a whole test or every single question etc.

Apart from already described functionality, one of our primary priorities was to take full advantage of ComAL communication capabilities allowing the software to function as part of managed system even without network connectivity. Such scenario is most often ignored by de-signers of modern e-learning solutions as it prohibits the use of web-based thin-client architecture. At the same time it is a very popular scenario in case of network-related courses, as obtaining network connectivity is often the final goal of laboratory exercises on the subject.

The module is fully capable of independent offline operation according to encrypted and protected configuration files and course packages. In such case results are stored in similar, protected files and can be decrypted by the module at teachers request for manual transfer to the server or other application such as MS Excel. The module can also detect available network connectivity and update its status by obtaining new configuration settings/course packages from the server and uploading cached test results automatically.

After through testing of this module and over one year of its extensive usage in real didactic environment, we decided to upgrade its functionality to better support a remote learning environment. Even in its initial version, the module can support remote knowledge

assessment activities, but it does not incorporate any dedicated mechanisms for the task, except these required to support various connectivity conditions.

Knowledge assessment in remote learning scenario is a very difficult task, because teacher has no direct control over student's working environment or even reliable information about it – for example, there is no guarantee, that the test was completed by a given student, that no prohibited resources were utilized (calculators, books, etc.), that it was attempted only once, that the questions were unknown prior to test, etc.

There were many solutions proposed to solve these issues (including real-time video monitoring and similar, difficult and costly solutions) but any of them can be bypassed with at most moderate difficulty, and their deployment and maintenance can be called "troublesome" at best. In this situation we decided to test a number of new approaches to the problem of enforcing test conditions in remote learning scenario, but none of them has been designed to provide guarantees – only to protect against most common exploits, but without inconveniencing student or teacher.

In our opinion there are 3 basic security elements of remote knowledge assessment:

1. Reliable and secure communication – aforementioned information should be transmitted to and from client in a secure (confidential and unmodified) and reliable way.
2. Stored data security and confidentiality – sensitive data, such as test content, answers and results need to be protected against unauthorized access, modification and deletion.
3. User identity verification – there should be some way to verify identity of a person taking the test.

The first two requirements can be easily solved with the new ComAL version, supporting Secure Storage functions. Secure communication mechanisms were included in ComAL from the beginning as a common requirement for e-learning systems. New Secure Storage procedures can prevent users from obtaining unauthorized access to local module data (encryption, signing, in memory file storage, secure delete), including a significant level of protection against multiple test attempts performed by backing up and restoring module data (file attribute/location verification, access counters).

To provide some means of user identification confirmation, we decided to employ keystroke dynamics analysis (Sungzoon C. et al., 2000) – method of identification based on measuring timing relations between keystrokes. This method is employed during user logon and in course of the test, if there are any questions that require typing. During our test deployment (about 100 tests), the method showed over 90% efficiency in detecting attempts to impersonate users. The required identification data (necessary to train underlying neural network) has been collected during preceding classroom exercises.

As a result we have at our disposal one of the most advanced (apart from adaptive choice tests) knowledge testing products utilizing choice test method, which can function as a standalone application or as a part of managed system regardless of network connectivity available, due to integrated ComAL functionality and provides dedicated support mechanisms for remote knowledge assessment.

6.2. Real-time grading module

Real-time grading module has been designed to support classical theoretical and laboratory exercises. It is an inter-face for a teacher, allowing him to grade students during such activities, by marking their progress through assigned tasks.

The main element of the module's user interface (Figure 6) is a table containing all students participating in the current class in the rows and task numbers in columns. A teacher can assign percent grades (including above 100% for exceptional performance) for specific tasks, and the system will calculate resulting total grade automatically and in real-time.

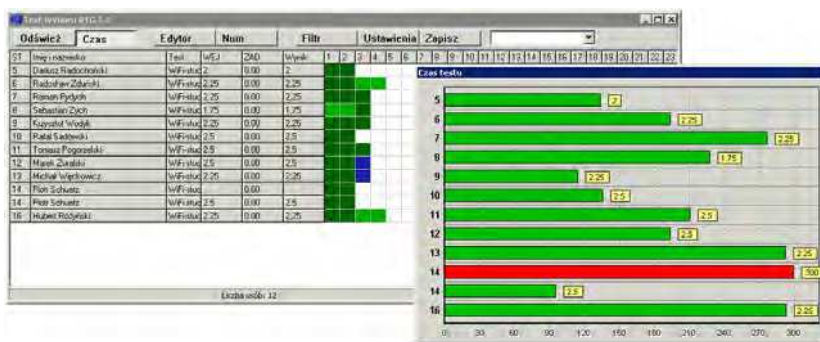


Fig. 6. Real-time grading module – user interface.

The list of students can be obtained from the system server or any other application with use of any ComAL-supported mechanisms including offline files. For example, if students are required to take an entry test before the class, their list, complete with additional information such as their place number and entry test grade will be displayed. Of course other means of creating such list can also be used (such as dedicated system module for checking in or a teacher prepared list) and provided to module by the server automatically or by simple file import.

Apart from user list, module can also take into account different point weights for different tasks, and such information is obtained by the module in the same way as the list of students.

This module can be an enormous help for a teacher, as he has complete information about his students at his disposal including name, place, and if entry test was taken, its result. That way he knows the entry level of theoretical knowledge of his students and can assign his attention accordingly. Also student's progress during current class is easy to track and graded automatically. Results can be uploaded to system server via any of ComAL transport methods.

6.3. Real-time grading module (mobile version)

Practical deployment of the system at Gdansk University of Technology proved great usefulness of Real-time grading module. It has been successfully utilized in a number of scenarios, including lectures, both theoretical and laboratory exercises and seminars. An ability to easily grade activity of students, without interrupting the flow of lecture/exercise (to ask for student's identity, for example) is a great asset. However the deployment also uncovered several limitations of the product – the most important being its dependency on a

computer hardware able to run full blown Java applications and equipped with standard console: high resolution monitor, full keyboard and optionally mouse.

Because of these limitations and aware of high popularity of Symbian based mobile phones, we decided to start development of a mobile version of real-time grading module. It supports all console types available in mobile phones: standard phone keyboard, full keyboard and touch screen.

The module utilizes a simplified version (only two communication models: online and offline) of ComAL for Symbian 6.0 v3 (still under early stages of development) which can provide connectivity with use of WiFi, GPRS/EDGE..., and Bluetooth. It is also able to transmit ComAL offline communication files through IrDA.

In online mode the module exchanges information with system in real-time, while offline mode allows teacher to download/upload information packages on demand.

Apart from the support for new console types, we also decided to include a new mode of entering the grades, which has proven an efficient one in case of large number of students. Instead of marking their progress on the graphical list (X axis/Y axis: tasks/students), teacher can simply enter ID in "row number/chair number" format, task number and grade. The module will provide him with student identity and information about his current grading along the way, if such information is available to the module.

With help of such module the most troublesome element of real-time grading is obtaining and entering an identification of place that a given student occupies. To help with the task, we are currently developing and testing two different solutions, both utilizing low level integration with wireless technologies.

The first mechanism, designed for computer laboratories, utilizes Bluetooth interface. It works by measuring the strength of Bluetooth signal from interfaces present in student's computers. Hardware address of a Bluetooth interface with the strongest signal is then checked with a list of such addresses to obtain a computer number (place number). In practice it means that all a teacher has to do is to approach a student instead of entering place ID manually – the module will display student's information and allow teacher to enter the grade.

The second one, designed for larger environments (for example lecture halls) is only available to the module in online mode. In contrast with the Bluetooth-based solution described above, it depends on WiFi access points (APs) and their ability to measure WiFi client's signal. Our test proved, that with 3-4 APs (we used Cisco Aironet 1100 APs) deployed in a simple (as far as radio propagation is concerned) environment (lecture hall) it is possible to pinpoint location of wireless client with about 5 meter accuracy. It is insufficient to identify a particular place in the hall, but the module uses this information to narrow down the list of possible place ID, which it provides to teacher to choose from.

Both of above solutions are in testing phase, but they are already functioning fairly stable. The main problem concerns a proper configuration of the required infrastructure: Bluetooth interfaces should be of the same type or at least transmit with similar power, they need to be configured as "shown" (as opposed to "hidden" where they will not transmit their identity information), the table of Bluetooth hardware addresses and their corresponding place IDs need to be constructed as well as the table allowing translation of WiFi signal strengths to locations, etc. To offset this problem, we are currently working on tools designed to automate many of the above tasks.

6.4. Script-based automatic grading framework

We have observed, that solutions of many tasks in laboratory environment can be verified with use of automatic programs or scripts. In case of our area of expertise – computer networks – we often employ such tools to verify the correctness of student's solutions. Simple scripts can check connectivity, measure network performance statistics (such as throughput, error rate, delay etc.), monitor availability of services (such as, for example HTTP, FTP, DNS, SMTP, POP, IMAP servers) or even remotely login into computers to check correctness of the configuration files. Such automation is an enormous help for a teacher, allowing him to both easily grade students and help them troubleshoot their solutions.

Seeing the usefulness of such tools, we decided to integrate them with our knowledge-assessment system. The main difficulty of this task originates from a multitude of available scripting engines, additional required libraries etc.

To offset this difficulty, we decided to implement the module as a framework designed to run scripts and programs supported by an underlying operating system and then capture and parse their returned results.

The module obtains configuration from a system server (through ComAL) or from a local XML file. Teacher can configure what programs and scripts are to be executed, how often and how to interpret their results, for example: "Every minute run iperf.exe with a number of parameters to measure network throughput through a student configured system. If the throughput is over certain threshold, mark task 5 as completed in 100%. If it is below this threshold, but here is connectivity mark task 5 as completed in 50%."

Many programs and scripts can be used to verify a single task creating complex and detailed checks – in such case checking scripts are chained with OR / AND conditions.

Teacher can also organize tasks in a hierarchical way, by defining a prerequisites for a given task – for example: "Completion of task 6 is not checked until task 5 is marked as completed". It prevents unnecessary checks and possible "false positives" by allowing the teacher to clearly specify the moment the task begins and/or correct starting configuration for a given task.

Test deployment of the framework in computer networks laboratory proved that it allows a completely automatic grading of many complex exercises, provided that teacher analyzes students' workflow and prepares a well designed and tested module configuration. However, as a result of a test deployment we decided to mark automatically assigned task grades in system's database as such, for optional verification and confirmation by a teacher.

7. Simulation-based client modules

A simulation product can be of great service in didactic process (Chwif, L. & Barretto, M.R.P., 2003) (Reichlmayr T., 2005) – it has its use in both remote learning (Kindley R., 2002) (Chiaming Y. & Wu-Jeng L., 2003) and classical learning activities (Lainema T. & Nurmi S., 2006) (Henning K. et al., 2005). Of course there are many types of simulation products, each suitable for different tasks: there are simulators devoted to detailed modeling of physical processes (often employed in scientific research), simulators created to train users in operation of various machines (for example: flight simulators), and a great many other products.

In our work at Gdansk University of Technology we find simulation an effective tool in both research and didactic tasks. Its usefulness in computer network science research is obvious

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

