E-Learning Function Integration with Corona 2

Marco Pedroni
CARID Università degli Studi di Ferrara, Italy

INTRODUCTION

The aim of the Corona project, which was launched in 2003 in the context of methodological-technological research on the subject of e-learning carried out in collaboration with CARID ("Centro di Ateneo per la Ricerca, l'Innovazione Didattica e l'Istruzione a Distanza", the research center of the University of Ferrara which studies e-learning methods and techonologies and applies them in real learning contexts) and the Omniacom Consortium, is to facilitate the integration of functional modules (synchronous, such as chat rooms and virtual classrooms, and asynchronous such as the production and provision of Learning Objects, forums, assessment and tracking tools), originating from different development environments, in a single online interactive environment.

Possible Choices for an E-Learning Platform

The project commences with an analysis of a situation common to training organizations aiming to provide e-learning courses: such organizations inevitably come up against the problem of what software tools to choose in order to support online interaction between content experts, tutors, and students.

There are three possible choices: acquisition of a commercial platform, implementation of an open-source platform, or proprietary development.

The first choice enables the purchaser to select from a wide range of products that are important as regards both quantity and quality: the market of this sector, far from being monopolized by the leading software houses, actually offers dozens of well-consolidated solutions.

Identifying the best product for the purchaser's goals and requirements is not a simple task, inasmuch as the enterprises that produce and provide training often have organizational peculiarities which are not easy to match up with the internal structure of platforms designed with general application criteria: the same type of learning proposal, if characterized by complex forms of modularity can come up against obstacles in the platform's standardized management.

The second choice differs from the first mainly as regards cost, not linked to the purchase of the licence but deriving from the need to have qualified computer science engineers for the installation, configuration and, even more important, adaptation of the product to specific requirements, and this entails a considerable investment in terms of time and resources.

The third choice obviously allows for maximum freedom and flexibility of development as well as extensive adaptation to already existing organisational structures, but requires the availability of a technical staff with the capacities to satisfy, by means of programming tools, the function, and interaction requirements desired by the methodological research and development department.

Concept of Interoperability

A further aspect to be carefully considered in the context of the suggested hypotheses is the degree of platform interoperability, that is, the capacity of the platform to implement content modules developed in different environments: this capacity does not, obviously, refer to the banal possibility to download such modules or to use them in stream by means of standard plug-ins, but rather the possibility to implement software procedures that support the tracking of user activities.

In conclusion, the limitations of these three possibilities are such as to render problematic a definitive judgment regarding the best choice to make in order to ensure the efficiency, efficacy, and cost-effectiveness of the results, even in relation to individual situations: from this panorama, the objective difficulty in embarking on such initiatives experienced by institutions involved in online learning methodological and technological research clearly emerges.

CORONA PROJECT

In an attempt to tackle the entire issue and identify possible solutions aiming at overcoming the operational and economic limitations posed by the various hypotheses, it is important to consider two fundamental aspects: the need to interact with preexisting databases and software environments and the need to select, customize or construct *ex novo* interaction functions able to support the teaching methods adopted.

In the light of the previous, within the context of the "Corona" project a thin slice of software in Java (the standard Web language, extremely high performing and portable) and PHP (language extremely widespread in online applications, distinguished by its user friendliness, and portability across all main Web-servers) aimed at enabling dual integration with the organisation's database (in which the data relative to the figures involved in the learning process, course areas, and access levels may be found) and with externally developed functions.

The reason for the interest in this project should not be sought exclusively in factors such as flexibility and freedom of choice (indeed, it is important to emphasise that composition of a methodologically consistent, efficient, and high quality environment is a complex activity which is not to be confused with the indiscriminate aggregation of software objects), but mainly in the possibility to improve, complete, and customize, on the basis of the various requirements, the interactive structure on which the online learning environment is based: not wishing, of course, to diminish the primary importance of the strategic design underlying every e-learning environment and platform.

The Beginning

The first version of Corona, which was developed and tested between 2003 and 2005, effectively demonstrated the possibility of developing functional modules that connect easily to the login and integration layer which composes the core of Corona, and similarly the possibility to adapt modules originating from university research, the development of open-source environments and commercial platforms to it.

The focus of Corona is the standardization of the method for transferring data and information between the organization's database and the individual functional modules, for which self-consistency is required (or rather, encapsulation, a fundamental feature of software objects, which may be defined as being the capacity to administer its own functions using only the data and methods existing within itself and the parameters included in the launching instructions), and to which the data relative to the learning context and to the user identified by means of a parameter containing the link to a file in XML format is sent (the following figure summarizes the operational diagram).

In requiring self-consistency of the modules, the first implementation of Corona, does not solve the problem linked to the need—frequent in e-learning environments—to transfer data between various functions (just

Figure 1. Scheme of Corona first version