Systematization of the WebLabs Development Process: Towards an Approach Proposal

Authors:

Luciano Antonio Mendes, Industrial and Systems Engineering Post Graduation Program, Pontifical Catholic University of Parana, Curitiba, PR, Brasil, 80215-901, l.mendes@pucpr.br
Marcio Luiz Debner dos Santos, Industrial and Systems Engineering Post Graduation Program, Pontifical Catholic University of Parana, Curitiba, PR, Brasil, 80215-901, marcio.debner@gmail.com
Marco Túlio Corrêa de Siqueira, Industrial and Systems Engineering Post Graduation Program, Pontifical Catholic University of Parana, Curitiba, PR, Brasil, 80215-901, marco. siqueira@ gmail.com

Abstract — *Technological education has premises such as qualified infra-structure and personnel, and students availability to engage studies and fulfill prescribed academic programs. If the first can be understood as the result of the educational institution strategic vision, the latter is tightly compromised with the common limiting problems met by student seeking knowledge updates: personal and professional routines, distance to the school, time for after class activities, personal budget limitations. The relatively recent Distance Learning (DL) concept, which addresses at least in part these problems, is growing fast powered by the advances in the supporting information and communication technologies (ICT) – particularly, the growth of the end user internet bandwidth. However, in engineering education, many courses contain practical laboratory activities, and offering student the experience of operating a real system from a distant location is a challenge. A solution for this is the use of remotely operated laboratories (WebLabs), where an experiment can be run through the internet and observed through a live videoconference link. This paper proposes an approach towards a systematic method to design and implement WebLabs, as the design of these systems must meet specific requirements related with the particular context of each project. The approach covers three layers - physical system, hardware and software - and is based on a similar proposition derived for the design of automated test systems. The main stages of the proposed model are: didactical, end user and technical requirements definition; system technical specification; conceptual synthesis; analysis, simulation and dimensioning; detailing and documentation; integration, testing and start up.*

Index Terms — *WebLabs, Distance Learning, Remotely Operated Laboratories.*

INTRODUCTION

Nowadays, a consensual important factor for the development of companies' competitiveness is the availability of qualified engineers. Even experienced engineers are required to keep updated with technological progresses in a scenario where products are designed by teams with complementary knowledge, as single-discipline consumer products become more and more rare. Technological education, like engineering graduation and *lato-sens*u post-graduation, has a major role to play in the required continuous education process necessary to avoid engineers' obsolescence and keep companies competitive.

Technological education is two-sided: on the first is the consolidation of modern educational structures, humanly and technically qualified to offer consistent and in-the-expected-depth knowledge; and on the other are the students with availability to fulfill the apprenticeship programs, either in graduation or post-graduation levels. If the first can be understood as the result of the strategic vision of educational institutions (or public policies), the second assumes the forms of the different difficulties met by students willing to improve skills and update technical knowledge (e.g. senior engineers): personal and professional routines, home-to-school distance, after-class activities and budget limitations. At least part of these difficulties is addressed through Distance Learning (DL), a relatively recent, growing-in-acceptance concept. DL is being powered by the evolution of Information and Communication Technologies (ICT's), as the didactical usefulness of these gains space and recognition in education and broadband connectivity expands worldwide. The creation of associations and councils to accredit DL institutions [1] is a direct consequence of such growth. Nevertheless, the compatibility of the basic forms of DL with different knowledge disciplines isn't uniform: if the interaction between a professor and his students can be fairly approximated with the use of cameras, projection screens, microphones and loud-speakers in humanistic disciplines, the same videoconferencing capability itself isn't enough for the practical experimentation activities throughout engineering courses.

By the time of the new millennium, Ko *et al.* [2] stated that a barrier to DL would be in the impossibility of practical laboratory activities. Surely, the direct and present experimentation of classroom concepts is an unsurpassed practice to provide students the closest-to-the-real-world experience, being the laboratory also a place for complementary skills development, such as practical thinking and problem-solving. However, many are the circumstances where the distance learning concept can provide much more than the available locally. Circumventing the problem of local unavailability of generally expensive laboratory facilities is the challenging issue for many educational institutions.

As Internet evolves to Web 2.0 and the broadband networks become available worldwide, the new standards of interactivity levels reaching multiple areas of human actuation are inevitably going to do the same in education. Videoconference links connecting such distinct environments as a modest classroom in a small town and an advanced laboratory in a research center, allied to computer interfaces containing buttons, controls, input/output data capabilities and other pertinent interactivity elements, are scaling possibilities: students testing concepts in equipment located elsewhere; an educational institution 'borrowing' for hours the laboratory of another (far away) during its idle time; laboratory tasks becoming after-class works; students actions being logged and then analyzed to verify how the used reasoning lead to the draw conclusions. The modes of actuation available on the real lab under remote control can vary largely in sophistication, from the simplicity of switching circuits or simple control experiments [3], to elaborated servo-mechanisms or even commanded (static or mobile) robotic manipulators [4].

Therefore, the ever-growing computing and networking technologies are enabling the spreading of the concept in which an experiment in a remote laboratory is operated over the Internet. Even if all the underlying implications are not completely known yet, one already feasible and of direct interest in engineering education is clear: students from remote locations learning and practicing engineering concepts that are only available far away. In order to engineering laboratory activities be incorporated into DL, the Remotely Operated Laboratory through the World Wide Web (WebLab) is already a feasible solution in a multitude of cases. The perceived value of such concept dissemination into the educational process is something in construction yet, and the analysis of the results of experiences with WebLabs is progressively feeding information for this important evaluation [5].

An alternative concept to web-based laboratories, known as 'Virtual Laboratories' [6], has a major difference to the WebLabs one: it consists of a simulated environment, running in a local or remote computer, but that doesn't return on-line real world results, which in general bring imperfections in a rather unpredictable way.

Besides the educational ends, research collaboration is another important possibility. Scientists and engineers can share infrastructures located in distant locations, observe experiments together in real time and interact actively without traveling. As pointed in the KyaTera Project [7], an initiative in Brazil for the development of advanced technological infrastructure to support web interaction in high quality (including fiber-optics infrastructure for a future WebLabs federation), WebLabs bring possibilities yet to be explored in laboratories sharing, e-collaboration, e-learning, distance learning and on-line remote presentation of experiments (e.g. in a conference).

The problem of remote operating a laboratory has already been successfully addressed in several cases. Gravier *et al.* [8] performed a state-of-the-art literature review on WebLabs and their evolutionary trend. Following are some representative examples of implementation cases: Selmer *et al.* [9] reports engineering students in the Cambridge University operating a heat exchanger physically existent in a laboratory on the other side of the Atlantic, in the Massachusetts Institute of Technology (MIT), where the remotely operated laboratories are known for the suggestive short 'iLab' [10]. Duro *et al.* [11] describe an integrated virtual and remote control lab of a three-tank system (UNED and Murcya Universities), in Spain. Also in this country, the DEUSTO WebLab (Deusto University) [12], which has been made open-source recently and is available on-line, allows the operation of mechanical systems and test of programmable electronic devices, among which are microcontrollers, FPGA-based circuits and pneumatics assemblies. This WebLab project was designed to be a scalable platform supporting multiple applications in the institution. The Porto University 'Remotelab' [13], in Portugal (Porto University), uses the LabVIEW^{TM} platform facilities. Pereira *et al.* [9] report a cooperative case between a Brazilian and Portuguese universities which resulted in an application including haptical features. From Italy, the 'G. Savastano' WebLab experience is reported in Andria *et al.* [15].

In the near future, most of educational institutions will deal with the impacts of these new technologies in the academic life. The specificities of each engineering area themselves determine preferential characteristics in the potential applications of WebLabs. Huang *et al.* [16] integrated-manufacturing WebLab is essentially a manufacturing engineering case. The works of Cmuk *et al.* [17], with a model for integration of remote laboratories in electrical engineering courses, and Selmer *et al.* [9], analyzing the potential use for chemical engineering, are just some examples of efforts to define scopes of use and the potential impact in engineering education.

WEBLABS ENGINEERING AND DESIGN METHODOLOGIES

The design of a WebLab depends essentially on the context defined by the experiments to be implemented, which are backed by the didactical objectives involved. This fact leads the WebLab design activity mostly to one-time projects with specific requirements. The degree of complexity is dictated by these requirements, and can vary from readily available sub-systems integration to dedicated, highly automated systems. Therefore, a WebLab development process rather has similarities with special projects, in opposition to serially manufactured systems or products.

WebLabs contain a necessary degree of automation, and so are likely to have the three typical layers found in multi-disciplinary systems (e.g. mechatronical and, in some degree, also many of the modern consumer products): physical system, hardware and software [14] [17]. Attending the requirements for a WebLab project means to engineer and build a system throughout these three layers in an integrated form, having for reference a consistent set of requirements and project specifications early defined. When the comprehension of the design problem in its early stages is higher, less design loops, reworks and other pitfalls are expected to delay the development process.

In the absence of a guiding methodology, the natural behavior of a team of engineers facing the beginning a new design task (a moment when uncertainties prevail) is normally to use the background experience to readily point solutions. The way through a development can be made less empiric if the process is worked out with the systematization provided by a design methodology, which is basically a prescription of stages and actions. In the second half of last century, several design methodologies have been academically proposed [18], coming mainly from the mechanical engineering area, such as Pahl and Beitz [19], VDI 2221 [20], Asimov [21], ASME[22] and Back [23] [24], to work as guides in the development of new products and technical systems. The problem of synthesizing (good) engineering solutions has beyond-technical aspects, like the creativity inherent to a group of designers, so that distinct solutions to a same design problem are likely to outcome from different minds. These models represent the first efforts of systematization of design tasks. The validity of a systematic workflow model is supported by a simple argument: the later a design problem is detected, the more expensive it is to fix it.

One recurrent tradeoff in prescriptive design methodologies models is their rather superficial character, which keeps the general applicability in different branches of technology, versus the ready usefulness in a specific kind of system. In other words, when prescriptions are deepened, the applicability is reduced in some degree. In this sense, the adaptation of a classic methodology model to the case of a given type of system or product is a 'customization' work left to the design team. This task may be easier for groups with experience in the common technical elements present, but even from these it can require higher efforts in first-time designs. If the design can be guided by a less generalist model, where the prescriptions, besides being deeper, are oriented and contextualized for a given type of system and contain valuable specific knowledge, then and advanced start point and a much more useful tool will be aiding the team designing such system. Increasing the degree of prescription of a design methodology model towards WebLabs' development is the intent of the present paper.

As a reference, the prescriptive Pahl & Beitz's Systematic Approach (PBSA) four major design stages are resumed here:

- *Clarification of the Task*: Collect information about the requirements to be embodied in the solution and also about the constraints. Its result is a specification, a reference document stating correctly the problem to be solved, thus responsible for the correct problem addressing;
- *Conceptual Design*: Establish the functions structure; search for suitable solution principles; combine into concept variants. A concept must result from this stage. The major creative efforts are made in this stage, and the taken decisions here will define the cost magnitude of the solution;
- *Embodiment Design*: Starting from the concept, the designer determines the layout and forms and develops a technical product or system in accordance with technical and economic considerations. A preliminar and a definitive layout are the results expected. Dimensioning, analysis and simulations are made to ensure the concept will attend the specifications;
- *Detail Design*: Arrangement, form, dimensions and surface properties of all the individual parts finally laid-down; materials specified; technical and economical feasibility re-checked; all drawings and other production documents produced. Its result is the project documentation.

PBSA model relative simplicity contrasts with finely-detailed, harder to apply models found in engineering design literature, and is a reference that influences the design process of companies worldwide. Based on the PBSA model, Mendes *et al.*[25] proposed a model with deeper prescriptions to automated test systems, another case of three-layer systems. With the appropriate adaptations, that structure is used in the WebLabs design approach proposal which is object of the present work.

THE PROPOSED MODEL

Figure 2 shows the six-step model proposed as approach for WebLabs development. The first two stages are focused on the project concise planning through the capture of the requirements and their conversion into a representative set of technical specifications, while steps three to six show an integrated engineering character determined by the physical system, hardware and software layers elements. In a consistent design process, the engineering decisions taken along all stages are in a connected chain of decisions governed by the requirements in the first stage. In the first two stages, a useful tool to organize the sets of requirements and performance measure is the 'House of Quality' matrix of the Quality Function Deployment (QFD) method [26] [27]. The QFD process aims converting non-exact information (requirements, demands) into technical solutions in the product, and is widely applicable for services and product planning. In a QFD model, the correlation between qualitative requirements and measurable performance indicators can be mapped, and an order of importance for both, ultimately determined by the project clients' viewpoint, can be derived. The technical specifications defined in the second stage can be stated as the target values for the performance measures selected for the system.

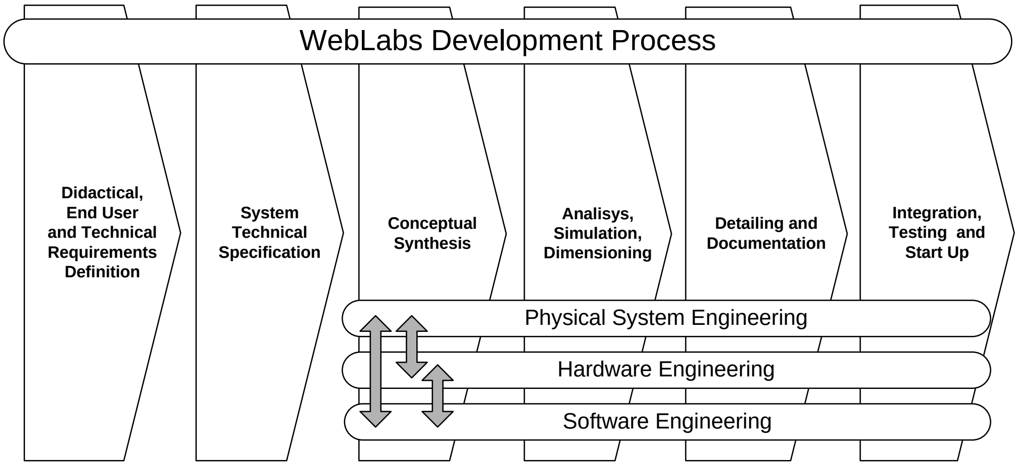

FIGURE 2
STAGES OF THE DEVELOPMENT PROCESS.

Didactical, End User and Technical Requirements Definition

Some authors that have analyzed requirements for WebLabs [8] [28] [29] [30]. A plurality of contributions and approaches characteristic of emerging concepts is evident. Fulfilling a consistent set of requirements, which have a rather qualitative format, in the beginning of a new design, is a challenge due to the degree of uncertainty at that time.

Therefore, the main objective in this initial stage is to clearly define the design task by collecting and organizing an as-complete-as-possible set of requirements. In particular, the underlying educational purposes of the project related with the didactical ends of the WebLab lead to important requirements.

Group	Requirements sources
Didactical	Experiments knowledge background Experiments context and objectives Didactical sequencing Commands and results cognition Expected experiments results
End User	Cognitive interface Dynamic physical system visualization Interaction modes and elements User actions guidance Constant (and enough) system performance Language options Multi-platform access Compatibility with other e-learning systems
Technical	Remote experiment setup Laboratory restrictions Applicable safety norms and procedures Experiment data formatting and presentation Calibration, maintenance and testability Upgradeability and scalability Maintainability (local and remote) Parallelization of tasks Time availability and use scheduling User access and permissions management Support to simultaneous multiple users Support to collaborative work Local technician interventions

TABLE 1
REQUIREMENTS SOURCES IN WEBLABS' DESIGNS

Table 1 contains a set of requirements sources to be taken into account in a WebLab design. The virtually unlimited WebLab applications is the reason to organize the requirements sources in a rather expanded sense, which can readily suggest the more specific demanded qualities in given development context. The process of requirements definition should be object of a multidisciplinary vision, including the final customer (external clients) and system engineers (internal clients).

System Technical Specification

The technical specification expresses the desired performance, in quantitative measures. At this stage, no particular solution has been selected, so that the performance measures should not be compromised with any particular solution. The requirements already defined are a start point for the specifications statement, but the fact that those are qualitative must be kept in mind to avoid misstatements. In Table 2, a proposed, non-definitive set of sources is given to guide the elaboration of the technical specifications list for a project.

Sources of Technical Specifications
Network connection bandwidth
Local and remote computers capabilities
Videoconference quality measures
Variables ranges and update rates
Allowable uncertainties for measured and output signals
Experiments limitations and execution times
Simultaneous independent or collaborative users
Physical system limitations and restrictions
Energy usage
General safety
WebLab availability

TABLE 2
COMMON SOURCES OF TECHNICAL SPECIFICATIONS IN WEBLABS' DESIGNS

At the end of the project, the specified performances should be verifiable. An indication of the means of test can be added to the specifications list.

Conceptual Synthesis

A macro framework of sub-systems usually present in a WebLab is shown in Figure 1. The user interface in the remote computer and the on-line videoconference link to the experimental system are the interactivity elements available to the remote user. The local computer is in charge of a server software application which is connected to the input/output hardware components, responsible for the digitalization / interfacing of sensors signals and for the output signals necessary for the actuation tasks. Part of the actuation is dedicated to the experimental system configuration, as this is necessary to setup a given experiment without the need for a local operator intervention. The software application ensures the safety of the commanded actions, to avoid misuse of the system or accidents. Physical safety elements must be present too, in accordance to safety norms.

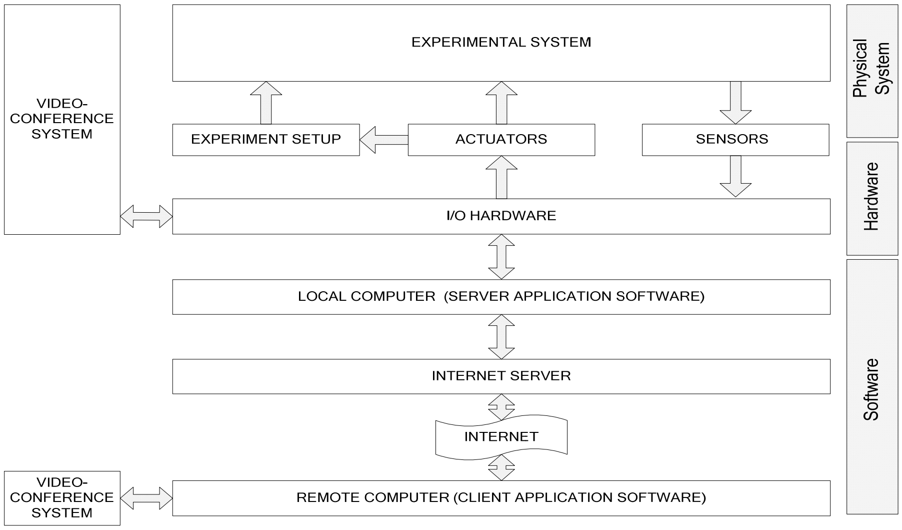

FIGURE 1
MACRO WEBLAB FRAMEWORK.

The experimental system often is an existent process plant to which will be added the remote operation functionalities, but can be specifically designed for a given project as well.

This macro framework is helpful in establishing a functional structure for the WebLab. The hierarchical functional decomposition is a valuable tool to map existent interactions amongst the WebLab sub-systems, and is context dependant, i.e., depends on the scenario of each project. The benefits of the decomposition rely on the fact that it's easier to solve local, small sub-problems, and then integrate the sub-solutions, than look for an overall solution directly.

The functionalities of a WebLab system are implemented through software. Designing the algorithms and integrating them into the system requires programming skills and, according to the degree of complexity, software engineering tools. Algorithms can be implemented in many computational languages, but the use of object-oriented languages may lower the efforts in building user interfaces, and if high level driver software functions are available, then integration with hardware will be easier too.

Several modern technologies can be selected to support both server and client applications over the Internet, each one having advantages and limitations. The following are some software platforms to be considered in WebLabs projects: ActiveX, Java Applets, (Adobe) Flash, Assincronous Javascript and XML - AJAX, and (National Instruments) LabVIEW^{TM}. The architectures growing in use are the Service Oriented Architectures (e.g. SOAP) and the Resource Oriented Architectures (e.g. REST). The REST architecture is becoming popular in Web Services programming, due to it has less protocol layers to deal with [31]. Also, there may be firmware code to run on specially designed micro-processed hardware, where the most convenient combination of language and platform (e.g. Assembly / C on microcontrollers, VHDL / LabVIEW^{TM} on FPGA real time hardware) for the stated requirements can be chosen.

Particularly, the LabVIEW^{TM} platform has a remarkable potential in the implementation of WebLab projects, and is being chosen for it in many of referenced WebLabs applications. Programs in the G language are called ‘virtual instruments’, and are conveniently organized in front panel, block diagram and connector pane. The block diagram is a graphical interconnection of blocks and terminals through wires that determine execution precedence, in a singular style of programming that can render better results if some design structures recommendations are followed [32] [33]. Front panel objects include graphical controls and indicators readily usable, fastening the user interface development. Allied to an easy integration with all major standards of modular hardware equipment and interfaced instruments, LabVIEW^{TM} is very suitable for a multitude of applications in test automation and R&D activities. The concept of using computers to design instruments with programmable functionality to provide customized solutions, in opposition to traditional, fixed-functionality interfaced instruments, became known as ‘Virtual Instrumentation’ [34] [35]. Recently, REST functionalities were added to LabVIEW^{TM}, and these, together with the integrated web server configuration capabilities, make the publication of applications over the Internet a task to be done through high level programming majorly.

Once the system's functional structure is defined, each sub-function can have optional local solutions associated. In Table 3, some usually present top-level functions are given as an initial, general proposition for the functional decomposition, as they consider relevant aspects typically present in WebLabs. Further subdivision of these in higher order sub-functions, in the context of the project, is necessary. The functional structure must be checked to verify its completeness, and should support the features implicit in the requirements and quantified in the specifications.

Layers	Top Level Functions
Physical System	Mechanical sustentation/positioning of sensors/actuators and interactive devices Supply energy Supply materials/substances Protect system components from potential hazards Prompt emergencies treatment
Hardware	Condition and transmit input sensors signals Output and amplify actuation signals Position the visual take of the physical system Live reproduce and capture audio / video Exhibit system state locally and remotely Computational capacity to execute the local (server-side) software applications
Software	Initialize and configure WebLab systems Manage I/O signals with hardware components and interfaced instruments Run experiment(s) algorithm(s) Manage experiment data exhibition, storage and remote retrieval Process acquired data into meaningful information Manage local network and web data transfers Calibrate sensors Ensure safe operation

TABLE 3
FIRST-ORDER FUNCTIONS IN WEBLABS' DESIGNS.

The next step is the association of solution principles to the partial functions in the functional hierarchy, inserting them in the left column of a matrix to form the so called Morphological Chart. The objective is to map the possible solutions to each partial function, and make explicit possible combinations. Eventually, polyvalent solution principles can simplify the process by attending groups of sub-functions simultaneously. This process is based on the creativity, experience and reasoning capability of the design team – good motives to have multidisciplinary design teams.

Some productive recommendations can be made for each of the three layers:

- Physical System layer: prefer standard mechanical / electrical accessories and frames, avoiding expensive special local designs whenever possible;
- Hardware layer: choose from easy-to-integrate I/O modules amongst the manufacturers options; prefer components-of-the-shelf and open-standard equipment; consider interfaced, wired/wireless instrumentation, re-programmable and permanent-firmware options;
- Software layer: think the solution principles as finite-functionality software routines and software architecture principles; balance the benefits of open-standards and proprietary systems, comparing their values to the project case.

After the morphological analysis is concluded, alternative conceptual solutions may be assembled by looking for feasible combinations of functions solution principles. There may be elective functions, present or not in the final configuration. The compatibility amongst the selected solution principles across the three layers must be checked at this time, as a panoramic view of the system is already possible and any changes still can be made at a low cost.

Next decision is the best concept selection. For a non-biased comparison of the alternatives, a leveraged depth of detail in the description of each conceptual candidate model must be kept. Using a decision criteria set (e. g. the project requirements, weighed by the project client’s perceived importance from the QFD method), the alternatives (conceptual solutions) can then be compared through Pugh’s Method for Alternatives Evaluation [36].

The decisions taken up to this point are determinant of the overall cost of the project.

Analysis, Simulation and Dimensioning

Once the concept is defined, the system must be submitted to the applicable engineering analysis and simulations, in order to be refined and optimized to the final form. The compliancy with applicable engineering technical and safety norms must be checked. Functional prototyping can be used to test performance. Eventually, the results of these evaluation and optimization tasks may feedback the concept.

- In Physical System: mechanical and electrical components and/or integrated sub-systems analysis, simulation and dimensioning; materials selection (by the required properties); components and/or integrated sub-systems layout optimization; tolerances and geometry optimization of parts to be manufactured;
- In Hardware: I/O sampling/update rates definition and optimization considering channels count and data transmission buses/lines; balance computational power and software application demands; watch for noise sources; dimension the network devices for the specified bandwidth and computers processing speed /storage capacities;
- In Software: prototyping of a simplified functional application; use of simulated external signals to test the prototype application; verification of I/O signals and interfaced instruments integration; progressive inclusion and debugging of sub-routines (adequately architect the software structure and sub-routines for scalable usage and maintainability, both remote and local); sub-systems firmware/software verification.

Detailing and Documentation

Detailing and documentation ensure the correct execution of the project and are a future reference for maintenance, modification and upgrade. Engineering standards should be used for appropriate documentation. Also, documentation is necessary to allow eventual intellectual property protection and user manuals elaboration.

- In Physical System: Mechanical/electrical detail drawings; cabling drawings;
- In Hardware: local I/O and network connectivity schematics; sub-systems specific hardware configuration parameters; sub-systems integration and connectivity schematics;
- In Software: network configuration parameters; software/firmware applications code documentation (server and client); version tracking and control.

Integration, Testing and Start Up

In the last stage, sub-systems are progressively assembled, tested and validated.

As WebLabs are intended to require minimal human intervention, the operational safety must be thoroughly tested. Network protection mechanisms, responses to hazardous combinations of commanded actions and to faulty conditions must pass extensive ‘stress tests’. Alarms and warnings must correctly activate in abnormal conditions, alerting responsible personnel through appropriated means.

It’s recommendable that all subsystems be individually tested before integration. Extensive testing has for objective to verify and ensure the performance of the implemented functions, both individually and collectively. If there are many combinations of functioning modes possible (e.g. many experiments variations), a test plan can be useful to minimize time and cost. Despite the careful conduction of design process, some reworks may be pointed necessary by the tests results – from minor adjustments to concept improvements.

Thoroughly tested, the system is ready for the start up, receiving the final adjustments. The verified performance can then be compared to the project's technical specifications.

Finally, learned lessons are used as feedback to the design process.

GENERAL DISCUSSION AND CONCLUSIONS

The proposed model is an effort to organize the elements and information typically present when architecting a WebLab application. The cascade form of prescribed stages and respective actions doesn't let explicit the common design loops found in any project, particularly in new ones – actually, the main expected benefit from the methodology is the reduction of design loops by avoiding the waste of efforts due to poor planning. The consideration of the sources of requirements indicated in the model should reduce the margin for errors in the early decisions taken by the design team. Naturally, the model cannot be read as a closed receipt, and rather be used as a reference, with the emphasis on the prescriptions varying with nature of each project. The prescriptions may find higher or lower applicability from a design to another, and their usefulness is expected to be the highest when the design team isn't experienced, a not-so-rare situation, especially in the case of small organizations.

The systematization of actions is the key to reduce design loops. The thorough study of the problem in the early stages of the development requires an anticipation of resources, mainly in the form of design team productive time, and this must be well understood by the upstream managerial instances. Another important reason for loops reduction is the problem clarification resulting from the morphological analysis, exposing the field of options for the solution, improving the conflict detection possibilities. The mentioned importance is accentuated if the multi-disciplinary character of WebLabs is remembered, as decisions taken in one of the layers influence the others, given the inter-dependencies amongst physical system, hardware and software in the synthesis of a solution. Treating these layers separately is a common source of problems.

Only engineering aspects of WebLabs' design have been considered: cost analysis, project management issues and team experience have not, as these are dependent on the involved organizations profiles, project size and on a surrounding funding scenario beyond the scope of this work.

Further elaboration of the present model can be expected as experience gained in real projects is turned into additional improvements, continuing a research process that aims an optimized approach proposal.

ACKNOWLEDGEMENT

The authors want to thank Pontifical Catholic University of Parana and CNPq for supporting this research.

REFERENCES

[1] India Education – Educational Institutions resource Bank, http://www.indiaedu.com/distance-learning, accessed May 02, 2010.

[2] Ko, C. C., Chen, B. M., Chen, S. H., Ramakrishnan, V., Chen, R., Hu, S. Y., Zhuang, Y., "A large-scale web-based virtual oscilloscope laboratory experiment", *Engineering Science and Educational Journal*, April 2000, pp. 69-76.

[3] Yeung, K.; Huang, J.," Development of a remote-access laboratory: a dc motor control experiment", *Computers in Industry* 52, 2003, pp. 305–311.

[4] Song, Y.; Wang, T.; Eagleson, R.; Meng, C.; Zhang, Q.; "A low-cost internet-based telerobotic system for access to remote laboratories", *Artificial Intelligence in Engineering* 15, 2001, pp. 265-279.

[5] García-Zubía, J. & Gomes, L. (Eds.), "Advances in remote laboratories and e-learning experiences", Deusto Publicaciones, Bilbao, 2007.

[6] Ma, J. & Nickerson, J., "Hands-On, Simulated and Remote Laboratories: A Comparative Literature Review," *ACM Computing Surveys*, vol. 38, no. 3, 2006, p. 7.

[7] KyaTera Network, http://www.kyatera.fapesp.br, accessed May 02, 2010.

[8] C. Gravier, C., Fayolle, J., Bayard, B., Ates, M., Lardon, J., State of the art about remote laboratories paradigms – foundations of ongoing mutations *International Journal of Online Engineering* – Vol. 4, Issue 1, Feb 2008.

[9] Selmer, A., Kraft, M., Moros, R., Colton, C., "Weblabs in Chemical Engineering Education", Trans IChemE, Part D, *Education for Chemical Engineers*, 2007, 2: 38–45.

[10] Massachusetts Institute of Technology, "MIT iCampus. iLabs: Internet access to real labs - anywhere, anytime", http://icampus.mit.edu/ilabs, accessed May 02, 2010.

[11] Duro, N., Dormido, R., Vargas, H., Dormido-Canto, S., Sánchez, J., Farias, G. Dormido, S., Esquembre, F., "An Integrated Virtual and Remote Control Lab - The Three-Tank System as a Case Study", *Computing In Science & Engineering*, IEEE CS / AIP, August 2008, pp. 50-59.

[12] WebLab Deusto; https://www.weblab.deusto.es/joomla/, accessed May 02, 2010.

[13] RemoteLab FEUP – Porto University, http://remotelab.fe.up.pt, accessed May 02, 2010.

[14] Bernardi, M., Bley, H., Schmitt, B.,"Integrating a Mechatronics-oriented Development Process into Development Department", *Proceedings of the 37th CIRP International Seminar on Manufacturing Systems*,Budapest, Hungary, 2004, pp. 265-270.

[15] G. Andria et al., "Remote Didactic Laboratory 'G. Savastano,' Italian Experience for e-Learning," *IEEE Trans. Instrumentation and Measurement*, vol. 56, no. 4, Aug. 2007, pp. 1135-1147.

[16] Huang, S. H., Su, Q., Samant, N., Khan, I. (2001). Development of a web-based integrated manufacturing laboratory. *Computer Applications in Engineering Education*, 9(4), pp. 228-237.

[17] Albers A. Conceptual Design of Humanoid Robots. http://www.sfb588.uni-karlsruhe.de/publikationen/2002_20.pdf. Accessed May 02, 2010.

[18] Cross, N., *Engineering Design Methods*, 2nd. Ed, Chichester, Ed. Wiley, 1994.

[19] Pahl G, Beitz W., *Engineering Design*., 2nd ed., London: Springer-Verlag, 1996.

[20] VDI 2221, "Methodik zum entwickeln und konstruieren technisher Systeme und Produkte.", VDI – Verlag, Düsseldorf, 1985.

[21] Asimov, M., "Introduction to design: fundamentals of engineering design", *Prentice Hall*, New Jersey, 1962. [15] ASME RESEARCH, "Design theory and methodology: a new discipline.", *Mechanical Engineering*, 1986, pp. 23-27.

[22] ASME RESEARCH, "Design theory and methodology: a new discipline", *Mechanical Engineering*, 1986, pp.23-27.

[23] Back, N.; Metodologia de projeto de produtos industriais. Guanabara Dois, Rio de Janeiro, 1983.

[24] Back, N., Ogliari, A., Dias, A., da Silva, J. C., Projeto Integrado de Produtos – Planejamento, Concepção e Modelagem. Ed. Manole, Barueri, 2008.

[25] Mendes, L. A., Back, N., Ooliveira, G.H.C., Designing automated test systems: An adapted methodology inspired on Pahl and Beitz's systematic approach. *Robotics and Computer-Integrated Manufacturing* v.25 pp. 945-950, 2009.

[26] Terninko, J., "Step-by-Step QFD – Customer-Driven Product Design", Ed. St Lucie, 2nd ed, 1997.

[27] Akao, Y.; Mazur, G. H.; King, B.; Quality Function Deployment: Integrating Customer Requirements Into Product Design; Ed. by Yoji Akao, 2004.

[28] García Zubía, J. Lopez-de-Ipiña, D. Orduña, P. Hernández, U. Trueba. "Questions and Answers for Designing Useful WebLabs", *International Journal of Online Engineering*, 2006.

[29] Nedic, Z., Machotka, J., Nafalski, A., Remote Laboratories Versus Virtual and Real Laboratories, 33rd ASEE/IEEE Frontiers in Education Conference, Boulder, 2003.

[30] Prieto-Blázquez, J., Arnedo-Moreno, J., Herrera-Joancomartí, J., "An Integrated Structure for a Virtual Networking Laboratory" *IEEE Transactions on Industrial Electronics*, Vol. 55, No. 6, June 2008.

[31] Ngolo, M. A. F., "Arquitetura Orientada a Serviços REST para Laboratórios Remotos", Master Degree Dissertation, Universidade Nova de Lisboa, July 2009.

[32] Blume, Peter A., "The LabVIEW style book.", Upper Saddle River: Prentice Hall, 2007.

[33] Conway, J., Watts, S., "A Software Engineering Approach to LabVIEW™.", Upper Saddle River: Prentice Hall ,2003.

[34] National Instruments. Virtual Instrumentation and Traditional Instruments, http://zone.ni.com/devzone/cda/tut/p/id/4757, Accessed May 02, 2010.

[35] Sumathi, S., Surekha, P., "LabVIEW based advanced instrumentation systems.", Berlin, Springer, 2007.

[36] Pugh S., "Total Design: Integrated Methods for Successful Product Design.", *Great-Britain: Addison-Wesley Publishing Company*, 1991.