ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOPHA002

A MODEL-DRIVEN SERVICE-ORIENTED WIZARD-BASED MULTI-TARGET DEVELOPMENT KIT FOR SUPERVISION SYSTEMS *

C. F. Afonso[1], C. Larizza, Department of Electrical, Computer and Biomedical Engineering
University of Pavia, Pavia, Italy
S. Foglio, S. Gioia, M. Necchi, M. Pullia, Centro Nazionale di Adroterapia Oncologica, Pavia, Italy
S. Toncelli, Consultant on behalf of Centro Nazionale di Adroterapia Oncologica, Novara, Italy
L. Casalegno, Consultant on behalf of Centro Nazionale di Adroterapia Oncologica, Como, Italy
[1] also at Centro Nazionale di Adroterapia Oncologica, Pavia, Italy

Abstract

The operation of particle therapy facilities requires complex control and supervision systems, often spanning several software development environments, each containing a large number of applications. While the component-based development approach brings many benefits, the integration of the parts is still left to the initiative of each developer without a repeatable path that can be demonstrated when the artefacts have to undergo the certification process.

Besides the development of such control and supervision systems is not any more limited to a network based on fixed workstations, but mobile devices have to be taken into account introducing an extra need for enhanced security.

As part of the technological update of one of its development environments, the Centro Nazionale di Adroterapia Oncologica (CNAO) chose to build a development kit to face these new challenges.

This development kit is based on models of applications and services that are managed by wizards that configure the general layout and create the connections among the components so that the integration of the parts is performed in a substantially unique way.

The developer is responsible to choose among the available models that already include the integration to the mandatory services and use the wizards to create the application and the project that is able to build the executable.

Developers are still able to add the specific business logic and the required interactions; nevertheless, they will be directed in doing so by a set of 'hooks' present in each model that shall guarantee repeatability of behaviors also in this area of the work.

In this document, we examine the development kit, highlighting the most valuable aspects that enable to build easily certifiable applications running in a multi-target system. The services that support the actions performed by each application are described in a companion paper [1].

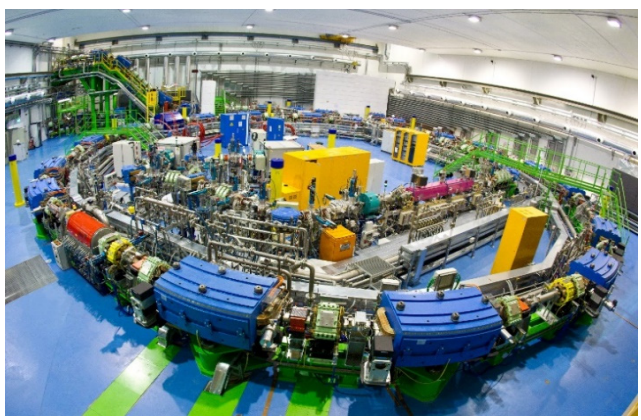
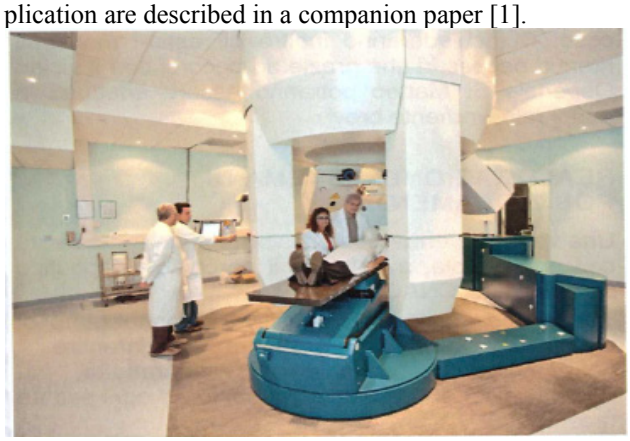

Figure 1: The CNAO accelerators and treatment complex.

MOTIVATION

CNAO (National Centre for Oncological Hadrontherapy) is a clinical facility, established and funded by the Italian Ministry of Health and the Lombardy Region, that uses hadrontherapy for cancer treatments. To date, about 2500 patients completed successfully the treatment.

Hadrons (basically protons and carbon ions) are characterized by a maximum of energy deposition at the end of their range and a sharp penumbra that allows to achieve a precise coverage of the target and an enhanced sparing of the surrounding healthy tissues. In addition, carbon ions are more suitable for the treatment of radio-resistant tumors.

The treatment has the aim to transfer a given amount of energy to a tumor. In doing so, the tumor is subdivided into several slices orthogonal to the beam direction. Each slice can be treated with several beams. Depending on the energy, beams have different characteristics. Thus, during

* This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement 675265

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOPHA002

each treatment, the physical conditions of the beam change many times. The beams are produced, maintained and delivered by means of particle accelerators. A cycle is the period of time the machine takes to produce a beam with planned energies, deliver the dose to the tumor slice and go back to the initial condition. The spill is the operation that transfers the beam from the machine into the tumor. A full treatment is composed of many cycles. Depending on the physical characteristics of the beam, the equipment has to be set into different running conditions. A running condition lasts, in general terms, the length of a single cycle. For each device, a mapping exists relating each beam characteristics with the equipment settings.

The main particle accelerator of CNAO is a synchrotron, a circular accelerator, of about 25 m in diameter (see Fig. 1). Inside the ring, the sources, the lines of injection and the linear accelerator are housed, so to realize a compact layout. The Electron Cyclotron Resonance (ECR) ion sources produce a low energy beam that is then injected into a linear accelerator where a first acceleration occurs up to 7 MeV/u.

Along the 78 m synchrotron circumference carbon ions and protons beams are accelerated up to 400 MeV/u and 250 MeV, respectively. Outside the main ring there are four extraction lines, about 50 m each, leading the extracted beam into three treatment rooms (see Fig. 1). In each of the two side rooms a horizontal beam is driven, while in the central hall both a horizontal and a vertical beam are directed. An experimental room is also under construction and shall be equipped so that the beam line can be devoted to research activities.

Because the plant activity affects human beings, building applications to control and supervise such target has to undergo a set of quality assurance activities that have the aim to demonstrate the consistency and the reliability of the artefact; additionally, evidence of the accelerator's correct behavior has to be provided to the regulatory body. Much stricter rules have been adopted by the regulatory body since the first construction of CNAO [2, 3]. Due to the fact the number of applications necessary to run the facility is quite high, the amount of work for upgrading or reproducing the center would be devastating without the help of an environment that helps to demonstrate and document the job being done.

Besides, the traditional way of supervising the activities in the plant, based on fixed workstation is changing in favor of more flexible tools such as tablets and possibly smartphones. These new devices impose a new approach to the security aspects and to the choice of development tools that are able to encompass all the different targets.

TARGETS

The goal was to create a development kit that can create end user applications to be used both on traditional devices and on mobile devices with different operating systems. We therefore chose a basic platform that natively offered this characteristic. The choice has fallen on Xamarin [4] in Visual Studio (MS). Xamarin allows creating applications in native format, starting from the same source code, on tools that have installed one of the following operating systems: Windows, Android, IOS (future use), MacOS (future use), Linux (future use) (see 3).

In addition, the services provided and introduced in [1] were made available to the two other development environments that have historically been used by CNAO for the construction of supervision applications, through appropriate interfaces:

- WINCC (Siemens/ETM SCADA)
- Labview (National Instruments)

Visual Studio (MS) has been selected for the characteristic of offering the possibility of building wizards directly within the production tool.

ARCHITECTURE

The present CNAO control system architecture is composed of 4 layers (Fig. 3Figure). The fourth and lowest layer contains the device controllers. The third layer aggregates the outcome of the underlying layer and creates a common interface based on OPC technology. The second layer contains the data that belongs to both the real-time and the project and configuration databases. The first layer is devoted to the end user applications [5]. From the point of view of the applications the second and third layer are optional and can be by-passed. Thus, the end user applications in general can communicate with a database or an OPC-UA [6] based controller (See Fig. 2 and Fig. 3).

The applications built with the development kit described in this paper, that we named CF2020, shall be part of level 1.

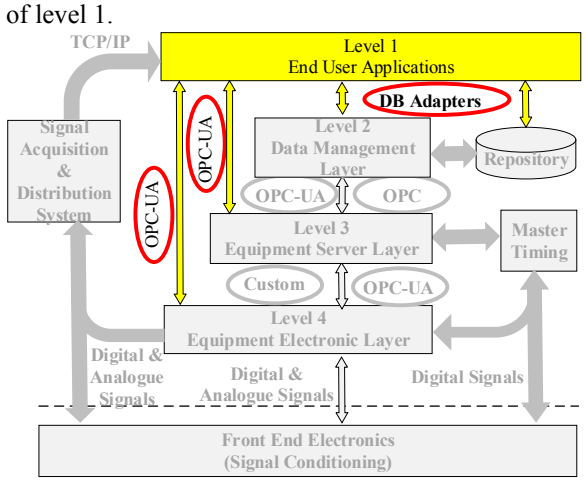

Figure 2: CNAO Control System Layered Architecture.

When building software for plants such as the CNAO medical center, several issues have to be taken into account: security and safety, interconnection protocols, high robustness for unpatrolled subsystems,

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOPHA002

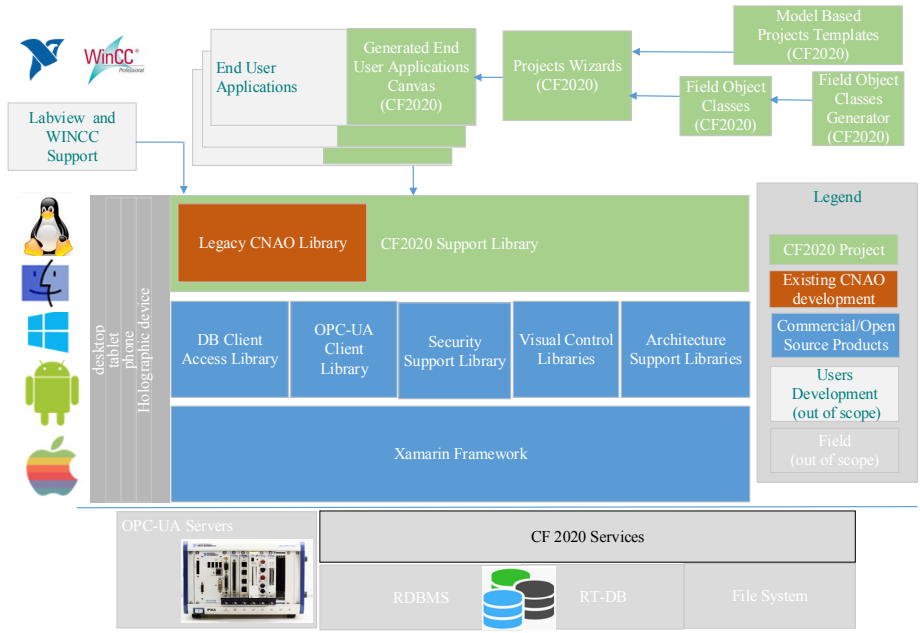

Figure 3: CF2020 Development Kit Architecture.

homogeneity of development regardless the devices and the operating systems on which the software will run, long term products life cycle encompassing decades and not simply years or months.

CF2020 is an innovative framework for building such mission critical software allowing nevertheless to keep a low time to market factor.

The applications produced using the CF2020 approach shall adhere to the MVVM paradigm [7].

The development kit includes (see Fig. 3):

- A set of model-based project templates. The templates are composed of:
  - Project files that are used to compile and build the final application.
  - C# source files, XAML source files and code snippets that implement the models: these files contain parameters that shall be translated into code by means of the information supplied by the developer when using the project wizards. The files shall be grouped into View, ViewModel and Model directories in the final project as recommended by the MVVM paradigm.
  - Files containing run time configuration settings for tuning the application in different target environments (e.g. DB name).
- A field class generator that is able to build C# classes that represent in a suitable way the field objects (i.e. database tables and views and OPC-UA server families: an OPC-UA server family is a group of OPC-UA objects having the same variables). The generated class files are kept in predefined libraries where can be found by the wizards.
- Project wizards, which are able to build the final project on the base of the chosen model by selecting the suitable files, substituting the parameters inside the files using the information supplied by the developer, generating the suitable code, building the run time configuration files and creating the project directories containing the generated files.
- A set of libraries that shall support the implementation of the behavior defined by the models, the client access to the field and to the ancillary services such as file system access and logging system access and take care of the security aspects. These libraries depend on other widely used support libraries that are open source or commercial.
- The development kit is completed by a set of web services that supply basic functionalities such as access to remote databases and file systems. The description of such services is the topic of another paper in this conference [1].

SECURITY

Applications that run in a distributed and open environment, possibly from mobile tools, need an advanced authentication and authorization system. Each model included in CF2020 has the ability of requesting the identification of the user. Once authentication is performed, these models are able to display only functionality that the user has permission to access via authorization tokens (i.e. enabling buttons, menu items, fields or full panels). The authentication and authorization of clients and users is performed using the OpenID Connect standard, which is based on OAuth 2.0 [8]. The details of the adopted mechanism are outside the scope of this document: the service is introduced in a companion paper in this conference [1].

MODELS

Model Driven Architecture has been an approach to building applications for a long time [9, 10]. It is an extension of design patterns to encompass the whole application. Models require a deep understanding of the requirements and the commonalities among applications that address a

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOPHA002

common domain. Thus, a survey of most of the applications presently running in the CNAO Control and Supervision System has been done before defining and implementing the tools that generate the application canvases. About 100 applications have been taken into account. The applications were grouped in the following categories:

- Applications managing data mainly contained in relational DBs or file systems for configuration and data management purposes.
- Applications managing the core business workflow, i.e. applications that are aimed to obtain the delivery of the treatments on the patients or applications that are able to show the amount of work done and the amount of work to be done.
- Applications managing a single device in a family (executing commands and displaying indicators of a single device).
- Procedures, i.e. sets of operations performed on groups of devices with a given sequence to obtain a change in the state of the plant or data to be analyzed in order to evaluate the state of the plant.
- Applications to manage groups of devices (executing commands and displaying indicators in a group of devices).
- Applications to show the status of group of devices (summaries of the status of devices).

For these categories of applications several models have been devised. In general, models are composed of pages on which data is displayed and managed, a navigation system among the pages, a set of menu items for each page and possibly a toolbar to execute standard commands such as search and filters. A page can be configured having several panels. The models define also the connection to all the services needed to manage the data of the applications. The models are briefly summarized as follows:

- A model that allows navigating between groups of entities of different types. For each group, a list of objects is displayed. Selecting an object displays information relating to that object. Objects can be added and deleted and the information relating to each object can be modified. The objects can be represented by information residing in an instance of a single DB table or in instances of different tables, one of which represents the main table and the others the details. The information on the objects can reside on a single page or on different pages. It is possible to order, search and filter the information presented. Information can be represented in visual controls, tables or graphs
- A model that allows data to be transferred from files contained in file systems to database tables. The model allows users to select a set of files. Afterwards, the model indicates if the data contained in the files are already present in the database, and if so, whether their values have been changed.
- A model that allows viewing and executing commands on a single instrument.
- A model that allows performing a series of commands on properties of field objects. The model allows selecting the set of objects and properties to operate on, to start the execution of the commands and to view the status of the going on operations for each object. The operations on the objects can take place in series (i.e. one object at a time) or in parallel by performing a set of processes at the same time. The execution of the commands can be suspended and resumed. This model is able to group operations into the following tasks: initialization, execution, analysis, completion and finalization. The model maintains a log system for each operation.
- A model that allows viewing the status of the operations to be performed on different physical resources. This model represents the status of each operation in a calendar for each resource, assigning different values and colors according to the state of each operation. The state of the operations is kept in a database where other applications can update the values.
- A model that allows managing a state machine. The machine can be started, suspended, restarted and stopped. The machine status data can be represented in visual controls, tables, graphs. The advanced version of the model allows adding detailed panels to the main page that shows the status of the application, therefore allowing for the management of extra data other than the state machine.
- A model that allows viewing the status of a large number of components and perform commands on them. The model can contain multiple pages and each page can be split into different panels to allow a clearer view of the data that can be displayed in visual controls, tables or graphs.
- A model that allows executing procedures on the system. The model is the composition of the previous model that constructs the state machine and the previous model that executes commands on field objects.

WIZARDS

The project wizards allow the user to select the desired model, to select the targets operating system on which the application will run, to choose the pages that will be included in the final applications, assign the model configuration parameters and define the menu and the menu items that will be included in each page [11, 12] (see Fig. 4).

The types of page(s) that are included into each model depend on the model. If the pages are enabled to contain detail panels the developer can add as many panels as needed and possibly choose their layout.

The outcome of the wizards is a skeleton of an application that is already a running executable. Consequently, the developer knows that all the connections and interfaces are respected and the basic behavior supplied by the chosen model is fulfilled.

The skeleton contains ‘hooks’ to which the developer can attach the code specific to the business logic of the application.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOPHA002

The wizard is not a one-way application. The developer can later modify the generated project adding pages, detail panels and menu items where available. Additionally, an undo mechanism has been developed, allowing developers to return the generated application to a previous version of the project whenever required.

The development kit can be extended adding new models and new wizards when the target environment requires new types of applications [13]

A set of test cases for validating the common toolkit behavior is supplied. These tests shall be always executed when a new application is built to verify that for any reason the basic functionality have not been disrupted.

Then the procedure already devised in CNAO that foresees the production of a test plan, test specifications and test reports shall be adopted [14].

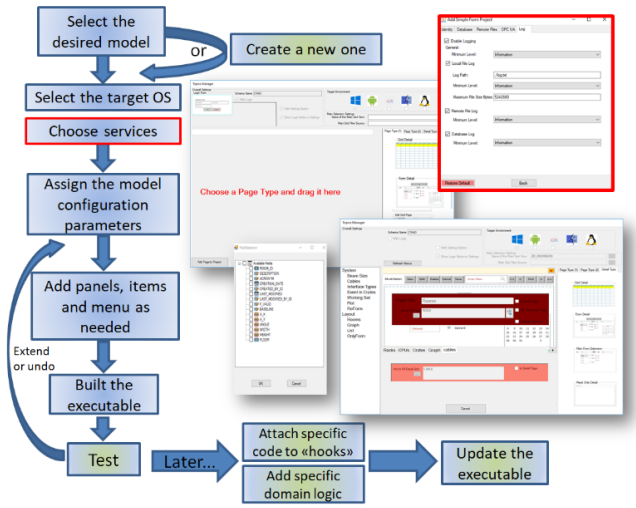

Figure 4: Building CF2020 Applications with the Wizards.

CONCLUSION

An environment such as the CNAO accelerators system involves a sizeable effort to validate and certify the software that manages the plant. Additionally, the introduction of mobile devices has extended the need for security and requires a development kit targeting many different devices.

A wizard model-based supported architecture can reduce the time required for validation and certification and allows creating projects that build solutions for different devices using always the same source code. The main advantages of this approach are:

- The integration of the architectural modules and the modules created by the wizards is made in a proved, validated and documented way.
- The developer has a well-defined and constrained path to integrate, validate and document the business logic of the application.
- The risks are mitigated by reusing software and building new software following rules enforced by automatic code generation.
- The wizards automatically take into account the differences of the targets and frees the developer from the task of adapting the code to the end devices.

REFERENCES

[1] C. F. Afonso *et al.*, "Integrating Mobile Devices Into CNAO's Control System, a Web Service Approach to Device Communication", presented at the 17th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'19), New York, NY, USA, Oct. 2019, paper MOPHA003, this conference.

[2] EN ISO 13485, Medical devices -- Quality management systems -- Requirements for regulatory purposes, 2016.

[3] IEC 62304, Medical device software – Software life cycle processes, 2016.

[4] C. Petzold, *Creating Mobile Apps with Xamarin Forms*, Microsoft Press, 2016.

[5] L. Casalegno, M. Pezzetta and S. Toncelli, CNAO General Control System, Organisation Document, Fondazione CNAO, 2004.

[6] OPC Unified Architecture Specification Part 1: Overview and Concepts, OPC Foundation, Scottsdale, AZ USA, 2017.

[7] The Model-View-ViewModel Pattern, `https://docs.microsoft.com/it-it/xamarin/xamarin-forms/enterprise-application-patterns/mvvm`

[8] User Authentication with OAuth 2.0, `htpps://oauth.net/articles/authentication/.`

[9] S. J. Mellor, K. Scott, A. UHL and D. Weise, *MDA Distilled*, Addison Wesley, 2004.

[10] M. Fowler, *Patterns of Enterprise Application Architecture*, Addison Wesley, 2012.

[11] C. Afonso and L. Casalegno, CF2020 Wizards Developer's Guide, CNAO Foundation, unpublished.

[12] C. Afonso and L. Casalegno, CF2020 VS Wizards User's Guide, CNAO Foundation, unpublished.

[13] C. Afonso and L. Casalegno, CF2020 Ctrl VS Wizards User's Guide, CNAO Foundation, unpublished.

[14] IST 707: Validazione Software di Prodotto, CNAO Foundation, unpublished.