Lean Automation (LA) – making the life better

Abstract *– Lean automation is a methodology developed by Spin for the integration of automation systems of electrical sites (substations, power plants, wind farms, etc.) that reduces the process cost, from its parameterization to its commissioning and increases the quality and robustness of the solution achieved. It can be used, not only for creating new applications in energy companies (GTD), but also to generate components which are the best practices for the use of IED families, such as protection relays, wind turbines, asset management IEDs, etc.*

Key Words: SCADA – Lean Automation – Communication Protocols – Systems integration - Substation Automation – Power Plant Automation – Wind Farms Automation – Asset Management.

1 INTRODUCTION

The integration of automation systems, using SCADA-type software, have among its costs associated with SCADA and IEDs parameterization activities, laboratory and factory environment tests, and commissioning. Using this methodology, as more standardized is the customer culture, lower will be the costs. This standardization means process mechanization, however, it depends on the culture of each company. Thus, if the target company is an energy utility, there will be several standards to be followed:

- Screens: the layouts, number of screens to represent a substation, equipment's designs, associated colors depending on the state of the equipment and voltage levels, bay types representations, screens navigation, substations standard reports, historic data queries, standard command screens, etc.
- Tags: rules for naming a site's tags based on its region, substation, bay type, function of the point, etc.
- Alarms /events: points that cause alarms / events, messages format, recognition standards, priority, etc.
- Historical records: rules to register both alarms / events in historian archives, and analogue variables, etc.
- Protocols used: protocols normally used for center to center communication, center to substation, substation to IEDs, etc. Due to the protocol used, rule for naming the IEDs and addresses of points within the IED. In the case of IEC 61850, is there standardization of the nodes used considering the point type?
- Databases: Does the company have a standard relational database? Is this a corporative database? Is there a database manager?
- Reports: what reports are used by the company, its frequency, and its formats?

Electrical automation systems integration companies, which provide services to several concessionaries, must maintain the different standards and different libraries.

The IEDs for electrical applications, such as relays, meters, wind turbines, circuit breakers, etc., are updated every year, presenting new models, with new functionalities that are almost never used in their fullness, due to the lack of knowledge of the integrators, clients and consultants. Generally, the basic of each IED is used by a culture of "copy / paste". On the other hand, the best practices when using the IEDs, is in the experience of some employees and not in the company. The Lean Automation (LA) technology, in addition to reducing costs and ensuring the quality and robustness of the solution, also keeps the knowledge of the best practices of using an IED in the company and not only with the technicians.

Thus, LA reduces costs of deploying an automation system, reducing the number of man-hours required, the level of expertise of the technicians involved, the duration of the project, as well as the cost of travels, hotels, rent cars, etc. More than these, the result has quality and robustness and the knowledge of the process is also in the solution and not only with the project´s technicians. The methodology can be repeated and improved over time, guaranteeing technological evolution without discontinuity of knowledge.

2 METODOLOGY DEVELOPMENT

2.1 Past inheritance – Old SCADA ActionView

Spin, the company responsible for the development of this methodology, emerged as a startup from the University of Brasília, at the end of 1992. At the time, the company's objective was to develop a SCADA software for niche markets. The result was a SCADA software oriented to the electric market. This software

* simoes@spinengenharia.com.br

was not based on any other, being created from the programming knowledge of the authors and the utilities clients who demanded a series of functionalities that would attend to their company's culture. Utilities who assisted in the creation of the SCADA´s requirements were CEB [1], CELPE [2], CEEE, CEMAR, CESP, CTEEP, DME, EMAE, CHESF, etc.

This software's organization didn't follow the normal pattern of a SCADA system and some concepts were created quite differently from what is found in other market solutions. An example of that are the analog and digital variables files that, unlike other SCADA, concentrate all the information at the attributes of each register, instead of having different table interrelated (Alarms, events, historic, etc.). For example, at the register of a digital point there were attributes that identified its alarm condition, the type of recognition, the address of the source variable, whether it was I/O, command conditioning, alarm conditioning, the trigger for the history, etc. From the way the relational database was created, the information was grouped in such a way to fulfill the functionalities demanded by the customer. This organization had advantages and disadvantages, and among the advantages, was the lightness and speed of the software, allowing an application with more the 150 thousand I/O points and hundreds of communication channels, to be executed in a laptop-type microcomputer. Another advantage derived from the system architecture was the creation of the "base groups" concept, which could be associated to IEDs, creating applications that were the best practices to utilize an IED. Those "base groups" had all the information of a bay, for example, the points address, the alarms and events associated, the triggers to record historic registers, tag names following the company's culture, etc.

2.2 The SCADA evolution

Over time, the disadvantages of the SCADA have outweighed the advantages, requiring a change. With the emergence and quick evolution of the internet, graphic libraries with state of the art objects (WPF), architectures of software oriented to services (WCF and SaaS), IoT, Dotnet environment, HTML5, etc., demanded the creation of a new product that would incorporate all these new trends / technologies. In this scenario a new SCADA was developed, without legacy code, all developed in Dotnet environment. The first application, in 2012, was the first substation in Brazil using iOS tablets as mobile IMM of the SCADA, running over a wireless network [3]. This new software was built with last generation technology, considering functionalities of other SCADA software of the market, but in the end, there were some missing features that anyone who had used, did not forget.

2.3 The best of two worlds

Missing functionalities and creative leisure led to the conception of the LA methodology applied to this SCADA. It was developed in a few steps, but in this article, we will show only its current state.

A LA application cell is called a component. An empty SCADA Project already comes with a component that generates several reports common to electrical applications such as current alarms, day events, operation logs, variables trends, etc. This component can be changed depending on the user's automation culture, and its development was all done in C#. An integrator with different customers, can have different empty projects, appropriated to each client's culture.

To this empty project you can add other components that give different functionalities to the application. Thus, for example, we present some developed components below:

– Utility A component: generate, in a few hours, a standard substation of the utility "A". In this component are defined the different types of bays utilized in substations of the utility and, automatically, all points of the application and standard screens of it are instantiated;
– SE Industrial component: generate, in a few hours, a typical industrial substation. In industries unlike utilities, the substations are associated to the relays model that control them. This application was made thinking about a relay manufacturer that supplies its relays for the industry and adds the SCADA solution that is parameterized in a few hours, without errors, minimizing the solution cost.
– "Wind Power" component: generate, in a few hours, an application that monitors wind farms, using two types of wind turbine, Suzlon S95 and GE/Alstom ECO-122. The component can be changed to include other wind turbine models. These wind turbines were selected because they deal with the automation of real complexes, which used these turbines [4].
– "Asset Monitoring" component: fast generation of an asset´s monitoring module, where the user defines the various IEDs that will be used, for example, to monitor substation transformers and as a result an

application, that reads all the IEDs used in the monitoring, is generated and it also serves as a gateway, sending the data to a centralized asset management application that monitors several substations. In addition to the gateway, the application generates HTML5 screens that can be locally accessed, at the substation, or even remotely, if the IP address is made available at the gateway.
IEDs from different manufactures can be used to monitor the assets of the substation. In case of communication failure with the central computer, data from several weeks can be stored at the gateway, to be sent when the communication is reestablished. In case of failure at the IEDs or sensor that this application controls, the gateway software has routines that help to identify the failure.

– "FLISR": The configuration of a FLISR application was automated using LA methodology, as can be observed in the linked video. In this case, a symbol library with all the elements of a distribution FLISR network was generated. The Toolkit module of the ADMS network generates the logical loop schemas and they are automatically imported, and their screens are created at the SCADA, as well as their instantiated tags via the deploy extension. This application was implanted in two concessionaires of the Neoenergia group [5].

After the creation of a component, is easy to adapt it to other similar application, for example, from the component of a concessionaire it's possible to easily generate components for other concessionaires. From the Wind Power component one can easily include other models of wind turbines.

2.4 The technology fixation in the solution

A component allows the user to generate applications in record time, minimizing errors and maximizing quality. More than this, LA ensures to keep the technology in the solution and not only with the company technicians. Thus, for example, the "Wind Power" component implements monitoring of wind farms, meeting all the requirements of the regulator agents, like ANEEL and ONS. Its creation is based in the implementation of more than 1 GW of Eolic energy in different wind farms.
A component can be used by a manufacture or a consultant to set the best practices for using any equipment, like a relay. Thus, the developer uses all the equipment available functionalities to make the best use of it and keep this IED´s knowledge use in the component that will be used in dozens of projects.
Considering assets management solutions, such as transformers monitoring [6], nowadays there are IEDs from different manufacturers used in a decentralized architecture monitoring this equipment. Many of these IEDs are already shipped on the transformer. In addition, many relays used for transformers protection, also have the function of monitoring parts of the transformer. Using this basic idea, the "Asset Monitoring" component was created, this component is generated from the inclusion of several components where each one treats a manufacturer's IED.

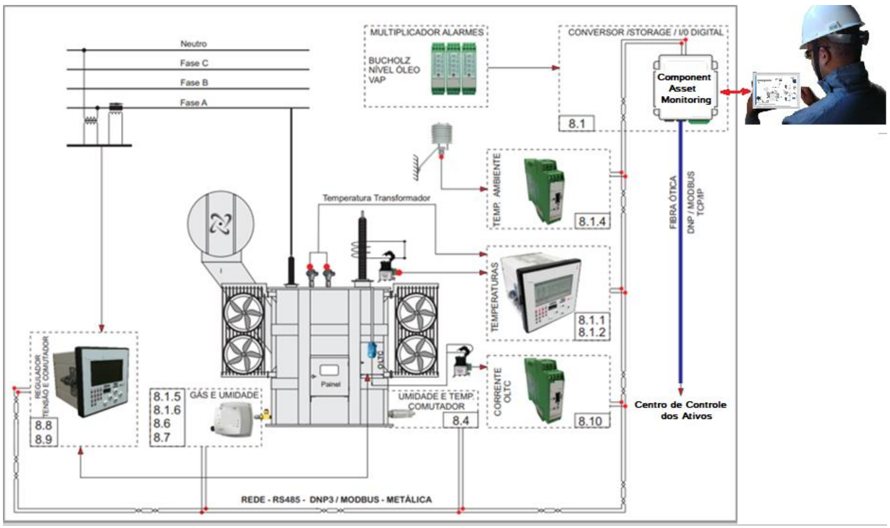


Fig. 1. Example scheme of the "Asset Monitoring" component

3 IMPLANTATION OF THE METHODOLOGY TO GENERATE AN APPLICATION

3.1 Summary of the implementation of an LA application

This methodology should be used in the creation of base components that will generate functionally similar applications of SCADA systems. To facilitate understanding, we will use one base application to generate wind farms.

A wind farm is composed of several wind parks, a wind park is composed of an anemometric tower and several wind turbine circuits. In most of the deployments we participate in, a circuit have between five and six wind turbines and a park usually has two circuits (10 to 11 wind turbines) and an average power of 25MW.

This example application is based on deployments made and we will use two kind of wind turbines and identical anemometric towers in all parks.

3.2 Identification of application data structures (templates)

In this base application we have five database structures:

- Structure 1: Wind turbine ECO-122 (GE/Alstom);
- Structure 2: Wind turbine S95 (Suzlon);
- Structure 3: Anemometric tower xxx;
- Structure 4: GE/Alstom Wind turbines farm. An ECO-122 farm has a single data concentrator with OPC protocol that contains an all wind turbine structures (structure 1 repeated for the n° turbines existent in the farm) and the data structure of the farm, which contains global data associated to a group of wind turbines (structure 4);
- Structure 5: Suzlon wind Turbines Park. Each park has a distinct Modbus data concentrator, with global park data and data from each wind turbine associated to that farm.

For each data structure we will create a template that has all the points of this structure, identifying for each point the rule of creation of its name, the alarm conditions, whether the variable goes for the history and the associated trigger, and the address of point on the IED where it's located.

In addition to the structures that describe points associated with IEDs, we may have other logical structures with calculated logical point groupings.

If, instead of a wind farm, we had a substation, the structures would be the substation bay type, such as line bay, transformer bay, capacitor bank bay, feeder bay, etc.

The figure below, for example, shows 3 templates, one for the anemometric tower data structure, one for the data structure of the ECO-122 wind turbine and one for the wind farm global data.

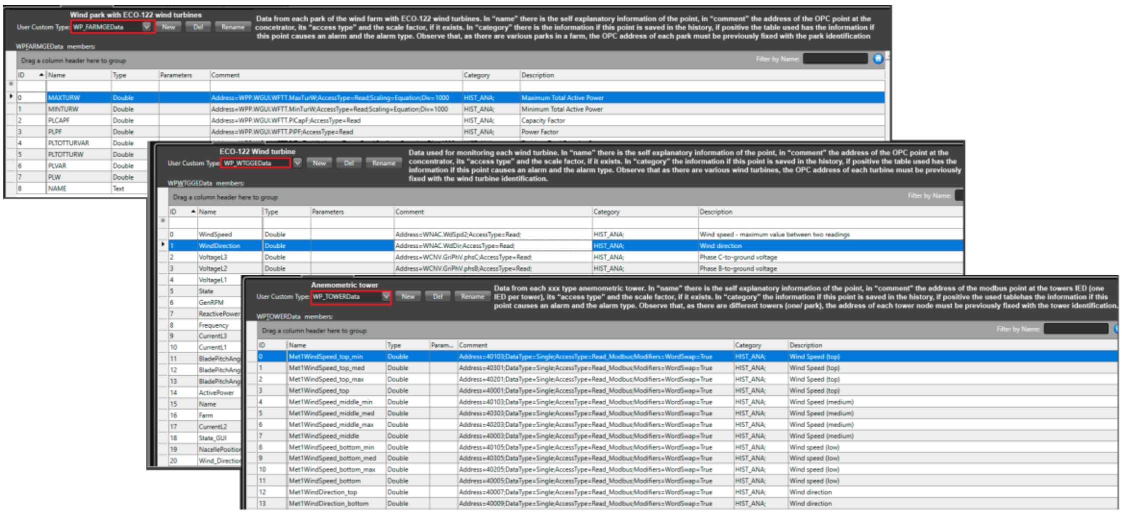


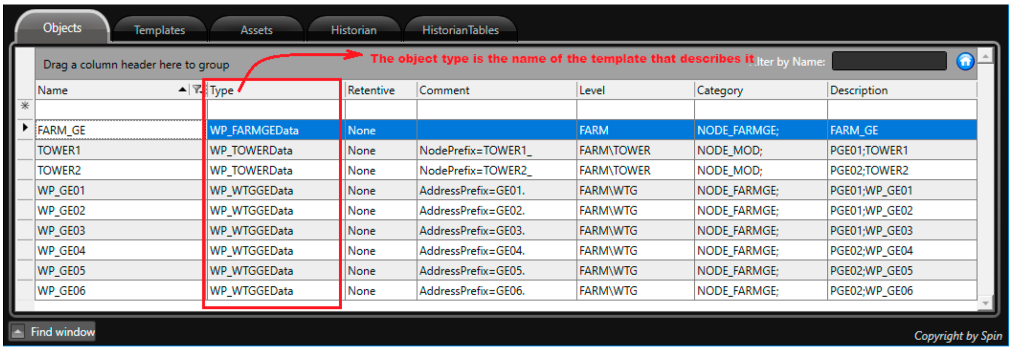
Fig.2 – Templates for a wind farm with ECO-122 wind turbines and XXX Anemometric Towers

3.3 Creation of the objects associated to the base templates

In the base component, for each type of object, comes an example line of how to define this object. The figure below shows the description of a GE wind farm with two anemometric towers and six ECO-122 wind turbines.

Note that in the object description there is a naming rule of the object according to the client definition, and the type of the object corresponds to the type defined in the template for that object.

In the "Comment" column are defined prefixes that are concatenated and added to the addresses of the points of the towers and wind turbines. That is, each tower has the same address associated with the same point, and a prefix must be added that differentiates one address from another. The same is true for wind turbines.



Name	Type	Retentive	Comment	Level	Category	Description
FARM_GE	WP_FARMGEData	None		FARM	NODE_FARMGE;	FARM_GE
TOWER1	WP_TOWERData	None	NodePrefix=TOWER1_	FARM\TOWER	NODE_MOD;	PGE01;TOWER1
TOWER2	WP_TOWERData	None	NodePrefix=TOWER2_	FARM\TOWER	NODE_MOD;	PGE02;TOWER2
WP_GE01	WP_WTGGEData	None	AddressPrefix=GE01.	FARM\WTG	NODE_FARMGE;	PGE01;WP_GE01
WP_GE02	WP_WTGGEData	None	AddressPrefix=GE02.	FARM\WTG	NODE_FARMGE;	PGE01;WP_GE02
WP_GE03	WP_WTGGEData	None	AddressPrefix=GE03.	FARM\WTG	NODE_FARMGE;	PGE01;WP_GE03
WP_GE04	WP_WTGGEData	None	AddressPrefix=GE04.	FARM\WTG	NODE_FARMGE;	PGE02;WP_GE04
WP_GE05	WP_WTGGEData	None	AddressPrefix=GE05.	FARM\WTG	NODE_FARMGE;	PGE02;WP_GE05
WP_GE06	WP_WTGGEData	None	AddressPrefix=GE06.	FARM\WTG	NODE_FARMGE;	PGE02;WP_GE06

In this example there is a farm (FARM_GE), two towers and six wind turbines
Observe that:
a) - In Level we have FARM\TOWER\WTG
b) - As there are 6 GE wind turbines, the OPC address of each wind turbine will be previously fixed with the wind turbine identification (GE01. GE02. Etc.)
c) - As there are 2 towers with 2 IEDs Modbus, each tower node will have as prefix the tower identification:
TOWER1_NODE_MOD
TOWER2_NODE_MOD
d) - As there is only one GE concentrator, its identification is NODE_FARMGE

Fig.3 – Objects created from the templates (1 farm, 2 towers, and 6 wind turbines)

3.4 Creating Symbols libraries

For each application type, generated through LA, a library of symbols that will be used to draw objects is created. Thus, for each type of wind turbine, tower, farm, etc. symbols will be created and placed in the library.
These symbols automatically generate references to their associated objects, i.e., if I copy a WP_GEO01 object, in the “objects” column (see figure above) and paste on a drawing screen, all variables associated with that object instance are mapped for this symbol. That is, the power generated by that wind turbine is the power of WP_GEO01, and the same is true for its current, its state, etc.

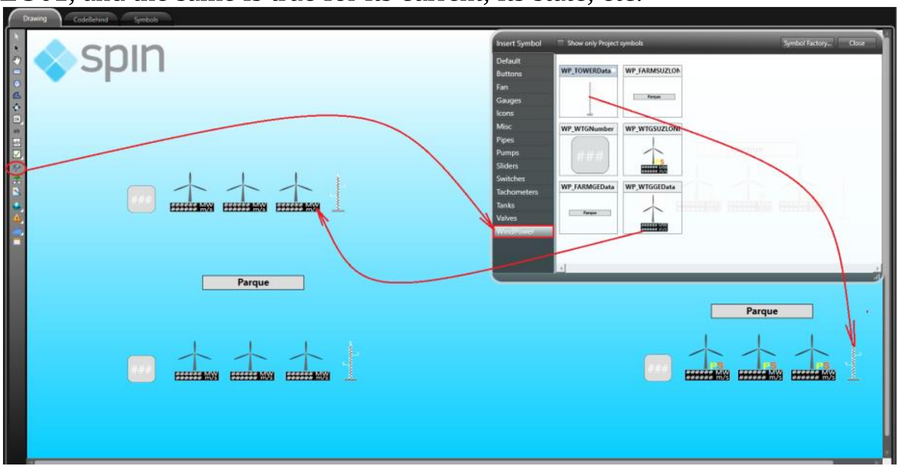


Fig.4 – Symbols library of a wind farm

In the figure below, using an LA library to generate utility substations, the copy of the CAX_21P1 feeder line of objects and their paste is shown on a drawing screen.
All animation rules are already made in this feeder symbol:

- A click on the feeder name take us to a detail screen;
- A click on the circuit-breaker or switches opens a command equipment window;
- A click on the feeder phase A current opens a window with the current in the three phases;
- A click on the adjustment tables allows us to change the settings of the relay feeder;
- A click on the feeder protections allows us to inhibit, block, simulate, etc.

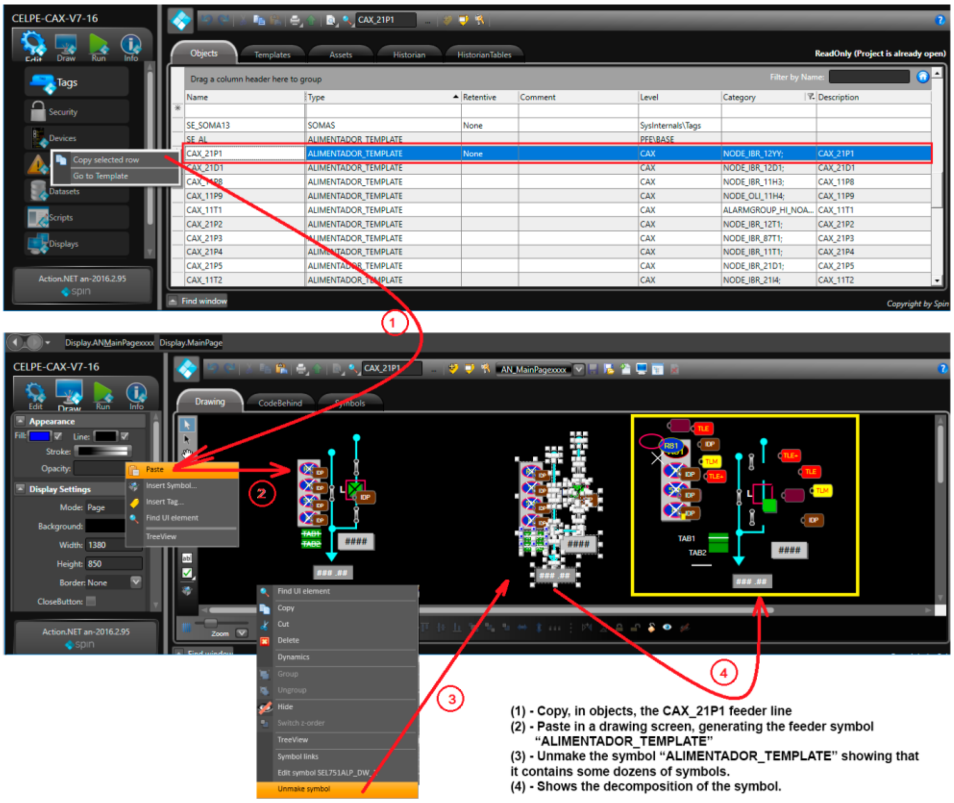


Fig.5 - The CAX_21P1 object turns into a drawing of a feeder span

3.5 Creating the application screens

After creating the templates of the base application and the objects associated with these templates, the application screens are created. For example, in the case of wind farms, seven screens have been created in addition to the existing reports in the empty project (alarms, events, operation log, tags, etc.).

These screens are:

(1) Screen of the example wind farm, with four wind parks. Each park has three wind turbines, an anemometric tower, a symbol that informs the number of wind turbines in operation at the moment and a button that allows to navigate to the totals of the farm. Several of these objects are animated with colors, highlighted symbols, etc. This screen is created in the application time, while the others are already pre-ready and are accessed from links generated, automatically, in the symbols of the objects.

(2) Details of an ECO-122 (GE / Alstom) wind turbine. If the mouse passes over a wind turbine, its figure is modified to a hand and if it is given a click, it navigates to the detail screen of that wind turbine.

(3) Details of an anemometric tower. The operation is similar to that of the wind turbines, click on a specific tower and navigate to the screen detail of the selected tower.

(4) GE/Alstom wind turbines farm details. This screen covers the production totals of the GE/Alstom wind turbines that are all in a single data concentrator, using the OPC protocol.

(5) Details of a Suzlon wind turbines farm, since each park contains a concentrator of its own. If we want to make a totalizer of the Suzlon wind farm (sum of the two farm), we must create a new logical structure.

(6) Details of an S95 (Suzlon) wind turbine, selected in the same way as ECO-122 wind turbines.

(7) In the case of S95 wind turbine, they have an alarm indicator that identifies hundreds of possibilities, with a description of the fault, corrective actions and suggestions.

For each type of application, a planning of these screens and their navigation is done, always attending the client's operational culture. Once the base application is tested, new applications are generated in record time, with quality, robustness and very few errors.

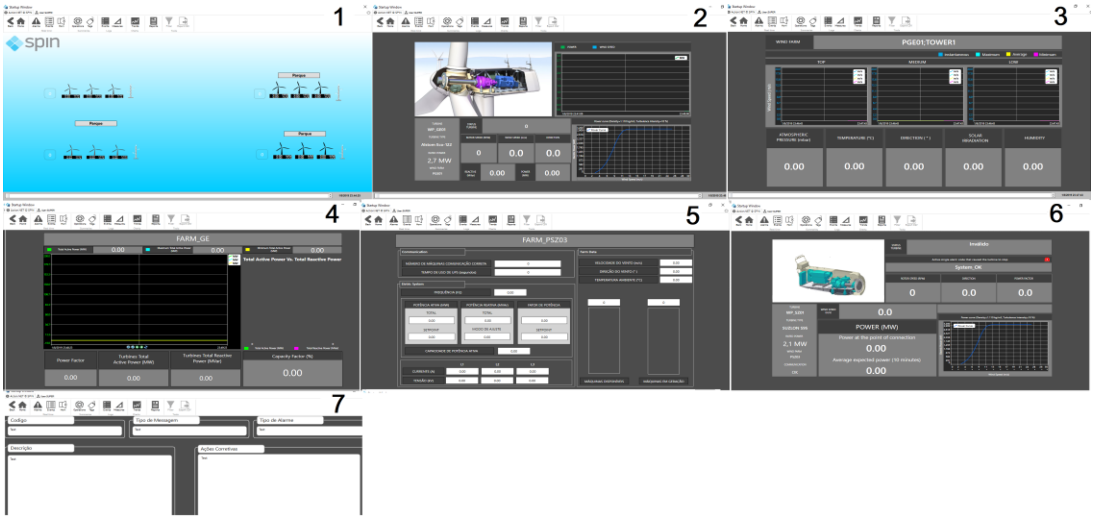


Fig.6 – Wind farm application screens (Example)

3.6 Creating associated channels and nodes

For each IED of the project should be created channels and nodes that will allow the connection with them at runtime. In the case of the example application, according to the figure below, we will have 7 channels, 1 OPC and 6 Modbus.
As in the address of the points the prefix of the node associated with the point must be added, in order to identify the associated IED, these prefixes must be included in the comments of the respective objects (see figure 3 - Comments).

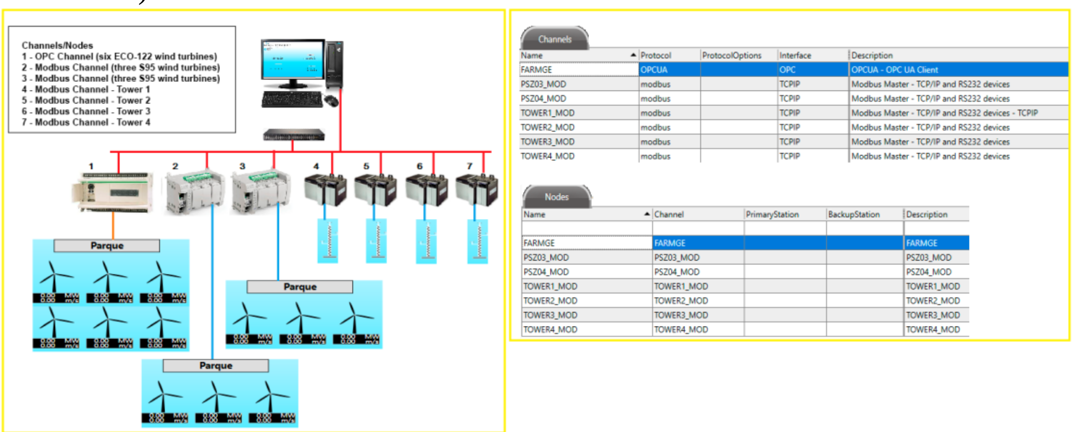


Fig.7 – Channels and Nodes from the example application

3.7 Deploy – Instantiation of alarms, addresses and history records

In sequence, after all the previous steps one must execute the Deploy module to instantiate all the items associated with each tag. The Deploy is one of several programs available in a library of extensions, which at project time allow you to import, export and process data associated with the application. In the specific case of this program, when executed it scans the Tag/Object lines (figure 3) and, for each line that has objects to be exploded (category and comments), it explodes automatically generating the items of the tables of alarms, history, and I/O points associated with the structure. After the explosion, it deletes the categories and comments associated with that line, allowing in the future, when creating a new object, only this one has its data instantiated.
As an example, considering the wind farm, for each object created (wind turbine, anemometric tower, wind farm, etc.) the alarm tables, history and associated I/O addresses are generated. The mechanics of this process minimizes errors and increases productivity, since thousands of items are automatically generated.

3.8 Applications main screens generation

As shown in the above item (3.5), the base screens are pre-made and assume the reference element when called, for example, there is a feeder detail screen that when called from a given feeder, assumes the

reference of the same, as well as wind generators, anemometer towers, etc. However, the main screens of the application, as a given substation, a wind complex, etc. are generated in each repetition of the process. Thus, the last step of the application is to copy all objects created on the object screen and paste on one or more design screens, as shown in the items above (3.4 and 3.5).
Once these procedures are completed, the application will be ready and, if the base application tests have been well done, the applications generated will be error-free and generated in record time.
In IEDs with fixed addresses, such as concentrators of wind turbines and wind farms, anemometric tower concentrators, meters, etc., all addressing will be 100% correct after testing the base application, also allowing the process knowledge to be kept in the application rather than in the technical staff. That is, anyone can generate a correct application, without ever having worked in this process or with these IEDs.
In IEDs that must be configured for each application, such as relays and PLCs, it is suggested that they be configured for functionality and, after tested, your program be saved to be repeated whenever the IED is used again, for the same function. For example, a feeder protection relay may have a few configurations depending on the feeder arrangement. For each arrangement it is suggested to configure the relay in the same way and using the appropriate standards. If the relay uses the IEC61850 protocol, the points must use the appropriate nodes and functions and not use generic points.

4 CONCLUSIONS

This methodology is in the process of being patented at INPI and has already been applied in several automation projects of electrical systems such as automation of utility substations, FLISR applications (fault localization, isolation, system restoration), wind farm automation and module asset management.
The figure below shows two real applications implanted with the LA methodology.

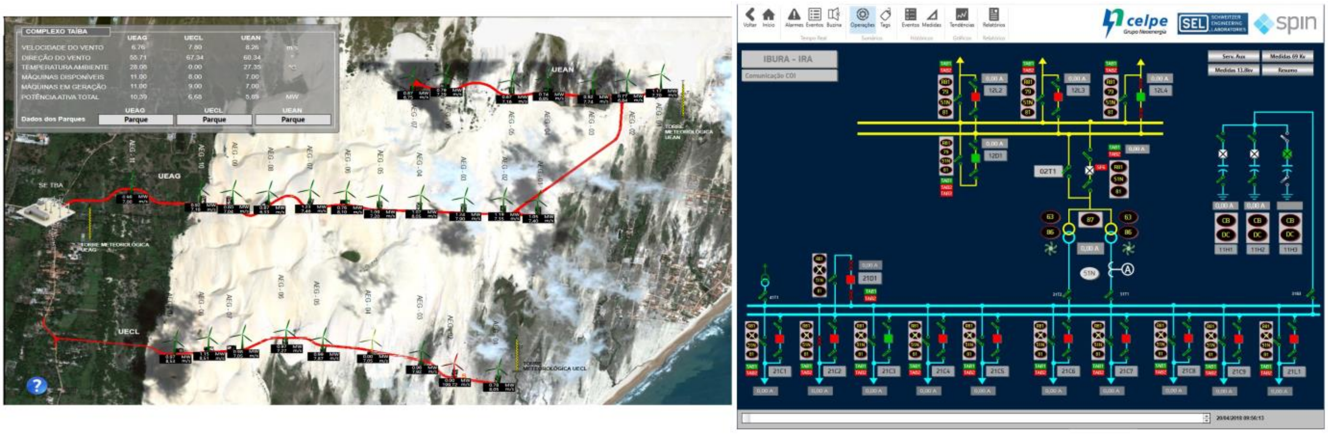


Fig.8 – Two real applications developed with the LA methodology – utility substation and wind farm

5 BIBLIOGRAPHY

[1] Simoes, C. "A integração completa entre COS, EMS, DMS, rede corporativa da empresa, CORs, subestações e chaves de poste", XV Simpósio Nacional de Automação de Sistemas Elétricos, Brasília, 2000.

[2] Simoes, C. e Regueira, B. "Implantação do centro de operação integrado do sistema CELPE", X ERIAC – Cigré, 2003.

[3] Batista. M. e Sylvestre, M.H, "Mobilidade do sistema de supervisão e controle em subestações", XII EDAO, Brasília, nov. 2012.

[4] Simões, C. At Alli, "Centro de operação integrado da Queiroz Galvão Energias renováveis", PowerGEN, SP-2014.

[5] Simões, C, Duarte.D.P, at alli, "Planejamento e implantação de um sistema que executa a função FLISR em redes de distribuição", CBQEE, 2017.

[6] Silva,C.A, Silva Junior.A.P, e Borges R.L, "Desenvolvimento de sistema especialista para gerenciamento de transformadores instalados em subestações", Cigré-Brasil, "IX Workspot - Workshop Internacional Sobre Transformadores de Potência, Equipamentos, Subestações e Materiais, 2018.