

On the Use of Artificial Intelligence for Prognosis and Diagnosis in the PROTEUS E-maintenance platform

PROTEUS WP2 Team

(L. Déchamp, A. Dutech, T. Montroig, X. Qian, D. Racocanu, I. Rasovska, B. Brézillon, F. Charpillet, J-Y. Jaffray, N. Moine, B. Morello, S. Müller, G. Nguengang, N. Palluat, L. Pelissier)

PROTEUS ITEA European Project

URL : <http://www.proteus-iteaproject.com>

Abstract - *The PROTEUS project has identified several maintenance tasks and functionalities where Artificial Intelligence tools or methods could bring many benefits. In this paper, we focus on the diagnosis and prognosis of faulty states in equipments. We show how the generic “PROTEUS AI Web Service” can be instantiated with different AI tools to do diagnosis or prognosis. Thanks to the PROTEUS architecture, these services can work with various data and knowledge sources (like FMECA, SCADA, CMMS, ERP) enabling the use of both maintenance and production data. The characteristics of these “Web Services” are then analyzed so as to specify the features of a “Meta-Tool” that, for a given diagnosis or prognosis task, would help decide which tools are best suited. This Meta-Tool itself could be built as an instance of the generic “PROTEUS AI Web Service”.*

Keywords: E-maintenance, Decision support, Diagnosis, Prognosis, Rule-based systems, Case-based reasoning, Self-organizing maps, Neuro-fuzzy systems, Hidden Markov Models.

I. INTRODUCTION

Proteus is a European ITEA project which aims at building a generic and general e-maintenance platform. In this framework, the more specific goal of the Work Package 2 (WP2) was to study the use of Artificial Intelligence (IA) for PROTEUS. This paper offers an overview of the WP2 work, i.e. motivations and specifications.

Knowledge based systems and artificial intelligence methods can support maintenance basically on three different levels. First, there are solutions which recognise alarms, evaluate the situation based on the available data from condition monitoring and execute appropriate actions if required. Second, there are solutions for diagnostic support. They are especially intended to help service technicians which are not yet really familiar with the maintained equipment, and, based on a hierarchical model setting relationships between errors or external phenomena, their possible reasons and adequate actions, they can be considered as knowledge bases where professional experience is stored. Third, there are solutions for analysis of measurement values which are not used in daily maintenance practice but in maintenance engineering

in order to get decision support for retrofit interventions or improvement of maintenance rules.

However, looking at implemented solutions in the industry, we find that installed solutions rarely use artificial intelligence even if it might seem so at first sight. Given the fact that the concepts for AI methods which are known today are 20 years old or more, and taking into account that the performance of current personal computers complies with the high requirements of these methods, it is somehow surprising to see that an application in the domain of maintenance is still waiting to be realised. But on the one hand, development costs for easily useable AI tools are high and neither equipment manufacturers nor customers want to assume them. On the other hand, few types of machines exist in numbers which are large enough to justify the development costs.

So, what the market needs are software packages including tools with a generic approach, which can be used in the most various application fields and which the user can adapt to his own needs even without expertise in the domain of AI. At the same time, the tools need to be largely self learning and have to be optimised on a minimisation of the data acquisition effort, since otherwise, the effort for system implementation would be higher than the maintenance effort which could be saved thanks to AI methods. Within the Proteus project, the attempt at developing AI based tools responding to these requirements has been made, and number of promising concepts have been elaborated. To describe their approaches will be the subject of the present contribution.

To illustrate how AI based tools are integrated into the PROTEUS project, we will focus on some “level two” tasks: Diagnosis and Prognosis. Section 2 will motivate this choice. Then, we will introduce the generic template of each PROTEUS AI Web service. Section 3 shows also various instances of this generic template for the purpose of diagnosis or prognosis. The real world example of Section 4 introduces the need for a special “Meta-tool” that can also be specified within the PROTEUS generic template. Section 5 brings some conclusions and perspectives to this work.

II. DIAGNOSIS AND PROGNOSIS

In this Section we motivate our focus on diagnosis and prognosis by showing their importance in the process of maintenance. The general architecture of the PROTEUS platform is then quickly exposed.

A. Motivations for Maintenance Practice

The Proteus project is aimed at bringing enhancements to maintenance organizations thanks to a distributed architecture including the main tools set up in industrial environments. Thus, this “generic platform for e-maintenance” must bring to a maintenance actor efficient tools enabling him to perform the right actions thanks to the most relevant information. In these actions, the starting point is mainly a failure appearing on an equipment.

Starting from a faulty state of an equipment raises the first problem before intervening: what is the reason why the equipment is not running as it is attended to? This first step is the most crucial, as it defines all the actions to be performed in the next steps. Two main cases can be presented here:

- **“basic” failure**: the maintenance operator is able to perform the diagnosis by himself and so the maintenance task may be performed quite quickly
- **“complex” failure**: the help of an expert or of a whole maintenance team may be needed in order to determine what actions will be necessary for the equipment to be recovered in a correct running mode

In these two cases, adding the help of a tool dedicated to diagnosis through artificial intelligence computations can bring many benefits. In “basic” cases, time and resources can be saved if failure and maintenance operation could be predicted by an AI tool rather than observed. In more “complex” cases, AI tool could probably help experts on the spot and bring them an automated way to capitalize their knowledge (sharing information between people, equipment, products...).

Furthermore, AI tools could be helpful from the “return of experience” point of view. For example, it could help in analyzing more precisely the high amount of available data gathered on the equipment through the sensors configured on it so as to enhance further maintenance tasks.

Actually, many tasks in maintenance can be improved thanks to such computing tools, as they will also give information to help a manager taking a decision, as maintenance is a succession of decision tasks. Thus, the intervention of such tools can be spread all along maintenance tasks: preparation, monitoring, decision help, intervention and experience feedback. To go further in this vision of AI activities, considering a company which is specialized in maintenance of installations, this feedback will also be useful for the next maintenance preparations when applied to similar equipment plants.

B. 2.2. Requirements and Usage

From these motivations, and the general requirements of the PROTEUS project, the architecture of the platform has been designed as depicted on Figure 1, where each “Platform tool” is indeed a Web service.

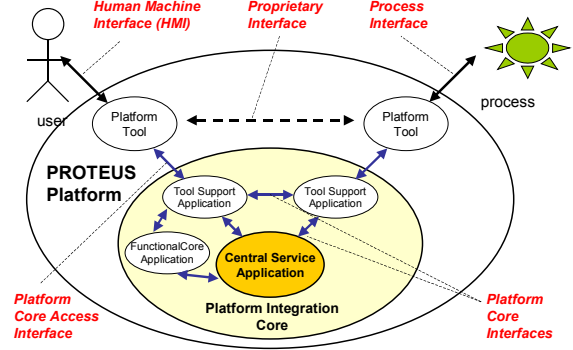


Figure 1 : the general architecture of PROTEUS

Several AI tools are implemented in the current platform. Mostly aimed at low level diagnosis or prognosis tasks, some of these tools can also be adapted to more high level tasks like for example “return of experience”. These tools use different AI methods but they all fit the generic Proteus AI template that is detailed in next section.

III. THE “PROTEUS/GENERIC AI SERVICE TEMPLATE” AND ITS INSTANCES

As the whole PROTEUS platform is centered on the use of Web services organized around a central core, a main contribution of the WP2 was to define a generic AI service template that all AI tools should instantiate. We describe this template and some AI tools we adapted to PROTEUS.

A. Generic Template

An exhaustive study of the maintenance field shows that a generic decision tool should provide the following functionalities (to be used either by a maintenance expert or by an operator depending on the operation considered): configuration, initialization, on-demand decision making, alarm raising, model enhancement and model storage.

Configuration: This “template” service enables the user to define a decision task; the maintenance expert first creates an association between the decision technology to be used, the target equipment on which maintenance decision has to be performed, the decision action to perform (such as diagnosis for instance) and the set of data containers required for data analysis (FMECA, production historical data, pre-processed data from history, fault history)

Initialization: This service enables the maintenance expert to effectively retrieve the data described in the corresponding configuration from the devices attached to the considered equipment and to create a decision model by training on the imported data. It generally includes the

manual configuration of the training parameters and a validation of the obtained decision model.

On-demand decision making: In the Proteus project, we have mainly focused on on-demand diagnosis on a given equipment. This service is mainly a presentation of decision results to the final maintenance operator.

Alarm raising: The user can configure an autonomous process to monitor a given equipment and issue appropriate alert signals when needed.

Model enhancement: This service, implemented by mostly all technologies used within Proteus project, refreshes the decision model using the latest available data. It can be an automated step and also include a manual feedback.

Update configuration: Tools implementing this service are able to adapt to a new configuration (like adding a sensor) without having to re-learn the model from scratch. As such, it is faster than doing everything from scratch and previous experience is not lost.

Model storage: Any AI service within Proteus must implement the centralized storage functionality to comply with the general architecture. This enables a totally distributed system and a full compliance with any IT architecture and enterprise workflow on the deployment target.

This "template" mechanism enables an easy integration of any AI tool that could formalize its functionalities using this template. Each of the following tools implements several of the functionalities depending on the specificities of each method.

B. Neuro-fuzzy system

The interest of neuro-fuzzy techniques for the diagnosis and prognosis is to combine the ability to learn of neural networks with the natural language formalism of the fuzzy logic. The diagnosis support tool so obtained uses industrial knowledge databases like CMMS, ERP, SCADA, FMECA, Failure Tree and is both reactive and easy to configure.

The proposed neuro-fuzzy diagnosis aid tool is composed by a dynamic neural network module and a neuro-fuzzy module [1]. The fault detection uses a recurrent radial basis neural network [2]. The output of this real time detection module is given by the failure mode or failure symptom. This symptom is used in the neuro-fuzzy diagnosis module. The fault tree of the critical part of the system is transformed into a fuzzy neural network. This neural network uses information from the CMMS historic and is able to learn any new maintenance situations (symptom-mode, localization, origin). In this way, we have a reactive and dynamic diagnosis system.

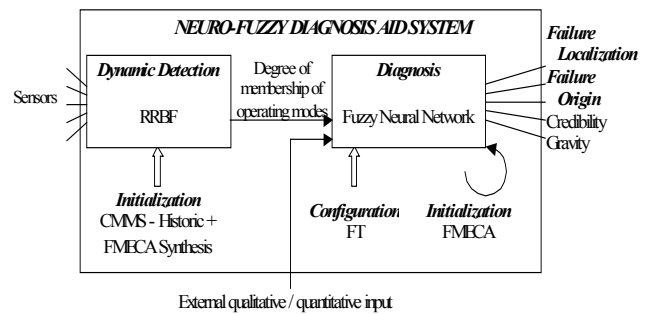


Figure 2: Principles of the neuro-fuzzy diagnosis aid system

During the process, the detection system scans continuously the critical element. When a breakdown or a failure occurs or when one is likely to occur, an alarm is raised and the diagnosis aid system starts. Detection keeps working. According to the information provided by the detection system, the diagnosis aid system proposes to the operator the possible causes of the problems as well as the fuzzy interpretation of these causes.

C. Case-based reasoning

The creation and realization of the case-based reasoning system passes by the representation of expert's knowledge and the exploitation of this knowledge. Case-based reasoning implements a case base which contains the solutions of already solved problems. In this case base one can look for the last experience similar to the problem to be solved. A retrieval phase makes it possible to find the similar cases in the case base. The solutions of the found cases are used as bases to solve the problem running, realizing an adaptation phase to take into account the differences between the reminded case and the new case. These cases are memorized and organized according to well defined criteria making it possible to find them effectively [3]. The feasibility of CBR for the decision help to the decision operator in industrial supervision, where the experience of last situations is re-used to manage new situations, was shown in the study of the decision-making process for the operator in a context of industrial supervision [4].

The decision process to help an operator consists in the analysis of a given situation and in the drawing of the list of the entities having a role to play (or having contributed) for the current (considered) situation. These entities represent equipment variables which are known from the monitoring system such as SCADA, functional analysis, failure tree and FMECA. Even the maintenance operator could be asked to answer some questions needed to complete the description of the problem. The retrieval algorithm looks for the similar description in the case base and proposes to the operator the origin of the failure and the repair action to do. The list of the found relevant elements is presented to the operator with the aid of special tools such as dashboards, documents, etc.

A knowledge base and a list of parameters used to describe cases in the databases must be constructed. Knowledge needed for the CBR process as such has been identified in terms of four containers. These are :

- The vocabulary used that describes the domain
- The case base, which means the set of cases describes in this vocabulary
- A measure of similarity between cases.
- Knowledge to adapt solution cases to current case.

Data post-processing corresponds to experience feedback exploitation based on intervention report capitalization.

D. RuleMaker : a rule-based decision tool.

The rule-based decision service provides on-demand diagnosis and recommendations on a given equipment. System knowledge that can be taken into account includes historical production, quality data and fault history. The decision model is a rule-set obtained by rule induction [5] on the selected data with an optional user defined preprocessing; the decision is obtained by a vote of the obtained rules. It also implements the automated enhancement of the decision model.

One of the specificity of this method is the white-box model (the rules) that is self-explanatory. Diagnoses may take the following form:

*Certainly in faulty mode ("Failure might occur within 4 months after last one")
with a confidence of 87 %.*

The maintenance expert, when initializing the decision process, decides on which criterion he will perform his decision (in the example above, the decision variable is a discretized time "since last failure", whose modalities have been manually classified as "correct running mode" or "faulty mode").

The decision is provided with an indicator for the confidence of whose definition in the decision strategy (that can be more or less aggressive depending on industrial context)

The recommendations take the following form:

As long as:

-Pressure in [15.6 ; 17.5]

Keep:

- Feeder = Open

- Time since last pipe change in [0 ; 3 months]

- Temperature in [35; 41]

To obtain:

Failure might occur between 12 and 17 months after last one.

This recommendation is based on an automated selection of the rule most adapted to the user's needs.

Such a recommendation can only be obtained if the service is based on a general data repository that stores the variable definitions (to specify if any industrial parameter can be industrially controlled or not).

E. Hidden Markov Models

Hidden Markov Models are specially well suited to sequential diagnosis with noisy data as it relies on a probabilistic model of the system. The mathematical framework of Markov Chains is the basis for several Artificial Intelligence tools among which Hidden Markov Models can be adequately used for the problem of diagnosis. Under this framework, the system to study is modeled as a finite state machine with probabilistic transitions. The state of the system is not directly observable and the external agent (i.e. the maintenance operator) has only access to some observations, that are probabilistic functions of the state of the system. The whole process of diagnosis is that of inferring the most likely hidden state from the current sequence of observations.

The more exact the model of the system is, the more pertinent the diagnosis will be. An appropriate algorithm, derived from the well known "Baum-Welsh" algorithm (see [6]), can learn a model from a corpus of (observation-sequences, diagnosis) examples while giving a semantic meaning to the hidden state of the system (i.e. normal, faulty, ...). Expert knowledge can greatly help the learning process by providing prior probabilities to the parameters of the model.

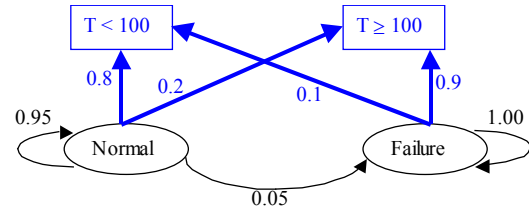


Figure 3 : HMM, circle are states, square are observations and each transition has probabilities

On the example of Figure 3, if we know that we start from a "Normal" state, the AI tool can warn that "a failure state is very likely when one observes a sequence of more than 3 high temperature ($T \geq 100$)."

F. Self Organizing Map

Self-organizing map (SOM) is an unsupervised learning and topology preserving neural network model. The SOM can map data from a high-dimensional input data space onto a two-dimensional network or grid of neurons.[7]

In fault diagnosis, the SOM has two functions. Firstly, the SOM can be used in detecting the fault and in identifying it. One can detect faults even if there are no measurements of the faulty states by investigating the quantization error between the SOM and the measurements. If the quantization error exceeds a pre-defined limit, a faulty state has occurred. If one needs also to identify faults, representative examples of the faulty situations must have been recorded.

Data plays a very important role in SOM using as a decision help tool. If only normal condition process data of a certain equipment, then fault detection / prognosis can be

done by comparing the trained model from normal process condition; if fault data available, then do fault identification / diagnosis with a few labelled samples of the faults or some general prior knowledge of the fault types.

Using SOM as a maintenance decision help tool needs to firstly initialize the trained model-so called “learning model” based on proper dataset and prior knowledge of the dataset, the aim is to learn the typical normal behaviour of a machine from measurements. Then comparing the “learning model”, it is possible to “decide” the operation state of new measure data of the equipment.

Input	Data	Boolean, numerical, symbols, free flow text, composite...
		Discreet, continuous, dynamical...
		History, sequences, single events
		Irregular, noisy, incoherent
Output	Needed knowledge	Functional decomposition; Model of the process, of the data
	Pre-processing	Necessity, availability,
	Type	Decision, Probabilities, Rules,
	Post-processing	Necessity, availability,
Task	Data	<i>See above</i>
	Type	Diagnosis, prognosis, monitoring, evaluation...
	Equipment	...
	User	Technician, Operator, Manager...
Task	Others	...

Table 1: Features that must be taken into account when choosing a PROTEUS AI Web Service

IV. DIAGNOSIS AID APPLICATION TO A FLEXIBLE MANUFACTURING SYSTEM USED FOR PROTEUS DEMONSTRATOR

In order to illustrate the efficiency and the usefulness of the proposed diagnosis help tools, we use an e-maintenance platform developed around the flexible manufacturing platform of one of the partner of the project (Figure 4). This system has a local industrial network between the Programmable Logic Controllers, linked with the Ethernet network by the mean of an acquisition board and a SCADA system working via an OPC server. The SCADA works in a web service approach, offering all necessary data to all clients of the e-maintenance platform. The CMMS (distant) system and a web portal participate to the diagnosis scenarios. Such possible scenarios are presented in the figures 2, 3 and 4 in a sequence diagram form.



Figure 4: Flexible production system of Besançon

G. Tool characteristics

The various PROTEUS AI Web Services quickly presented here have all been developed within the PROTEUS platform. They can all be used in the general context of diagnosis and prognosis, but depending on many parameters, each one can be more or less efficient on a specific instantiation of a task.

The application boundaries of the tools, which appear more or less explicitly in the descriptions above, can be characterized by many features. Table 1 gives a non-exhaustive list of such characteristics.

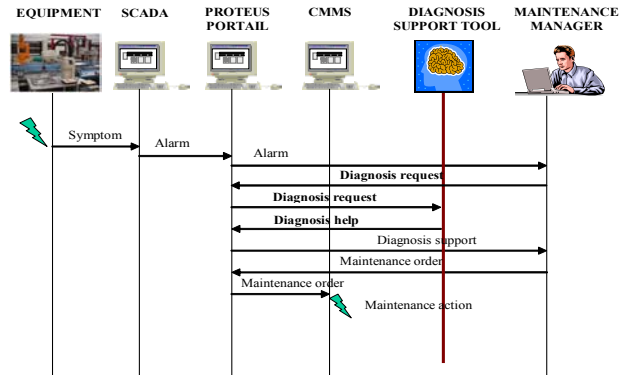


Figure 5. First possible sequence diagram (alarm emitted by the SCADA)

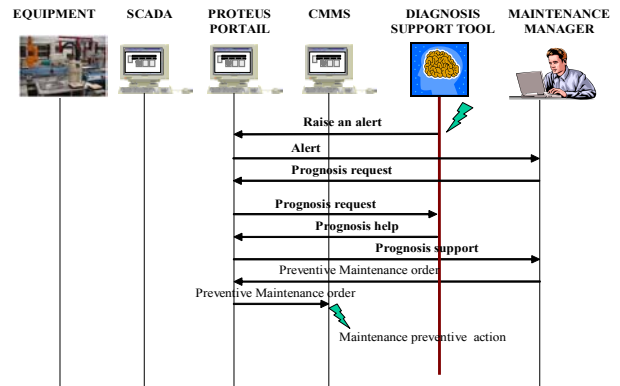


Figure 6. Second possible sequence diagram (alarm emitted by the diagnosis help tool – Prognosis case)

In this scenario, we notice that the diagnosis tool is used after raising an alarm or after a maintenance manager request. We obtain also several cases:

- the SCADA raises an alarm (Figure 5),
- the diagnosis tool raise an alarm (Figure 6). This case corresponds also to a prognosis scenario in which the diagnosis help tool acts like an alert raising system before the occurrence of the failure and the failure appearance.
- the request comes directly from the maintenance manager or operator. This case can also correspond to a consultation of the diagnosis tool in order to identify the possible cause of a symptom.

The question now is: which AI tool is best adapted to each case? Between the proposed artificial intelligent diagnosis help tools, it would be nice to have a “super one” able to answer this question, or better, to automatically choose and configure an appropriate AI tool for each scenario.

A. Meta – tool specification

As the final goal of the Proteus project is to provide a generic platform, each configuration of a PROTEUS AI Service requires a concrete answer to the following question:

“Which technology will best suit my needs?”

For example, in the above example, one could use HMM in the first case, RuleMaker in the second and case-based reasoning in the third.

The answer will of course take into account the relative strengths and weaknesses of each considered technology, relatively – for example – to every feature described in Table 1. This configuration process can also be seen as a maintenance process and could benefit from the help of another PROTEUS AI Service. We call this hypothetical service a “meta-tool”.

Such a meta-tool intends to integrate the AI expert expertise to handle the AI tool choice. It enables the following evaluations based on the system description (reachable through a general Proteus information repository describing its ontology):

- Is dynamical data provided? Is it necessary for the decision in the given context? If yes, which technology can handle it?
- Is historical data provided? Event-driven data?
- Is it necessary to handle irregular data sampling?
- Does the technology require a functional analysis of the equipment? Is it provided?
- Which pre-processing capabilities are available on the data acquisition servers implemented on site? Are they sufficient for the targeted decision technology?
- What is the nature of the provided data? Mostly symbolic? Free-text, manually collected? Boolean? Numerical? Composite ? Does it include missing data or data of uncertain quality?

The meta-tool will then provide the best matching possible between the given decision context (available data and intended goal) and the available technologies, based on the

description provided with each tool. This “meta-tool” mechanism also ensures the evolution of the Proteus platform as new AI tools can be integrated since they implement a self-descriptive interface.

Such a meta-tool could itself be specified in the general template of the PROTEUS AI Service. For example, by using Case Base Reasoning as the underlying “intelligence” of the Meta-Tool:

Configuration: Specify the AI Tools that exists, the type of services (diagnosis, prognosis, ...)

Initialization: gather examples from actual use of AI services within the PROTEUS framework.

On-demand decision making: ask advice for diagnosis on a new equipment.

Alarm raising: not useful for the meta-tool.

Model enhancement: when a given service has been used for some time, one can update the case base.

Update Config.: take into account a new kind of AI tool.

Model storage: example of the case base need to be saved.

V. CONCLUSIONS

In this paper, we presented an overview of the integration of artificial intelligence tools in the PROTEUS European project on e-maintenance. The main contribution is the definition of a generic AI template for specifying how AI tools must be integrated into the platform. Several examples of AI tool instantiations were shown in the field of diagnosis and prognosis, as well as the need of a “meta-tool” that can fit itself into the generic template.

The project opens many perspectives beyond actually implementing a meta-tool. The most obvious would be the development of tools to deal with other aspects of maintenance like maintenance engineering, intelligent e-doc access, optimization of maintenance procedures, etc.

VI. REFERENCES

- [1] N. Palluat, D. Racocanu, N. Zerhouni, “Diagnosis using Neuro-Fuzzy Systems, Intelligent Maintenance System,” in IMS 2004, 14-15 juillet 2004, Arles, France.
- [2] R. Zemouri, D. Racocanu, N. Zerhouni, “Recurrent Radial Basis Function Network for times series prediction, Engineering Applications of Artificial Intelligence,” The International Journal of Intelligent Real-Time Automation, journal IFAC ed. Elsevier Science, vol. 16, Issue 5-6, 2003.
- [3] B. Fuchs. « Représentation des connaissances pour le raisonnement à partir de cas : Le système ROCADE ». *Thèse au Laboratoire Image Signal Acoustique de CPE Lyon*, 1997.
- [4] A. Mille. “Raisonnement basé sur l’expérience pour coopérer à la prise de décision, un nouveau paradigme en supervision industrielle ». *Thèse d’université*, Saint Etienne, 1995.
- [5] Smyth P., Goodman R.M. “Rule induction using information theory”, in *Knowledge Discovery in Databases*, G. Piatetsky-Shapiro and W. Frawley (eds.), The MIT Press, Cambridge: MA, pp. 159-176, 1991.
- [6] L. Rabiner and B. Juang. “An introduction to Hidden Markov Models”. IEEE ASP Magazine, Vol 3(1), 1986.
- [7] Olli Simula, http://www.cis.hut.fi/projects/ide/good_project.html