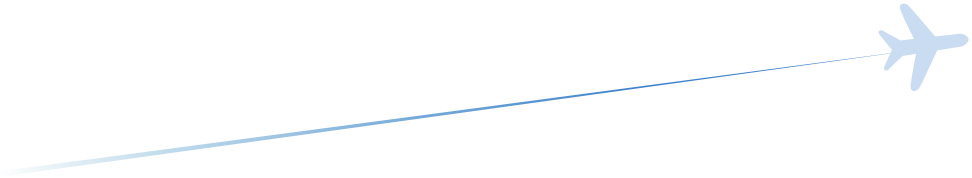

D3.1 Preliminary Metamodeling Methodology

Deliverable ID:	**D3.1**
Dissemination Level:	**Public**
Project Acronym:	**NOSTROMO**
Grant:	**892517**
Call:	**H2020-SESAR-2019-2**
Topic:	**SESAR-ER4-26-2019**
Consortium Coordinator:	**CRIDA**
Edition date:	**20/11/2020**
Edition:	**00.01.00**
Template Edition:	**02.00.02**

Founding Members
EUROPEAN UNION EUROCONTROL
SESAR JOINT UNDERTAKING

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Francisco Antunes / DTU	WP3 Member	25/09/2020

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
Javier Poveda Barbero	CRIDA	21/09/2020
Faustino Tello	CRIDA	21/09/2020
David Mocholí González / Nommon	Researcher - WP3 Member	29/09/2020
Abderrazak Tibichte	ISA	30/09/2020

Approved for submission to the SJU By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
Mayte Cano	CRIDA	30/09/2020
Ricardo Herranz	NOMMON	Silent Approval
Gelard Gurtner	UoW	Silent Approval
Jordi Pons Prats	UPC	Silent Approval
Francisco Camara	DTU	30/09/2020
Sandrine Molton	ISA	30/09/2020

Rejected By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
N/A		

Document History

Edition	Date	Status	Author	Justification
00.00.01	09/09/2020	Draft	DTU	Document creation and first draft for review
00.00.02	25/09/2020	Draft	DTU	Version for partners approval
00.00.03	30/09/2020	Final Draft	DTU	Minor adjustments. Version for SJU delivery
00.01.00	20/11/2020	Final	DTU	Version approved by SJU

NOSTROMO

NEXT-GENERATION OPEN-SOURCE TOOLS FOR ATM PERFORMANCE MODELLING AND OPTIMISATION

This Project Management Plan is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 892517 under European Union's Horizon 2020 research and innovation programme.

Abstract

This document aims at describing a preliminary version of the metamodeling methodology to be employed within this project. Due to its draft nature, the provided version herein should be revisited in the future as the project evolves and as the results are iteratively obtained. Small adaptations, performance improvements, and fine-tuning procedures are likely to be required.

In this deliverable, we also provide brief descriptions of two core concepts that compose the base structure of the proposed methodology, namely, active learning and simulation metamodeling itself.

Table of Contents

List of Figures

1 Introduction

1.1 Purpose of the Document

Simulation constitutes a well-known and established tool to model complex real-world systems, such as urban and transportation environments. However, despite its clear practical advantages, simulation models, when embedded with enough detail and realism, can become computationally expensive to run. This shortcoming may hinder the exploration of its input-output behaviour when many variables are at play.

To tackle the mentioned computational challenge, simulation metamodels can be employed to estimate the underlying function inherently defined by the simulation model and used as a modelling proxy for the latter. In turn, this allows for a reasonable number of exhausting computer experiments to be bypassed during the exploration process.

The problem of expensive simulation run has a clear resemblance with modelling scenarios where labeled data tends to be particularly difficult or time-consuming to obtain. In such scenarios, active learning has historically proved to be a powerful learning paradigm to be adopted. Its main objective is to attain high prediction performances with as few data points as possible.

This document provides the key aspects of a preliminary metamodeling methodology developed in conjunction with a straightforward active learning scheme.

1.2 Intended readership

This document is intended to be used by SESAR JU and NOSTROMO members.

1.3 Terminology and Acronyms

Term	Acronyms
ANNS	Artificial Neural Networks
ATM	Air Traffic Management
ECAC	European Civil Aviation Conference
E-OCVM	European Operational Concept Validation Methodology
ER	Exploratory Research
GP	Gaussian Process
KPA	Key Performance Area
KPI	Key Performance Indicator

Term	Acronyms
NOSTROMO	Next-generation Open Source Tools for peRfOrmance Modelling and Optimisation
SESAR	Single European Sky ATM Research Programme
SJU Work Programme	The programme which addresses all activities of the SESAR Joint Undertaking Agency.
SESAR Programme	The programme which defines the Research and Development activities and Projects for the SJU.
TMA	Terminal Manoeuvring Area

2 Active Learning

Active learning [1], also called query learning, among several other designations, is a subfield of supervised machine learning that primarily consists of an iterative process that aims to attain higher prediction performance with fewer selected training data points. This process allows the algorithm to choose, according to some given criteria, the data points from which it learns. Thus, it is particularly useful for modeling tasks where the labeled data is difficult or expensive to obtain.

The general idea of the active learning approach is to optimally select the most informative data points in an active manner in order to not only boost the model training efficiency but also to improve its predictive performance, therefore labeling as few data points as possible. Hence, it is essentially focused not only on improving the overall prediction performance of the underlying machine learning model but also on controlling the associated costs of acquiring new labeled data. The entire process is guided by an oracle, traditionally, but not limited to, a human annotator oracle, whose task is to provide labeled data instances that are successively incorporated in the initial training data set.

An arbitrary active learning approach can be formally defined as the following quintuple (L, U, M, O, Q) [2]. First, L is the labeled training set. Then, the set of unlabeled data points is represented by U. Generally, U#>>#L, i.e., the number of unlabeled points is much higher than the labeled ones. Note that U represents the explored area of the feature space. M is the machine learning model. Depending on the nature of the problem being modeled, it can be a classification or a regression model, which in turn affects the nature of the set L of being discrete or continuous, respectively. The oracle is represented by O. In the case of the NOSTROMO project, the oracle role will be played by an ATM simulation model, which will be a provider, or generator, of labeled data points, that is to say, simulation input-output tuples.

Finally, Q is the query function that encodes the strategies and criteria for finding and selecting the most informative instances from U to be added to L. There are many query strategies formulations available within the related literature which essentially encompass different perspectives to approach the problem in study. Depending on both the nature of the problem and the model being used, several query frameworks can be adopted. 1 summarily depicts the general active learning scheme.

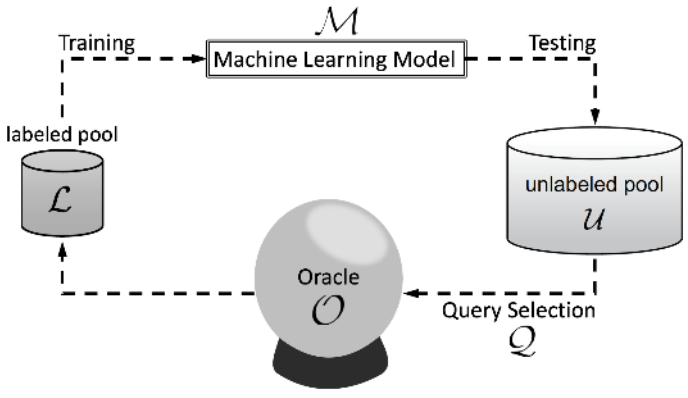

Figure 1- A general illustration of the active learning paradigm.

3 Simulation Metamodels

The development of simulation metamodels [3] has been around since the early '70s [4]. Their primary purpose is to serve as surrogates, emulators, or even response surfaces, for simulation models so that expensive simulation runs can be avoided. Specific features such as mathematical simplicity, speed, and interpretability are usually attributed to metamodels. Consequently, the application of metamodels within simulation analysis provides an additional level of understanding of the underlying system, as well as of the relationships between the system input and output variables, while maintaining a computationally economical and straightforward approach to the problem.

Simulation metamodels are essentially input/output functions that approximate the true, and usually much more complex, unknown function inherently defined by the simulation model itself. Commonly, many of these inputs are shared with those of the simulation model, although it is not entirely necessary. The metamodel can have extra input variables defined as functions of the original simulation inputs.

Simulation metamodeling can encompass four possible major goals, namely, problem entity understanding, simulation output prediction, optimization, and verification/validation [5]. This project is not only but mostly concerned with understanding the underlying real system and with assessing the prediction performance of the metamodel. The assumption is that the ATM simulation model of interest is perfectly validated, verified, optimized, and thus calibrated with respect to the real problem under study. To this end, two ATM simulation models, namely FLITAN and MERCURY, previously developed by some of the consortium's partners, will be used as test cases. Figure 2 depicts the relations and dependencies between real-world problem, simulation model and simulation metamodel.

The ultimate objective is to fit a metamodel that is capable of predicting, with reasonable accuracy, the output values of the base simulation model, so that it can be used as a valid modeling replacement. Generally, the metamodeling methodology consists of three main steps, namely, the definition of the experimental design, metamodel specification, and metamodel learning/fitting. The first step consists of strategically sampling the input space to generate a data set for model training. Then, steps 2 and 3, which are usually coupled, are conducted sequentially. While the former involves selecting a family of metamodel functional forms, the latter regards fitting the selected metamodel to the data set obtained from step 1.

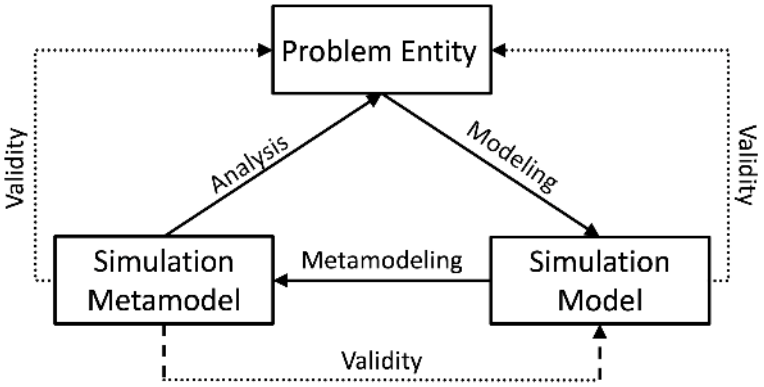

Figure 2 - Relationship between problem entity, simulation model, and simulation metamodel. Adapted from [5].

4 Preliminary Methodology

The current modeling strategy is based on an active learning scheme built on the top of a simulation metamodeling approach, and it is specially designed to extract relevant information regarding the underlying ATM simulation model under study with as few simulations runs as possible. As such, it combines the best of both worlds in order to jointly address the challenge of exploring the simulation input space, within the context of computationally expensive simulation models.

A Gaussian Process (GP) [6] is employed as a simulation metamodel. Thus, GP is considered to approximate the simulator's behavior and then used to explore the simulation input space. The fitness of the GP is then iteratively improved with active learning via simulation requests by decreasing the associated variance of the given predictions over the simulation input region of interest. Nevertheless, other machine learning models can be considered and adopted in the future, such as, for example, Artificial Neural Networks (ANNs).

The GP framework is a well-known modeling tool widely applied in numerous research and application fields. Due to its Bayesian formalism and highly non-linear properties, it constitutes an excellent option for designing active learning strategies based on simulation metamodeling settings. A critical characteristic of GPs is that they generate predictions in the form of Gaussian distributions, instead of pointwise ones, allowing for increased handling of uncertainty inherently present in any prediction process.

Any GP prediction is univocally defined by a Gaussian distribution with specific mean and variance parameters, oftentimes estimated via maximum likelihood conditional upon the training data. These parameters are also usually called predictive mean and predictive variance. Here, as mentioned earlier, the predictive variance has the crucial role of encapsulating the information potential of unlabeled points and their contribution to the underlying learning process. Notice, however, that this does not mean that the GP treats the target variable as to being normally distributed. Only the predictions themselves are generated as such. When these distributions exhibit relatively high variance around the mean value, a wider range of possible estimates are provided. This can be due to high variability naturally present in the observed data or to model uncertainty. The latter can be addressed by acquiring new data points through simulation runs in strategic input regions such as those associated with high predictive variance. From the model learning point-of-view, these unlabeled regions potentially encode more information than those in which the model is more certain and thus generally exhibiting low predictive variance values.

In this first version, a straightforward pool-based active learning strategy is adopted, similar to those seen in [7] [8]. The experimental design is depicted in Figure 3. Here the unlabeled data set U is entirely available for querying and represents the simulation input region in which we aim to explore the simulator's behavior. It should correspond to the simulation input region in which we aim to explore the simulation's output behavior. Ultimately, it should be suggested by domain experts and according to the concrete real-world problem being studied.

The pool of labeled instances L is comprised of simulation results, i.e., input-output tuples. The machine learning model M is a GP, whereas the query function Q is based on the analysis of the predictive variance provided by the latter at each point in U. Particularly, this function can be designed to query, via simulation requests, the points with the highest predictive variance or, in other words, with greatest prediction uncertainty.

The general idea within this experimental design is to assume that the functional relationship between the simulation input vector and the output is described by a GP. After the GP is fitted to L, the provided conditional distribution is used to predict the output values over U. This makes it possible to bypass simulation runs and to approximate the simulator behavioral structure, making the exploration process more efficient. The predictive variance is used as a measure of fitness, and it should decrease as the iterative process evolves. Finally, this trained GP model is used as a simulation metamodel to explore the behavior of the simulator and then to conduct policy analysis and assessment.

This approach is divided into three main blocks or steps. First, the simulation metamodeling approximates the simulation in question using a GP. Then, an active learning strategy is used to iteratively increase the fitting quality of the GP, by decreasing, for example, the total predictive variance across the unlabeled input simulation region. This metric is computed by summing the predictive variance provided to each unlabeled data point within U. Other metrics, relying on the predictive variance or not, can be considered in the future.

Finally, when the stopping criteria are verified, policy analysis by means of the provided meta-simulator is conducted. Here, notice that the predictive variance is used as a proxy measure of informativeness of each unlabeled data point, that is to say, that high variance points within the search space U potentially encompass more information.

The stopping criteria can be designed in various forms. A relatively straightforward approach is to consider it as a function of the predictive variance insofar that the addition of new training points will eventually have no significant impact on the model's performance. In other words, it is expected that the model will ultimately reach a prediction performance plateau where the further expansion of the training set does not justify the computational effort of acquiring new labeled data points. Hence, when the addition of new training points effectively translates into a low impact on the reduction of the overall predictive variance across U, the learning process may be stopped.

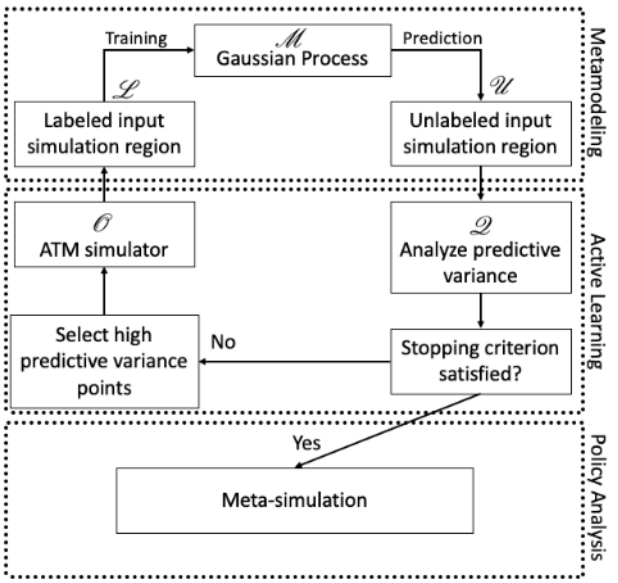

Figure 3 - Preliminary High-level Active Learning Metamodeling approach.

In practice, this methodology will require a constant link between the algorithmic script and the simulator platform itself, either through a configuration file (usually a text-based file such as txt or csv) or via an API. The former approach constitutes the simplest way to proceed and will be adopted in the preliminary stages of the methodology, as illustrated in Figure 4.

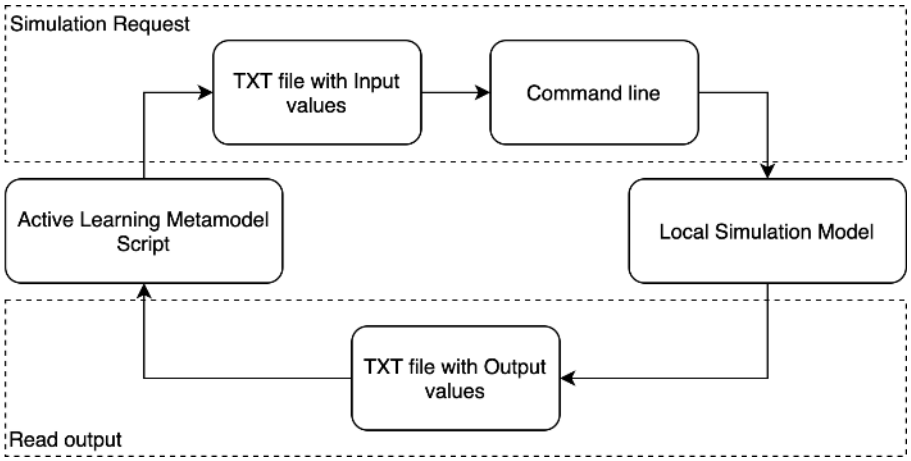

Figure 4 - The dynamic communication link between script and simulator.

5 Data – Test Cases

As mentioned previously in this document, FLITAN and MERCURY simulation models will be used to develop test cases and to conduct preliminary exploratory experiences.

This first approach should employ simplified versions of these simulators. One way to attain this is to fix most of the inputs while exploring the output behavior in relation to only a small set of variables. A set of 1-3 inputs and outputs (or KPIs) would be ideal to start exploring and assessing the performance of this methodology and eventually draw future lines of fine-tuning and improvement. Whereas these test cases may be unrealistic from the domain application point-of-view, they constitute a proper sandbox for the first iterations of the metamodeling methodology.

The GP framework can be used either as a regression or a classification tool, essentially depending on the nature of the output metric of interest. However, continuous variables are typically better handled when search methods are required. Thus, this first iteration should explicitly deal with continuous variables, although it is not entirely needed.

In the case of FLITAN, the TMA departure time can be used as the single variable input, whereas for the MERCURY, the Turn-around time would be a good option.

The selection of an output metric of interest, or several, does not pose a real concern insofar that they will always be produced and accessible after each simulation run, regardless of the input variables under study. Multiple GPs can be easily employed to model different output variables independently. In the future, however, a multiple-output approach can be considered.

6 References

[1] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009

[2] Xizhao Wang and Junhai Zhai. Learning with Uncertainty. CRC Press, 2016

[3] L. W. Friedman, The simulation metamodel. Springer Science & Business Media, 2012

[4] Russell R Barton. Simulation metamodels. In Simulation Conference Proceedings, 1998. Winter, Volume 1, pages 167–174. IEEE, 1998

[5] Jack PC Kleijnen and Robert G Sargent. A methodology for fitting and validating metamodels in simulation. European Journal of Operational Research, 120(1):14–29, 2000

[6] C. E. Rasmussen and C. Williams, Gaussian processes for machine learning (Adaptive computation and machine learning). The MIT Press, 2006

[7] F. Antunes, B. Ribeiro, F. Pereira, R. Gomes, Efficient transport simulation with restricted batch-mode active learning, Transactions on Intelligent Transportation Systems pp 1–10, 2018

[8] Antunes, F., Amorim, M., Pereira, F. C., & Ribeiro, B., Active learning metamodeling for policy analysis: Application to an emergency medical service simulator. Simulation Modelling Practice and Theory, 97, 2019


CRIDA
I·D·I - ATM


NOMMON


UNIVERSITY OF
WESTMINSTER


UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH


Technical University
of Denmark
DTU


ISA
Innovation for Sustainable Aviation