**Hackathon n° 3**

**Questionnaire sur un cas d’école parmi les applications de votre entité**

**Partie I - Spécifications fonctionnelles (matinée)**

1. Pour chacun des acteurs suivants, dire quels sont ses buts dans l’utilisation **actuelle** des outils de réduction de modèles **de votre entité**

|  |  |
| --- | --- |
| **Utilisateur boîte noire** | Buts (en 1 phrase) |
| Utilisateur du modèle réduit en boite noire, sans le connaître | * Créer et utiliser un modèle réduit à partir d’un DoE * Utiliser la réduction de modèle dans les problème d’optimisation. * ………. * ………. * ………. |

|  |  |
| --- | --- |
| **Utilisateur connaissant le modèle complet et/ou la physique** | Buts (en 1 phrase) |
| Utilisateurs connaissant le modèle complet et/ou la physique représentée | * ......... * ......... * ………. * ………. * ………. |

|  |  |
| --- | --- |
| **Utilisateurs sachant établir un modèle réduit** | Buts (en 1 phrase) |
| Utilisateurs sachant élaborer le modèle réduit à partir du modèle complet | * ......... * ......... * ………. * ………. * ………. |

1. Pour chacun des acteurs suivants, dire quels sont ses buts (non listés précédemment) dans l’utilisation **future** des outils de la bibliothèque Mordicus **par votre entité**

|  |  |
| --- | --- |
| **Utilisateur boîte noire** | Buts (en 1 phrase) |
| Utilisateur du modèle réduit en boite noire, sans le connaître | * Communiquer directement avec les logiciels de simulation par APIs * ......... * ………. |

|  |  |
| --- | --- |
| **Utilisateur connaissant le modèle complet et/ou la physique** | Buts (en 1 phrase) |
| Utilisateurs connaissant le modèle complet et/ou la physique représentée | * ......... * ......... * ………. |

|  |  |
| --- | --- |
| **Utilisateurs sachant établir un modèle réduit** | Buts (en 1 phrase) |
| Utilisateurs sachant élaborer le modèle réduit à partir du modèle complet | * ......... * ......... * ………. |

1. Parmi ces buts, choisissez un « cas d’école », emblématique (ou du moins typique) de la pratique de la pratique de la réduction de modèle dans votre entité.
   1. Cocher les fonctionnalités génériques suivantes, listées lors du premier Hackathon, auxquelles votre cas serait susceptible de correspondre ou de faire appel :

*Utilisateur boîte noire*

Usage d’un modèle réduit en plan d’expérience

Utilisation d’un modèle réduit avec des mesures in-situ (en laboratoire, sur site de production...)

Disposer d’un “clone digital” basé sur un modèle réduit

Générer un champ complet à partir d’un modèle réduit

Permettre la visualisation des résultats du modèle réduit

Archiver une liste de modèles réduits

Gérer une taille mémoire prescrite pour l’élaboration d’un modèle réduit

Couvrir avec un modèle réduit un sous-domaine d’usage du modèle haute fidélité

Garantir qu’un modèle réduit fonctionne de la même manière que le modèle haute fidélité sur un sous-domaine

Couplage spatial entre un modèle réduit et un modèle haute fidélité

*Utilisateur connaissant le modèle complet et/ou la physique*

Création d’un modèle réduit pour effectuer un plan d’expérience (sans exigence de méthode particulière de réduction, mais avec un contrôle de sa fiabilité)

Comparer un modèle réduit 1 avec un modèle haute fidélité 2

Optimisation de loi(s) de comportement sur structure complexe

Faire interagir un modèle réduit et des opérations de Data Science

*Utilisateurs sachant établir un modèle réduit*

Création d’un modèle réduit avec des informations provenant de mesures

Déduire un modèle réduit 2 à partir d’un modèle réduit 1

Permettre le calcul multi-échelles ou multi-physiques de systèmes représentés par des modèles réduits

Permettre la visualisation du modèle réduit lui-même

Construire un modèle réduit 3 en combinant deux modèles réduits 1 et 2

Évaluer l’intérêt de la démarche “Offline+Modèle réduit” par rapport à l’utilisation directe du modèle haute fidélité

Construire un modèle réduit lorsque le maillage (voire la géométrie) change entre les différents snapshots

Utilisation de tous les calculs intermédiaires existants pour effectuer la réduction

* 1. Si votre cas ne s’apparente à aucune des fonctionnalités ci-dessus, proposer une formulation générique de votre cas :

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

1. Décrivez votre cas d’usage en remplissant le tableau suivant (exemple en page …)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE #** | < the name is the goal as a short active verb phrase >  Evaluate the response of new parameter set from the data generated by simulation. | | |
| **Context of use** | Users simulate their problems with various parameter set. Nodal value will be saved as training data. The solution of new parameter set is evaluate by using model reduction. | | |
| **Scope** | <what system is being considered black box under design>  The model reduction module of ODYSSEE. | | |
| **Level** | <one of : Summary, Primary Task, Subfunction>  Primary Task | | |
| **Primary actor** | User with knowledge of the physics problem, knowledge of model reduction is not necessary . | | |
| **Stakeholder & Interests** | **Stakeholder** | | **Interest** |
|  | <stakeholder name>  …………………………….  ……………………………. | | <put here the interest of the stakeholder>  …………………………………………………………………………………………  ………………………………………………………………………………………….. |
|  | <stakeholder name>  ……………………………  ………………………….. | | <put here the interest of the stakeholder>  …………………………………………………………………………………………  ………………………………………………………………………………………….. |
| **Preconditions** | <what we expect is already the state of the world>  Have available:   * Result fields or requested responses for all time steps. * The DoE used for generating the results/ responses. * New parameter set to evaluate results/ response. | | |
| **Success End Condition** | The expected data is produced | | |
| **Failed End Protection** | Process is terminated, users get informed about computation time-out. | | |
| **Trigger** | Users enter the model reduction modul of ODYSSEE | | |
| **DESCRIPTION** | **Step** | **Action** | |
|  | 1 | Users fill the requested inputs : 3 csv files | |
|  | 2 | A csv file which contains the DoE user for generating the training data (*input*) | |
|  | 3 | A csv file which contains actual training data (*output*) | |
|  | 4 | A csv file which contains the new parameter set used to evaluate the results (*new\_input*) | |
|  | 5 | Users demand the execution of the module to obtain *new\_output* | |
|  | 6 | System verifies if the provided data is valid | |
|  | 7 | System generates the reduced model from *input* and *output* | |
|  | 8 | System computes the response of new set of parameter (*new\_output)* and save in a csv file | |
|  | 9 |  | |
|  |  |  | |
| **EXTENSIONS** | **Step** | **Branching Action** | |
|  | 1a | Submitted data is corrupted or wrong format :  Notification for users | |
|  |  | Non matching dimensions between input files:  Notification for users | |
|  |  | ………………………………………………………………………………………………………………………………..  ……………………………………………………………………………………………………………………………….. | |
|  |  | ………………………………………………………………………………………………………………………………..  ……………………………………………………………………………………………………………………………….. | |
|  |  | ………………………………………………………………………………………………………………………………..  ……………………………………………………………………………………………………………………………….. | |
|  |  | ………………………………………………………………………………………………………………………………..  ……………………………………………………………………………………………………………………………….. | |
| **I/O VARIATIONS** |  | **Branching Action** | |
|  | 1 | <list of variations> | |
|  |  | …………………………………………………………………………………………………………………………………..  ………………………………………………………………………………………………………………………………….. | |
|  |  | …………………………………………………………………………………………………………………………………..  ………………………………………………………………………………………………………………………………….. | |
|  |  | …………………………………………………………………………………………………………………………………..  ………………………………………………………………………………………………………………………………….. | |
| **RELATED INFORMATION** |  | <list here related documentation to this use case, and whether or not you are ready to make it available to other members of Mordicus> | |
|  |  | …………………………………………………………………………………………………………………………………..  ………………………………………………………………………………………………………………………………….. | |
|  |  | …………………………………………………………………………………………………………………………………..  ………………………………………………………………………………………………………………………………….. | |
|  |  | …………………………………………………………………………………………………………………………………..  ………………………………………………………………………………………………………………………………….. | |
|  |  | …………………………………………………………………………………………………………………………………..  ………………………………………………………………………………………………………………………………….. | |

**Exemple (sous forme de tableau)**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE #EDF.01** | Apply the offline stage for a linear parametric parabolic problem | | |
| **Context of use** | Expert user establishes the needed autonomous data structure for running a reduced-order simulation of a parametric parabolic problem, with operators linearly dependent of the parameters and the state. For instance, a linear transient thermal analysis with an explicit parameter variation of the conductivity, capacity or exchange coefficient falls into this case. | | |
| **Scope** | The offline component of Mordicus library | | |
| **Level** | Primary Task | | |
| **Primary actor** | User with knowledge of reduced-order modelling techniques | | |
| **Stakeholder & Interests** | **Stakeholder** | | **Interest** |
|  | User of a reduced-order model in his field | | wants a reduced-order model that behaves as the full-order one from his point of view;  wants clear and direct access to varying parameters and physical case description (for control) |
|  | Developer of a leightweight field-specific application | | wants an clear and portable output data structure with minimal software dependencies |
| **Preconditions** | Have available:   * a nominal full-order case of study of a phenomenon governed by parabolic equations * its result fields for all targeted time steps at least one parameter value, possibly in different folders but in the same filesystem * pre-computed full-order operators : C (unit “capacity” matrix), K (unit “conductivity” matrix), f (unit “source” right-hand side vector), fb (unit “Robin condition” right-hand side vector, zone-wise on the border), Kd (unit “Robin condition” right-hand side matrix, zone-wise on the border). Each right-hand side vector is accompanied by one time-evolution file | | |
| **Success End Condition** | The expected autonomous data structure was produced | | |
| **Failed End Protection** | Only the responsible process is stopped. User gets informed about missing or erroneous input data, or algorithm configuration at failure | | |
| **Trigger** | User enters the offline component of Mordicus while announcing a parabolic problem | | |
| **DESCRIPTION** | **Step** | **Action** | |
|  | 1 | User: fills user input for the offline stage  Demanded input data: parameter values (at least one) and a path to one time-history result file for each; pre-computed operators C, K as path to different files; truncation criterion; path for output data structure  Optional input data: f and fb with their time-evolution files; Kb (zones should be consistent); format for the description of the operators in the output data structure (one of ‘MED’, ‘numpy’… numpy is default); format for time-evolution files (csv…); path for log files of the offline phase (for diagnosis); capacity, conductivity or exchange coefficients if fixed | |
|  | 2 | User : demands execution of the offline phase | |
|  | 3 | System : verifies that input data is provided and valid. Extracts initial state from provided results. | |
|  | 4 | System : applies the progressive RB algo (greedy selection, appeals to incremental SVD only), produces a reduced-order basis | |
|  | 5 | System : produces compressed operators for unitary M, C, f and initial state vector f0 by algebraic projection | |
|  | 6 | System : packs the following into an output data structure: compressed operators, time-evolutions, fixed coefficients, and either links to high-fidelity file results or assumed mesh and time discretization. Writes it to a file at the location specified by the user. | |
|  | 7 | System : notifies use of completion | |
| **EXTENSIONS** | **Step** | **Branching Action** | |
|  | 3a | Submitted data is incomplete (some required input is not provided) :  ask missing data from the user | |
|  | 3b | Submitted data is erroneous due to wrong file path:  Inform user of erroneous file path, save existing input data, clean exit | |
|  | 3c | Not all provided results refer to the same mesh or zones of provided operators Kd, fd are not consistent :  Inform user and clean exit | |
|  | 3d | Time-history of the input computation results are not consistent with each other or with the provided time-evolutions :  Inform user and clean exit | |
|  | 4a | Dimensions of input results are not compatible :  Inform user of responsible files and time steps, exit | |
|  | 5a | Incompatible size error occurs on the projection:  Inform the user of the nature of the responsible matrices/vectors, exit | |
|  | 6a | Invalid or taken output file location:  Generate valid file path, save there, inform user, continue | |
| **I/O VARIATIONS** |  | **Branching Action** | |
|  | 3 | I/O variation: Initial state may be given explicitly by the user if not present in the initial results | |