Uncertainty Estimation for Human-System Interactions – Recent Developments and Application of the MCDET Crew Module

Tanja Eraerds[a*], Inés Mateos Canals[a], Jürgen Hartung[a], Josef Scheuer[a], Jan Soedingrekso[a]
[a] Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, Garching b. München, Germany

Abstract: An integrated dynamic probabilistic safety analysis (IDPSA) combines the enhanced realism of a deterministic safety analysis (DSA) with the advantages of a probabilistic safety analysis (PSA). The GRS software tool MCDET (*M*onte *C*arlo *D*ynamic *E*vent *T*ree) for dynamic PSA allows to analyze and quantify the influence of (aleatory and epistemic) uncertainties on the behavior of dynamic systems over time. It can be used both to detect unforeseen accident sequences as well as to quantify the dependence of certain end state scenarios on the regarded uncertain input parameter. It is particularly suitable for analyzing scenarios that have both a strong dependence on the exact timing of events and an inherent uncertainty of this timing, such as accident scenarios involving mitigating human actions. To simulate human action sequences and to model the uncertainty regarding duration and success of these, the MCDET is complemented by the MCDET Crew Module. Recent developments have transformed the Crew Module from a MCDET dependent tool to a stand-alone deterministic simulation code that can be coupled with MCDET like other deterministic simulation codes, such as ATHLET (*A*nalysis of *T*hermal *H*ydraulics of *Le*aks and *T*ransients).
This paper describes the improved MCDET Crew Module and presents an example application for an accident scenario during mid-loop operation. Based on this example application, a comparison between the functionality and capabilities of the former and the improved MCDET Crew Module is provided.

Keywords: MCDET, Crew Module, uncertainties, integrated dynamic probabilistic safety analysis.

1. INTRODUCTION

Complex technical facilities such as nuclear power plants (NPPs) are susceptible to a variety of different parameters, such as the void fraction in the core, the pressure of the primary cooling liquid or the heat flow in the heat exchanger. Accidents, such as a break in a steam generator heating pipe, or e.g., a prolonged loss of offsite power (LOOP), affect the plant operation and the values of these parameters. Nuclear accidents cannot only be induced by plant intern events, but also by external hazards such as, earthquakes, external floods or aircraft crashes, or by internal hazards such as fires, explosions or internal flooding.

Human actions are often involved in mitigating the consequences of accidents; however, faulty human actions can also represent root causes of accidents. Human actions can be the subject of errors of commission by performing an undesirable action, errors of omission by failing to perform an action correctly and, in a more graded approach, errors by performing a desired action too late or by choosing a sub-optimal action sequence. Unforeseen effects of single or combined hazards can lead to unfamiliar situations in which the derivation of the optimal solution depends on the knowledge and experience of the operators, their physical and psychological well-being, and the complexity of the problem. The limited time available to resolve these potentially complex issues increase the stress level of power plant personnel and affects their ability to provide the optimum resolution strategy.

To provide a realistic estimation of the safety of such installations, these uncertainties and their consequences need to be assessed according to best practice. The time available, and therefore the stress level and the success probability of a human action, depend on the system state, which is likely to depend on previous mitigating actions and on random events (aleatory uncertainties). Therefore, the whole process can be regarded as dynamic in nature. For calculating the probability of different end system states, it is important to take this dynamic nature into account and to consider the interaction of different aleatory uncertainties, such as random events and the timing of human actions, as well as epistemic uncertainties, such as a lack of knowledge about the precise initial state of a plant or the uncertainty regarding the failure probability of a human action. An IDPSA combined with a human reliability analysis (HRA) allows to consider these time dependent interactions between human actions, system states and random influences. The GRS software tool for IDPSA, MCDET [1], [2], combines the advantages of the realism of a deterministic safety analysis by performing best-estimate simulations of the plant behavior with the effective treatment of discrete aleatory uncertainties of a probabilistic safety analysis.

MCDET uses dynamic event trees to account for the effects of discrete aleatory uncertainties while maintaining the realism of a deterministic simulation. Starting from the root simulation process, at each point in simulation time where a discrete aleatory uncertainty affects the simulation, branching occurs, with each new branch corresponding to a potential state of the parameters under consideration. The state of the parent simulation is copied, and the values of the affected parameters are adjusted. In addition, uncertain parameters can be sampled using Monte Carlo techniques, allowing classical statistical analysis techniques such as tolerance bounds to be applied to estimate the probability of different end state scenarios. Each set of sampled parameters corresponds to one dynamic event tree (DET). Moreover, MCDET allows sampling during the analysis process; in this way the current system and process states can influence the sampling process, e.g., the mean value of the sampling distribution. MCDET itself can sample from the following distributions: polygonal, discrete, histogram, log-histogram, (truncated) normal, (truncated) log-normal, uniform, log-uniform, triangular, log-triangular, (truncated) Weibull, beta, (truncated) gamma, (truncated) exponential, extreme values. In case no sampling during the analysis is required the sampling can also be performed in a pre-processing step using the GRS software SUSA for sensitivity and uncertainty analyses [3], [4], which allows to take into account dependencies between uncertain parameters.

MCDET is accompanied by a method for modelling and simulating the interaction between system states and human behavior, the Crew Module [5]. The structure and input format of the classic Crew Module are outlined in the following Section 2. Section 3 details the changes made to the classic Crew Module and the new functionalities. Section 4 presents an example application and based on this example application, the modelling capabilities of the Crew Module. Finally, Section 5 provides a conclusion and an outlook on future developments of MCDET and the Crew Module

2. CLASSIC MCDET CREW MODULE

2.1. Main Objectives and Implementation Consequences

The approach chosen for the MCDET Crew Module is outlined in detail in [5]. The two main objectives are to assess the point in time when a complex human action is performed, and the probability of an end system or process state resulting from a complex human action. Such a complex human action could be e.g., performed to mitigate the consequences of a mid-loop accident scenario.

To get an as far as possible realistic estimate of the completion time of the action and the probabilities of different end states, the action is decomposed into several basic actions. The reasoning behind this decomposition is that even though the complete action may be too complex to be reliably assessed by experts, the granularity of the basic actions can be chosen such that a reliable estimate of the duration is possible. It is therefore advisable to choose the granularity of the basic actions carefully to achieve the desired final accuracy. In the context of [5], a basic action is assumed to be performed by a single person. Further developments have extended this concept to include actions performed by a group of people working on a single task, e.g., a team of firefighters working together to successfully suppress a fire. The main attributes of a classic Crew Module action are the subject of the action, the object of the action, which can also be a technical component, and the duration of the action. It is assumed that some actions always follow each other irrespective of the boundary conditions. These are so-called action lists (Als), with each AL being associated to a separate ID. In addition to ALs, certain branch points can be specified at which aleatory uncertainties, system or process states could influence the further sequence of actions, i.e., the next AL. Each branch point must be assigned to a condition under which the branching takes place. In addition, at each branch point, the values of a set of Crew Module variables are considered, and the subsequent AL is selected based on these values. Each set of variable values is associated with a probability. These probabilities could, for example, depend on the assumed stress level, thus allowing the interdependence of human reliability and stress to be modelled. These probabilities can be modelled either as scalar values or as a probability distribution from which the final probability can be sampled.

The classic Crew Module also allows to simulate actions to be performed in parallel. This parallelism is implicitly considered if different people are involved in subsequent actions in an AL. In this case, these subsequently listed (consequential) actions are started together in the modelling by the Crew Module. The

Crew Module also uses the information about the subject and object of the actions in an AL to implicitly model the order in which these actions are performed. The time and duration of the process in which a person is informed about a task, i.e., is the object of the communication process, defines the starting point in time at which this person can perform the task. It must be noted that both the communication process as well as the task performance are typically modelled as actions in the Crew Module. During the time when a person is actively involved in an assigned task, it is assumed that this person cannot perform any other action. The action that starts first will therefore be completed before the person involved can start another task (action).

2.2. The Classic Crew Module and Its Interface to MCDET and SUSA

The classic Crew Module is written in FORTRAN. It reads directly from a supplied input file the sampled time values for those actions whose duration is assumed to be uncertain and to follow a probability distribution function. It also reads in the sampled probability values for the probability of certain sets of Crew Module variables at the Crew Module branch points. The input format of the sampled time and probability values corresponds to the output format of the GRS software SUSA for sensitivity and uncertainty analyses [3], [4]. In case the duration of the action or the branching probability depends on the system state, different sets of variable values are sampled for the different relevant system states. An example is the influence of stress. The stress factor can depend on the process and the system state and affect the success probability of the actions modelled in an AL and thus the probability of the subsequent ALs. The classic Crew Module can also be used to sample some simple distributions, e.g., a uniform distribution. This capability is used to simplify the process of sampling from many similar distributions, i.e., the many time distributions needed to model a complex human action. These time distributions are usually assumed to be uniform, for simplicity reasons.

The classic Crew Module calls the classic FORTRAN based MCDET version. It calls MCDET at specifically marked branch points where information about the state of the process is required for action sequences to proceed. The MCDET software checks the shared Crew Module state and, depending on this state, provides sets of modified Crew Module variables. For each of these sets, the Crew Module proceeds to iterate through the following ALs and corresponding actions, updating time and Crew Module variables as defined in the Crew Module input.

2.3. Input Format

The latest input format for the classic Crew Module is a mind map based graphical user interface (GUI) using the free software FreeMind [6]. The mind map structure lends itself to modelling the tree-like structure inherent in the Crew Module input layout. The Crew Module has been extended by a program ‘readMindMap’, which reads the information provided in the GUI and passes it on to the Crew Module. Rules have been defined [5] ensuring that the ’readMindMap’ program can analyze the input provided. A part of one exemplary Crew Module input is presented in Figure 1, with the time progressing from left to right. The first three ALs are specified. As shown in Figure 1, each AL starts with a line giving the ID of the AL, e.g., AL 1.0, possibly followed by a '//' sign and a description of the AL. The next lines of the AL specify the condition, indicated by the keyword ‘Cond: &’ followed by the logical condition to be fulfilled for this AL. This logical condition must include the identifier of the previous AL, indicated by the keyword ‘ALcont =’, and the ID of the previous AL. The logic condition can also contain conditions for other Crew Module variables which need to be fulfilled, e.g., ‘b_recFalseStim = 1’, where b_recFalseStim is one of the Crew Module variables. In case of the first AL the predecessor identifier is ‘ALcont = 0’, meaning that no predecessor AL exists. The different parts of a condition are separated by an ‘&’ meaning an ‘and’ concatenation. No other concatenation is possible in the classic Crew Module. The remainder of the AL field contains identification information for the actions that make up the AL. Separated by a '/' specifier, the information is:

- the ID of the action subject,
- the ID of the action object; if the action has no object, this is indicated by a '-' sign,
- the minimum duration of the action,
- the maximum duration of the action,

possibly followed by a '//' specifier and further description of the action.

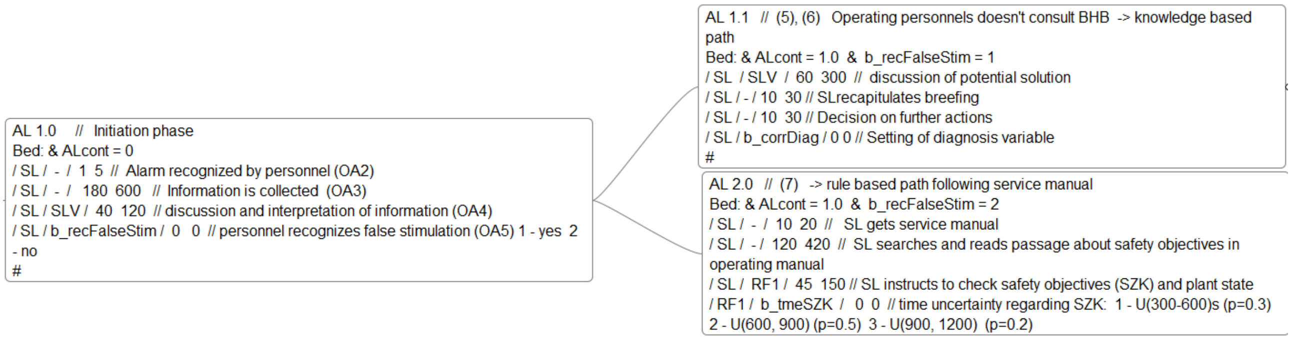

Figure 1. Part of a Crew Module input, containing the first action list AL 1.0 and its child action lists AL 1.1 and AL 2.0

The classic Crew Module also allows key signs to be used instead of an action object. These key signs could be specified in a separate input file and were typically negative numbers. For example, '-99' indicated the end of a sequence. Other negative numbers indicated that an interaction with MCDET was required at that point. Depending on the value of the key sign and the state of the Crew Module, new branches have been generated with different values of the variable linked to the key sign and different probabilities. In the GUI, new AL boxes can be generated as child nodes of existing AL boxes. These new ALs are linked to the parent ALs by thin lines. However, these lines are not necessary for the Crew Module to link one AL to another one, as this is entirely based on the 'ALcont' identifier in the head of the AL. In this way, different ALs could lead to the same child ALs, which is not inherently supported by the FreeMind software.

3. NEW MCDET CREW MODULE

MCDET has undergone several changes in the recent past [2]. Much of its functionality has been ported from FORTRAN to Python, its structure has been reordered and modularized, and a completely new scheduler system has been added allowing to run different DET branches in parallel. Currently, the estimator core of MCDET is being ported to Python. The estimator core is responsible for evaluating the state of a simulator and providing so-called change sets, where each change set corresponds to a new set of simulator variables and corresponds to a potential new branch of a DET (the branch is only generated if its probability is higher than a predefined cut-off probability).

The classic Crew Module does not work with the new MCDET version. Therefore, and to make the Crew Module maintainable and its results more reproducible, it was decided to rewrite the Crew Module. To achieve the intended maintainability, the Crew Module was modularized choosing an object-oriented approach. The aim was that the new Crew Module should be able to perform the same tasks as the classic one and be able to read the same input files for quality assurance. The classic Crew Module was a mixture of a probabilistic and a deterministic simulator, this has now been disentangled. The new Crew Module is a purely deterministic simulator that can be linked to MCDET. Based on the provided mind map input files, default values are determined for each Crew Module variable. Without further interaction with MCDET, the Crew Module would iterate through the ALs as predefined by the default values. The probabilistic part of the classic Crew Module has been completely removed from the Crew Module. As with all coupled deterministic simulators, this probabilistic part, e.g., the branching conditions, and the resulting variable change sets are now defined in the MCDET input. The MCDET input can also be used to specify whether the continuous aleatory variables are to be sampled by MCDET or read in from another source. Typically, the GRS software SUSA is used for external sampling of the uncertain aleatory or epistemic variables, in which case all SUSA distribution functions, as well as the possibility of defining dependencies, can be used. As a rule, the GRS software SUSA is used for external samples of the uncertain aleatory or epistemic variables. In doing so, all SUSA distribution functions and the option of defining dependencies can be used.

3.1. Structure of the New Crew Module

The main components of the new Crew Module are:

- **Resource:** Each entity involved in an action is an object of the resource class, e.g., an acting person or an object necessary for an action. A resource object corresponds to the subject and object of the classic crew module. The main attribute of a resource is its state, which can be busy if the resource is actively participating in an action, waiting or idle.
- **Action:** An object of the action class is anything that changes the state of the Crew Module. The state of the Crew Module consists of the state of all variables declared in the crew module input and the time. An action has an ID, it uses resources, it causes a change of state, and it has a duration.
- **Condition:** The condition class is used to contain the logical condition under which an AL is selected. Currently it contains a comparison between a variable, a threshold, and a stimulus. The stimulus is an indicator of whether the variable value should be larger than, less than or equal to the threshold value. In the future, this will be replaced by a logical expression that can be evaluated in Python.
- **Action List:** An object of the AL class corresponds to the AL concept of the classic crew module. An AL consists of an ID, a list of associated actions, a list of necessary conditions and a list of possible next ALs. It also contains the ID of the currently active action and the ID of the last active action.
- **Action Getter:** An object of action getter class is responsible for providing the next possible action to be performed, its attributes are the list of all ALs and the ID of the currently active AL.
- **Mind Map Reader Module:** The 'produce_crewmodule_input' function of the mind map reader module reads the mind map input file and produces the necessary input for the crew module. It returns a dictionary of all ALs, a dictionary containing the default state of each Crew Module variable and the first AL.
- **Crew Module**: An object of the Crew Module class provides all the necessary requirements for a simulator to be compatible to MCDET [2]:
  - It provides restart capabilities such that the whole Crew Module can be set and started from a predefined state and that the current state of the Crew Module can be saved. Its attributes are a restart file to which the Crew Module state can be written and from which a previous Crew Module state, the Crew Module state itself and an object of the action getter class can be read to get the next action to be performed. To start and restart a Crew Module run, the Crew Module model is read from a mind map file. Care has been taken to set up the new Crew Module so that it can run an analysis based on a mind map file written for the classic Crew Module. Consistency between classic and new results has been tested for selected examples.
  - It provides a loop in which new actions are called from the action getter and these are applied to the state of the Crew Module. It therefore runs in predefined time steps.
  - The state of the Crew Module can be accessed externally. Hooking points have been set in this Crew Module event loop. At these hooking points, the Crew Module simulation is halted, and the state of the Crew Module is shared with MCDET. MCDET evaluates the current state of the Crew Module, determines if branching is necessary, and provides new sets of modified variables for each of the new branches. For each new branch, the Crew Module clones itself, resets each clone to the last saved restart point and applies the provided sets of changed variables on top. Then MCDET restarts the parent simulation, therefore MCDET steers the progression of the Crew Module.
  - The loop can be stopped based on the state of the shared Crew Module variables. This allows MCDET to trigger a simulation stop.

3.2. Crew Module/MCDET Output

Such as the GRS thermal hydraulic simulation software ATHLET [7] or the simple tank simulator, the new Crew Module can be controlled via MCDET. MCDET's probabilistic module can sample different values for continuous aleatory variables, such as the duration of a maneuver, and the scheduler generates different event trees for the sets of sampled variables. The variables are sampled from a probability distribution specified by the user. For discrete aleatory variables, e.g. the success or failure of a control action, MCDET generates branches of an event tree. The output format is hierarchically structured in hdf5 tables. Each table is connected to a branch, even information about branches whose probability is below a predefined threshold is stored in tables. For each branch a so-called event table and a state table exist. The event table contains information about events which affect the structure of the event tree, such as transitions (branch points) and sequence terminations. The state table contains information on the values of parameters of the coupled deterministic simulation for each time step of the simulation. The variables for which information shall be stored in the MCDET output have to be defined in the MCDET input. MCDET provides several post-processing tools to analyze such hierarchical hdf5 files. The Crew Module run in stand-alone mode provides error and warning logging messages as well as information about the state of the Crew Module variables after each action in form of json files. Figure 2 shows a part of the hdf5 state table associated with one branch of a decision tree generated for the example application outlined below in Section 4. Each row corresponds to one Crew Module time step and therefore to one performed action of the modelled AL. The length of a Crew Module time step depends on the time of the performed action. Of the presented columns the columns ‘time’ (simulation time), ‘action’ (action ID) and ‘action_list’ (AL ID) are available for each Crew Module run. The other columns in the table vary from application to application and depend on the input modelled by the user. The columns shown in Figure 2 are just part of the 24 simulator variables needed for the application below. Each time a branch point is reached, one or more new branches and therefore new hdf5 tables are created. The state of the presented simulator variables such as ‘b_corrdiag’ (the correctness of the diagnosis performed) are modelled as depending on the regarded branch and are not changed by the performed actions.

	time	action	action_list	b_recFalseStim	b_corrdiag	b_setnkk	b_set2nkk	b_jr41
0	3.0	0.0	1.0	2.0	1.0	1.0	1.0	1.0
1	393.0	1.0	1.0	2.0	1.0	1.0	1.0	1.0
2	473.0	2.0	1.0	2.0	1.0	1.0	1.0	1.0
3	473.0	3.0	1.0	2.0	1.0	1.0	1.0	1.0
4	488.0	0.0	2.0	2.0	1.0	1.0	1.0	1.0
5	758.0	1.0	2.0	2.0	1.0	1.0	1.0	1.0
6	855.5	2.0	2.0	2.0	1.0	1.0	1.0	1.0
7	855.5	3.0	2.0	2.0	1.0	1.0	1.0	1.0

Figure 2. Part of the hdf5 state table associated with one branch of one decision tree generated by a MCDET/Crew Module run for the new Crew Module for the application presented in Section 4; not all state variables are visible in the part of the state table presented.

4. EXAMPLE APPLICATION: ACCIDENT DURING MID-LOOP OPERATION

In the following, the initiation phase, and the problem-solving attempt (PLA) of an Operator Action Model (OAM) and the corresponding MCDET / Crew Module input are derived by means of an example for the application of the Crew Module for an analysis of the human influence on accidents in mid-loop operation. This example is intended to mainly illustrate the Crew Module modelling approach; however, in case the input modelling is affected by differences between the classic and the new Crew Module these differences are discussed.

Mitigating human actions can be highly relevant for accidents during low power and shutdown operation, as shown in past PSAs for NPPs [8]. Mid-loop operation is of interest for reactor safety due to the inherent decoupling of the reactor protection system (RPS) and the reduced availability of the instrumentation and control (I&C) systems. In addition, it can be argued that the resulting increased workload and importance of the operating personnel in turn increases the likelihood of human errors. Currently, an IDPSA is being performed using MCDET, the classic Crew Module and the GRS thermal hydraulic software ATHLET [7] to assess the human influence on the end states of a mid-loop operation accident scenario for a generic Western design1300 MWe pressurized water reactor. The corresponding research and development (R&D) project is entitled “Advanced Dynamic PSA Methods for Assessing the Effectiveness of Human Actions for Accidents in Mid-Loop Operation” (ADAMO). Information on the status of this R&D activity can be found in [9] to [11]. Basis for the research is the accident scenario “failure of the residual heat removal due to erroneous actuation of the reactor protection system”, which has a high safety significance (cf. [10]), as a major contributor to the core damage frequency.

The IDPSA for ADAMO consists of two main parts, namely:

1. a pre-analysis of human action time periods and the corresponding success probabilities applying MCDET coupled to the Crew Module, and
2. a simulation of the thermal hydraulic process and the influence of certain predefined aleatory uncertainties applying MCDET coupled to ATHLET [7]. MCDET can be coupled to different

simulation codes meeting certain general requirements provided in Section 2.1.6 of [2]. ATHLET is a thermal hydraulic code developed by GRS, it analyses the entire spectrum of probable incidents and accidents in nuclear reactors and allows to assess the safety of reactors of different construction lines. For analyzing core degradation / core melt MCDET would need to be coupled to the ATHLET extension ATHLET-CD (Core Degradation) [7]. This principally possible and has been tested but out of scope of the presented example application.

In the pre-analysis, the cumulative density functions for the time required to reach certain end states and the probability of these end states are derived. The GRS software SUSA is used to sample from these distributions. The sampled values serve as input for the MCDET/ATHLET simulation run. The three-phase-model of the human problem-solving process developed by Fassmann et al. [5], [12], presented in Figure 3, provides the basis for the development of the OAM.

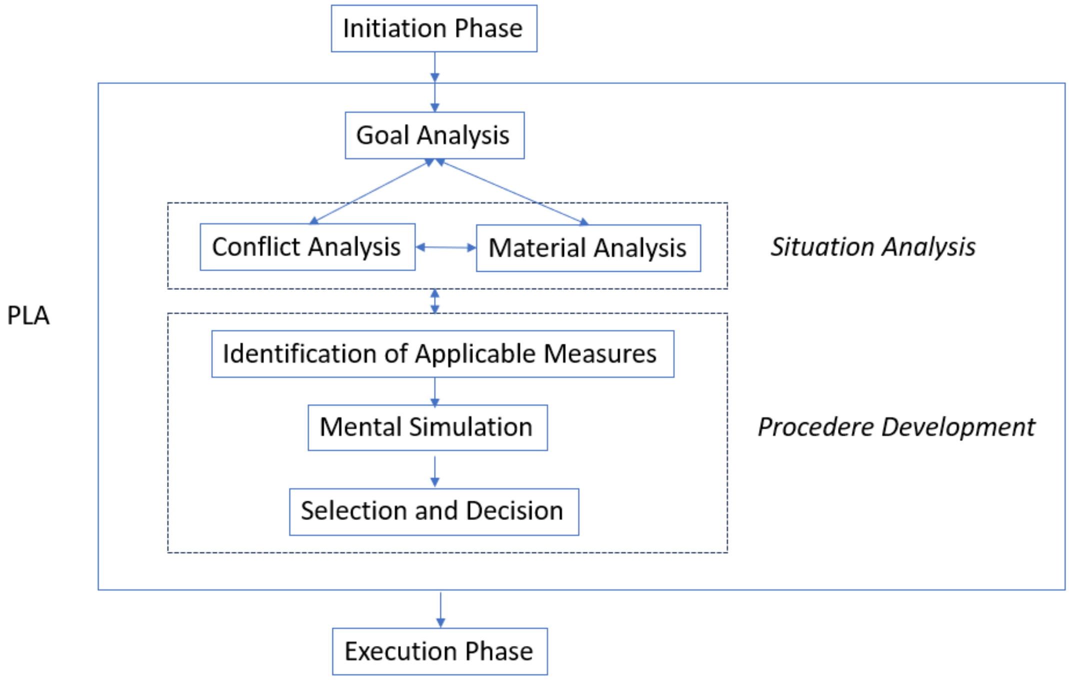

Figure 3. Flowchart based on the generic model of problem-solving [3]

The three phases are:

- **Initiation Phase:** This phase, sometimes also called “Pre-Phase“, initiates the problem-solving attempt, it is the state just before such an attempt is deemed necessary. The conditions in this initiation phase influence the success of the next phases, i.e., stress might for example reduce the probability of a successful problem-solving [13].
- **Problem-solving Attempt (PLA):** In this phase, the goals, the available knowledge, and resources as well as potential conflicts are analyzed (Situation Analysis). A procedure for solving the problem is developed employing mental simulation of different potential problem-solving strategies. It must be noted that the goal analysis is in turn influenced by the conflict and the material analysis, meaning that the goals can be adapted depending on the available material and potential conflicts.
- **Execution Phase:** The selected problem-solving strategy is executed.

The OAM derived includes aleatory uncertainties for the execution times of human actions and the dependence of human actions on stochastic influences and/or system and process states as well as human error probabilities (HEP). The basis for the quantification of execution times and HEP are the HRA methods ASEP (*A*ccident *S*equence *E*valuation *P*rogram) [14] and THERP (*T*echnique for *H*uman *E*rror *R*ate *P*rediction) [15]. Expert judgement was combined with ASEP recommendations to calculate execution times. An OAM is built from so called operator actions (OAs). These consecutive OAs are translated into the actions of an AL in the mind map input of the Crew Module.

The following two fundamental assumptions have been made in the development of the OAM for ADAMO:

- **FA1:** An action resulting from a misdiagnosis with a diverging character from the success-oriented paths is interpreted as “omission”. In this case, it is assumed that no further mitigating actions will be performed. This assumption has been made as the variety of potentially misleading follow-up OAs and their associated probabilities has been deemed as too large to be predicted and quantified.
- **FA2:** The OAM is consequence oriented, meaning that uncertainties in the behavior of operational personnel are assessed under consideration of their consequences on following OAs and ultimately on the systems’ end state.

Initiation phase: The reactor is in mid-loop operation; the emergency core cooling system (ECCS/RHR) is in “residual heat removal” function. One steam generator (SG) is full and available. The accident unfolds with the failure of a primary side pressure transducer. The next OAs are the following:

- **OA 1:** The failed transducer resets the overriding of the “pressure vessel level low” signal in the RPS, thus triggering the flooding signal in the RPS and leading the RHR to switch to reactor safety injection state. An alarm is triggered.
- **OA 2:** After a certain time, the personnel recognize the alarm signal.
- **OA 3**: An orientation phase commences during which the personnel start to collect visual information in the main control room.
- **OA 4**: The operators reflect, discuss, and interpret the information received.

The corresponding AL is shown in Figure 1 (AL 1.0), it starts with OA2 since it is assumed that the alarm is the starting point of the OAM. The last action in this AL is only necessary in the classic Crew Modul, this line triggers the call to the classic MCDET and the branching with two new states. In one new state ‘b_recFalseStim’ equals 1 (meaning that the false stimulation is recognized) and, in the second, ‘b_recFalseStim’ equals 2 (meaning that the false alarm stimulation is not recognized), the original value of ‘b_recFalseStim’ is 0 (false alarm has not yet been evaluated). The last action in this list is not necessary if the new Crew Module is used, as the new Mind Map Reader Module gathers all Crew Module variables and potential states from the conditions of all ALs and the point at which a branching occurs is specified separately in the corresponding MCDET input.

Attempt of problem-solving (PLA): Depending on whether the operators recognize the false stimulus, two different sequences are initiated. If at least one operator believes they have correctly identified the fault, it is assumed that a knowledge-based path is chosen. If this is not the case, it is assumed that the personnel will follow a rule-based procedure and consult the operating manual. The actions in the knowledge-based path fulfill the following conditions:

- They do not pertain to procedures and training provided for the personnel to deal with the task at hand.
- They belong instead to those actions and procedures in which the personnel have been trained for other tasks.

The probability of choosing a knowledge-based path has been estimated based on THERP [15] (Table 16-1). Use of written operating procedures under abnormal operating conditions). The different OAs for the two paths are listed in Table 1. It has been assumed that in case of the rule-based path, when the operators identify the false alarm stimulus, the subsequent OA is OA8, thus the OAM returns to the knowledge-based path.

Figure 4 illustrates the relationships between the different OA paths. The “end” node visible in Figure 4 corresponds to the state when the further development is assumed to no longer depend on human action but on system availabilities, like e.g., the availability of the safety valves and the pressure release valve.

5. CONCLUSION AND OUTLOOK

The MCDET Crew Module structure has been modularized increasing its maintainability and adaptability. Care has been taken to ensure that the new Crew Module can read the classic Crew Module input files and provide the same results. It has been shown how an uncertainty estimation for human-system interactions can been performed by applying the classic Crew Module or the new Crew Module and by using the three-phase-model of the human problem-solving process as starting point for the modelling process [12]. For the first time this approach has been used to model the human-system interactions for the IDPSA with MCDET of an accident scenario during mid-loop operation. The development of the OAM for this scenario has been described in detail and serves as exemplary application of the MCDET Crew Module. In addition, the tasks and the implementation consequences of the Crew Module have been presented and a comparison between the classic and the new Crew Module has been made.

Table 1. OAM for the problem-solving attempt for ADAMO

Operator Actions	
OA 5: Decision if the operator manual should be consulted or not	
Knowledge-based Path	**Rule-based Path**
OA 6: At least one operator believes to have identified the fault.	OA 7: The operations personnel are consulting the plant operating manual and systematically controls the safety objectives and the plant state.
OA 8: A decision block that determines the correctness of the diagnosis. Misdiagnosis is interpreted as “omission” based on FA1 and leads to an end failure term-	OA 9: Optional return point to knowledge-based path at OA 8.
	OA 10: The operational personnel searches for event and / or state-oriented procedures in the plant operating manual. In case no suitable procedure is found the manual is reinvestigated. Failure of recovery leads to an end failure term consequential to FA1.
	OA 11: If a suitable procedure is identified the operational personnel controls associated system requirements.

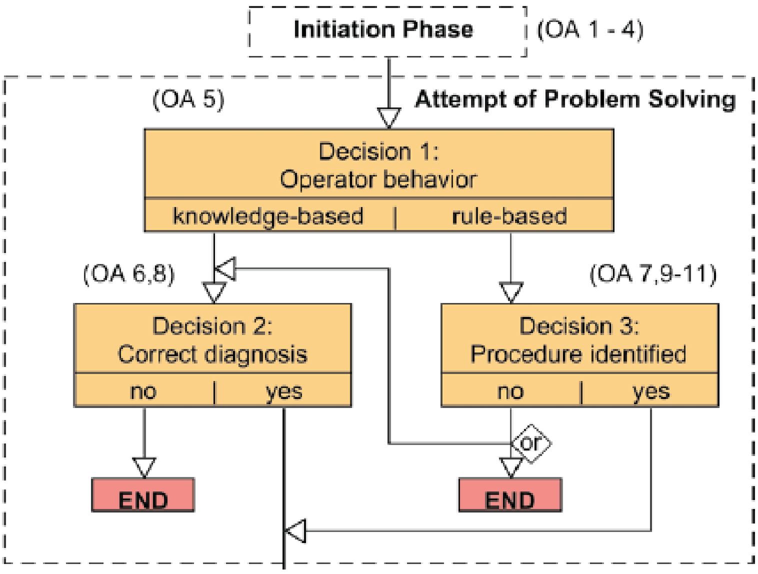

Figure 4. Presentation of the OAM for the initiation phase and the attempt of problem-solving for ADAMO based on the ENSI 2021 annual report [9]

The next steps for further enhancing the Crew Module will be built upon the new modularized Crew Module structure, some of the intended abilities have been described in Section 3.1. One of the major goals of the current MCDET advancement project is to allow conducting more than one simulation tool at a time. Once this ability is implemented, the Crew Module could indeed react to system and process states instead of being a preprocessor step. Currently, several Crew Module models need to be provided for the different system states which might emerge in the thermal hydraulic simulation run. It is therefore necessary to predict which scenarios might occur. However, in the interacting scenario one Crew Module input would be sufficient. This Crew Module input would need to contain sensitive AL conditions to allow for a rule-based handling of the different emerging system and process states. The second major working point is to optimize the GUI to better support the user in modelling the desired human-system interactions. Human-system interactions remain a key component of the safe operation of a complex facility such as a NPP and tools need to be provided to assess the uncertainty inherent in these interactions.

Acknowledgements

The enhancements of MCDET and its Crew Module presented in this paper have been performed within the research and development project UMRS1618 funded by the German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV). The application of MCDET and its Crew Module for analyzing the spurious activation of the emergency cooling during mid-loop operation has been carried out within the research and development project CTR00629 funded by the Swiss Federal Nuclear Safety Inspectorate ENSI.

References

[1] Kloos M. and Peschke J. MCDET – A Probabilistic Dynamics Method Combining Monte Carlo Simulation with the Discrete Dynamic Event Tree Approach. Nuclear Science and Engineering. 153, 137-156. 2006.

[2] Peschke J. et al. MCDET – Methode zur Integralen Deterministisch-Probabilistischen Sicherheitsanalyse, GRS-520, ISBN 978-3-947685-05-9, www.grs.de/sites/default/files/publications/grs-520_digital.pdf. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, Köln, 2018 (in German).

[3] Kloos M. et al. Main features of the tool SUSA 4.0 for uncertainty and sensitivity analysis. Proceedings of the 25th European Safety and Reliability Conference (ESREL 2015). Zürich, 2015.

[4] Kloos M. et al. Weiterentwicklung des Analysewerkzeugs SUSA, GRS-634, ISBN 978-3-949088-23-0, www.grs.de/sites/default/files/2021-12/GRS-634.pdf. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, Köln, 2021 (in German).

[5] Peschke J. et al. Analysis of Knowledge-based Behaviour in Dynamic Situations – Method and Exemplary Application, Proceedings of PSAM 2017 Topical Conference. München, 2017.

[6] FreeMind. https://freemind.sourceforge.io/wiki/index.php/Main_Page. 2024.

[7] Wielenberg A. et al. Recent improvements in the system code package AC2 2019 for the safety analysis of nuclear reactors. Nuclear Engineering and Design. 354, 110211. 2019.

[8] Babst S. et al. Sicherheitstechnische Bedeutung von Zuständen bei Nicht-Leistungsbetrieb eines DWR, Technischer Bericht, GRS-A-3114, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, Köln, 2003 (in German).

[9] Wenzel S. et al. Application of Advanced Dynamic PSA Methods for Assessing the Effectiveness of Human Actions for Accidents in Mid-Loop Operation, Erfahrungs- und Forschungsbericht 2021. Eidgenössisches Sicherheitsinspektorat (ENSI), Brugg, 2021 (in German).

[10] Mateos Canals I. et al. Application of Advanced Dynamic PSA Methods for Assessing the Effectiveness of Human Actions for Accidents in Mid-Loop Operation, Erfahrungs- und Forschungsbericht 2022. Eidgenössisches Sicherheitsinspektorat (ENSI), Brugg, 2022 (in German).

[11] Mateos Canals I. et al. Application of Advanced Dynamic PSA Methods for Assessing the Effectiveness of Human Actions for Accidents in Mid-Loop Operation, Erfahrungs- und Forschungsbericht 2023. Eidgenössisches Sicherheitsinspektorat (ENSI), Brugg, 2023 (in German).

[12] Fassmann W. Probabilistic assessment of knowledge-based behaviour by use of cognitive science and concepts developed by Swain. 11th International Probabilistic Safety Assessment and Management Conference and the Annual European Safety and Reliability Conference (PSAM11 ESREL 2012). ISBN: 978-1-62276-436-5. Curran Associates Inc. Red Hook, NY. 2012.

[13] Kuhl J. Lehrbuch der Persönlichkeitspsychologie – Motivation, Emotion und Selbststeuerung. Hogrefe, Göttingen, 2010 (in German).

[14] Swain A D. Guttmann H E. Handbook on Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications, Final Report, NUREG/CR-1278, www.nrc.gov/docs/ML0712/ML071210299.pdf. United States Nuclear Regulatory Commission (NRC), Washington, DC, August 1983.

[15] Swain A D. Accident Sequence Evaluation Program – Human Reliability Analysis Procedure, NUREG/CR-4772, https://doi.org/10.2172/6370593, U.S. Nuclear Regulatory Commission (NRC), Washington, DC, 1987.