Proposal for a Project on
Knowledge-based Decision Support for Water Treatment

Peter Struss
Technical University of Munich
struss@in.tum.de

Abstract

The proposal is related to Goal 6 of the SDGs, "Clean water and sanitation". The general goal of the proposal is promoting the establishment of facilities for water treatment, improving their scientific and technical foundations, and providing education and advice to local operators of plants, which might be non-experts. This is meant to be achieved by a web-based decision support system (DSS) that contains a repository of formal representations of treatment technologies and relevant natural processes and, based on them, an environment that supports different tasks, such as the design and operation of treatment systems.

Problem Addressed: Access to Drinking Water

In its resolution 70/1 "*Transforming our world: the 2030 Agenda for Sustainable Development*" [UN70/1, 2015], the UN general assembly committed to "*the human right to safe drinking water*" and established as part of Goal 6 "*Clean water and sanitation*": "**By 2030, achieve universal and equitable access to safe and affordable drinking water for all**". Here, "all" means 100 % of the people living on this planet.

In 2018, the general assembly emphasized in its resolution 73/226 [UN73/226, 2018] "*that water is critical for sustainable development and the eradication of poverty and hunger*", but had to note *"that the world is not on track to achieve water-related Sustainable Development Goals and targets at the global level by 2030 at the current rate of progress*".

The Sustainable Development Goals Report 2022 [SDG report, 2022] reviews the progress achieved in the 2030 Agenda. Regarding Goal 6, it reports: "*The proportion of the global population using safely managed drinking water services increased from 70 per cent in 2015 to 74 per cent in 2020. Still, 2 billion people were without such services that year, including 1.2 billion people lacking even a basic level of service. … At the current rate of progress, the world will reach 81 per cent coverage by 2030, missing the target and leaving 1.6 billion people without safely managed drinking water supplies*" (p. 38). It concludes that "*To reach universal coverage by 2030, current rates of progress would need to increase fourfold*", and that "*Achieving these targets would save 829,000 lives annually.*"

Not surprisingly, suffering from this situation is not evenly distributed over the planet. The report states that "*Eight out of 10 people who lack even basic drinking water service live in rural areas, and about half of them live in LDCs.*" (LDC: Least developed countries) – a conflict with the *Leaving No One Behind* (NLOB) action framework which declares *Equality and Non-Discrimination at the Heart of Sustainable Development* [LNOB, 2016].

As a consequence, improving the situation and speeding up the progress towards the 2030 goal has to focus on rural areas, exp. in the LDCs. Reaching the goal requires a number of actions, such as regulations and technological solutions that help to prevent pollution, improve water harvesting, reduce excessive freshwater withdrawal, water-use efficiency, and establish a nexus of water, energy, and food production. The problem is not simply access to a sufficient quantity of water, but to **safe** drinking water (or water for other purposes, such as irrigation), **facilities for water treatment** are needed. Esp. in rural areas, treatment facilities have to be distributed and run locally to avoid problems in transporting water over long distances.

An obstacle to establishing a larger number of treatment plants in places where they are most urgently needed is, besides the lack of financial resources, that in LDCs and esp. their rural areas, there may be a lack of expertise in designing, building and operating such plants. Even though there may be some standard technology available, there could be a need for adaptation to specific local conditions. Also, when facing disturbances of the plant operation, less experienced operators may need support.

In line with the LNOB policy "*Cooperate in technology transfer to promote greater equality*", our proposal is to develop an intelligent decision support system (DSS) ([Dhar-Stein, 1997], [Sanchez-Marre, 2022]). Such systems have been built for several domains, including water treatment ([Poch et al., 2012], [Mannina et al., 2019]). Our proposal aims at making technological knowledge and scientific results more accessible, improving the transfer of experience and best practices to other locations, and providing problem solving algorithms that support or automate the performance of various tasks during the life cycle of water treatment facilities.

Targeted AI Contributions

The general goal of the proposal is promoting the establishment of facilities for water treatment, improving their scientific and technical foundations, and providing education and advice to local operators of plants, which might be non-experts. This is meant to be achieved by a web-based decision support system (DSS) that contains

- a repository of formal representations of treatment technologies and relevant natural processes and, based on them,
- an environment that supports different tasks, such as the design and operation of treatment systems.

In contrast to other decision support systems in the area of drinking or waste water treatment that provide support to controlling and troubleshooting special (standard) kinds of treatment plants (activated sludge, constructed wetlands, …)(see [Poch et al., 2012], [Mannina et al., 2019]), our aim is to cover a wide range of combinations of different technologies and process steps and, in particular, to support the task of configuring solutions tailored to particular conditions and requirements, rather than taking the plant structure as given.

The Knowledge Repository

In the proposed project, we take a model-based approach ([Heller-Struss, 2002], [Wotawa et al., 2010]): expert knowledge about the water treatment domain is not represented in terms of verbal descriptions, data charts etc. but in the form of models, i.e. executable formal expressions. These models are not describing complete treatment systems, but, in a reductionist way, individual process steps in a context-free manner, stating their preconditions and inputs and their outcome, i.e. some cause-effect relation. This way, such model fragments can be assembled (automatically or manually) to form a plant model. In addition, the repository has to comprise models of the natural (physical, chemical, biological) phenomena that occur and have an impact on the performance of the systems, including ones that might disturb or prevent the proper operation of a plant. Finally, descriptions of possible human interventions (such as changing the amount of added substances) can be part of the repository.

In the project, we build on a previously developed theory and prototype ([Heller-Struss, 2002], [Roque et al., 2003]) which adopted the approach of process-oriented modeling [Forbus, 1984]. The model fragments ("**process types**") in the repository are considered to be the elementary phenomena in the domain, in particular, treatment steps and natural processes that may occur intentionally or due to abnormal conditions in the plant. A process is represented as a pair of conditions and effects, which both contain assertions about structural aspects, i.e. existing objects and their relations (such as particles of a certain kind contained in the water), and about resulting restrictions on quantities associated with the objects (e.g. the concentration of a substance is reduced to zero). Turning the informal semantics of a process, namely that the effects will be established whenever the preconditions are satisfied, into logic, a process becomes an implication:

$$\text{StructuralConditions} \wedge \text{QuantityConditions}$$
$$\Rightarrow \text{StructuralEffects} \wedge \text{QuantityEffects},$$

QuantityEffects can contain special expressions, called influences, that capture the impact of a process on the dynamics of the systems, i.e. how quantities change, but, nevertheless, are beyond the expressiveness of differential equations. In an approximate way, influences specify a partial derivative of a quantity. The actual change of a quantity can only be determined when all influences on it have been determined (which involves a closed world assumption; see [Heller, 2001].
for details).

Assembling a model of a system from instantiated process types in the repository requires that their representation uses a particular ontology, which is the second ingredient of the repository. Otherwise, expressions in effects and conditions could not be matched, e.g. to detect that one process triggers another one or that several processes affect the same quantity. This ontology has to introduce types of objects, their characterizing quantities along with the respective domains and types of relations between objects, specifying their signature in terms of object types and their properties, e.g. being symmetric.

Example

In the water treatment domain, the involved **types of objects** include

- water containers, basins etc.
- devices, such as valves, pumps, and mixing devices
- water bodies: inflow/outflow, water in containers
- ingredients of the water, like organic matter, dissolved substances, pollutants
- substances added during the process (oxidation agents, coagulants, …).

The **relations** are mainly needed to express

- connectivity of containers/water bodies
- component connections
- containment in water bodies (suspended_in, dissolved_in).

Typical **types of quantities** involved in the description of conditions and effects are

- attributes of water bodies (pH, temperature, , …)
- attributes associated with relations, mainly concentration specifying a containment relation.

As a side note, a design decision has to be taken whether to represent the water ingredients explicitly as objects and tie their concentration to the containment relation or to represent the various concentrations just as quantities associated with water bodies (refer to the challenges section of this paper).

The various kinds of process steps fulfill mainly the task of removing particular unwanted elements from the water or modifying them, usually in a sequential manner as depicted in Figure 1.

Process-oriented models of the steps have to capture the transformation of the water, relating the types of input properties with those of the output. Since conditions and effects of different processes refer to the same features of water, they can capture the treatment by the entire plant.

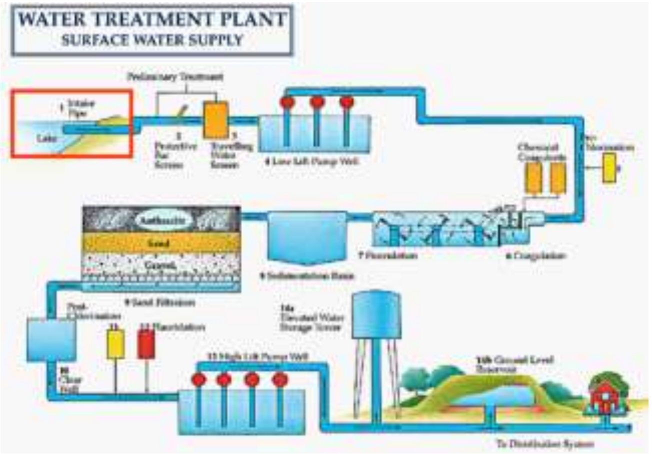

Figure 1 A typical treatment plant (Source:Drewes, Lecture Notes “Advanced Water Treatment Engineering and Reuse”, TUM 2021)

To illustrate the above, we consider the removal of colloidal particles (with a size between 0.001 mm and 0.01 mm), which can be carried out by the sequential steps of coagulation, flocculation, and sedimentation (refer to Figure 1). (Also, larger parts, up to 0,1 mm, may be treated here).

In the first step, coagulants (e.g. ferric sulfate or aluminum chloride) are added with the effect of neutralizing the charges of the particles. Thus, repelling forces between them are eliminated which enables the step of flocculation: under the influence of mixing devices (which have to be run with an appropriate speed) the discharged particles collide and aggregate to form larger and heavier flocs, which, in the sedimentation step, sink to the ground and are, thus, extracted from the flow of water.

In a simplified description of the process types, the preconditions of the **coagulation process** include the incoming water body and the contained colloidal particles with a particular concentration (zero or positive) and the added coagulant, while the effects specify (ideally) a zero concentration of (charged) colloidal particles in the outflow and uncharged particles contained with a concentration equaling the concentration of the incoming colloidal particles. Of course, all other objects contained in the water inflow will remain unaffected and simply transported to the output. Implementing this trivial, but essential feature turns out to be an instance of the infamous frame problem and is actually a challenging task, as discussed in the respective section.

The effects of the **flocculation process** include a zero concentration of discharged particles and flocs, whose concentration (qualitatively) equals the concentration of the incoming particles, with properly working mixers also in the precondition. Note, if (mis)behavior of involved devices, such as the mixers in flocculation (or their power supply and so on) are to be considered, e.g. in trouble shooting, we need to embed behavior models of components in the process-oriented modeling paradigm (again, refer to the challenges section).

A **sedimentation process** has larger particles (including e.g. clay, silt, etc., but also flocs) in its input, and the effects specify that the concentration of particles with a higher specific weight compared to water in the output will be zero, while the amount of the sediment is increased or stable (which may by modified by a removal process). Particles with a lower specific weight will just be moved from input to output. This context-independent representation of the process allows us to use it in a flexible way. For instance, in practice it is also used before the coagulation step.

This way, the repository contains elements whose combination yields an executable model. It differs from other simulation systems, because it potentially expands its structure by including process instances that are entailed by others.

It forms a firm theoretical and technical basis for various task-specific tools which support problem solving with different degree of automation, as outlined in the following sections.

Plant Design

There is a well-established set of treatment steps and a fairly standard mainly linear arrangement of these steps to form a treatment plant. Its individual treatment steps are captured as process types in the repository. In addition, there other types of treatment systems (e.g. constructed wetlands, delivering purified, but non-potable water) and more advanced technologies, such as membrane processes. For a particular area and application, designing a proper system means deriving a selection and arrangement of process steps that reflect the specific characterization of the incoming water and the operational conditions, as well as a set of requirements on the quality of the output water.

In our solution, this means finding a combination of elements from the repository that transforms the input into the output. Based on the cause-effect representation of the process types in the repository, the DSS can assist manual design by a human in offering candidate processes whose effects imply (some of) the output requirements.

When given the structure of a designed system, S, and a specification of the input and the contextual conditions (such as ambient temperature), INPUT, the DSS can create a system model MODEL(S, INPUT) as a collection of processes. Note that it has to be “causally complete” in the sense that it does not only contain the intended process steps of S, but also all processes that are triggered by them under the specified INPUT (recursively). I.e. the DSS constructs the “deductive hull” of the causal structure given the repository and, thus helps to reveal potential unwanted “side-effects”.

If the intended operation is specified by a set of requirements, GOALS, which are usually restrictions on the output water (thresholds for concentrations of substances, etc.), the DSS can check whether the designed system solves the task, i.e. the GOALS are entailed by the model:

$$\text{MODEL}(S, \text{INPUT}) \vDash \text{GOALS} \qquad (1)$$

Since the repository is considered to be complete, i.e contains all available water processing steps as well as natural phenomena relevant to the domain design proposals could, in principle, also be automatically generated by the DSS, which may be less complex than expected, because the search is focused by both INPUT and GOALS. This may generate novel solutions, which, however, may be unintuitive or violating restrictions that cannot be expressed in the repository or in GOALS (e.g. because they are related to structural aspects and not local w.r.t. individual steps). Therefore, the first case studies will aim at interactive solutions.

Trouble Shooting

We assume that a system, S, that is deployed has been properly designed, which means if all elements of the plant work as expected and the contextual conditions stay within the anticipated range, the intended effects will be accomplished, which is expressed by (1) in the previous section. Observations of the actual system performance, OBS, may indicate a deviation from the expected operation, which is detected by the DSS as a contradiction between the assumption of the nominal INPUT and the system working according to MODEL(S, INPUT) and OBS:

$$\mathrm{MODEL}(S, \mathrm{INPUT}) \wedge \mathrm{OBS} \models \bot \qquad (2)$$

For an operator, the task may then be identifying the cause behind the deviation from nominal behavior, if this is considered significant. In the DSS, this means hypothesizing

- an unanticipated INPUT_f (e.g. pH outside the expected range) that triggers unwanted or inhibits intended process steps, and/or
- a fault in the structure, S_f, (e.g. a valve being stuck, or a mixing device without power) which impairs the nominal operation.

Finding such causes, which we call **situation assessment**, can be guided by the repository by checking whether preconditions of expected processes could be invalidated or hypothesizing additional influences created by processes whose preconditions are satisfied unexpectedly (and then. perhaps, recursively searching for reasons for this).

As for design, the DSS may just be supportive to a human analyst in offering elements from the repository that might be involved in the disturbance. Alternatively, it might itself generate solutions and offer them to the operator for assessment (there will often be several potential explanations). The foundation for this are consistency-based diagnosis techniques, that were first developed for finding component faults [de Kleer-Williams, 1987] and then extended to process-oriented models ([Collins, 1993], [Heller, 2001], [Struss, 2008]). An illustrative example is presented in [Heller-Struss, 2002], [Struss 2020].

In any case, the criterion for a solution, i.e. a pair (S_f, INPUT_f) is that the hypothesized modification is consistent with the observations:

$$\mathrm{MODEL}(S_f, \mathrm{INPUT}_f) \wedge \mathrm{OBS} \not\models \bot \qquad (3)$$

which, again, can be automatically checked by the DSS.

Although this indicates the plant operates in an unexpected way, this does not necessarily imply that the GOALS cannot be achieved (The behavior could be simply unexpected, but not harmful). This can be done again by the DSS in a model-based way by checking whether the result of situation assessment (definitely or possibly) violates the GOALS:

$$\mathrm{MODEL}(S_f, \mathrm{INPUT}_f) \wedge \mathrm{GOALS} \models \bot \qquad (4)$$

or, weaker,

$$\mathrm{MODEL}(S_f, \mathrm{INPUT}_f) \wedge \neg\mathrm{GOALS} \not\models \bot \qquad (5)$$

This means fault detection can be performed by the system, esp. in cases where not all GOALS are monitored explicitly continuously.

Intervention Proposal

If a (potential) violation of requirements has been detected in the previous step (by (4) or (5), remedial actions may need to be carried out that trigger a mitigation of the negative impact and/or a re-establishment of the proper performance. Actions that can possibly be carried by an operator can be included in the repository in a smooth way be representing them as processes that have a described effect, but no preconditions other than the decision to carry them out. It turns out that determining appropriate actions is similar to situation assessment (and can use the same algorithm), but aiming at consistency with GOALS, rather than with OBS(see (6) below).

The first question to be answered is which GOALS may require corrective actions. This can be answered by the DSS as a result of the checks (4) or (5), which will not only derive an inconsistency with the entire set, but with individual requirements. This determines a starting point and focus for searching the repository.

In an interactive solution, the DSS is able to identify active processes in the model that have an impact on the deviation from a violated goal and also ways to weaken or strengthen this impact by manipulating its input. Furthermore, it can identify process types from the repository that might have effects that counteract the deviation when introduced, e.g. an oxidation process reducing the concentration of dissolved iron which exceeds a certain threshold. Usually, actions will affect quantities only via a causal chain of triggered (natural or technical) processes (e.g. the action may be opening a valve, which triggers a flow of chlorine into the tank, which starts an oxidation process, which reduces the iron concentration).

Like in design, the DSS is able to apply the criterion for a solution, i.e. a set of interventions, ACTIONS, which, when applied to S_f promises to re-establish the GOALS:

$$\mathrm{MODEL}(S_f, \mathrm{INPUT}_f \cup \mathrm{ACTIONS}) \models \mathrm{GOALS}' \qquad (6)$$

which in a way shows intervention proposal as a form of redesign.

An important remark is that, in this step, we deliberately refer to a modified set of GOALS'. This reflects the fact that if a continuous quantity has a value that violates a certain requirement, it will do so for a while. Actions usually cannot cause discontinuous changes, and, hence, cannot be consistent with the original goal, but, rather, replaced by a restriction on its derivative in order to bring the magnitude into the proper range - ultimately.

On the other hand, the non-violated goals should be maintained, such that the check (6) can reveal if proposed actions restore some goals, but have side-effects that violate others.

As for the other tasks, the DSS functions can be exploited on demand as a support to a human, but also as a completely automatic search for a solution (see [Struss 2020] for an example), which will terminate, because the repository and the set of objects is finite, unless a modeling fault allows for the unbounded creation of object instances.

Education and Training

The knowledge captured by the repository and the functions that perform reasoning on its basis can support the education of non-experts in several ways.

The simplest form of supporting education is retrieval from the repository, e.g. by searching for processes that have an impact on particular characteristics of the processed water. This is actually planned to be the first function to be realized in the project, because it provides a benefit right away and is also necessary for populating and debugging the repository.

Beyond this, by supporting a What-if analysis, the DSS would critically analyze design activities of students and trouble shooting and corrective actions in hypothetical situations by plant operators.

Explanatory Capabilities

In particular, for educational purposes, it is important to note that the DSS does not just offer a solution or deny a proposed one, but can **generate comprehensible explanations** of its results and judgements. This is due to the fact that the model has a causal structure, as opposed to, for instance, a numerical simulator that can only generate data (sequences) based on equations.

For instance, if a design is refuted due to the violation of requirements, the DSS cannot only identify the violated goals, but also display the underlying causal structure (or the lack of such a structure). If an intervention is proposed, the system can explain in what way it contributes to achieving the goals in terms of a causal chain.

Challenges for AI Research

Building the envisioned DSS comprises a number of software engineering tasks regarding a web-based, multi-lingual solution, editors and GUIs, data storage for individual applications, etc.

Beyond this, producing a useful and useable tool, raises number of issues challenging AI, some of which are instances of more general and classical AI problems, which, however, need to and can be solved in the context of the special approach followed in the project. Our work can build on previous and ongoing research and some prototypical solutions and case studies ([Roque et al., 2003], [Struss-Selvamani, 2022]). Currently, the foundation for the repository is developed in a joint project of researchers and students from the Technical University of Munich and the Vellore Institute of Technology in Chennai.

These activities have shown the principled feasibility of the approach, but also highlighted a number of limitations and problems that need to be addressed – not for the sake of academic merits, but in order to be able to deliver a tool that provides real support in practice. We discuss what we consider to be the most important ones, in the order of urgency as we assess it at this stage. Indeed, one of the first tasks of the project will be producing a pragmatic plan for tackling them, in balancing the benefit w.r.t. the project objectives, i.e. ultimately measurable progress regarding SDG Goal 6, and the feasibility of obtaining a working solution in due time.

- **Integration of Component-oriented and Process-oriented Modeling**: While the dynamics of the treatment plant can be essentially represented by the combination of certain process steps, the structure of the plant is described by a number of components, such as containers and pipes, and the performance of the processes depends on the functioning of components like valves, mixing elements, etc. Hence, we need a systematic and seamless integration of component-oriented and process-oriented modeling and diagnosis (A proposal for such an integration is presented in [Struss-Selvamani, 2022]). Such a representation is mandatory for trouble shooting, because component failures may be the root cause of a malfunction of the plant. In design and education activities, it will usually be assumed that all elements function correctly and an explicit representation of components will be dispensable (unless the response of the system under a fault is to be analyzed in order to assess its resilience).
- **The Frame Problem**: A fundamental classical AI problem is raised in our context due to very practical requirements on how to represent the process steps in the treatment, which usually involve transportation of water from input to output. Such a step transforms only certain ingredients of the water while leaving others unaffected and transporting them to the output. What we would like to express in a formal way is “the step transforms ingredients a, b, c to a’, b’, c’, and all others are transported unchanged to the output”. The problem lies in representing “all others”. Although the ontology will capture what can potentially be contained in water, listing them as being simply transported by the water flow would not only lead to large models of process steps that have to deal with many ingredients that are not relevant in a particular problem, it is not feasible, if we consider that the ontology will evolve and that, for instance, adding new substances would require to modify all process types. We need to find not a general solution to the frame problem, but a manageable one in the restricted context of our approach.
- **Boundary of a Model and the Reasoning**: Constructing the system model in a “causally forward” direction means iteratively including newly triggered process instances and their effects. For a well-defined process type repository which does not allow loops in creating new object and relation instances, this process will always terminate. Trouble shooting and intervention proposal, however, include expanding the model in a “causally backward” direction (perform abductive reasoning). The

underlying algorithm, after having found a cause, will attempt to find a cause for it, and, hence, tends to be unbounded, chaining "why?" questions as children often do. It will terminate if there exist no process types whose instances provide a causal account, and will declare the model as inconsistent. Our current solution addresses this problem by allowing some elements to be "introducibles", i.e. they do not require a causal explanation in the model. However, the problem arises how to determine the introducibles. It will usually be impossible to expect a user to define them in a comprehensive way beforehand. After all, this would require anticipating the potential causal explanations generated by the DSS. The only feasible solution appears to be an interactive one, where the user decides on the fly, whether or not something needs further causal analysis.

- **Temporal Reasoning:** The current solution supports only snap-shot-like analysis, i.e. it assumes that for a particular situation, a causal explanation can be constructed within a (qualitative) temporal snapshot. More technically, the analysis does not go backward beyond integration steps. If they are included, there could be concurrent changes in the system, and different orders of their temporal occurrences would have to be considered. The resulting complexity may render the analysis (practically) intractable. Similarly, the generated interventions are currently only collections of actions, executed in parallel, rather than in a particular order or a certain point in time.
- **Focused Reasoning**: The automatic composition and analysis of a model aims at being comprehensive and, hence, will often include aspects and causal interdependencies that are relevant for solving a particular problem; overcoming this deficiency requires mechanisms for focusing. A number of problems studied in the AI fields of reasoning about actions and time and planning need to be solved – not in principle, but in the context of the chosen model-based approach.
- **Human-Machine Interaction:** the creation of the repository and its underlying ontology requires support to non-AI users in displaying their content in a natural, comprehensible way and allowing navigation through it. Also, generating explanations of solutions or inconsistencies and deficiencies is non-trivial, because it has to avoid excessive detail and address the user's view on the problem and systems. Semi-automatic solutions that involve human decisions at certain steps do not reduce, but emphasize the problem, because the user needs to be provided with information about the internal state of the problem solving.

Project Schedule

Our proposal aims at a contribution to speeding up the establishment of clean drinking water facilities. Given that activities related to this goal are significantly behind the schedule of SDG, the project cannot be run in a way, that it works on solving research problems for a while and after several years delivers a tool (or not). It has to be run like some kind of anytime algorithm, i.e. produce first results quickly that already have a practical impact and over time deliver a sequence of tools each of which adds to the functionality of the DSS. The ultimate criterion for planning this has to be the impact on the number of people who get access to safe drinking water as early as possible.

Therefore, in a first planning phase, the project has to

- determine a focus on treatment technologies that are expected to be the easiest available and most effective ones for the targeted regions and conditions
- assess the time needed to develop the various DSS functions, distinguishing between different features, esp. concerning the degree of automation.
- produce a project schedule based on a combination of the two criteria
- define appropriate case studies that allow to assess the respective solution.

Obviously, the first tasks to be carried out are the **realization of the representation of the repository** along with editors and retrieval functions as well as creating the software engineering foundations for the web-based solution. Actually, the former has already been started in the mentioned collaboration of the Technical University of Munich and the Vellore Institute of Technology.

The result is a prerequisite for the domain experts' task of populating the repository, but also allows to use it for **education and training** purposes.

With respect to other DSS functions, reflecting the feasibility of solutions, we currently propose to continue by realizing the **design support** function in an application where the user configures a plant based on the retrieval of process steps, and exploits the DSS for checking the result (according to (1) in section 2.2). The justification for this is that this solution requires only having the system build the model in the "causally forward" direction, i.e. collecting the impact of the proposed structure, and then checking its consistency with the requirements. In contrast, letting the DSS search for a solution, involves searching in the "causally backward" direction, is more complex and will require interaction with the user.

A similar argument applies to the **trouble shooting** task: a user could generate hypotheses about causes for behavior deviations which are then checked by the system. However, retrieving reasonable hypotheses is certainly more difficult for the user than selecting water treatment steps from the repository. Therefore, the user would benefit from the system extracting more information from the model of the misbehaving system, which lets this task appear more difficult than design.

Finally, **intervention generation** could also be driven by the user exploring the impact of hypothetical actions. However, this requires the result of situation assessment and also an appropriate representation of actions in the repository.As a result, we obtain an order of the high-level tasks. Please, note that the implementation of the algorithms solving the different tasks share a significant amount of software, in particular the automatic model configuration and the consistency check.

There are at least two dimensions that guide the expansion of the achieved results:

- **Growth of the repository:** start with the commonly available and effective technologies, then for trouble shooting add disturbances and/or add more technologies, for intervention proposal add actions to the repository
- **Degree of automation:** from user driven problem solving to more autonomously generated (partial) solutions.

Summary

The work on the proposed project does not start from scratch, and some development activities have already started. However, it needs additional resources to be able to contribute have an impact on progress regarding SDG Goal 6 in reality. This holds, in particular, for the acquisition of domain expertise and opportunities for carrying out realistic case studies in order to be able to focus the work on accomplishments that are needed and effective.

The project is intended to be very focused. Regarding the application, it will first consider drinking water treatment. We anticipate that much of the principled solutions can also be applied to waste water treatment. With respect to the methods and techniques applied, the first solutions will be exclusively exploit process-oriented modeling and problem solving. In the future, other techniques may be applied, for instance case-based reasoning (e.g. for proposing an initial design), data analysis and abstraction (to feed the high-level representation used in the DSS), or numerical modeling.

While there are still problems to be solved by, we are confident that we can nevertheless produce a sequence of results that promote the establishment and improved operation of treatment facilities with increasing power.

Acknowledgments

Previous work that forms the foundation for the proposal has been supported by Bavarian-Indian Center in its Virtual Collaboration Program, the Global Incentive Funds of Technical University of Munich, and Vellore Institute of Technology.

References

[Collins, 1993] J.W. Collins. Process-based diagnosis: An approach to understanding novel failures. PhD thesis, Institute for the Learning Sciences, Northwestern University,1993.

[de Kleer-Williams, 1987] de Kleer, J., Williams, B.C.: Diagnosing multiple faults. Artificial Intelligence 31(1) (1987)

[Dhar-Stein, 1997] Dhar, V. and Stein, R. Intelligent Decision Support Methods: The Science of Knowledge Work. Prentice-Hall, 1997

[Forbus, 1984] Forbus, K.: Qualitative process theory. Artificial Intelligence 24 (1984)

[Heller, 2001] Heller, U.: Process-oriented Consistency-based Diagnosis-Theory, Implementation and Applications. Dissertation, TU München (2001)

[Heller-Struss, 2002] Heller, U.,, Struss, P.: Consistency-Based Problem Solving for Environmental Decision Support. Computer-Aided Civil and Infrastructure Engineering 17, 79-92. (2002)

[LNOB, 2016] https://digitallibrary.un.org/record/1628748.

[Mannina et al., 2019] G. Mannina, T. Ferrerira Rebouças, A. Cosenza, M. Sànchez-Marrè y K. Gibert (2019). Decision Support Systems (DSS) for Wastewater Treatment Plants – A Review of the State of the Art.

[Poch et al., 2012] M. Poch, U. Cortés, J. Comas, I. Rodríguez-Roda, M. Sànchez-Marrè. (2012). Decisions on Urban Water Systems: some Support. Servei de Publicacions, Universitat de Girona. ISBN 978-84-8458-401-8. September 2012.

[Roque et al., 2003] Roque, W., Struss, P., Salles, P., Heller, U.: Design de um sistema de suporte à decisão baseado em modelos para o tratamento de ÁGUA. In: I Workshop de tecnologia da informação apliacada ao meio ambiente - cbcomp2003, 2003, Itajaí, SC , Anais do III Congresso Brasileiro de Computação, pp. 1894-1906. (2003)

[Sanchez-Marre, 2022] M. Sànchez-Marrè. Intelligent Decision Support Systems. Springer Nature Switzerland AG, March 2022. ISBN: 978-3-030-87789-7.

[SDG report, 2022] https://ustats.un.org/sdgs/rport/2022/The-Sustainable-Development-Goals-Report-2022.pdf

[Struss, 2008] Struss, P.: Model-based Problem Solving In: van Harmelen, F., Lifschitz, V., Porter, B. (eds.). Handbook of Knowledge Representation, Elsevier, pp. 395-465. (2008)

[Struss 2020] P. Struss: Model-based Decision Support Systems - Conceptualization and General Architecture. In: Trends in Artificial Intelligence, 33rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2020, Kitakyushu, Japan, September 22-25, 2020

[Struss-Selvamani, 2022] P. Struss and R. Selvamani, Decision Support for Water Treatment – Challenging Model-based Reasoning, 35th International Workshop on Qualitative Reasoning

[UN70/1, 2015] https://www.un.org/en/dveloment/desa/population/migration/generalassembly/docs/globalcompact/A_RES_70_1_E.pdf

[UN73/226, 2018] https://documents-dds-ny.un.org/doc/UNDOC/GEN/N18/460/07/PDF/N1846007.pdf?OpenElement

[Wotawa et al., 2010] Wotawa, F., Rodriguez-Roda, I. & Comas, J.: Environmental decision support systems based on models and model-based reasoning. Environmental Engineering and Management Journal 9(2), 189–195 (February 2010).