International Environmental Modelling and Software Society (iEMSs)
2012 International Congress on Environmental Modelling and Software
Managing Resources of a Limited Planet, Sixth Biennial Meeting, Leipzig, Germany
R. Seppelt, A.A. Voinov, S. Lange, D. Bankamp (Eds.)
http://www.iemss.org/society/index.php/iemss-2012-proceedings

Ontologies, Rules, Workflow and Predictive Models: Knowledge Assets for an EDSS

D. Sottara [a], **S. Bragaglia** [a], **P. Mello**[a], **D. Pulcini** [b], **L. Luccarini** [b] **and D. Giunchi** [c]

[a] *DEIS, University of Bologna, Viale Risorgimento 2, 40136 – Bologna, Italy*
(davide.sottara2@unibo.it, stefano.bragaglia@unibo.it, paola.mello@unibo.it

[b] *ENEA – UTVALAMB–IDR, Via Martiri di Monte Sole 4, 40129 – Bologna, Italy*
(dalila.pulcini@enea.it, luca.luccarini@enea.it)

[c] *HERA S.p.A., Via Balzella 24, 47122 – Forli, Italy*
(daniele.giunchi@gruppohera.it)

Abstract Given the complexity of Waste-Water Treatment Plants (WWTPs), both from the environmental, legal and economic point of view, Environmental Decision Support Systems (E-DSSs) are getting wider adoption to monitor and manage the plants in real time. From a cognitive perspective, the knowledge required by an E-DSS may be encoded in different forms. In this paper, we argue that the operational domain and its most relevant concepts should be defined in a proper ontology, providing a vocabulary to encode inferential or operational knowledge in the form of decision-making rules. The rules process information extracted from data, acquired through sensors and possibly processed using predictive or analytic models. Eventually, the rules themselves and the actions they recommend, can be orchestrated as business processes, using workflow models. Moreover, we argue that standard formats should be used to facilitate the formalisation and exchange of knowledge between different systems, including OWL 2 (ontologies), RIF/RuleML (rules), BPMN 2.0 (workflows) and PMML (predictive models). Finally, we present a use case modelling a periodic plant monitoring routine which is necessary to check that the plant emissions are compliant with the national legislation. The system, implemented using the open source Knowledge Integration Platform *Drools*, exploits a hybrid knowledge base but relies on a unified data model and execution environment.

Keywords: Environmental Decision Support Systems; Knowledge-Base Systems; Waste-Water Treatment Plants; Hybrid Systems

1 Introduction and Related Works

For many years, the management of Waste-Water Treatment Plants (WWTPs) has been performed manually by human operators with a deep and practical knowledge of these plants. Since well-skilled and experienced personnel is not always available, great importance has been given to automation techniques in order to improve the efficiency of the processes and to reduce the operational costs.

Even if automation is the natural domain of control theory, WWTP management has also become a relevant field for researchers in Artificial Intelligence (AI). Many researchers have applied AI-based technologies, often involving Artificial Neural Networks (ANNs) and/or Fuzzy Logic (FL) [Garcia et al., 2007; Shi and Qiao, 2010], to implement "intelligent" control systems. These techniques can be applied with good results in terms of performance and robustness, especially when the relationships between inputs, system states and control

actions are not linear, as in the WWTP case. They can be used both as an alternative to model-based controllers, or combined with more traditional approaches to create hybrid control systems [Sottara, 2010].

Since the management system would effectively act as an expert operator, the choice of an "Expert System (ES)" [Riley and Giarratano, 1998] can be considered. We will focus on knowledge-based "Decision Support Systems (DSSs)" as an evolution of ES: in particular, we will consider Environmental Decision Support Systems (E-DSSs), i. e.those DSSs applied to environmental use cases and, even more specifically, WWTPs. An E-DSS operates on different types of information, at different levels of abstraction. At lower levels, reactive computations need to be executed in real time on quantitative data; moving towards higher levels of abstraction, the processed information becomes more structured and symbolic and can even be shared and exchanged with the human operators.

E-DSSs have a much broader scope than control algorithms encapsulated in control modules. An E-DSS should be able to incorporate and apply any well-established and available form of knowledge, but it should also be able to adapt and learn new pieces of information which were partially or totally unknown at the time of its creation. In general, it should be able to combine *top-down* (information-driven, imparted by teaching) and *bottom-up* (data-driven, acquired by learning) strategies. Moreover, it must be observed that the data and knowledge regarding a WWTP are often characterised by a significant amount of uncertainty: measurements, whether collected using sensors or acquired through laboratory tests, are often imprecise, uncertain or erroneous and do not allow to identify the operating conditions precisely. Likewise, management policies are often vague, do not always guarantee to reach the desired goal and are subject to exceptions. Even if the supervision of a domains expert can be leveraged, the uncertainty can be reduced, but it can rarely be eliminated.

Given such complex settings, it is almost impossible to meet all the requirements using a single technology [Sottara, 2010]. Instead, most real applications of E-DSSs (e. g. [Cortes et al., 2001]) integrate several AI, Data Mining (DM) and statistical methods such as rules, case-based reasoning, ANNs [Shi and Qiao, 2010], Bayesian Networks (BNs) and (environmental) ontologies [Ceccaroni et al., 2004]. A common practice to coordinate the components is to consider each one a service provider, part of a Service Oriented Architecture (SOA) (e. g. [Cortes et al., 2001; Sottara et al., 2009]). As an alternative, provided that the software modules are autonomous, reactive, pro-active, and "social", decentralised and asynchronous agent-based architectures can be adopted.

While the choice between a synchronous and an asynchronous interaction model is a fundamental design parameter, it affects only marginally the internal "business" logic of a module. Unless thin, granular modules are used, each one incapsulating a very specific function (like we did in [Sottara et al., 2009]), the problem of assembling knowledge-intensive modules appears again at a more nested level. To deal with this class of issues, we are developing an open source infrastructure supporting both hybrid services and agents. This framework is built on top of the popular production rule engine *Drools* [1] which, due to its rich set of functionalities, community support and friendly licensing model makes it a good candidate for both academic and commercial applications. In particular, here we will focus on the internal architecture and the design principles of hybrid, knowledge-based modules which can be assembled to provide a service implementation or an agent's behaviour, specifically providing examples and a use case based on a real WWTP.

[1] http://www.jboss.org/drools/

2 A KNOWLEDGE INTEGRATION PLATFORM

To manage a complex domain such as a WWTP, a non trivial amount of knowledge from different sources is required. To facilitate its collection, sharing and reuse, it should be acquired using appropriate techniques and tools, and expressed using standard formats. A knowledge-based framework, then, will import that knowledge and will be able to execute it against the data coming from external sources. In particular, we currently support and advocate the following knowledge assets.

Ontologies An ontology is a formal description of some applicative domain. It defines the concepts and entities involved, as well as the relations between them. It is founded on Description Logics (DL) and expressed using languages defined by the Web Ontology Language (OWL) 2 [2] standard. Ontologies range from simple vocabularies to taxonomies (e. g. [Ceccaroni et al., 2004]) to more complex forms, and can be authored using open source tools such as *Protege* [3]. In an Object Oriented (OO) and *Java*-driven setting, capturing the concepts defined in an ontology with a class model is not trivial. We have implemented a custom transformation framework which extends the approach in [Meditskos and Bassiliades, 2008]. Due to the restrictions imposed by the language, it generates interfaces rather than classes, as well as a reference implementation of those interfaces. The same interfaces, however, have been designed to work with dynamic proxies, emulating a loosely typed system with multiple inheritance, following an object--triple integration principle similar to the one presented in [Stevenson et al., 2011]. In (Bragaglia et al. [2010]), it was also shown that the ontology can be used to generate rules implementing a tableau algorithm to classify and recognise objects.

Predictive Models The term "predictive model" describes quantitative, data-driven knowledge, usually resulting from a data mining process. The category includes both predictors and classifiers, implemented using techniques such as neural networks, decision trees, clustering, support vector machines, regression models, etc. A predictive model can be trained using any DM tool (e. g. *KNIME* [4]) or algorithm, encoded using the XML-based interchange format PMML [5] and then imported into a *Drools* session, where it is converted in a set of rules equivalent to the model itself, following the approach described in [Sottara et al., 2011].

Workflows Workflows are a graphical notation for the description of Business Processes (BPs). A BP is a model of the actions required to achieve a goal, defining the interdependencies and the responsibilities for each action. At design time, workflows are abstract specifications which can be exchanged and validated between domain experts; at runtime, they can be used by a process engine to create, execute and monitor process instances involving actual data and actors. *Drools* is tightly coupled with an open source business process engine, *jBPM* [6], and supports the BPMN 2.0 standard [7], providing a compliant graphical workflow editor.

Business Rules "Business Rules (BRs)" are production rules used to capture domain-specific criteria. Unlike workflows – which focus on when and who should act – business rules deal with whether and what should be done (*consequence*) under given conditions (*premise*). They are usually authored and validated by domain experts and exported using

[2] http://www.w3.org/TR/owl2-overview/
[3] http://protege.stanford.edu
[4] http://www.knime.org/
[5] http://www.dmg.org/
[6] http://www.jboss.org/jbpm
[7] http://www.bpmn.org

standard interchange languages such as RuleML [8] or Rule Interchange Format (RIF) [9]. As a production rule engine, *Drools* supports business rules natively, either expressed in its own native language or in the form of decision tables. The rules can be applied both to raw sensorial data and inferred data; moreover, they can be be used to control external systems, closing control loops.

3 USE CASE

The platform we are working on allows to create hybrid knowledge bases, where ontologies and predictive models can be used for qualitative and quantitative inference, workflows control and orchestrate the execution of processes and business rules define policies and managements actions; in the background, additional production rules can be added to tune and extend the system as well as fixing any integration issue between the different components; finally, the engine has built-in complex event processing capabilities, so it is suitable to to build online applications.

To discuss the potentialities of such an integrated system, we introduce a simple but concrete scenario. According to the national legislation, waste water chemical parameters (e. g. Nitrogen (N) and Phosphorus (P) compound concentrations, pondus Hydrogenium (pH), etc.) must fall within some fixed limits, which may also depend on the area where the water is collected, treated and then discharged. Plant operators must ensure that the limits are respected and are subject to independent, third-party verification. To ensure that the plants are operating effectively, protecting the environment while avoiding legal controversies, several solutions can be applied: a continuous, online monitoring of the relevant parameters using “hardware” or “software” sensors is recommended, provided that the reliability of such devices can be guaranteed. Nevertheless, not all plants are adequately equipped, so an independent sampling and analysis process, performed by a certified laboratory, is planned so that a minimum number of controls is performed each year.

The frequency, number and type of controls which have to be performed is partly determined by the legal requirements and the partly by the policies of the plant manager. The procedure itself can be outlined as follows: given a plant and its layout, a number of potential collection points is defined. For each collection point, a specific set of chemical analysis has to be performed, set which depends on the plant type, size and location. For each analysis, the spot value has to be compared against the limits set by the law: any violation or missing value has to be recorded in dedicated registries for further reference. Since checks have to be carried out a minimum number of times each year, a dataset is eventually recorded for each plant which can be mined for further information.

Procedures like this, which are usually delegated to specialised technicians, can be almost completely automated, since the only part requiring a human is the collection and analysis of the samples (and even then, only assuming that the adequate probes are not available on a plant). The decision on which samples must be collected, which analysis have to be performed on each one and the interpretation of the results can be executed by an E-DSS endowed with the necessary knowledge. In the remainder of this section, we will describe our proposed representation, based on a hybrid model. In particular, we will refer to the municipal WWTP of Trebbo di Reno (Bologna, Italy), a continuous-flow 3600PE plant managed by *Hera S.p.A.*

[8] http://ruleml.org/
[9] http://www.w3.org/2005/rules/wiki/RIF_Working_Group

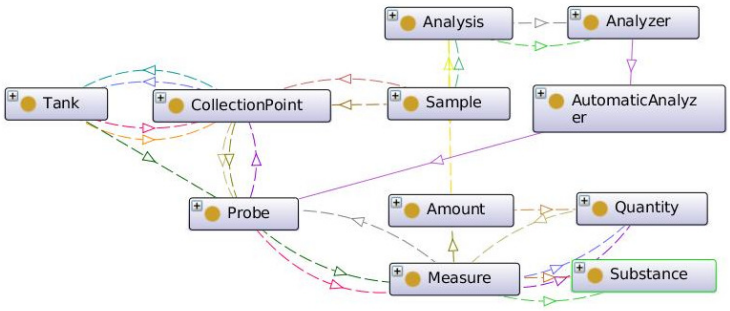

Figure 1: Use case's ontology fragment.

```
NitrificationTank = Tank ⊓ dedicatedTo some Nitrification.
NitrificationCP = CollectionPoint ⊓ locatedIn some NitrificationTank.
Tank(tank3).
CollectionPoint(cp5).
tank3 ⊏ dedicatedTo some Nitrification.
locatedIn(cp5, tank3).
```

Listing 1: DL-based collection point definition and recognition.

Use Case Modelling We have developed the core of a WWTP ontology: unlike other existing ontologies [Ceccaroni et al., 2004], it is mainly focused on the structural components, the data collection process and the control and management aspects. Figure 1 shows a fragment of the current version: Probes [10] *locatedIn* Tanks perform online Analysis on Samples, in order to acquire the Measurements of Quantities of one or more Substances. For example, pH and concentrations of Nitrates ($NO_3 - N$s) would fall under these categories.

More specifically, Tanks are *dedicatedTo* ProcessPhases such as Gritting, Nitrification or Sedimentation, which allows to provide definitions of dedicated tanks. Likewise, CollectionPoints are *locatedIn* Tanks, and so they can be classified against specific definitions. In this way, it is sufficient to provide factual and structural information regarding the layout of a plant, the processes and the position of the collection points: a semantic reasoner, then, will be able to characterise the nature of the tanks and their collection points automatically, as well as checking the consistency of the provided data. For instance, the statements of DL in Listing 1 allows to recognise *cp5* as a NitrificationCollectionPoint (located in a NitrificationTank).

One of the artefacts obtained processing the ontology is a *Java* class model reflecting the concepts, relations and individuals defined in the ontology. So, we created an semantic description of the Trebbo di Reno plant, importing the concepts from the general WWTP plant, and transformed it into an OO model. This description is static, since it only contains the structural layout of the plant, and can be safely processed offline. The resulting model, instead, can be used to create and populate fact instances from the dynamic (raw) data collected online from the plant: these object can then trigger (production) rules written using patterns derived from the same concepts in the ontology.

The knowledge on the available CollectionPoints is necessary to determine which Samples should be collected on a plant routine control and which Analysis should be performed. The list of Samples is created using rules such as the one in Listing 2. Given a plant in a non-sensitive area, one rule is required for each collection point type and plant size.

[10] Concepts, *relations* and constructs taken from the ontology are formatted accordingly.

```
rule "Determine Required Analysis - NitrificationCP"
ruleflow-group "selectAnalysis"
when
  $plant: Plant( $lines: hasLine, critical == false, $pe:
   populationEquivalent > 2000 && <= 10000 )
  $line: PlantLine( this memberOf $lines, $tanks: hasTank )
  $tank: Tank( this memberOf $tanks, $collectionPoints:
   hasCollectionPoint )
  $acp: NitrificationCollectionPoint( this memberOf
   $collectionPoints )
then
  Sample s = factory.createSampleImpl();
  s.addIsCollectedFrom($acp);
  // for each required analysis
  Analysis ax = factory.createAnalysisImpl();
  // configure Analysis
  s.addRequiredAnalysis(ax);
  // eventually add to global
  requiredSamples.add(s);
```

Listing 2: Required sample listing.

Figure 2: The process for this use case.

The actual rules are not compiled manually, but defined using Rule Templates (RTs) and Decision Tables (DTs): in fact, these rules are parametric in the *populationEquivalent* range and required `Analysis`. Condition patterns (size and collection point type) and actions (analysis to be scheduled) are determined from the values in the table cells and used to instantiate a rule template. *Drools*, through its guided editor, conveniently allows to present a table where only the relevant columns are exposed, namely two columns for the size limits, one column to select the `CollectionPoint` type and one column to include/exclude each possible `Analysis`. For example, our case study plant has 12 potential collection points, 6 of which require 1 to 5 different laboratory measurements each, including Biochemical Oxygen Demand (BOD_5), Chemical Oxygen Demand (COD), Total Suspended Solids (TSS) and N compounds concentrations.

The rules in this format are not plant-specific, so the same criteria will be applied to all plants managed by the same company. The selection of the `Samples` and `Analysis`, however, is only the first step of the monitoring procedure. We have modelled the whole process using a workflow, shown in Figure 2. The BP, triggered by an event (usually scheduled on a regular basis), requires that, after determining the required `Samples`, a human operator collects and performs the necessary `Analysis`.

The process engine, integrated with the rule engine, keeps track of the process state in a persistent way, and propagates the data between the various tasks. The manual activities – *collection* and *analysis* – are modelled as BPMN 2.0 *Human Tasks* and managed by a *Human Task* server, which will call back the process engine when each activity has been completed (or aborted for some reason), updating the process progression and the appropriate data structures. Notice that, among their other advantages, human tasks allow to trace the identity of the person who executed the task.

```
rule "(In)Validate [NO3] values"
ruleflow-group "validate"
when
  $a: Analysis( $res: results )
  $x: Measurement( this memberOf $res, ofQuantity isA Concentration,
    ofSubstance isA NO3Substance, $val: hasAmount.hasValue > 20,
   hasMU == "mg/L" )
then
  invalidMeasures.add($x);
end
```

Listing 3: Lab analysis validation.

For each resulting `Analysis`, the resulting value is eventually validated against the limits imposed by the law. As previously, we use rules (see for example Listing 3) derived from a DT where the threshold values can be conveniently set for each of the 22 legally relevant parameters. Whenever a measure is not compliant, a call will be issued to a dedicated service, managing a registry (database).

Eventually, the process terminates calculating a summary score, namely the ratios of completed (w. r. t. required) and valid (w. r. t. completed) analysis. For various reasons, it may not always be possible to perform all the necessary tests, so in practice the two indicators have values lower than 1. In order to improve the overall quality, it is possible to plugin soft sensors or estimators implemented using predictive models such as neural networks, regression models or support vector machines, i. e. any model covered by the Predictive Model Markup Language (PMML) standard. In particular, regression models would be used to estimate the missing values and classification models could be used directly to estimate their compliance, without trying to assess the exact value. While we think that this is a promising approach, at the moment we can not confirm its validity on the test plant since the dataset we have collected to this date are too small to draw any conclusion with a sufficient level of confidence.

4 Conclusions

We claim that the use of an integrated, knowledge-oriented platform is a convenient option for the development of a hybrid Environmental Decision Support System (E-DSS). From a cognitive perspective, we have shown that even standard management use cases can benefit from an appropriate modelling. A semi-declarative approach facilitates the development and increases the maintainability and robustness of an E-DSS, allowing to bring the domain experts into its development process. The use of standard languages allows to reuse and exchange knowledge both within and between organisations: in particular, it allows to import pre-existing knowledge and, when appropriate, allows to use knowledge elicitation techniques and authoring tools. Such knowledge can then easily be imported in the runtime execution environment, where the engine will apply the knowledge to the data it receives from external sources. Thus, a minor effort is required on the integration platform, saving time and resources to improve the business logic. In fact, building a Hybrid System (HS) using components of different nature raises many design and integration issues, to the point that the framework may become more complex and critical than its core components. Letting the middleware take care of this issue, instead, decouples the domain models from the runtime architecture. The platform makes it possible to map the various pieces of knowledge onto a unified data model, which can be processed by the hybrid engine.

The main limitation is that, to this date, the platform does not provide a complete coverage of the existing knowledge representation standards, so it may impose some constraints on the cognitive models it can support. Future works, then, will focus on increasing the coverage and robustness, allowing for more complex use cases. Moreover, it will be necessary to collect additional data from the plant under examination, in order to build the more sophisticated statistical and predictive models which would be useful to better estimate and diagnose its operating conditions.

ACKNOWLEDGMENTS

We wish to thank *HERA S.p.A.* for providing financial and logistic support for this work, which was also partly funded by the DEIS "DEPICT" Project of *University of Bologna*.

REFERENCES

Bragaglia, S., F. Chesani, P. Mello, and D. Sottara. A rule-based implementation of fuzzy tableau reasoning. *Semantic Web Rules*, pages 35–49, 2010.

Ceccaroni, L., U. Cortés, and M. Sanchez-Marre. OntoWEDSS: augmenting environmental decision-support systems with ontologies. *Environmental Modelling & Software*, 19(9): 785–797, 2004.

Cortes, U., M. Sanchez-Marre, R. Sangueesa, J. Comas, I. R.-Roda, M. Poch, and D. Riano. Knowledge management in environmental decision support systems. *AI Communications*, 14(1):3–12, 2001.

Garcia, C., F. Molina, E. Roca, and J. Lema. Fuzzy-based control of an anaerobic reactor treating wastewaters containing ethanol and carbohydrates. *Industrial & Engineering Chemistry Research*, 46(21):6707–6715, 2007.

Meditskos, G. and N. Bassiliades. A rule-based object-oriented OWL reasoner. *Knowledge and Data Engineering, IEEE Transactions on*, 20(3):397–410, 2008.

Riley, G. and J. Giarratano. *Expert Systems: Principles and Programming*. PWS, 1998.

Shi, X. and J. Qiao. Neural network predictive optimal control for wastewater treatment. In *Intelligent Control and Information Processing (ICICIP), 2010 International Conference on*, pages 248–252. IEEE, 2010.

Sottara, D. *Integration of symbolic and connectionist AI techniques in the development of Decision Support Systems applied to biochemical processes*. PhD thesis, University of Bologna, 2010.

Sottara, D., A. Manservisi, P. Mello, G. Colombini, and L. Luccarini. A CEP-based SOA for the management of wastewater treatment plants. In *Environmental, Energy, and Structural Monitoring Systems, 2009. EESMS 2009. IEEE Workshop on*, pages 58–65. IEEE, 2009.

Sottara, D., P. Mello, C. Sartori, and E. Fry. Enhancing a production rule engine with predictive models using PMML. In *Proceedings of the 2011 workshop on Predictive markup language modeling*, pages 39–47. ACM, 2011.

Stevenson, G., S. Dobson, A. Rosi, S. Dobson, M. Mamei, G. Stevenson, J. Ye, F. Zambonelli, J. Ye, G. Stevenson, et al. Sapphire: Generating Java runtime artefacts from OWL ontologies. In *Proceedings of the Third International Workshop on Ontology-Driven Information Systems Engineering*, volume 1, pages 1–8. ACM, 2011.