International Congress on Environmental Modelling and Software

7th International Congress on Environmental Modelling and Software - San Diego, California, USA - June 2014

Jun 17th, 3:40 PM - 5:20 PM

A hybrid, integrated IEDDS for the Management of Sequencing Batch Reactors

Davide Sottara
Arizona State University, davide.sottara@asu.edu

Stefano Bragaglia
University of Bologna, stefano.bragaglia@unibo.it

Dalila Pulcini
Politecnico di Milano, dalila.pulcini@mail.polimi.it

Paola Mello
University of Bologna, paola.mello@unibo.it

Luca Luccarini
ENEA, UTVALAMB-IDR, luca.luccarini@enea.it

International Environmental Modelling and Software Society (iEMSs)
7th Int. Congress on Env. Modelling and Software, San Diego, Ca, USA,
Daniel P. Ames, Nigel W.T. Quinn and Andrea E. Rizzoli (Eds.)
http://www.iemss.org/society/index.php/iemss-2014-proceedings

A hybrid, integrated IEDDS for the Management of Sequencing Batch Reactors

Davide Sottara [a], **Stefano Bragaglia**[b], **Dalila Pulcini**[c], **Paola Mello**[b] **and Luca Luccarini**[d]

[a] *Arizona State University, Biomedical Informatics Department, 13212 E Shea Blvd 85259 Scottsdale (AZ), (davide.sottara@asu.edu)*

[b] *DISI - Department of Computer Science and Engineering, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy, (stefano.bragaglia@unibo.it, paola.mello@unibo.it)*

[c] *Department of Civil and Environmental Engineering, Politecnico di Milano, Piazza L. da Vinci 32, Milano, Italy, (dalila.pulcini@mail.polimi.it)*

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

Abstract: A Sequencing Batch Reactor (SBR) is a particular kind of wastewater treatment plant (WWTP), where all treatment processes take place in a single reactor tank, according to a fixed temporal sequence. SBR offers several advantages in terms of reduced costs, minor impact and greater flexibility with respect to traditional WWTPs. However, an optimal cost/performance ratio can only be achieved if the treatment processes are continuously monitored and controlled. In this paper, we present a hybrid, distributed, knowledge-based (Intelligent) Environmental Decision Support System (IEDSS) specifically dedicated to the management of SBRs. The IEDSS is responsible for verifying, ensuring and enforcing the compliance of the processes with the optimal operation policies and the current regulations. The core of the IEDSS is composed by a hybrid, declarative knowledge base that encodes the knowledge and best practices for the management of the plant. It relies on OWL ontologies to describe the plant and its hardware equipment, business processes to model the plants treatment cycles, business rules to encode decision-making policies, an improved variant of Event Calculus (EC) to manage the temporal aspects and a compliance mechanism based on extended Event-Condition-Action rules (ECA rule) to monitor and check the compliance of its evaluations and decisions. The system as a whole has been implemented using open source technologies and has been tested on data coming from a pilot plant fed with real urban wastewater.

Keywords: IEDSS; Event Calculus; Business Processes; Ontologies; SBR.

1 Introduction

Wastewater treatment plants (WWTPs) exist in different configurations, but two of the most diffused ones are SBR and Conventional Activated Sludge (CAS). The processes involved in these plants are the same: denitrification, nitrification and organic substance removal, while the main difference is the plant set up. In CAS plants, the nutrient removal involves the execution of nitrification and denitrification processes, which take place at the same time, but in different tanks. A SBR, instead, is a one tank reactor which relies on an alternate sequence of nitrification and denitrification reactions. The processes take place in the same tank but at different times: the main reaction phases, the anoxic phase and the aerobic phase, are alternated cyclically as new loads of water are accumulated for treatment. The time required to complete the reactions varies with the amount of pollutants in the load, so the the duration of the phases is usually set to a fixed-time, based on a worst-case scenario. If it was possible to monitor the biological processes, the durations could be optimized, saving time, energy and increasing the overall performance of the treatment system [Luccarini et al., 2013]. Most of the existing applications are focused on detecting the completion of the process reactions, such as Luccarini et al. [2010] and Sottara et al. [2009]. However, biochemical plants such as a SBRs are complex system and their optimal management cannot be reduced this problem alone. Recently, much interest has been shown in the development of remote control infrastructures, including data acquisition systems, data storage facilities, control channels to issue commands to the actuators such as pumps and blowers, and remote user interfaces. Such infrastructures are desirable since plant management includes activities such as the diagnosis of malfunctioning or the regulation of control parameters, where the experience of an expert operator is fundamental. Dürrenmatt and Gujer [2012] investigating on the applicability of various data--driven modelling techniques to support WWTP operation, have concluded that a high degree of expert knowledge is available, which could be transferred to automated, knowledge-based systems and integrated with the remote control infrastructures. Advanced EDSSs can interact with the plant managers or even control a plant independently, without an external intervention (Rodriguez-Roda et al. [2002].

In this paper, we propose one such knowledge-based EDSS for the management and control of SBR, particularly focused on the monitoring of the state of the process. In fact, we argue that the identification of the state of the process is a necessary precondition for the application of the optimization policies. Even more importantly, the isolation of anomalous operating conditions is required to apply the emergency measures to ensure the safety of the plant and its environment.

2 Materials and Methods

2.1 Pilot plant

The experimental activity was carried out on a pilot scale SBR plant which is located side-stream to the municipal WWTP of Trebbo di Reno (Bo). The reactor, having a working volume of 500 litres, has been fed with real sewage drawn after grit removal. The plant has been equipped with a mechanical mixer, a variable-flow blower connected to a membrane diffuser, two peristaltic pumps for influent loading and effluent discharge (flow rate = 6 l/min) and a pump for sludge wastage (flow rate 1 = l/min). Besides, it has been rigged with a digital modular multiparameter system for the measurement and the on-line acquisition of pH, redox potential (ORP), dissolved oxygen (DO), temperature. All signals have been acquired in current (4 - 20 mA) by analog inputs of a multi-function data acquisition device (National Instruments 6052E), while the electrical components have been actuated by the digital outputs of the data acquisition device. A more accurate description of the experimental activity is reported in Luccarini et al. [2013].

2.2 Event Calculus

The Event Calculus (EC) is a logic formalism that was introduced for the first time by Kowalski and Sergot in 1986 and later extended in several works by Miller and Shanahan. It takes advantage of concepts like *event* and *fluent* to reason about *actions* and their *effects* on a domain. In particular, it has been exploited in a variety of domains, such as business process management [Weske, 2012] and service-oriented computing [Huhns and Singh, 2005]. The success of EC, which after more than 25 years from introduction is still subject of active research, is probably testified by its intrinsic simplicity, its ability to deal with complex problems in a modular fashion, its implicit robustness and, last but not least, its great versatility.

2.3 Event-Condition-Expectation rules

ECE rules [Bragaglia et al., 2011] are a proposal to extend ECA rules to add a richer semantics to common production rules. In *event driven architectures* and *active database systems*, ECA rules traditionally consists of three parts: (i) an **event**, a signal that triggers the activation of a rule, (ii) a **condition**, an additional logical test that constraints the execution of a rule, and (iii) a set of **actions**, a sequence of updates or commands on the local data. In addition to this, ECE rules express an ideal, desirable state that is expected after some *event* happens when a certain *condition* holds. The actual state of the world will fullfill or violate the expectation.

Table 1. Events

Event	Properties	Description
`Sample`	Signal, Amount, Filtered	Probe-sampled data
`TrendChange`	Signal, Type, Extension, Slope	Characteristic points (max, min, ...)
`EndReaction`	ReactionId	Reaction complete
`NewCycle`	CycleId	Start of treatment cycle
`NewPhase`	PhaseId	Start of process phase
`Switch`	PhaseId	Process phase to set
`Next`		Switch to next phase

2.4 Software architecture

The architecture we propose is based on an Enterprise Service Bus (ESB), where events are collected and routed to those subsystems for which they are relevant. The approach, in fact, is based on a combination of a Service-Oriented Architecture (SOA) and Event-Driven Architecture (EDA), where the interaction between the services can either be tightly coupled, loosely coupled or even decoupled as needed. In particular, we are interested in the "knowledge-base", the module which contains the knowledge and the reasoning capabilities. Its implementation is based on the open source knowledge integration platform Drools (`http://jboss.org/drools`), a flexible, object-oriented inferential engine based on a production rule system extended with support for complex event processing [Luckham, 2001], workflow management and, more recently, other types of knowledge representations [Sottara et al., 2012]. While it has already been used in some of our previous works to formalise and enact some control policies for the SBR, here we propose a new model that extends and reconciles the available expertise in a more general, robust and elegant way.

3 Results and discussion

The analysis of the data acquired and stored during the experimental activity confirmed that a relationships between pH, ORP and DO concentration trends and the evolution of the biological processes exists. Moreover, the variability of the input sewage characteristics was reflected in the acquired signals. This variability even led to several "failures" in the process. It is evident, then, that an optimal management of the plant should exploit the information coming from the indirect signals, in terms of values, trends and change points [Luccarini et al., 2010]. Moreover, the internal structure of the EDSS should be flexible and modular to deal with the various sub-tasks. At its core, the role of the SBR control system consists essentially in monitoring the state of the process and any factor influencing it, tracing its progress and ultimately trying to ensure that its outcome matches the plant operator's expectations in terms of effectiveness and efficiency. This goal can be reached by *i.* defining an adequate event and fluent "ontology" to model the state of the process and its relevant events; *ii.* incorporating a (complex) event processing system which delivers or generates the events appropriately; *iii.* using the EC formalism to infer the state of the process on the basis of the detected events; *iv.* using the ECE rule to define the desired behaviour and link the policies and actions to be executed in case of fulfillments and violations. These four steps will be discussed in greater detail in the rest of this section.

3.1 Domain "Ontology"

In order to provide a formal model of the domain, we are in the process of authoring a full (OWL) ontology which contains and defines the concepts related to the automation and control of WWTPs in general. This activity is out of the scope of this paper, so we will use a simplified version based on

Table 2. Fluents

Fluent	SubTypes	Description
`Trend`	Rising, Stable, ...	Signal trends
`Process`	Denitrification, Nitrification, ...	Chemical processes
`Phase`	Idle, Load, ...	Treatment phases
`Cycle`	UrbanCycle, ...	Cycles

the relevant concepts. In our model, all events have a timestamp and may have different "payloads" to deliver information. We generally use the following modelling pattern: if `X` is the payload, an `XEvent` is defined as an `Event that hasPayload some X` (respectively for Fluents). An outline of the relevant events and fluents is shown in table 1 and 2. In particular, events and fluents are defined at different levels of abstraction, according to the principles of Complex Event Processing. At the lowest level of abstraction, events deliver the values collected by the probes installed in the plant. The raw samples are first preprocessed (denoising, outlier filtering, error interval estimation, etc.) and then analyzed to extract more relevant events and states. When dealing with SBR, `Trend` and `TrendChange`s are particularly relevant: a trend is defined in terms of the first time derivative of the time series, and may be further qualified as `Stable`, `Rising` or `Falling`. Other trends might be defined (e.g. `Oscillating`), but are not usually part of the SBR domain. Dually, a `TrendChange` is an event marking the transition from one trend to another: in particular, we are interested in `Apex` (local minima and maxima) and `Knee` (stabilising rising/falling trends followed by a new stabilising/falling trend) since they usually allow to identify characteristic points in the process. At an intermediate level of abstraction, we use events and fluents to model the bio-chemical processes taking place within the tank, such as the cited nitrification and denitrification. On top of this, we model the cycle and its phases, using one fluent for each phase. So, the state of `IdlePhase`, `LoadPhase`, `AnoxicPhase`, `AerobicPhase`, `SettlingPhase`, `DrawPhase` and `DischargePhase` fluents determine the current phase. The commutation between two phases is marked either by the `Switch` event, between two explicit phases, or by the `Next` event, implicitly moving from the "current" phase to the "following", as defined by the canonical treatment cycle.

3.2 Complex Event Detection

Our system is completely agnostic with respect to the way the events are generated or detected. In general, dedicated modules are deployed, which can be roughly divided in two categories. Pattern-matching techniques try to map a sequence of `Sample` to an `EndReaction` event; Characteristic point techniques detect `TrendChange` events in sequences of `Sample`, which are then mapped to `EndReaction`.The former uses sub-symbolic data processing techniques (e.g. neural networks, PCA, clustering, etc…) to analyse the time series, while the latter include an intermediate step where characteristic points in the signals (correlated to the process) are first detected and then used to recognise the state of advancement of a process.

The second category can be revisited in the context of CEP and EC. For example, it is well known that the contemporary detection of a local maximum in the pH and a falling knee in the ORP during the anoxic phase is an indicator of the completion of the denitrification reaction. This logic can be modelled easily in terms of reactive EC in a form similar to the one shown in Listing 1. However, our framework supports the stronger form shown in Listing 2. This alternative states that, while the denitrification process is taking place during the anox phase, the detection of a local maximum in the pH can only happen around same time a knee is detected in the ORP. If this is the case, the end of the process is recognized, otherwise it is considered a violation.

Listing 1. EC-based recognition of the EndOfDenitrification event.

```
rule "EndOfDenitrification - EC"
 when
  $max: Max( $time : timestamp, signal == "pH" )
```

```
  $dkn: FallingKnee( signal == "ORP",
                       this overlaps[-5m,5m] $max )
  ?denitrificationHolds( $denitro; )
  ?anoxPhaseHolds( $curr; )
 then
  insert( new EndOfDenitrification( ... ) );
end
```

Listing 2. ECE rule based recognition of the EndOfDenitrification event.

```
rule "EndOfDenitrification - ECE"
 when
  $max: Max( $time : timestamp, signal == "pH" )
  ?denitrificationHolds( $denitro; )
  ?anoxPhaseHolds( $curr; )
 then
  expect $dkn: FallingKnee( signal == "ORP",
                   this overlaps[-5m,5m] $max )
  fulfillment
    insert( new EndOfDenitrification( ... ) );
end
```

3.3 State Management

The cyclic nature of the SBR plant allows to model its core functioning with the seven fluents introduced in the domain ontology, one for each phase. The current state determines univocally the plant configuration for that phase, which is obtained issuing an appropriate set of commands to the plant actuators. While production rules and Drools in particular can manage this aspect directly, we preferred to separate the state management from the control aspects. In the current implementation, the business process traces the current state of the plant and coordinates the actuations, while the fluents model the process from the perspective of the monitoring system. In particular, the monitor enforces the current state, reflected by the business process, generating the events texttSwitch and `Next`. `Switch(From,To)` assumes that the fluent associated to `From` is currently holding, *terminates* it and *initiates* the fluent associated to the phase `To`. The event `Next`, instead, is used to abstract the canonical sequence of the phases, potentially allowing to reconfigure the sequence to apply different treatment processes. Internally, `Next` is mapped into `Switch` by simple rules which could be expressed in pseudolanguage as `Next() and ?anoxPhaseHolds()` ⇒ `Switch(anoxPhase,aerobicPhase)`. The use of `Next` and `Switch` allows to decouple the act of commuting from one phase to another and supports the canonical policies. First of all, the events can be generated manually, through some user interface. More optimal policies mandate that the `Next` event is generated when an `EndReaction` event is detected while the appropriate `Phase` *holds*. In addition to these two strategies, it is possible to set a fixed, maximum duration for a phase, as shown in Listing 3. We assume that a notification event is generated upon entering a new phase, such as the anoxic phase: this event triggers an expectation for a `Switch` within a deadline, corresponding to the maximum allowed duration for that phase. In case no explicit `Switch` is detected before the deadline, the system will be forced to commute to the next phase. This representation has several advantages: first of all, it allows to express the basic policy, which is more often than not delegated to a component different from the EDSS, in the same logic framework as the more “intelligent” ones; second, it can be integrated with other policies which will usually override it; third, it will work in case none of the more optimal policies is applicable. Moreover, a commutation triggered by a violation might be used to trigger some kind of diagnostic process.

Listing 3. Maximum phase duration policy (example).

```
rule "Anox Phase Watchdog"
 when
  $init: NewPhase( $id : id ="anox" )
  anoxHolds()
 then
```

```
  expect $swt: Switch( from == "anox",
                this after[0,90m] $init )
  violation
    insert( new Next() );
end
```

3.4 Process Compliance

The rule in Listing 3 is a relevant example of a more general class of rules which can be added to the EDSS knowledge base. In fact, water treatment processes are characterised by a high degree of variability and noise, which has to be added to the measurement noise introduced by the probes and the observation noise introduced by the estimation algorithms. For this reason, the consistency of each measured or estimated quantity or states should be checked. The notion of expectation is particularly suitable in this context, since it allows to match the ideal behaviour of the system in a given circumstance, as defined by the domain experts, with what is actually observed. The fulfilment (respectively the violation) of these expectations may lead to the execution of appropriate management policies, or – in the case of violations – diagnostic and recovery procedures. In practice, a relevant amount of domain and process knowledge can be formalised in terms of expectations, and used to define (soft) constraints on the whole execution of the process. For a process to be considered ideal, several expectations are involved at different levels of abstraction: these expectations, then, can be formalised as ECE rule which use the domain events and fluents as triggers and conditions. For each of the events defined in Section 3.2, one or more expectations can be defined.

- `Sample`: All signals have expected ranges during the various process phases. The DO concentration is expected to be close to 0 during the anoxic phase, around 2 mg/l during the nitrification process and then saturate to around 6 mg/l for the rest of the aerobic phase. The pH is expected to stay within the range 6-8. Redox potential is expected to be negative during the denitrification process (roughly -200 mV), strongly negative during the anoxic phase, after the end of the denitrification process, but to become positive (around 200 mV) during the aerobic phase.

- `TrendChange`: Trend changes are expected to be present and correlated temporally over different signals. Other trend changes, or the same trend changes without an appropriate temporal correlation, are generally not expected.

- `EndReaction`: The end of denitrification is expected to be recognised only during the anoxic phase; likewise, the end of nitrification is admissible only during the aerobic phase.

- `Switch`: Some transitions are not allowed or meaningful, such as commuting to the load phase from any phase different from the idle phase (following the discharge which is expected to have emptied the tank).

4 Discussion

In this paper, we present a knowledge-based, model-driven architecture which is the natural evolution of a preexisting EDSS used to analyze the data acquired from an SBR plant [Sottara, 2010]. The main argument of this paper is that while EDSSs are excellent candidates for a knowledge-based implementation, the complexity of a domain such as a WWTP is likely to require a heterogeneous knowledge base, where structural, temporal, normative, deliberative and quantitative information has to co-exist. Since each aspect can be better captured by dedicated (and possibly standard) languages and models, we recommend their adoption. As the different modules are developed independently, the coupling is reduced while the scalability and the maintainability of the system increase. For example, it is easy to add new control policies and new data processing algorithms without affecting the rest of the system. The presence of a runtime engine which integrates the models automatically facilitates the deployment of the application. A critical role is played by the underlying domain ontology, which provides the vocabulary and the schema to represent the data and the concepts. The actual SBR ontology is part of a broader ontology, OntoPlant [Sottara et al., 2014], which is being developed to describe WWTPs, their instrumentation and control systems. The use of the ontology, together with the modularity of the system, allowed to create an extensible and scalable architecture. As more EDSSs are built to be compatible with domain ontologies which can be aligned to each other Orgun et al. [2006], it will be less difficult to share data and logic.

5 Conclusion

SBR may represent a sustainable, effective and reasonably economic solution for waste water treatment. However, the optimal cost/performance ratio can only be achieved if the processes are continuously monitored and prompt decisions are made on how to configure and operate the plant. To this end, the use of integrated, distributed, knowledge-based EDSS represents an appealing solution. An EDSS can exploit common, cheap and readily available hardware infrastructures. The use of well-known formal knowledge representation techniques allows to capture most management policies in a way that can easily be validated by the domain experts and customised as needed. The proposed framework allows to model the “worst-case”, fixed duration policies, as well as the ones based on adaptive phase durations and founded on a variety of process observation techniques. The strategies can be integrated, either individually or in ensembles, thanks to the monitoring component. This module observes, tracks and diagnoses the state of the plant, controlling which policies should be applied according to the actual operating conditions, dealing with anomalies as they occur. From a modelling perspective, the use of well known logic formalisms in place of arbitrary rules improves the declarativeness of the knowledge base, clarifying its intended semantics and preventing ambiguities. The possibility of executing all the decision and control logic in a standard runtime engine eliminates the need of a further implementation phase and reduces the deployment time. Finally, the use of standards and open source technologies facilitates the diffusion and the adoption of the system, allowing to run it both in proprietary contexts and public settings. Nevertheless, we think that the complexity of the domain may require additional degrees of expressiveness, so we are also considering the integration of techniques such as fuzzy and defeasible logics.

References

Bragaglia, S., Chesani, F., Fry, E., Mello, P., Montali, M., and Sottara, D. (2011). Event condition expectation (ece) - rules for monitoring observable systems. In Olken, F., Palmirani, M., and Sottara, D., editors, *Rule-Based Modeling and Computing on the Semantic Web*, volume 7018 of *Lecture Notes in Computer Science*, pages 267–281. Springer Berlin / Heidelberg.

Dürrenmatt, D. and Gujer, W. (2012). Data-driven modeling approaches to support wastewater treatment plant operation. *Environmental Modelling and Software*, 30:47–56.

Huhns, M. and Singh, M. (2005). Service-oriented computing: Key concepts and principles. *Internet Computing, IEEE*, 9(1):75–81.

Kowalski, R. A. and Sergot, M. (1986). A logic-based calculus of events. *New generation computing*, 4(1):67–95.

Luccarini, L., Bragadin, G., Colombini, G., Mancini, M., Mello, P., Montali, M., and Sottara, D. (2010). Formal verification of wastewater treatment processes using events detected from continuous signals by means of artificial neural networks. case study: Sbr plant. *Environmental Modelling & Software*, 25(5):648–660.

Luccarini, L., Pulcini, D., Valentini, M., and Sottara, D. (2013). Investigation of energy consumption and costs saving for a pilot scale sbr plant. In *Proceedings of Asset management for enhancing energy efficiency in water and wastewater systems*, pages 225–232. IWA Publishing.

Luckham, D. C. (2001). *The power of events: an introduction to complex event processing in distributed enterprise systems*. Addison-Wesley.

Miller, R. and Shanahan, M. (1999). The event calculus in classical logic-alternative axiomatizations. *Electronic Transactions on Artificial Intelligence*, 4.

Orgun, B., Dras, M., Nayak, A., and James, G. (2006). Approaches for semantic interoperability between domain ontologies. In *Proceedings of the Second Australasian Workshop on Advances in Ontologies - Volume 72*, AOW '06, pages 41–50, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Rodriguez-Roda, I., Sànchez-Marrè, M., Comas, J., Baeza, J., Colprim, J., Lafuente, J., Cortés, U., , and Poch, M. (2002). A hybrid supervisory system to support wwtp operation: Implementation and validation. *Water Science and Technology*, 45(4-5):289–297.

Sottara, D. (2010). *Integration of symbolic and connectionist AI techniques in the development of Decision Support Systems applied to biochemical processes*. PhD thesis, PhD School in Electronics, Computer Science and Telecommunications, University of Bologna. XXII cycle.

Sottara, D., Bragaglia, S., Luccarini, L., Mello, P., and Pulcini, D. (2012). Ontologies, rules, workflow and predictive models: Knowledge assets for an edss. In *R. Seppelt, A.A. Voinov, S. Lange, D. Bankamp (Eds.) (2012): International Environmental Modelling and Software Society (iEMSs) 2012 International Congress on Environmental Modelling and Software. Managing Resources of a Limited Planet: Pathways and Visions under Uncertainty, Sixth Biennial Meeting, Leipzig, Germany. http://www.iemss.org/society/index.php/iemss-2012-proceedings. ISBN: 978-88-9035-742-8*.

Sottara, D., Colombini, G., Luccarini, L., and Mello, P. (2009). A pool of experts to evaluate the evolution of biological processes in sbr plants. In *Proceedings of the 4th International Conference on Hybrid Artificial Intelligence Systems*, HAIS '09, pages 368–375, Berlin, Heidelberg. Springer-Verlag.

Sottara, D., Correale, J. C., Spetebrot, T., Pulcini, D., Giunchi, D., Paolucci, F., and Luccarini, L. (2014). An ontology for the wwtp instrumentation, control and automation infrastructure. In *Proceedings of 7th International Congress on Environmental Modelling and Software (iEMSs) June 15-19, 2014, San Diego, California, USA - Bold Visions for Environmental Modelling*.

Weske, M. (2012). *Business process management: concepts, languages, architectures*. Springer-Verlag Berlin Heidelberg, 2nd edition.