IPHV7I20062X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

EVENT STREAM OBFUSCATION THROUGH MULTIPLE CORRELATION STEPS

[#1]V K SUDHAKAR YADLAPALLI - M.Tech Pursuing,
[#2]B SASI KUMAR - Assistant Professor,
Department of Computer Science and Engineering,
KKR AND KSR INSTITUTE OF TECHNOLOGY AND SCIENCES (KITS) GUNTUR

Abstract: **-** Present event processing systems not have methods to protect privacy constraints of inward event streams in a chain of consequently applied stream operations. This is a problem in important distributed applications like a logistic chain where event processing operators may be widening over many security domains. This paper addresses security in multi-hop event processing networks and planned a solution to lock this gap. More precise, this paper offers an approach that allows the inheritance of access requirements, when events are interrelated to complex events. The algorithm includes the obfuscation of information, which can happen during the correlation process, and uses the obfuscation value as a decision-making basis whether inheritance is desired. This paper presents an implementation based on Bayesian Network. It enhances the obfuscation calculation and methods to boost the Bayesian Network size and measures obfuscation over more than one correlation steps.

Keywords: - *Access policy, event processing, Complex Event Processing, Obfuscation Threshold.*

I.INTRODUCTION

An event is a message that indicates the occurrence or change in an organization. Event processing is a method of tracking and analyzing streams of information about things that happen and deriving a conclusion from them. In company processes, it is required to notice inconsistencies or failures before time. For instance, in manufacturing and logistics processes, items are tracked always to sense thrashing or to redirect them during transfer. To counter this need complex event processing (CEP) systems have evolve as a key standard for business and industrial applications. The complex event processing identifies meaningful events and responds to them as quickly as possible. CEP systems permit to sense situations by performing operations on event streams which come out from sensors all over the world, e.g. from packet tracking devices. While, conventionally event processing systems have useful powerful operators in a middle way, the rising increase of event sources and event patrons have risen the necessitate to diminish the communication load by dispersed in-network processing of stream operations. In addition, the joint nature of today's wealth fallout in large scale networks, where diverse users, companies, or groups swap events. As an effect, event processing networks are mixed in terms of processing capabilities and technologies, consist of differing participants, and are broaden across several security domains. However, the rising interoperability of CEP applications raises the issue of security. It is not possible for a middle instance to cope access control for the entire network. Instead, each producer of information should be able to manage how its created data can be accessed. A corporation could limit certain information to a division of authorized users. Present effort in providing security for event-based systems covers already confidentiality of individual event streams and the authorization of network participants. In CEP systems, though, the supplier of an event looses power on the allocation of dependent event streams. This constitutes a major security difficulty, allowing an opponent to deduce information on confidential inward event streams of the CEP system. The opponent generally referred to be a person, group or force that opposes and or attacks.

Figure 1. Access Control & Event Dependency

As an instance consider the logistics practice illustrated in Figure 1 where a manufacturer needs to send an item to a target. The shipping company determines a warehouse nearer to the destination, where the item will be shipped to earlier than it will be sent to the customer. The logistic process is supported by an event processing system, where operators are hosted in the field of each party and swap over events together with potentially private information (e.g. the item's destination is sent to the shipping corporation). If now a third party receives events associated to the warehouse, it may depict conclusions about the original event data (i.e. destination), in spite of the manufacturer declaring this information as extremely secret and only given that the shipping company with access rights to it. The aim of this work is to set up access control that ensures the isolation of information even over several processing steps in a multi-domain, large scale CEP system.

In particular, the offerings are i) an access policy inheritance mechanism to impose access policies over a chain of dependent operators and ii) a scalable method to measure the obfuscation imposed by operators on information exchanged in event streams. This allows describing as piece of the access policy an obfuscation threshold to specify when the event processing systems can disregard access restrictions, therefore raising the amount of events to which application components can respond to and this way raising also the service of the CEP system. Obfuscation means making communication confusing and hiding of intended meaning in communication which makes harder to interpret.

II. BEYOND WORK CONSERVING POLICIES

With the growing attractiveness of event-driven systems, a lot of effort has been spent to make the systems safe. For instance, a role-based access control is planned inPesonen et al. and Bacon et al. talk about how publish/subscribe systems can be protected by introducing access control policies in a multi-domain construction. They explain how event communication between the domains can be supported. Opyrchal et al. present the concept of event owners that can be specified. These are used to offer access to *their* events. Tariq et al.

IPHV7I3010X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

advise an answer to present authentication and confidentiality in broker-less content-based publish/subscribe systems. This work is based on the earlier work which makes event communication protected among several entities in the system. We imagine the occurrence of a system that can handle access control on events. Based on this, we use policy composition in order to obtain the required access policies at any point during the event processing steps. Access policy composition has found a lot of concern in distributed systems. Bonatti et al. defined a well known algebra for composing access policies. Particularly in the area of web service composition, the composition of security policies plays a vital role, as several policies have to be united for every grouping of web services. We take up some of these concepts into our distributed CEP system, which allows us to inherit access limitations during the several processing steps in the operators of our system. To understand our concepts we make use of techniques from statistical inference. More precise, we calculate the Bayesian inference after creating a Bayesian network and learning the dependencies. While Bayesian inference is a complex calculation, several Monte-Carlo algorithms have been projected to estimate the inference value(s). They all have in common to arbitrarily pick samples from the Bayesian network probability distribution, and guess the values based on the samples. The accuracy of the estimated inference values is dependent on the number of samples. A commonly used technique is the Gibbs sampler.
Sampling techniques can be used to guess the conditional probabilities of the Bayesian network. However, their accuracy depends powerfully on the number of samples taken from the network, and no rough calculation scheme exists that allows drawing samples in polynomial time to attain certain accuracy. This makes the rough algorithms infeasible for safety applications, since no guarantees can be made in suitable time. On the other hand the difficulty of calculating *accurate* inference can be condensed by storing partial results of the inference calculation which otherwise would have to be calculated many number of times. Though, the advantage of these optimizations is deeply dependent on the structure of the Bayesian network.

EXISTING WORK:

A role-based access control is proposed. Pesonen et al. and Bacon et al. discuss how publish/subscribe systems can be protected by introducing access control policies in a multi-domain architecture. They explain how event communication between the domains can be supported. Opyrchal et al. present the theory of event owners that can be specified. These are used to provide access to *their* events. Tariq et al. propose a solution to provide authentication and confidentiality in broker-less content-based publish/subscribe systems. This work is based on the earlier work which makes event communication safe among diverse entities in the system. We guess the existence of a system that can grip access control on events. Based on this, we use policy composition in order to obtain the required access policies at any point during the event processing steps.

PROPOSED WORK:

This paper addresses security in multi-hop event processing networks and planned a solution to lock this gap. More precise, this paper offers an approach that allows the inheritance of access requirements, when events are interrelated to complex events. The algorithm includes the obfuscation of information, which can happen during the correlation process, and uses the obfuscation value as a decision-making basis whether inheritance is desired. This paper presents an implementation based on Bayesian Network. It enhances the obfuscation calculation and methods to boost the Bayesian Network size and measures obfuscation over more than one correlation steps.

III. NETWORK FUNCTION

We imagine a distributed correlation network, where dedicated hosts are interconnected. On these hosts we set up operators, which are executed to collaboratively sense situations and shape the distributed CEP system. The supportive behavior of the operators is modeled by a directed operator graph$G=(\Omega,S)$ which consists of operators $\omega\in\Omega$ and event streams $(\omega i,\omega j) \in S\subseteq(\Omega\times\Omega)$ directed from ωi to ωj. Thus, we name ωi the event producer and ωj the consumer of these events. Each event contains one or more event attributes which have distinct values. Every operator ω implements a correlation function $f\omega$: $I\omega\rightarrow O\omega$ that maps incoming event streams $I\omega$ to outgoing event streams $O\omega$. In particular, $f\omega$ identifies which events of its incoming streams are chosen, how event patterns are recognized (correlated) between events, and finally how events for its outgoing streams are created. The correlation function fsc is applied to events received from and created by ωm on created items in the manufacturing domain.

IV. ACCESS CONTROL FOR CEP

This approach allows inheriting access requirements by passing on them to event attributes in appearance of an *access policy*. This allows preserving requirements through any chain of dependent correlation ladder of operators in *G*. In addition, an obfuscation policy allows specifying an *obfuscation threshold* for event attributes. In each correlation footstep, the obfuscation of event attributes in created events is dogged by the planned access policy consolidation protocol. Once the obfuscation threshold is reached for an event attribute, the attribute's access requirements can be disregarded. In the following, we specify the concepts in the wake of access policies and obfuscation policies, and honor the security goal.

A. Access Policies

Access control allows specifying access rights of subjects (operators) for the set of available objects (event attributes). These access rights are provided by the owner of an object (e.g. the producer of an event stream) and may be approved to operators based on an *access requirement*. Such a requirement may be a role, a location or a domain affiliation. Requirements are usually not direct *properties* of the operators, but of the hosts where the operators are deployed. Officially, we indicate the access rights within an *access policy AP* for an operator ω as a set of (attribute, access requirement) pairs:

$$AP_{\omega} = \{(att_1, ar_1), ..., (att_n, ar_n)\} .$$

If there is no requirement specified for an attribute, any consumer in the network will be able to right to use it. Note that we think about attributes to be different even if they use the same name, but are created at two distinct operators.

An access requirement is a tuple of a property *p*, a math-ematical operator *op* and a value set *val*: $ar = (p, op, val)$, where $op \in \{=, <, >, \leq, \geq, \in\}$. *val* can be specified by a range or a set of values. For the sake of ease, in this paper access requirements are only referring to domain affiliation and have a structure like this:

$$ar_1 = (domain, \in, \{domainA, domainB\}).$$

In our paradigm scenario, the manufacturer's event attributes have diverse access requirements. While the information about the item's destination is available by the customer, information about where the item is created and when it can be picked up is confidential to the shipping company. So, the attached AP is defined as follows:

$$^{AP}manufacturer = \{(\text{destination},(\text{domain},\in,\{\text{shippingComp},\text{customer}\})),$$

IPHV7I20062X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

(pickup time, (domain,=,shippingComp)), (production place, (domain,=,shippingComp))}

With the enforcement and guarantee of access policies at each producer, a consumer will be qualified to access (receive) an attribute only if the consumer's properties match the access requirements defined for the particular attribute. In this case the consumer is trusted to use the attribute in its correlation function and adopt the requirements specified for the attribute in its own access policy for all produced events.

B. Obfuscation of Event Information

though access policies let a producer to indicate access requirements in a fine-grained way, the inheritance of requirements in a chain of successive operators is at times very limiting and can edge the efficiency and applicability of the CEP system: in each correlation step of this chain, the number of access requirements may raise by the consolidation of requirements from various producers. Each consolidation step can then increase the number of involved consumers which are prohibited from access to the event attributes of created event streams. This does not replicate the nature of event processing systems where basic events like single sensor readings may have only little influence on the outcome contained in a complex event representing a explicit situation.

In our logistics instance, f_{sc} uses *destination*, *production place* and *pickup time* to decide the expected day of delivery. As a result, the customer has no entrance to the *expected day of delivery* of the ordered item, since she does not achieve the access requirements for *production place* and *pickup time*. Yet she has a sensible interest in this information. And one may claim, that knowledge of the day of delivery does not essentially permit to illustrate a applicable conclusion on the *creation place* and *pickup time* attribute values. We say, the attribute values get *obfuscated* throughout the correlation process and depending on the achieved level of obfuscation, the access requirements of an attribute may no longer be required. In our approach, the level of obfuscation is a gauge, to which level a consumer of the produced attribute (*estimated day of delivery*) can deduce the value of the original attribute (*production place*). It can be simply seen in the instance, that obfuscation is not only dependent on the values of the attributes, but also on the knowledge of the consumer. Because the *destination* value has led to the *day of delivery* as well, knowledge of the destination would be of great help when trying to deduce the constrained attribute *production place* since the delivery time of the item is almost surely related to the distance among destination and production place. In this work, we will use *obf* (att_{old}, att_{new}, ω) to pass on to the obfuscation achieved by att_{new} for att_{old} given the knowledge obtainable at a consumer $\omega \in \Omega$.

We permit every operator to state with its access policy also an obfuscation policy. The obfuscation policy contains obfuscation thresholds for the attributes the operator produce. During the dispensation of an event attribute, its obfuscation w.r.t. each potential consumer is designed. Once, the obfuscation threshold for a consumer is reached, the event attribute can be delivered in spite of contradictory access requirements. Officially, we name the obfuscation policy *OP* for an operator ω as a set of (attribute, obfuscation threshold) pairs:

$$OP_{\omega} = \{(att_1, ot_1), ..(att_n, ot_n)\} .$$

For instance, the obfuscation policy

$$OP_{manufacturer} = \{(destination, 0.9)\}.$$

C. Security Goal

Assume $att_{old} \rightarrow_{\omega} att_{new}$ denote that

1) at some operator $\omega \in \Omega$, att_{old} is taken as input to the correlation function f_{ω} , and
2) f_{ω} produces att_{new} in dependence of att_{old}. In addition, $att_{old} \rightarrow^{*} att_{new}$ denote the transitive closure of the dependency relation. For any pair of attributes with $att_{old} \rightarrow^{*} att_{new}$ we state that att_{new} is *dependent* on att_{old}. Our main aim is to protect the privacy of event attributes over multiple correlation steps with respect to the dependency relationship among the attributes created by the CEP system. In particular, access requirements must not be practical solely to the attribute att_{old}, but have to be inherited to all dependent attributes (att_{new}) except a enough obfuscation threshold for att_{new} has been reached.

More officially, given for each attribute *att* an initial set of access requirements denoted by $AR_{init}(att)$. We require for any policy consolidation algorithm two circumstances to be met:

Circumstance 1:

For all attributes att $\in O_{\omega}$ created at ω

$$AR_{init}(att) \subset AP_{\omega} .$$

Circumstance 2:

For all attribute dependent pairs

$$(att_{old}, att_{new}) \in \rightarrow^{*}$$

1) ω_i has created att_{old} with access requirement $AR(att_{old})$ and obfuscation threshold $(att_{old}, x) \in op_{wi}$
2) att_{new} is created by ω_j
3) att_{new} is utilized by ω_k

the access requirement in $AP_{\omega j}$ yield $AR(att_{old}) \subset AP_{\omega j}$ if obf $(att_{old}, att_{new}, \omega_k) < x$.

A policy consolidation algorithm needs to ensure circumstance 1 and circumstance 2 in the presence of adversary who try to draw from event attribute

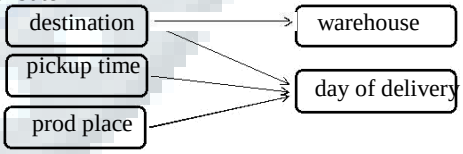

Figure: Dependency Graph of the Shipping Company Operator

values they are by policy not permissible to contact directly. We would like to keep away from that hosts unkindly or inadvertently gain information from event streams for which they have no authorization. Note, by accessing event streams according to the specified system model, hosts may still be capable to predict event attributes of illegal event streams from legally expected event streams. An adversary in our system is consequently restricted to the behavior described in the system model. The adversary is authenticated and can only access streams according to its properties. The resulting event output follows the operator requirement and the access requirements for each executed operator. Each adversary is bound to analyzing outgoing event streams which it is allowed to access, for inferring any extra information.

V. EVENT OBFUSCATION

To reach the security objective of our approach establish secure event streams among each pair of operators in *G*. For establishing secure event streams we rely on mechanisms offered in position of the art publish/subscribe systems as well as our own work. Our approach it is only important to understand that each consumer ω_c

IPHV7I3010X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

desires to request essential event attributes. The requests are handled at the producer ω_p and ω_c will need to validate itself against ω_p for the equivalent event attribute. After successful authentication ω_p will forwarded to ω_c

1) only those events matching the request of ω_c,
2) Only those events containing attributes *att* s.t.
   a) the access policy of *att* allows ω_c access to *att*,
   b) *att* has achieved a sufficiently high obfuscation,

i.e. $\forall(att_i, ot_i) \in OP_{\omega p}\ obf\ (att_i, att, \omega_c) \geq ot_i$

To this end ω_p will have to achieve on its inward streams an access policy consolidation to make sure all essential access policies can be inherited and a computation of the obfuscation values *obf* (att_i, *att*, ω_c). In the subsequent we will show the approach to access consolidation by modeling all possible dependencies among inward and outgoing event streams in an event dependency graph and compute obfuscation policies by relying on a Bayesian network.

VI. ALGORITHM

Instead of secretarial for a global Bayesian network, we suggest exploiting local knowledge accessible at each host. This allows us to decrease the number of relationships of inward.

Algorithm 1 Global Obfuscation Calculation

Procedure INITIALIZE(ω)
for all operator ω
do

$D_\omega \leftarrow$ FINDMULTIPATHOPERATORS(ω) end for
for all $\omega \in D_\omega$ do
relAtts ←FINDRELATEDATTRIBUTES for all
(att_{new} , att_{old}) $\in$ *relAtts* do

TRANSMIT P(att_{new} |att_{old}) TO ω end

for
end for
end procedure
procedure broadcast(w)
for each w UPDATE(D_w)
end procedure
procedure PONRECEIVEEVENT(e) for all
att $\in$ *e* do
if $\exists$ mulitPathDependency(att) then
CALCULATEGLOBALOBFUSCATION(ATT)
end if
end for
end procedure

The design of our approach is that a host in the CEP network creates a Global Bayesian network for each of its deployed operators. The handling (i.e. forwarding) of the event is based on the globally achieved obfuscation. This computes accepting that obfuscation is calculated over multiple correlation steps, and therefore some events may be treated more restrictive than actually desired.

A. Measuring global Obfuscation

In this approach, every host measures obfuscation for the globally known attribute dependencies (i.e. $att_{old} \rightarrow_\omega att_{new}$) in contrast to calculating the obfuscation for only locally known pair of attributes (i.e. $att_{old} \rightarrow^* att_{new}$). This has three major drawbacks: i) a larger dependency graph, ii) more communication overhead, and iii) the network is multiply connected, because there exist many paths of length n. But this approach has great benefit when compared to drawbacks i.e. each attribute calculates global obfuscation so that the event can choose the path at one to reach destination without calculating the obfuscation at each node. As a consequence, every host can create a *global dependency graph* on its own instead of creating a local dependency graph for only local dependent attributes. Furthermore, we can efficiently calculate the exact inference probabilities by applying variable elimination optimization for single connected networks to efficiently determine the obfuscation value.

Even in a local approach for obfuscation calculation the multi-path dependencies of attributes require to be measured. Attributes may reach the recipient by means of multiple paths (i.e. parallel chains of operators in a multiply-connected correlation network). An adversary that can pledge to such attributes may be able to predict the original value by combining the event information received through the multiple paths. For every attribute pair with multi-path dependencies the operators that exist in on distinct paths exchange the dependency functions w.r.t. the attributes.

B. Correctness

As our work addresses mainly how to establish producer centric access policies in CEP in a scalable way, we give only informal correctness arguments under the limitations for the adversary. Three main properties guarantee that the proposed approach is correct in terms of the defined security goal:

1) According to our assumptions, an adversary tries to predict extra information by analyzing all event streams which it is allowable to access. The proposed algorithm considers the whole knowledge the consumer *might* have. That means, it is considered that every attribute influencing the requested gocal obfuscation (*obf* (att_{old}, att_{new}, ω_c)) that is accessible to the consumer is known.
2) In accordance to Property 1, every path from att_{old} to att_{new} is considered in the algorithm. That means, every part of information an adversary may access in order to infer att_{old} is included when calculating the inference.
3) Locally unknown events (which may occur in multi-path dependency calculations) are always handled as a worst-case-consideration. We always use the value in our calculations which would give an adversary the most inference information, i.e. the value resulting in the worst obfuscation.

While all sources of event information which might control the obfuscation value of any operator are considered in our approach, the obfuscation value intended at an operator cannot further be lowered by any means. Hence, with the presented approach, we guarantee: If the consumer does not satisfy the access requirements for an attribute att_{old}, it will also not be able to access any attribute att_{new} if the attributes depend on each other ($(att_{old} \rightarrow^* att_{new})$ unless a sufficient obfuscation threshold for att_{new} has been reached. We do not guarantee, though, that the consumer will receive every attribute that has achieved a sufficient obfuscation.

VII.IMPLEMENTATION ISSUES

There are several implementation issues are used in user to manufacture communication some of the important implementation issues are

1. ***Complex Event Processing***
   CEP applies to a very broad spectrum of challenges in information systems. Like business process automation and computer systems to automate scheduling and control network based process and processing.

2. ***Manufacturer***

In this module the manufacturer, insert the product details and view the request from the shipping company. Send the details to the shipping company to delivery date and pickup time. Most commonly applied to industrial production, in which raw materials are transformed into finished goods on a large scale. Such finished goods may be used for manufacturing other, more complex products.

3. ***Shipping Company***
In this module shipping company, view product request from customer. Then company forward the request to manufacturer or reject the request. Shipping agents will usually take care of all the regular routine tasks of a shipping company quickly and efficiently.

4. ***Customer***
In this module customer is the recipient of a good, service, product, or idea, obtained from a seller, vendor, or supplier for a monetary or other valuable consideration.

VI.CONCLUSION

This paper addressed the inheritance and consolidation of access policies in heterogeneous CEP systems. We recognized a deficient of security in multi-hop event processing networks and projected a key to lock this crack. More precise, we offered an approach that allows the inheritance of access requirements, when events are correlated to complex events. Our algorithm includes the obfuscation of information, which can happen during the correlation process, and uses the obfuscation value as a decision-making basis whether inheritance is needed. We presented an implementation of our approach, based on Bayesian Network calculations.

REFERENCES

[1] A. Buchmann and B. Koldehofe, "Complex event processing," *it - Information Technology*, vol. 51:5, pp. 241–242, 2009.

[2] Access Policy Consolidation for Event Processing Systems Bj¨orn Schilling∗, Boris Koldehofe∗, Kurt Rothermel∗ and Umakishore Ramachandran†∗ *Institute for Parallel and Distributed Systems, Universit¨at Stuttgart.*

[3] A. Hinze, K. Sachs, and A. Buchmann, "Event-based applications and enabling technologies," in *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, ser. DEBS '09. New York, NY, USA: ACM, 2009, pp. 1:1–1:15.

[4] P. Pietzuch, "Hermes: A scalable event-based middleware," Ph.D. dissertation, University of Cambridge, 2004.

[5] G. Li and H.-A. Jacobsen, "Composite subscriptions in content-based publish/subscribe systems," in *Proc of the 6th Int. Middleware Conf.*, 2005, pp. 249–269.

[6] G. G. Koch, B. Koldehofe, and K. Rothermel, "Cordies: expressive event correlation in distributed systems," in *Proc. of the 4th ACM International Conference on Distributed Event-Based Systems (DEBS)*, 2010, pp. 26–37.

[7] B. Koldehofe, B. Ottenwalder,¨ K. Rothermel, and U. Ramachandran, "Moving range queries in distributed complex event processing," in *Proc. of the 6th ACM International Conference on Distributed Event-Based Systems (DEBS)*, 2012, pp. 201–212.

[8] B. Schilling, B. Koldehofe, U. Pletat, and K. Rothermel, "Distributed heterogeneous event processing: Enhancing scal-ability and interoperability of CEP in an industrial context," in
Proc. of the 4th ACM International Conference on Distributed Event-Based Systems (DEBS), 2010, pp. 150–159.

[9] B. Schilling, B. Koldehofe, and K. Rothermel, "Efficient and distributed rule placement in heavy constraint-driven event systems," in *Proc. of the 10th IEEE International Confer-ence on High Performance Computing and Communications (HPCC)*, 2011, pp. 355–364.

[10] L. I. W. Pesonen, D. M. Eyers, and J. Bacon, "Encryption-enforced access control in dynamic multi-domain publish/subscribe networks," in *Proc. of the 2007 ACM International Conference on Distributed Event-Based Systems (DEBS)*, 2007, pp. 104–115.

[11] J. Bacon, D. M. Eyers, J. Singh, and P. R. Pietzuch, "Access control in publish/subscribe systems," in *Proc. of the 2nd ACM International Conference on Distributed Event-Based Systems (DEBS)*, 2008, pp. 23–34.

[12] M. A. Tariq, B. Koldehofe, G. G. Koch, I. Khan, and K. Rothermel, "Meeting subscriber-defined QoS constraints in publish/subscribe systems," *Concurrency and Computation: Practice and Experience*, vol. 23, no. 17, pp. 2140–2153, 2011.

[13] S. Rizou, F. Durr,¨ and K. Rothermel, "Providing qos guaran-tees in large-scale operator networks," in *High Performance Computing and Communications (HPCC), 2010 12th IEEE International Conference on*, 2010, pp. 337 –345.

[14] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 2nd ed.* Prentice Hall, 2002.

[15] S. Geman and D. Geman, "Stochastic relaxation, gibbs dis-tributions, and the bayesian restoration of images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-6, pp. 721 –741, 1984.

[16] A. E. Gelfand and A. F. M. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 398–409, 1990.