39

Hierarchical Knowledge and Meta-Observations

Reinhard Riedl
Department of Information Technology, University of Zurich

Abstract: We present a model machinery for the generation of aggregating views of knowledge, the coupling of localized knowledge, and for carrying out meta-observations, which record the dynamics of a knowledge generating system. The machinery is based on the only assumption, that we can monitor events in a system. Thus it applies to all e-business applications with client/server computing.

1. THE COGNITIVE PERSPECTIVE

E-business often requires the exchange of textual information between partners in differing situations and with a different social, educational, or cultural background. Information is exchanged then as text data, whose interpretation depends on the problem context of the producer, and of the receiver, respectively, as well as on their respective ontologies. Thus, if context and/or ontology differ, the exchange of information is likely to fail. This fudamental problem, is often described as the problem of meaning and relevance. Seen from the perspective of cognitive science, one may understand meaning as a pointer, which affiliates an affordance in an ecological niche with an icon, which is usually a data object, e.g. a word in a text. On the one hand, pointers may be interally valued by a numerical inference value, which desribes the relevance with respect to affordances. On the other hand, pointers as a whole may be externally valued by relevance number which describes the relevance of the icon (embedded in a larger information object) for the receiver of information.

Inference network models with two layers may be used to subsumize and model these pointers, and affinity matrices can be used to represent the relationship between affordances and icons. Inherence network models have been used before in information retrieval and information filtering [1], while affinity matrices have been used for data-affinity based load balancing in high performance distributed computing. In both cases, we have formal representations at hand, which can be generalized to similar structures. A survey of these formal modeling approaches can be found in [11].

The relevance of icons can equivalently be modeled with inference networks consisting of two layers, or with affinity matrices, respectively. The canonical composition of inference network models, and the canonical matrix multiplication, respectively, may thus be used to jointly model meaning and relevance. Thus any larger textual information object can be modeled straightforwardly, as long as its components do not interfere with each other. The latter is for example true for digitalized personal documents in inter-organisational e-government [12]. We shall elaborate on these representations below.

In our picture, knowledge implies the ability to identify the pointers and to learn from experience how to improve that identification. The identification is a process rather than a single activity, which may be performed in a moment or in a series of steps. Knowledge management technology can facilitate faster or more accurate identification by

- supporting the human identification process
- automating the process and providing computed results to the human user or
- collecting and presenting experience, which can be used to construct or update automatic identification algorithms or human knowledge on how to perform the identifications.

In this paper, we shall primarily focus on the third task, and we shall explain how monitoring in client/server computing can be exploited to come up with formal knowledge representations.

2. THE ENGINEERING PERSPECTIVE

Recently, there has been started considerable work on higher order knowledge mining, compare for example the approach in [13] based on clausal form logic [6]. While most of these approaches are founded in database engineering, our approach stems from distributed systems engineering.

Distributed systems[59] are message-oriented and adhere to the service paradigm. This is both true for high performance computing and for information sharing systems, and it even applies in parts and to some extent for distributed mainframe environments. There are two basic communication paradigms implemented, namely request/reply interaction and messaging based on some queueing service. The Middle-ware is responsible for the execution of communication and in doing this, it usually supplies additional support for distributed computing, such as security services, transaction semantics, and computing context (e.g. virtual synchrony).

[59] equivalently, we may speak of client/server systems

Events and event models play an essential role in Middle-ware managed systems. For example, next to the naming service, the event service is a primary service in CORBA or CORBA-like systems. Among others, Middle-ware provides us event tracing mechanisms, and with basic functionality to evaluate these recordings and to apply them, e.g. for non-repudiation services. While most of this add-on functionality primarily generates order relations, it is also possible to put services on top of it, which supply us with inference relations, e.g. for data-affinity based load balancing in distributed computing environments.

As these capabilities are creating demand, considerable work is being done on the extraction of knowledge from these event tracings. The reasons for doing that are quite diverse, ranging from meta-benchmarking approaches over performance tuning and information system re-engineering to actual knowledge mining. The natural next step of this work is to develop tools for engineering of the knowledge extracted and for the monitoring of the observation extraction process itself.

Part of this monitoring is the creation of meta-observation on the internal structure of knowledge bases and knowledge representations, which constitutes some particular form of higher-order knowledge. In this paper, we present a modeling machinery, which merges and generalizes some of the knowledge models applied for actual engineering tasks in distributed systems.

3. THE MODEL BASELINE

First, we present the basic framework for our approach to knowledge modeling. The standing thesis in behind is, that knowledge can be generated from the clustering of observations of well identified and classified events. Thus knowledge grounding is performed by event classification. This defines clear restrictions on the scope of applicability of our theoretic framework, or rather, it identifies the place of our 'tool-box' in the chain of knowledge generation.

The input to our model are the observations of classified events, i.e. events[60] with well defined, observable a priori conditions, and well defined a posteriori conditions. The output of our model, or, more precisely, the output of a machine implementing (part of) the machinery defined by the model, are views of the knowledge represented by the classified observations, couplings of different, localized knowledge representations, and meta-observation reports.

Both views and meta-observation reports represent higher-order knowledge to a different extent: Views plainly describe aggregated knowledge, while meta-observation reports inform us about the internal structure of observations, e.g. their dynamics. Moreover, our model framework also supports the coupling of localized knowledge, represented by views. This includes the coupling of automatically

[60] The term event does not imply that it has no extension in time.

generated knowledge[61], i.e. grounding categorization, with human expert knowledge.

We shall proceed as follows: We first define the basic model and we introduce two representations of it, one in terms of hierarchical graphs and the other in terms of numerical matrices. The representation with hierarchical graphs enables us to define skew views of the knowledge, where part of the knowledge is aggregated and another part is not.

The representation with matrices enables us to give numerical pictures of these graphs as well as of their aggregated views. Finally, we explain how local knowledge may be combined technically with other knowledge. Combinations are worked out on the the level of representations only, in order to avoid a mixing up of the physical observations and their homomorphic (virtual) images in our model.

3.1 Events, a priori observables, a posteriori observables, and profiles

In our approach, knowledge is a localized model of the correlation between a priori observables and posteriori observables, which is formally represented by inference matrices, as they appear in inference network models ([5]), where they are also sometimes called link matrices. This model is drawn from the observation of events and the structuring of these events. Each event is associated the same a priori observables and the same a posteriori observables. In the following, the event set will be denoted by E. The set of a priori observations is denoted by O- and the set a priori observable states is denoted by S-. The set of a posteriori observations is denoted by O+ and the set a posteriori observable states is denoted by S+. Formally, observables are mappings between the event set and the set of states. We omit a detailed formal introduction and we refer to [11]. For the two sets of states, we introduce the concept of profiles. Profiles are sets of states. We requite that the set of a priori profiles, and the set of a posteriori profiles are atomic lattices (whose order structure is defined by set inclusion). As usual, a subset of disjunct profiles is called a partitioning (with respect to a given lattice of profiles). Partitionings define new observables, when the original observables are composed with a mapping representing the set inclusion of the original states into the new states. In the following, we shall assume that the lattices of profiles consists of a (finite) sequence of refined partitionings.

Definition 1: We define knowledge as the quintuple (**E,** O-, O+, T, P), where E denotes the set of classified observations or observed, classified events, O- and O+ denote the observables, and T and P denote lattices of profiles.

[61] where automatic generation is based on available classification

3.2 Graph representations

Knowledge as defined above can be represented as a VH-graph ([7]). We shall not bother the reader with explicit definitions, but rather we shall explain the concept to be applied. We may represent states as nodes in a bipartite base-graph, and we may represent events as edges. VH-graphs have in addtion a hierachical, i.e. tree-like, structure, where the nodes of the base-graph are embedded into the hierachy. Cuts are non-comparable sets of elements of the hierachy, and views corresponding to a cut collapse all structure of the base-graph "below" the cut. Thus, parts of the base-graph may be aggregated, while other parts may be shown in full detail. That is, we may zoom into the system at critical parts and still work with a moderately complex view, when we aggregate other parts.

Due to our assumptions above, the union of the both lattices of profiles provides us with an appropriate hierachy, and thus we can create hierachical views of the knowledge quintuple defined above. Please note that if we allow various edges between the same pair of nodes, the VH-graphs are in one-to-one correspondence with the knowledge qintuples. Further, we can merge two knowledge quintuples represented as bipartite graphs, by applying the construction principle for the composition of relations. This results in a representation for the composed knowledge as a new bipartite graph, whose nodes consist of the a priori states of the first quintuple and of the a posteriori states of the second quintuple, while for each pair of edges ((u,v), (x,y)), v=x, (u,v) an event in the first quintuple, and (x,y) and event in the second qintuple,there is an edge (u, y) in the new bipartite graph.

The graph representation reveals that our approach to formal knowledge representations is in the spirit of rule-based systems. This defines some constraints on its applicability, but on the other hand it links event monitoring directly with an autmated generation of rule sets, which reflects what is actually happening in a client/server system. The main advantage of graph representations is that they allow a visualisation of rules, where hierachical views provide an opportunity to focus attention on selected parts of the system and to aggregate others.

We conclude with

Definition 2: Given a knowledge quintuple (**E,** O-, O+, T, P), the affiliated hierachical knowledge is the union of all knowledge quintuples, which can be constructed from it by applying the technique of generating views.

By abuse of terminology, hierachical knoewledge is a set homomorphic images of the original knowledge, whose size is determined by the profile latices.

3.3 Matrix representations

Assume again that we have given a knowledge quintuple (**E,** O-, O+, T, P) as above. We now introduce the footprint matrix representation and the reference matrix representation of knowledge according to the following scheme

- lines correspond to states of a priori observables
- columns correspond to states of a posteriori observables
- entries correspond to events, where integer counting leads to reference matrices and Boolean counting leads to footprint matrices
- there is no semantic meaning in the enumeration of lines and columns
- states may be replaced by sets of states, namely profiles

Again we omit details. The scheme sketched enables us to provide matrix representation for knowledge quintuples, as well as for the affiliated hierachical knowledge. Note that forming a homomorphic image is done by addition of lines and/or columns, and that the combination of knowledge corresponds to matrix multiplication.

If we reverse the order of relations and if we normalize entries such that we get a probabbility matrix, then we obtain the inference values for an inference network model consisting of two layers. Again, combination is done by matrix multiplication.

3.4 Some Remarks

In many application scenarios, one type of view is exactly what one is interested in. However, if we are interested in the internal structure of a set of observations it is exactly the diversity of view, which enables us to 'measure' this structure [9].

There is some duality between a priori observables and a posteriori observables. They are essentially a view position. Exchanging them changes corresponds to a matrix transposition. Ignoring normalization issues, the change of view corresponds to the change between reference matrices and inference matrices.

Our framework essentially stems from the modeling of decision problems with decision tasks, a priori observables, and a posteriori observables. Decisions are made upon the observation of a priori observables and the quality of the decisions is a function of the a posteriori observables. Therefore, it is critical for good solutions to understand the correlation of a priori and a posteriori observables. (Cf. [10]). There are various different interpretations possible. Our basic understanding is, that any matrix entry specifies something like a rule linking the a priori state and the posteriori state. This rule may be understood as an inference rule, or some conditional probability.

4. IMPLEMENTATION TECHNIQUE

Next, we discuss basic formal techniques for the implementation of knowledge generators relying on our model framework.

4.1 Generation of hierarchies

Hierarchies of a priori profiles arise from the clustering of natural partitionings. Above we have defined a machinery to represent knowledge with respect to such hierarchies. However, which hierarchies are we to choose?

This depends on the problem context, of course. The general strategy will be to find satisfactory solutions for the following problem: Find a hierarchy of profiles such that each a priori profile is as homogeneous as possible with respect to the posteriori observables of their class members (because otherwise the inference rule associated had little relevance), that any two a priori profiles are as distinct as possible with respect to their a posteriori observables (because otherwise the associated inference rules would have little meaning), and that all profiles are neither too small (as then they would lack statistical relevance) nor too large (because we are interested knowledge as fine-grained as possible). Note that analogously, we can also ask for a posteriori profiles fulfilling the same requirements for a priori observables.

There are various different possibilities to formalize these requirements, and there are various methods to obtain the desired hierarchies: classical clustering algorithms based on a distance function on the set of a posteriori observables (in case of a priori profiles) or on the set of a priori observables (in case of a posteriori profiles), neural network clustering, algorithms choosing a configuration whose matrix representation is as close as possible to a block diagonal matrix, or which optimizes some independent reference model (cf [2]), and some further more. A discussion of these issues is beyond the scope of this paper.

4.2 Meta-observations

Meta-observations concern the event system as a whole. We use them to judge on the relevance of observations. There are various possibilities for analyzing observations

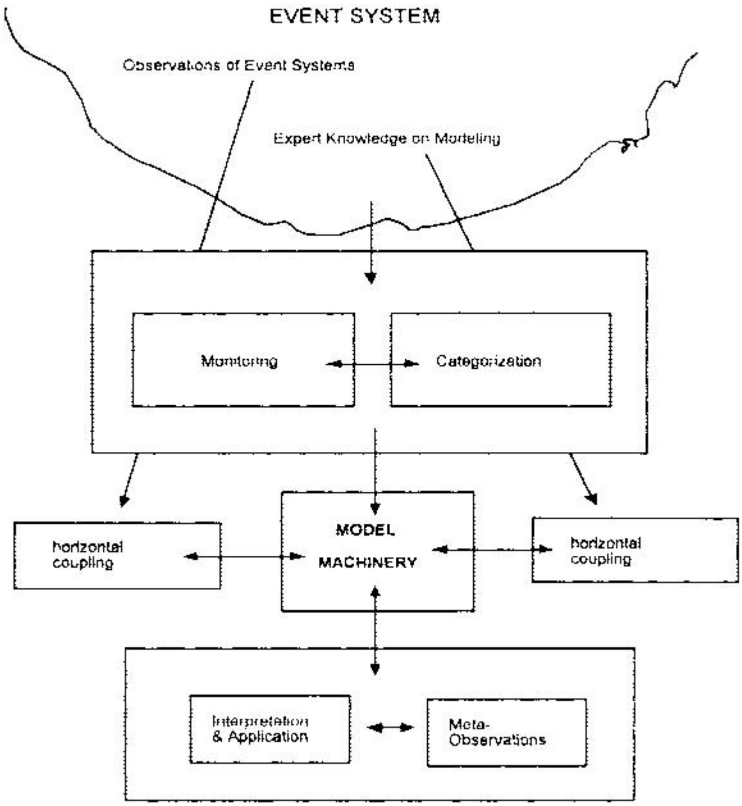
- the intrinsic algebraic structure as it is addressed by the optimization problems indicated above
- the intrinsic algebraic structure as it is represented by semigroup role structure (cf[9])
- the comparison with natural languages measuring the Heaps and the Zipf exponent (cp [11]).
- One example of the last is the footprint growth function, which stems from Heap's law:

Definition 3: The relevance of the observations are described by the growth of the footprint of the (ordered) events with respect to the a priori and a posteriori observables.

Heaps observed that in natural language texts this functions grows like a polynomial with exponent one half, i.e. similar as the supremum function of one-dimensional Brownian motion.

4.3 Architecture

Finally we depict the architecture for knowledge mining, where the model machinery discussed is situated in.



The architecture consists of four levels: the system, where measures are taken (called event system above), the layer with monitoring and primary categorization tools feeding our model framework with data, the implementation of our model framework, which makes use of automatic and expert categorization, and the layer with tools for interpretation and application of the hierarchical views and the couplings of different knowledge representations. There, the meta-observation tools are situated as well, since they are needed for the relevance ranking of knowledge representations.

4.4 Applications

There is a wide range of applications for our machinery in e-business:

- customer profiling in market research
- psychological analyses of user behaviour in virtual venues or communities
- design of information agents supporting users of large Intranets, design of e-brokers for customer-supplier matching in virtual markets, or personalied guidance of users through administration processes in e-government
- optimization of native code generators in Bytecode compilation
- optimization of load distribution in high performance transaction processing
- sociological analyses of information societies

In all these cases, we can monitor natural events, such as client requests, data object accesses, etc. and hierachical knowledge representation plays an important role for the design and the implementation of applications. Typically, a priori profiles describe the identity of an active person or object, while a posteriori profiles desribe what is really done, or which ressources are really requested. Indeed, applications im e-business with monitoring of client requests range from psychology and sociology to performance management.

We have performed meta-observations for various different scenarios, such as DB/DC transaction processing, accesses to web-pages in web-sites, and requests for search-engines. This revealed that Heaps and Zipf's Law are more or less valid in all scenarios, but the coefficients depend on the actual scenario. In some scenarios they clearly reflect the dynamics of the system, while in other scenarios they may also represent static patterns of access distributions.

5. CONCLUSIONS

We have presented a formal machinery for knowledge engineering. Due to its genericity it applies to a wide range of knowledge mining scenarios in distributed computing and distributed information systems. Part of this machinery has been implemented in joint research projects with industry. While the workflow for knowledge extraction is well understood in parts, so far little is understood of the internal structure of event systems.

6. REFERENCES

[1] Belkin, N.J., and Croft, W.C., Information Filtering and Information Retrieval: Two sides of the Same Coin, Comm. of the ACM, Vol 35, No 2, 1992;

[2] E. Born, T. Delica, W. Ehrl, L. Richter, R. Riedl, Characterization of Workloads for Distributed DB/DC-Processing, J. on Information Science, 1997;

[3] J. Heaps, Information Retrieval - Computational and Theoretical Aspects, Academic Press 1978

[4] C Lueg and R. Riedl, How Information Technology Could Benefit from Modern Approaches to Knowledge Management, Proceedings of the 3rd Conference on Practical Applications of Knowledge Management - PAKM 2000, Basel 2000

[5] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1998

[6] T. Richards: Clausal Form Logic: An Introduction to the Logic of Computer Reasoning, Addison Wesley, 1989

[7] M. W. Richter, VH-Graphs - A New Approach to Hierarchical Graphs and their Application to Object-oriented Programming, Thesis, Univ. of Zurich, 1998

[8] R. Riedl: Agents for Customer Support in Electronic Commerce: Agents as Performing Actors, ACM SIGUCCS Conference 1999, Denver, Colorado 1999;

[9] R. Riedl: Usage of Trace Data for the Deduction of Role Structures and the Comparison of Knowledge Societies, 3rd Int. Conf. on Cognition Technology, San Francisco 1999

[10] R. Riedl: The Impact of Workload on Simulation Results for Distributed Transaction Processing, Proc. HPCN Europe '99, Amsterdam 1999;

[11] R. Riedl: Need for Database Trace Benchmarks, in R. Eigenmann, Performance Evaluation with Realistic applications, MIT-Press, 2001

[12] R. Riedl, Document-based Interorganisational Information Exchange , accepted for Proceedings of SIGDOC 2001, Santa Fe 2001

[13] H. Spiliopoulou, J. F. Roddick: Higher Order Mining: Modelling and Mining the Results of Knowledge Discovery. In Proc. Second International Conference on Data Mining Methods and Databases, Cambridge, UK, WIT Press, 2000