World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:4, No:3, 2010

Understanding and Designing Situation-Aware Mobile and Ubiquitous Computing Systems

An Interdisciplinary Analysis on the Recognition of Situations with Uncertain Data Using Situation Templates

Kai Häussermann, Christoph Hubig, Paul Levi, Frank Leymann,
Oliver Siemoneit, Matthias Wieland, Oliver Zweigle

***Abstract*—**Using spatial models as a shared common basis of information about the environment for different kinds of context-aware systems has been a heavily researched topic in the last years. Thereby the research focused on how to create, to update, and to merge spatial models so as to enable highly dynamic, consistent and coherent spatial models at large scale. In this paper however, we want to concentrate on how context-aware applications could use this information so as to adapt their behavior according to the situation they are in. The main idea is to provide the spatial model infrastructure with a situation recognition component based on generic situation templates. A situation template is – as part of a much larger situation template library – an abstract, machine-readable description of a certain basic situation type, which could be used by different applications to evaluate their situation. In this paper, different theoretical and practical issues – technical, ethical and philosophical ones – are discussed important for understanding and developing situation dependent systems based on situation templates. A basic system design is presented which allows for the reasoning with uncertain data using an improved version of a learning algorithm for the automatic adaption of situation templates. Finally, for supporting the development of adaptive applications, we present a new situation-aware adaptation concept based on workflows.

***Keywords*—**context-awareness, ethics, facilitation of system use through workflows, situation recognition and learning based on situation templates and situation ontology's, theory of situation-aware systems.

I. Introduction

CONTEXT-aware applications require information about their environment and must adapt their behavior according to this information so as to be really adaptive or smart. Since acquiring, modeling and managing context information is a tedious and expensive task, it is beneficial to share this information between different kinds of application. So-called *spatial models* are intended to represent or mirror certain parts/aspects of the real world as closely as possible thereby serving as a shared, common basis for different context-aware applications and systems. Creating, managing, updating, and merging of spatial models has been a heavily researched topic in the last years aiming at providing highly dynamic, consistent and coherent spatial models at large scale, a world wide space in analogy to the WWW (see e.g. [1], [2], [3], [4], [5]). In this paper however, we want to concentrate only on how context-aware applications could easily use this information to adapt their behavior according to the situation they are in. Or to be more precise: 1) How to facilitate/improve the basic process of situation recognition within spatial model infrastructures and 2) how to allow for a safe situation reasoning with uncertain data gained from the spatial model infrastructure.

Concerning 1) the basic idea is to provide the spatial model infrastructure with a situation recognition component based on generic *situation templates*. Thereby, a situation template is an abstract, machine-readable description of a certain basic *situation type* specified in XML, which comprises a) a description of the context information considered as being relevant for the situation type in question and b) a description of how to calculate/infer from these values the existence of a concrete situation. Situation templates holding a description of a basic situation type – as part of a much larger database of other situation templates – could then be easily reused by different kinds of applications to evaluate quickly the situation / the situations they are currently in and to adapt their behavior accordingly.

Concerning 2) since the underlying spatial model infrastructure is distributed itself, the recognition system should run in a distributed manner either. Furthermore, quality information of processed context data (if available) should be incorporated in the reasoning process thus allowing for conclusions on the overall *quality of the situation recognition* and the probability of the existence of a concrete situation. Further improvements can be achieved by providing the recognition component with a *learning algorithm to adjust situation templates continuously* leading to an overall system, which, as a whole, could be considered as viable and sound.

Kai Häussermann, Oliver Zweigle, and Paul Levi are with the Institute of Parallel and Distributed Systems, Department of Image Understanding, University of Stuttgart, Germany, {firstname.lastname}@informatik.uni-stuttgart.de

Matthias Wieland and Frank Leymann are with the Institute of Architecture of Application Systems, University of Stuttgart, Germany, {firstname.lastname}@informatik.uni-stuttgart.de

Oliver Siemoneit and Christoph Hubig are with the Institute of Philosophy, University of Stuttgart, Germany, {firstname.lastname}@philo.uni-stuttgart.de

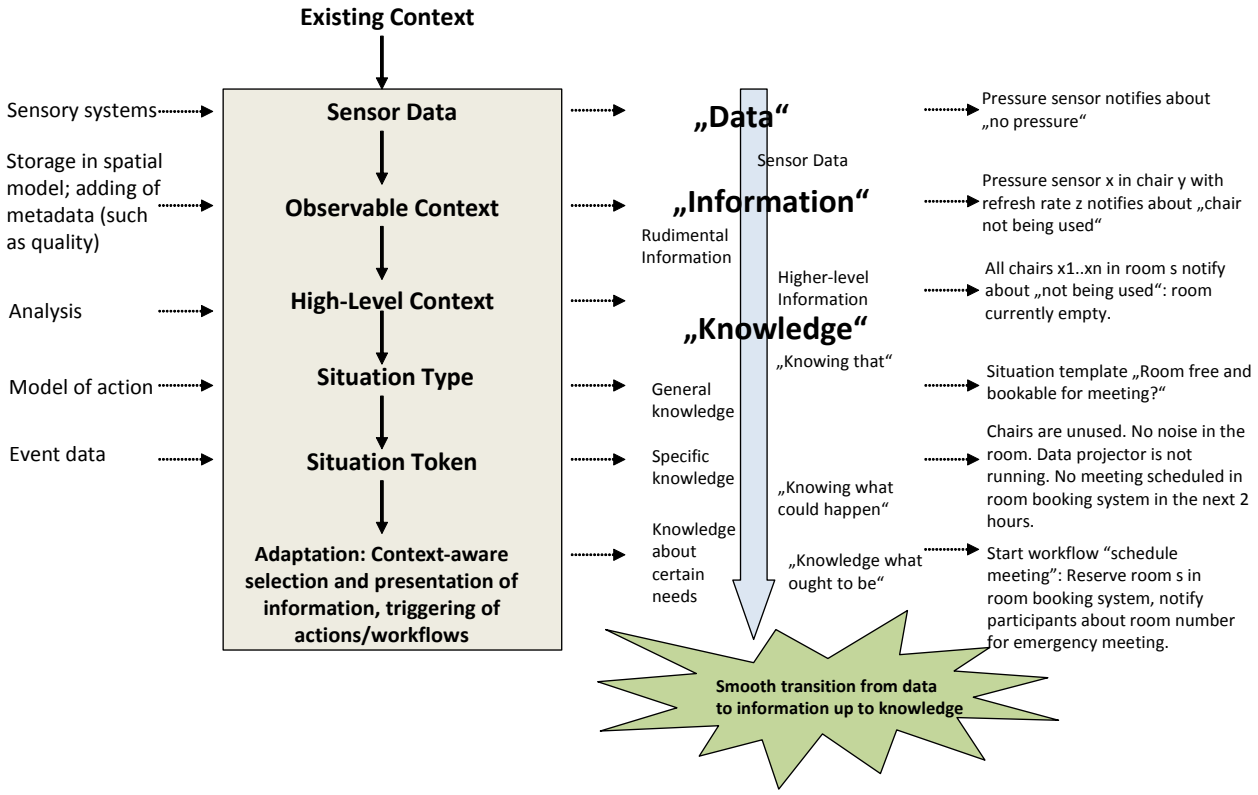

Fig. 1 Level Model of Situation Recognition

However, all this needs further detailed explanation. In Section II therefore, first, important related work is summarized thereby setting the field for our own work. Section III introduces important fundamentals mainly of terminological nature. Section IV presents the general architecture of the situation recognition component in the spatial model infrastructure. A possible realization of the situation templating approach, a learning algorithm for the automatic adaptation, and a workflow based application using situations templates are explained in Section V. Section VI discusses the approach from the view of technology ethics. Finally, Section VII summarizes our work and points out important future research directions.

II. Related Work

Due to the affinity to human ways of clustering information and the historical development of rule-based expert systems, most (context) reasoning systems [28], [29] ,[30] still use ontology-based and predefined rule-based approaches today. Compared to our approach, most of the existing context-aware systems are supposed to cover only a limited geographical area or support only a specific use case scenario [31], [32].

In this paper we focus our attention to a novel system which enables both the reasoning process in a general way, that is able to handle both with uncertainties during the situation detection process and using self-learning-approaches for the automatic adaption of these uncertainty-values during runtime by supervised-learning methods.

Furthermore our reasoning-approach bases on probabilistic networks where we distinct between two different uncertainty-metrics (confidence and probability), similar to other existing reasoning approaches (e.g. [19], [20]).

Beside the adaption of these metrics, we furthermore assume that a human expert initially designs the situation template and so we have to cope with ontology and design errors. In our previous work [21], a method for ontology-based learning is described. In order to be able to correct ontology and constraint errors, they have to be located in the ontology first. One possibility of locating errors is to accomplish data by new measurements several times until the fault location can be clearly determined. Corresponding work can be found in [23], [25], and [26]. In [24] an approach for recognizing and predicting context by learning from user behavior is described.

Despite of these methods we assume in this work that the ontology is correct – e.g. without any design-errors or constraint-errors. Thus, we are able to adapt automatically the uncertainty-parameters of our probability network approach we are using for reasoning. In statistics, many general inference techniques [38], [39] have been developed that have been applied to learning of probabilistic networks. Because we assume that the constraints and the ontology are correct, we examine the parameter-fitting problem – e.g. learn the parameters from data. These fitting algorithms exist for probability network (i.e. Bayesian networks) in the cases of complete and missing data [40], [41], and [42].

Because of the uncertainty and the non-observability in this reasoning domain, we only consider methods that can deal with missing data. Therefore, the Expectation-Maximization (EM) algorithm [42], [43] is a common and proven approach for dealing with incomplete information when building statistical models. Other common techniques in this domain are the iterative proportional fitting algorithm [41] and approaches using evolutionary algorithms [44], [45].

Furthermore we want to exploit the existing overlaps between learning of Bayesian Networks and Neural Networks described in [35], [36], [37] in our novel approach (see section V.B) for an automatic adaption using a only a simple Boolean feedback-information based on supervised-learning-methods.

For the development of the applications using the detected situations and presenting them to the user (or adapting the User Interface) any existing programming language can be used (like for example Java, C++, …). However, in our approach, we want to model the applications instead of programming them, to allow the users themselves to change the applications control flow. Because of that we are using the concept of situation-aware workflows (see section V.C) that orchestrate different automated services and integrate users with human task management for the realization of the situation-aware applications. This has the further advantage that the users can inspect the control flow to review what the application does in what situation.

III. Fundamentals

The use of context could be considered as a driving force for innovative products in mobile computing as well as a key enabling technology for pervasive computing, ubiquitous computing, and ambient intelligence. In the realm of mobile computing, portable devices make use of context to provide their users for example with important information dependent on the places where they are (so-called location-based services). In the realm of pervasive computing, where computing devices have become very small and where computing devices are invisibly embedded or "woven" in the environment, context-awareness is an important building block to make these environments intelligent, smart and adaptive. However, in order to better understand how context could be used and how the development of different context-aware applications (in the realm of mobile computing as well as in the realm of pervasive computing) could be facilitated, it is first important to understand what context is, what a situation is and what exactly constitutes a context-aware or situation-dependent system.

The term "context" is an intensively researched topic [6]. Originally, B. Schilit and B. Theimer [7] introduced the term into the field 1994. In the aftermath, a lot of different definitions of context have been proposed respectively what a context-aware system is and what it does (see [6] for an intensive survey and analysis on that including an overview on the discussion of the different basic types of context). Today, the definition given by A.K Dey and G.D. Abowd could be regarded as the commonly accepted one since it is rather formal and abstract thus being applicable in different areas of research on context-awareness. Dey/Abowd define *context* as "any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or object" [6]. Context-aware applications are therefore applications which adapt their behavior according to the situation they are in by a) automatically selecting information / discovering services b) automatically presenting information / services in an situation-dependent manner and/or finally c) by automatically triggering actions / executing services or workflows [6].

Having our situation template approach in mind, we could now extend the definition of Dey/Abowd by precisely saying what a situation is. A *situation* (better: a *situation type*) is the characterization of a specific, recurring circumstance or constellation in the real world that 1) could be described in an idealized way and that 2) serves as an evaluation basis for the adaption and reaction of context-aware applications. This idealized description of certain structural conditions (including a description of different parameters considered as relevant, a description of their thresholds, and instructions of processing/inferring the existence of a situation from these parameters) is hold within a *situation template*. Filling a situation template of a certain situation type with concrete data and processing it allows for the diagnosis of the existence of a particular situation, a *situation token*[1]. Based on the proved existence of a certain situation token, the application can now react and adapt its behavior accordingly.

Fig. 1 gives again the gist of inferring a certain situation (or more precisely: a certain situation token) from data in spatial model infrastructures using situation templates (thereby also showing a gradual rise from data to information up to certain kinds of knowledge).

All in all, one could say, that situation-dependent systems (using situation templates) are a subset of context-aware systems. In contrast to classical context-aware systems, a) the expert knowledge required to infer a certain kind of situation is not hidden in the code but swapped out in a separate situation template thus making this knowledge easily reusable, adjustable, extensible and b) the amount of expert knowledge required for inferring a certain situation is normally higher compared to classical context-aware systems thus enabling the explicit modeling and inference of complex, "difficult" everyday situations.

IV. General Architecture of a Situation Recognition Component for Spatial Model Infrastructures based on Situation Templates

[1] The distinction between types and tokens is in our mind a useful one – also in the realm of context-aware systems – and was first introduced by the philosopher C.S. Peirce [7]. Roughly speaking, the distinction describes the difference between some general sort of things ("types") and its particular concrete instances ("tokens")

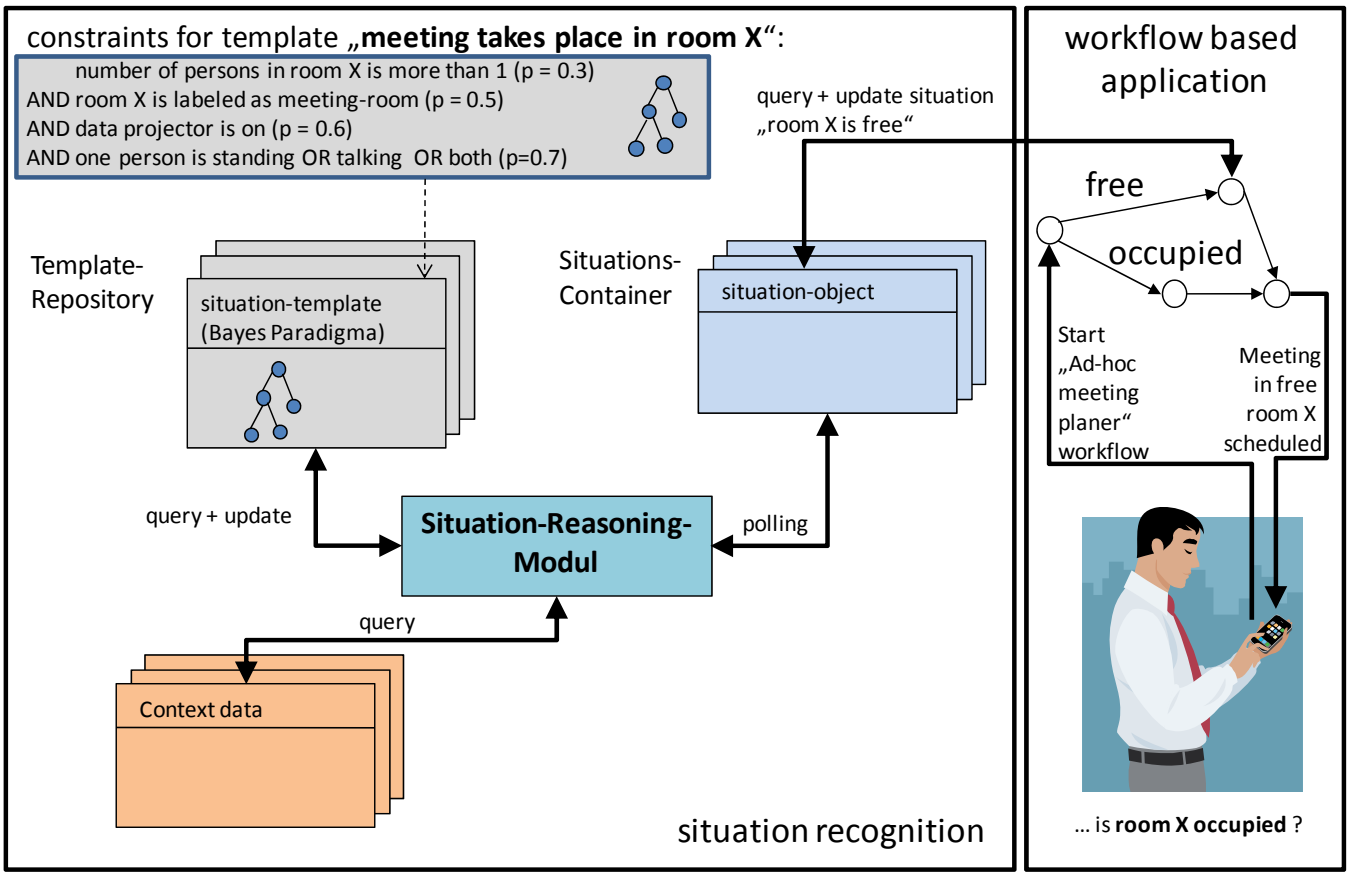

Fig. 2 Spatial model infrastructure architecture with situation recognition component

A situation template is – as already mentioned above – an abstract description of a certain situation type holding different constraints and their coherences for automatically recognizing the existence of a particular situation.

The Nexus Project [1] is a large-scale context-aware system envisioning a World Wide Space, which provides the conceptual and technological framework for sharing context models in an open platform and handling important challenges in such a platform including distributed processing of streamed context data, situation recognition by distributed reasoning, efficient management of context data histories, and quality of context information.

In spite of the goal of distributed processing in Nexus we disregard the distribution aspects in this paper and focus our approach on the representation of a situation as a template, the query-interface of the applications and show the reasoning of the situation under uncertainty, based on quality afflicted sensor values.

To consider the uncertainty and quality of context is a central point in this work in all areas, from the sensor values to the context data derived from the sensor values and the situations derived from the context data, and of course, at the end the application also pays attention for the confidence of the situation recognition.

As already mentioned above we distinguish between different uncertainty metrics in our reasoning architecture. One advantage of such a distinction is the modularity, the replace ability of the inference mechanism below and the possibility to expand the uncertainty metric for new ones.

Therefore, in this work we draw a distinction between the *probability*- and the *confidence*-metrics.

Hence, we define the *probability-metric* as a value, which represents the probability of the occurrence of the situation from the recognition-process view. For example a probability value of 0.8 means that the situation-recognition-process assumes that the situation is occurring with a probability of 80 percent.

Furthermore, we define the *confidence-metric* as a normalized value between zero and one, which reflects the quality or the correctness of the used situation-template. For example a confidence value of 0.99 means, that the recognition-process will detect the situation (independent of occurrence or nonoccurrence) with a high correctness. In contrast a confidence of 0.2 means that the quality of the template (or the underneath context-data) is poor.

We differ between these two metrics because of the possibility of different applications and their different handling of the confidence. E.g., a security-application for access-control to a banks vault should insist on a high confidence to prevent false positive results.

On the other hand, an application that cautions a blind person against an obstacle in front of the person should handle with less confidence because the blind person can put up with a false positive warning with less risk.

How these metrics are represented and passed through the system is a central part of the algorithms and the situation template schema presented later.

Fig. 2 show how the situation recognition component is integrated into the existing spatial model infrastructure. On the top of the figure, an example of a predefined situation template (only according constraints) is shown to detect a situation "whether a meeting takes place in the room X" or not. This and all other predefined templates are stored in a so-called *Template-Repository*. Each situation corresponds to a unique situation-object that links bidirectional to a situation template. Inside the spatial model infrastructure, the *Situation-Reasoning-Modul* uses the situation templates to evaluate the metrics of the situations continuously. Hence, the user detects a situation by using an application that queries the metrics of the corresponding situation-object, which is updated by the Situation-Reasoning-Module. For the user it is to difficult to use the situation recognition component directly, because of that an application has to be provided. In the example in Fig. 2 the application is running on the user's mobile device. The application only starts a *situation-aware workflow* (see section V.C) that implements the "Ad-hoc meeting planer" application. After the workflow is started, it looks up the situation of all available meeting rooms and reserves the first free room. For the situation evaluation step some context data and sensors are used. Which exact sensors and context data are used to detect the situation is defined in the situation template.

During the recognition-process the Situation-Reasoning-Module evaluates the situation template, query the needed context and sensor-data and infer the confidence- and the probability-metrics of the template according the used inference-method (e.g. Bayesian reasoning).

The results (here: probability and confidence) of the template-metrics are stored to the situation-object. Furthermore, the user can give a simple Boolean feedback to the system if the recognition was correct or not. This feedback is also written in the situation-object and can be used later in a batched learning-adaption-step to improve the recognition process.

However, one question remains open: What characterizes a good and sound situation template allowing for the safe inference of a situation token? The thesis is – from the point of view of philosophy of science – that a good situation template combines *different modes of logical reasoning and inference*, namely abductive, inductive and deductive ones (whereas abductive reasoning is the most basic one and also most unknown one).

The term *abduction* was coined by the philosopher C.S. Peirce – as the aforementioned distinction between (situation) type and (situation) token. Peirce himself gave in his "Collected Papers" the following tentative account of abduction: "The surprising fact, C, is observed; But if A were true, C would be a matter of course, Hence, there is reason to suspect that A is true" (CP 5.189) [9]. To be able to better understand this rough explanation of abduction, imagine a person in a house standing in a room some meters away from a window. Through the window, the person looks on the street and sees that the street is totally wet. The wetness of the street is our "surprising" fact. The "puzzlement" of the person could now be easily resolved if one remembers that when it rains, the street normally becomes wet. Thus, there is reason for the person to suspect that it has rained. However, this mode of inference is not a safe one: there are many other reasons why the street has become wet, such as that someone has spilled water there or that a street cleaner has driven past or those children have been playing there with squirt guns etc. Or to put it short: Abduction are ampliative but uncertain modes of inference indicating not for a probable but only a possible reason.

So far so good. Nevertheless, how is this all related to our topic of situation recognition? It is related to our topic insofar, as the use of sensory systems could be modeled as a form of abductive reasoning. E.g. using a humidity sensor and placing it somewhere in the environment is – from a logical point of view – the same as looking out of the window into the street and diagnosing that it has rained. However, this could be a "false positive" as well as someone smoking underneath a smoke detector triggering a fire alarm. Therefore it is common to build a *net of abductive reasoning* by combining different sensory systems so as to make the inference more safe (e.g. by looking not only through one window into the street but two or three respectively using more than just one humidity sensor or smoke detector). Moreover, abductive reasoning could be made safer when combining it with inductive and deductive reasoning. Concerning *induction*, one could look at the probability of raining depending of the time of day / time of year. Therefore, if someone sees through the window that the street is wet and he knows that it is April and that in April the probability of raining is very high, there is reason to suspect, that it really has rained. The recognition process could be made even safer by using *deduction*. By knowing the basic "law of nature" "When it rains, the street becomes wet", and the fact that is has really rained (e.g. by querying the internet, checking weather reports and rainfall radar pictures), one can deduce for sure that the street is wet because it has rained. [All in all, abductions – especially the combination of abductive reasoning with deductive and inductive reasoning and how these different modes "interact" – is a difficult and heavily researched topic [9], [10], [11], [12]. To put it short, if one has a close look at it, there is no way round on abductive reasoning since perception of something as something special has to be modeled as an abduction. Therefore deductions and inductions could not help but make use of abduction e.g. by diagnosing something as being of the same case (induction) or that is has really rained (deduction). However, that is what epistemology is all about: Figuring out how correctly, we can perceive the world through the window of our house and how safe our knowledge is. However, as K.R. Popper mentioned: Our knowledge of the world is constructed on "swampy land" [13] – in everyday life and in science.

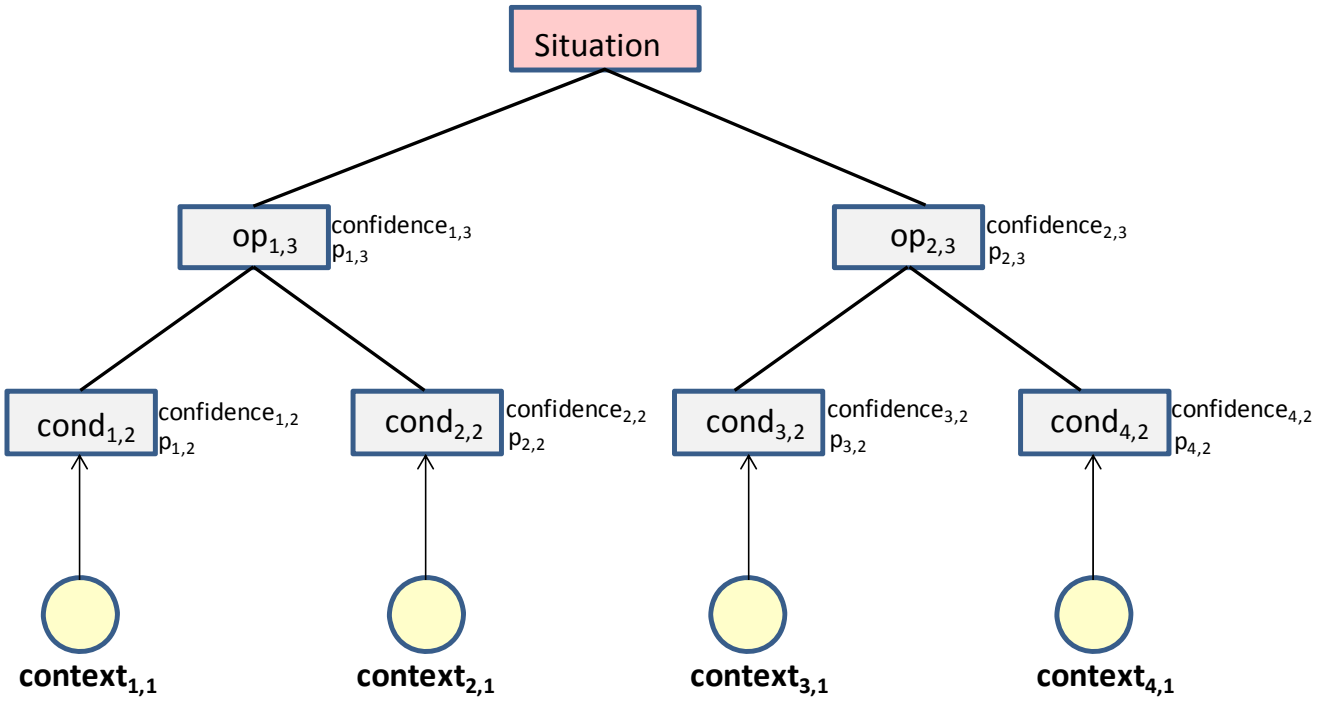

Fig. 3 Situation template

V. Realization of Situation Aware Systems

In this section, the realization of the previously introduced fundamentals and the architecture components is described. First in A it is explained how the situation types can be specified using a situation template. These types defined as templates are then used to perform the reasoning based on parameter-learning as described in B. The reasoning produces the previously defined situation tokens. Based on this situation tokens implemented as situation objects the application adapts their behavior. Hence, in C we describe how an application can be implemented using workflow technology that is enhanced in order to allow situation tokens to influence the control flow decisions.

A. Situation-Template

Due to the basic necessity of the handling with uncertainties we implement a probability network. Hence, the representation of a predefined situation-template is modeled as a directed graph. An example of a Situation Template is shown in Fig. 3. Because of the resulting polytree structure, the graph is called *Situation Aggregation Tree* (SAT).

Thus in more detail the graph is defined as SAT= (O, E), where $O = \{o_1,o_2,...,o_n\}$ represents a set of nodes. Furthermore the nodes can be differentiated into two different types of operators, the *constraint-validators* $CV = \{cv_1,cv_2,...,cv_n\}$ and the *subsituation-nodes* $SN= \{sn_1,sn_2,...,sn_n\}$. We define constraint-validators as operators which checks whether their input-value (e.g. sensor values) satisfy a predefined condition (using a specific operator-function *f*). Subsituation-nodes are operators that aggregate several nodes in one higher aggregated subsituation-node. As a consequence of the tree-structure, all paths and nodes are aggregated in a single top node, which represents the actual situation of the template.

The set $E=\{e_1,e_2,...,e_n\}$ represents a set of directed edges or links between the nodes which represents the relations between them. Furthermore $e_i=(o_k,o_j)$ is a directed edge from node o_k to node o_j. In other words, o_k is the parent node of o_j, thus we will say formally $pa(o_k) = o_j$.

The reasoning-mechanisms, e.g. the calculation of the probability- and confidence-metrics are done by exploiting the polytree-structure of the situation templates, using the Bayesian Theorem and the message-passing-algorithm of Pearl [33], which realize the distributed belief propagation in Bayesian Networks.

According to the Nexus-Idea, the information of the uncertainty of probability and confidence of every single context data can achieved from the Nexus Context Servers initially. We don´t want to go in detail how the nexus framework determine the uncertainty metrics of the sensors, but it should be clear, that these uncertainties can be obtained from the sensors technical documentation or it can be determined by comparing the sensors observations through training and statistical calculations.

The according metrics of each context-data (based on the template) is combined with the uncertainty metrics of the operators in the SAT itself using Bayesian methods during the reasoning-process. While this process the cumulative uncertainty-values are propagated to the top of the SAT to evaluate the situation. Because of the SAT structure and the stochastic independence of the context data, the uncertainty-metrics are propagated in the tree structure bottom-up to the top node. To aggregate the uncertainties during evaluation a set of probability distributions has to be added for each uncertainty metric to the structure of the current SAT. So we get the new extended structure SAT = (O, E, P, C) with

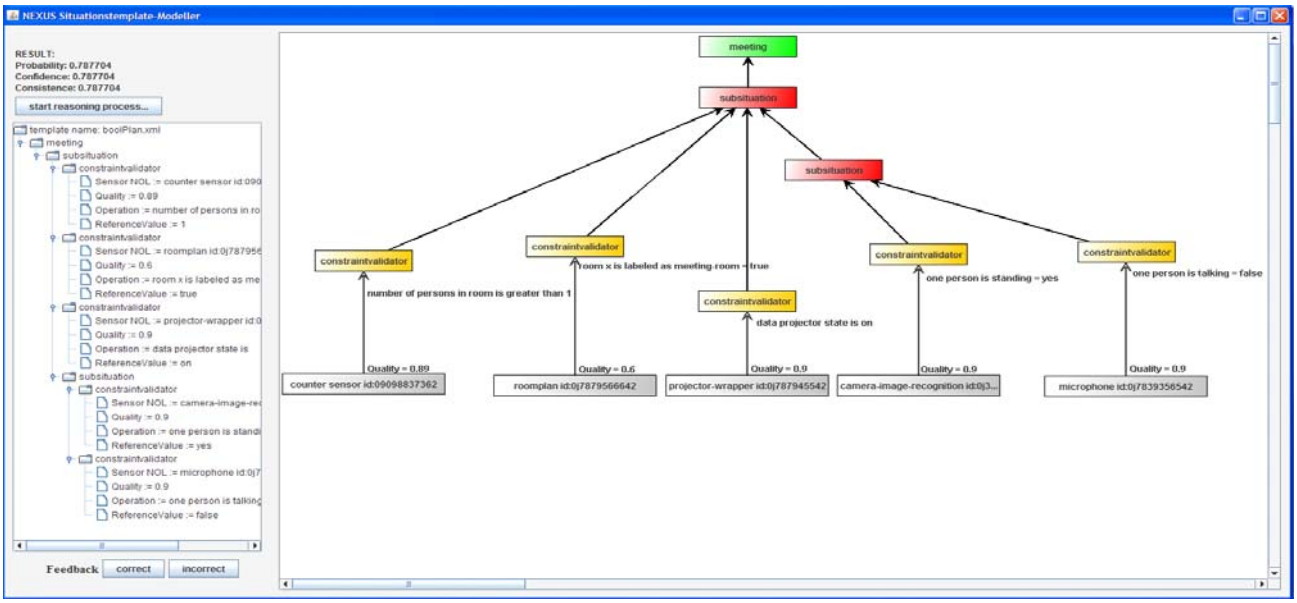

Fig. 4 Screenshot of Template-Designer

$P = \{CPT^P(o_1),..., CPT^P(o_n)\}$ and $C = \{CPT^C(o_1),..., CPT^C(o_n)\}$.

Whereas P stands for the probability metric and C stand for the confidence metric. Furthermore, $CPT(o_i)$ equates to the Conditional Probability Table of node o_i of the indicated metric. In detail, the CPT makes assertions about the according conditional probability of the according metric of node o_i in correspondence to its parent nodes $pa(o_i)$. Using this extension a Bayesian Belief approach is implemented.

The situation-template itself is stored in the template repository in a XML-structure, exploiting the tree-structure of the SAT. For fast designing of situation templates and further evaluation-purposes we implemented a simple prototype ("Template Designer") as shown in Fig. 4.

B. A learning algorithm for the parameter adaption of situation templates

In this section, we want to introduce the method of the automatic uncertainty adaption-process that runs immediately after the actual reasoning-evaluation to improve the probability network the template bases on.

As mentioned above the reasoning-method for situation-detection is done by a Bayesian inference approach using the $CPT^X(o_i)$ of node o_i and metric $X=\{P,C\}$ for all o_i in O. An approach for adapting the constraint-validators has already been published in a previous work [21]. Next, we want to describe how to exploit the Boolean user-feedback for adapting the CPTs of each uncertainty metric of each node.

Because of the fact that there is some overlap between learning of Bayesian Networks and Neural Networks [35], [36], and [37] and because of the advantage in exploitation our existing tree-structure of our SAT there, we decided to use a neural network learning method in this approach.

As already mentioned, each metric $X=\{P,C\}$ is represented via an individual and node specific Conditional Probability Table. An expert predefines these CPTs or they can be set randomly initially. After each reasoning-process the system gets a Boolean user-feedback, which determine, whether the situation actual occurred or nonoccurred.

With this simple feedback, we are able to adapt the uncertainty-metrics in the following way:

During the adaption-step (- e.g. immediately after the reasoning-evaluation) we transform the SAT into a neural net and interpret each CPT entry as a weight of an according edge of the neural net (NN) introduced in the work of [34]. Furthermore, we use a back propagation-algorithm to teach the NN weights. For each metric we use an own NN because the CPTs, the input-values and the teaching-vector is different for each metric. To adapt the probability-metric we use as input-values for the NN the context-probability-values given by the spatial model infrastructure. As adaption or teaching-vector of the NN, we use the actual Boolean user-feedback whether the situation occurred or not. Thus, we are able to adapt the weights iterative using a simple NN-Back propagation-Algorithm according to the given context-data (input-value(s) of the NN) and the information of the actual occurrence of the situation (output-value of NN).

Furthermore to adapt the confidence-metric we use instead as input-values of the NN the context-qualities provided by the spatial model infrastructure. As teaching-vector, we use a statistical measurement of the correctness of the corresponding template based on multiple training instances by summarizing the numbers of true-positives and true-negatives results divided by the number of all reasoning-processes.

After the optimal adaption of the weights of the NN, we transform the adapted neural network back to the SAT-structure, by interpreting the weights of the NN to the corresponding CPT-entry.

C. Development of Situation-aware Applications

To allow users to benefit from the situation recognition they

need convenient applications hiding the complexity of the described system. Of course programming such applications is possible, but very difficult, because many aspects have to be thought of. 1) A situation-aware program has very many branches for each situation at least one. 2) The adaption logic of such programs is very complex because it has to change the program code while executing and the change has to "fit" to all active situation-branches.

Because of that we think that workflows are very good concept for modeling such applications. Therefore, our vision is to develop a workflow language that naturally integrates and uses situations. In the following two concepts for integration of situations in workflows are described.

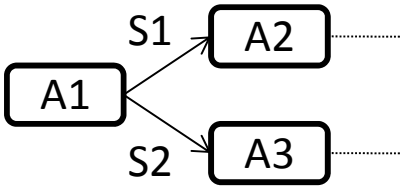

Fig. 5 Situation Transition

The first workflow construct that could be realized using situation recognition is a situation-transition (see Fig. 5). This construct can be used comparable to a normal transition condition, but instead of a normal Boolean condition, the existence of a situation would be used for deciding where to continue in the workflow. This could be used for example in a workflow for planning of an ad-hoc meeting. If somebody wants to start a meeting the workflow could than determine all meeting rooms where the situation "Room is free" is valid and build up a list for the user to select a room (see example on Fig. 2).

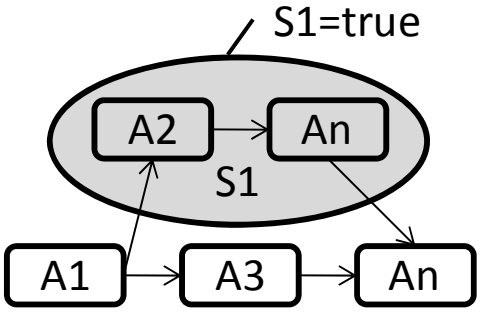

Fig. 6:Situation Sphere

The second one is more complex; it is the situation-sphere (see Fig. 6). Here a region of locality is defined in a workflow and is called a situation-sphere. Here local variables exist and a workflow can handle errors while the sphere is executing. A sphere annotated with a situation has the meaning that the situation has to be valid (or true) all the time while the sphere is executed. For example, a workflow for meeting planning could annotate a sphere in the workflow where a room is reserved with the situation: "Room is free". Than if this situation if evaluated to false a "Situation not valid" fault would occur. In this case, a standard fault handler can be modeled in the workflow for handling this case. This fault handler could reserve another free room automatically and inform the organizer of the meeting about the changed room. Nevertheless, how the error is handled can be defined by the user themselves by modeling the fault handler flow.

If a situation changes while execution of a workflow, which can happen often because workflows are mostly "long-running" (days to years) the workflow has to be adapted to that change. This can be done by exchanging parts of the workflow by other parts defined for the new situation. This is a very complex topic and cannot be detailed in this paper.

VI. Discussion

From the point of view of technology ethics, the automatic coupling of situation recognition and the performance of system actions needs more thinking since a new type of error could occur: the apparently correct execution of ready-made routines, which is completely situation inadequate due to an erroneous recognition of a situation. In sociology of technology, G.I. Rochlin [14], [15], extensively discussed this. Rochlin argues that it needs to be thought very well which tasks to automate and which not – especially to keep massive harm from people (physical as well as monetary one). Rochlin shows that there are critical tasks which need the permanent involvement of the user thereby creating systems which are "smart", "adaptive", "context-ware" in a reduced sense but which are able to better react (due to the permanent involvement of the user) to non-anticipated situations. The idea of rather permanently assisting the user instead of trying to replace him and to degrade him to a passive observer or bystander monitoring smart, autonomous systems is also put forward by J. Weyer since passive yet attentive observers are often utterly incapable of dealing with a critical, unforeseeable situation or case of emergency [16]. Concerning intelligent, smart, autonomous environments, where computers are invisibly weaved into the fabric of everyday life, the users furthermore even have not the chance to notice, what is going on [17]. Therefore, something needs to be implemented into the system, which we call the Stuttgart Concept of Parallel Communication [17], [18]. First – especially in smart, intelligent environments – there needs to be a means for the user to create transparency "on demand" what the system is currently doing and what it is not doing. Second, there need to be means – especially in critical tasks – to check the correctness of the recognition of a situation token by using different information channels lying "in parallel", because human perception – as the basis for decision making – is of holistic nature only: Perception is often the perception of a certain subjective overall impression or atmosphere whereas it is often difficult to tell by what it is really caused and which senses contributed to it. It is the problem of technical mediated communication that it dissects this holism and therefore goes along with a reduction and/or loss of information "channels". E.g., when using a phone – compared to the natural face-to-face situation – communication is restricted to the aural

channel only making it often complicated to judge or interpret the statement given to us by others. Therefore, in technical mediated communication, new channels of information need to be introduced which compensate for this loss.

VII. Conclusion and Future Work

In this paper, we presented an interdisciplinary analysis of the recognition process of situations with uncertain data using predefined generic templates. Thereby we discussed both different technical, ethical and philosophical issues for a better understanding of situation dependent systems. Finally, we used these issues to design an implementation of a situation recognition system, which allows for the reasoning with uncertain data using situation templates. Furthermore, we presented a method of an automatic adaption using learning algorithm for the parameter refinement of the template.

As applications are needed for using the situation recognition system, because this is too cumbersome to do it manually, we presented the vision of situation-aware workflows that allow the easy modeling of such applications.

In the future, we want to make the recognition system described above more general and more robust. The presented approach only works fine with the assumption that exactly one template is available for one situation. As a further step, we want to generalize this constraint by allowing more templates for the same situation. On the one hand, the advantage is a more robust and powerful recognition system. On the other hand, there could be more problems – i.e. if several templates are inconsistent with one another. Consequently, we have to consider the consistence as a new uncertainty value for the recognition process. Specifically, we have to recognize, describe and resolve inconsistency, which is due to the already existing uncertainty, a non-trivial task in the future.

Acknowledgment

This work was developed within the Nexus project (collaborative research centre/SFB 627), which is supported by the German Research Foundation (DFG).

References

[1] R. Lange, N. Cipriani, L. Geiger, M. Großmann, H. Weinschrott, A. Brodt, M. Wieland, S. Rizou, and K. Rothermel. Making the world wide space happen: New challenges for the nexus context platform. In Proceedings of the 7th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2009), Galveston, TX, USA, 2009.

[2] M. Großmann, M. Bauer, N. Hoenle, U.-P. Kaeppeler, D. Nicklas, and T. Schwarz. Efficiently managing context information for large-scale scenarios. In Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications, 2005.

[3] K. Henricksen and J. Indulska. A software engineering framework for context-aware pervasive computing. In Proceedings of the 2nd IEEE International Conference on Pervasive Computing and Communications, 2004.

[4] M. Roman and R. H. Campbell. Gaia: Enabling active spaces. In Proceedings of the 9th ACM SIGOPS European Workshop, Kolding, 2000.

[5] D. Salber, A., K. Dey, and G. D. Abowd. The Context Toolkit: Aiding the development of context-enabled applications. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: The CHI is the Limit, 1999.

[6] A.K. Dey and G. D. Abowd. Towards a better understanding of context and context-awareness. In Proceedings of the Workshop on The What, Who, Where, When, and How of Context-Awareness, as part of the 2000 Conference on Human Factors in Computing Systems (CHI 2000), The Hague, The Netherlands, April 3, 2000.

[7] L. Wetzel. Types and Tokens. In the Stanford Encyclopedia of Philosophy (Winter 2008 Edition), Edward N. Zalta (ed.), URL=http://plato.stanford.edu/archives/win2008/entries/types-tokens/

[8] C.S. Peirce. Collected Papers. Harvard University Press.

[9] C. Hubig. Die Kunst des Möglichen. Band 1: Technikphilosophie als Reflexion der Medialität. Transcript.

[10] C. Hubig. Abduktion – Das implizite Voraussetzen von Regeln. In G. Jüttemann (ed.), Individuelle und soziale Regeln des Handelns, pp. 157-167, Asanger.

[11] C. Hubig. Interdisziplinarität und Abduktionenwirrwarr. In N. Gottschalk-Mazouz/N. Mazouz (eds.), Nachhaltigkeit und globaler Wandel. pp. 319-340. Campus.

[12] G. Schurz. Patterns of Abduction. In Synthese Vol. 164, No. 2, 2008, pp. 201-234

[13] K.R. Popper. The Logic of Scientific Discovery. Routledge.

[14] G.I. Rochlin. Iran Air Flight 655 and the USS Vincent. In T.R. La Porte (ed.), Social Responses to large scale technical systems. pp. 99-125. Kluwer.

[15] G.I. Rochlin. Trapped in the net. The unanticipated consequences of computerization. Princeton University Press.

[16] J Weyer. Die Risiken der Automatisierungsarbeit. Mensch-Maschine-Interaktion und Störfallmanagement in hochautomatisierten Verkehrsflugzeugen. In Zeitschrift für Soziologie, Vol. 25, 1997, pp. 239-257.

[17] C. Hubig. Selbstständige Nutzer oder verselbstständigte Medien. Die neue Qualität der Vernetzung. In F. Mattern (ed.), Total vernetzt. Szenarien einer informatisierten Welt. pp. 211-230. Springer.

[18] C. Hubig. Die Kunst des Möglichen. Band 2: Ethik der Technik als provisorische Moral. Transcript.

[19] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and contextawareness. In HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, pages 304–307, London, UK, 1999. Springer-Verlag.

[20] S. McKeever, J. Ye, L. Coyle, and S. Dobson. A context quality model to support transparent reasoning with uncertain context. In 1st International Workshop on Quality of Context (QuaCon), Stuttgart, Germany, 2009.

[21] Zweigle, Oliver; Häussermann, Kai; Käppeler, Uwe-Philipp; Levi, Paul: Extended TA Algorithm for adapting a Situation Ontology. In: Proceedings of the FIRA RoboWorld Congress 2009, Progress in Robotics. Communications in Computer and Information Science; 44, pp. 364-371, Incheon, Korea: Springer Verlag, August 18, 2009.

[22] de Kleer, J. Using crude probability estimates to guide diagnosis. pages 118–123, 1992.

[23] de Kleer, J. and Williams, B.C. Diagnosing multiple faults. pages 372–388, 1987.

[24] Mayrhofer, Rene; Radi, Harald and Ferscha, Alois. A.: Recognizing and predicting context by learning from user behavior. pages 25–35, 2003.

[25] Reiter, R. A theory of diagnosis from first principles. Artif. Intell., 32(1):57–95,1987.

[26] Hou, Aimin. A theory of measurement in diagnosis from first principles. Artif.Intell., 65(2):281–328, 1994.

[27] Cheng, R; Prabhakar, S: Managing uncertainty in sensor database. SIGMOD Rec., 32(4):41–46, 2003.

[28] Chen, H. and Finin, T. and Joshi, A.: An ontology for context-aware pervasive computing environments, In: The Knowledge Engineering Review, vol. 18, no 3, pages 197-207, 2004

[29] Wang, X.H. and Zhang, D.Q. and Gu, T. and Pung, H.K.: Ontology based context modeling and reasoning using OWL, In: Proceedings of the second IEEE annual conference on pervasive computing and communications workshops, Vol. 18, 2004

[30] Henricksen, K. and Indulska, J. and Rakotonirainy, A.: Modeling context information in pervasive computing systems, in: Lecture notes in computer science, pages 167-180,2002

[31] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer. Easyliving: Technologies for intelligent environments. pages 12–29. Springer Verlag, 2000.
[32] S. Pado and M. Lapata. Constructing semantic space models from parsed corpora, 2003.
[33] J. Pearl. Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference. Morgan Kaufmann, September 1988.
[34] Sowmya Ramachandran, Raymond J. Mooney: Revising Bayesian Network Parameters Using Backpropagation, IN: Proceedings of the 1996 IEEE International Conference on Neural Networks, pp.82-87, Washington D.C., June 1996.
[35] R.M. Neal: Connectionist learning of belief networks, In: Artificial Intelligence, vol. 56, pp. 71-113, 1992.
[36] L.K. Saul, T. Jaakkola, and M.I. Jordan: "Mean field theory for sigmoid belief networks", Technical Report 9501, Computational Cognitive Science, MIT, 1995.
[37] W. Buntine: Operations for learning with graphical models, In: Journal of Artificial Intelligence Research, vol. 2, pp. 159-225, 1994.
[38] M.A. Tanner, Tools for Statistical Inference, Springer-Verlag, New York, second edition, 1993.
[39] R.E. Kass and A.E. Raftery, \Bayes factors and model uncertainty", Journal of the American Statistical Association, vol. 90, pp. 773-795, 1995.
[40] D. Edwards, \Hierarchical interaction models", Journal of the Royal Statistical Society B, vol. 51, no. 3, 1989.
[41] R. Jirousek and S. Preucil: On the effective implementation of the iterative proportional fitting procedure", Computational Statistics and Data Analysis, vol. 19, no. 2, pp. 177-189, 1995.
[42] S.L. Lauritzen: The EM algorithm for graphical association models with missing data, Computational Statistics and Data Analysis, vol. 19, no. 2, pp. 191-201, 1995.
[43] A.P. Dempster, N.M. Laird, and D.B. Rubin: Maximum likelihood from incomplete data via the EM algorithm, In: Journal of the Royal Statistical Society B, vol. 39, pp. 1-38, 1977.
[44] Myers, J.W. and Laskey, K.B. and DeJong, K.A.: Learning bayesian networks from incomplete data using evolutionary algorithms, In: Proceedings of the Genetic and Evolutionary Computation Conference, vol 1,pp:458-465, 1999
[45] Tian, F. and Lu, Y. and Shi, C.: Learning Bayesian networks with hidden variables using the combination of EM and evolutionary algorithms,In: Lecture notes in computer science,pp:568-574, 2001