Dealing With Poorly Mapped Entities in Adaptive Object-Oriented World Modeling

Achim Kuwertz*
*Institute for Anthropomatics (IFA)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
Email: achim.kuwertz@kit.edu

Jürgen Beyerer*†
†Fraunhofer IOSB - Institute of Optronics,
System Technologies and Image Exploitation
Karlsruhe, Germany
Email: juergen.beyerer@iosb.fraunhofer.de

***Abstract*—As a prerequisite to supporting situation assessment and awareness tasks, information about considered situations must be adequately represented in respective support systems. A meaningful representation can be provided by world modeling systems like the Object-Oriented World Model (OOWM). The OOWM models the current state of an application domain based on observed information while relying on domain knowledge for adding semantics to the model. This domain knowledge is usually designed a priori by human experts. Yet, in complex real-life domains, the occurrence of a priori not considered domain entities is likely during system operations. Therefore, the OOWM has to be able to deal with a potentially open world – by adapting the domain model in reaction to current and past circumstances. To this end, an approach for adaptive knowledge management has previously been proposed. Extending the approach, this contribution introduces a holistic process for managing and dealing with unforeseen entities supporting model adaptation and details a method for acquiring new concept definitions based on poorly represented entities.**

***Keywords*-adaptive knowledge management; concept acquisition; model evaluation; object-oriented world modeling;**

I. Introduction

When building support systems for facilitating tasks like situation assessment and the consequent decision making, a structured approach for representing information about the current state of a considered application domain should be applied. World modeling systems like the Object-Oriented World Model (*OOWM*, [1]) are able to provide such a representation to higher-level components charged with situation assessment and awareness tasks. The *OOWM*, summarized in Sec. II, is a semantic world modeling system designed to represent the current state of an application domain and is backed by a conceptual domain model. In this domain model, the semantics of domain entities is captured, including all entities which a priori are considered to be relevant. Since situational assessment tasks are often performed in complex application domains, the OOWM is likely to encounter domain entities not considered during the a priori model generation process. In order to cope with such circumstances, an adaptive approach to managing the OOWM domain model and the encoded knowledge is required. The approach taken for the OOWM is denoted as *adaptive knowledge management* and will be summarized in Sec. III. Basic building blocks of this approach have been presented in [2] and [3], including methods for evaluating the ability of a domain model to describe the currently observed domain state and methods for identifying those entities that are only poorly described by the model. Building on this approach, the current contribution focuses on performing purposeful adaptations of the domain model in the case of degrading model quality. Sec. IV presents the proposed holistic approach, while details on the acquisition of new concepts are given in Sec. V. A main idea of this contribution is the controlled knowledge adaptation, similar to approaches for cognitive systems like [4], where visual categories are acquired in a self-organizing fashion driven by communication needs, or [5], also extending knowledge models. For purposeful adaptation, the similarity of observed entities has to be determined – a task related to fields of Conceptual Clustering and Numerical Taxonomy (e.g., [6], [7], [8]).

II. The Object-Oriented World Model

The OOWM is a probabilistic framework designed for general world modeling tasks, able to integrate observations provided by heterogeneous sensing systems into a combined, consistent representation. It consists of two components responsible for handling input, i.e., observations of domain entities, on the one hand, and representing the semantic domain knowledge, on the other hand. Fig. 1 gives an overview of the OOWM and its components. The OOWM processes observations of domain entity features like their size, position, etc., and stores those observations in the *World Model*, which constitutes the dynamic modeling component of the OOWM. Observed entities are associated to the concepts stored in the semantic domain model of the OOWM. This second OOWM component is denoted as *Background Knowledge* and contains the a priori domain model as e.g. generated by human domain experts.

All information in the OOWM is represented in a probabilistic fashion following a Bayesian approach. The representation of an observed domain entity in the World Model is denoted as *representative* R and stored as the set of observed entity attributes $\mathcal{A}_R := \{A_1, A_2, \ldots, A_n\}, n \in \mathbb{N}$.

http://dx.doi.org/10.1109/CogSIMA.2014.6816545 ©2014 IEEE.

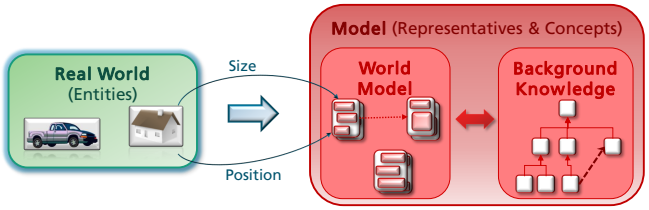


Figure 1. The Object-Oriented World Model consists of two components: the World Model is responsible for storing observations of real-world entities, while the Background Knowledge constitutes a conceptual domain model used for capturing the semantics of the application domain.

Each attribute is represented by a probability distribution $p_{A_i}(a)$ describing the degree of belief (*DoB*) in the observed attribute value. Either discrete probability distributions p_A^d with domain $\mathcal{S}_A$ or continuous Gaussian and Gaussian mixture distributions p_A^c are employed to represent attribute values. The probability distributions form the basis of the probabilistic information processing employed in the OOWM, like linking new observations to representatives using data association methods, or updating stored observation values using Bayesian fusion methods. The World Model in total consists of the set of all representatives, denoted as $\mathcal{R}$.

The counter-piece to representatives in the World Model are object-oriented class-like descriptions of relevant domain entities denoted as *concepts*. A concept C is represented in the same way as a representative, i.e., characterized by a set $\mathcal{A}_C := \{A_1, A_2, \ldots, A_m\}$ of concept attributes and, in addition, a number of relations to other concepts. Concept attributes are again represented by discrete and continuous probability distributions $p_{A_c}(a)$ interpreted as DoB. The Background Knowledge can be seen as the concept set $\mathcal{C}$.

For connecting World Model and Background Knowledge, a probabilistic classification approach is employed. This approach associates a given representative R to a concept C using the conditional probability distribution $p(C|R)$. This so-called *association probability* constitutes a discrete probability distribution over all the concepts C in $\mathcal{C}$, rating how well the observed attribute values of $\mathcal{R}$ correspond to the concept attributes. As proposed in [3], this association probability can be calculated according to

$$p(C|R) = \frac{1}{z} \cdot \prod_{A_i \in \mathcal{A}_R} \left(\int_{\mathbb{R}} p_{A_i}(a) \cdot p_{A_c}(a) \, \mathrm{d}a \right), \quad (1)$$

where z is a normalization constant and A_c is the concept attribute corresponding to the attribute A_i of R. Further details on the OOWM approach in general are given in [1].

III. Quantitative Adaptive Knowledge Management

The OOWM Background Knowledge constitutes a modeling artifact created by human domain experts. It includes all the concepts that were considered as relevant for OOWM operations at design-time. As a consequence of being a priori modeled, the Background Knowledge is only able to represent concepts from a closed part of the application domain. For complex and dynamic domains, this approach may be insufficient, e.g., when observing unforeseen domain entities. For handling such situations, the OOWM must be able to deal with unforeseen entities and must provide a solution for adaptively extending its domain model in order to handle an open world.

The OOWM open-world modeling approach is called *adaptive knowledge modeling* and has been introduced in [3]. Adaptive knowledge modeling is a management process consisting of several sub-tasks, like the quantitative evaluation of model goodness, i.e., its ability to represent an observed application domain, presented in [2]. Continuously evaluating this *model quality* allows to determine when the quality is degrading and, in return, trigger respective measures for counteracting this degradation. Details on how such countermeasures can be designed and realized are given in the Sections IV and V. As a prerequisite for such countermeasures, the management process must be able to identify those representatives which are only poorly described by the domain model. Those representatives are potential starting points for model improvement. In the following, previous results for model evaluation and representative identification are summarized, as they form the basis for the proposed adaptive knowledge management process.

A. Quantitative Measures for Evaluating Model Quality

Being able to quantitatively rate how well a domain model is suited for representing the observed domain entities is a central task in controlling the adaptive knowledge management process. In [2], a framework for quantitatively rating this model quality was proposed, based on the principle of Minimum Description Length (*MDL*, e.g. [9]). In this framework, the approach for rating the overall Background Knowledge quality combines two separate measures. The first measure, denoted as model complexity $L(\mathcal{C})$, rates the description length of the Background Knowledge $\mathcal{C}$, and is used to penalize overly complex models. The second measure rates the length of the description needed to represent the observed entities using the domain model, denoted as model correspondence $L(\mathcal{R}|\mathcal{C})$. The *overall model quality* in adaptive world modeling is then calculated as

$$Q(\mathcal{R}, \mathcal{C}) = L(\mathcal{C}) + L(\mathcal{R} \,|\, \mathcal{C}) \,. \quad (2)$$

Model complexity $L(\mathcal{C})$ is calculated as the sum of concept complexities $L(C)$, $C \in \mathcal{C}$, which are rated as the sum of the specificity $L(p_{A_c}(a))$ of their attribute distributions – in total, described by the cumulative measure

$$L(\mathcal{C}) = \sum_{C \in \mathcal{C}} \sum_{A_c \in \mathcal{A}_C} L(p_{A_c}(a)) \,.$$

The specificity of DoB distributions is measured based on concentration measures like Shannon entropy $H(\cdot)$, e.g., according to $L(A^d) = \log(|\mathcal{S}_A|) - H\big(A^d\big)$ for a discrete attribute A^d. Continuous attributes are handled by using a discretization approach.

Model correspondence, the second addend in the quality measure $Q(\mathcal{R}, \mathcal{C})$, is calculated as the expected sum

$$L(\mathcal{R}|\mathcal{C}) = \sum_{R \in \mathcal{R}} \sum_{C \in \mathcal{C}} p(C|R) \cdot L(R|C) \tag{3}$$

of representative-to-concept correspondences $L(R|C)$, using the association probability $p(C|R)$ as a weight. The concept correspondence $L(R|C)$ measures how well the attributes A_i of a representative R and the corresponding attributes A_{c_r} of a concept C are matching, according to

$$L(R|C) = \sum_{A_i \in \mathcal{A}_R} w_{A_{c_r}} \cdot L(A_i|A_{c_r}) \tag{4}$$

allowing for individual weights $w_{A_{c_r}}$. The attribute correspondence $L(A_i|A_{c_r})$ constitutes a function for comparing probability distributions, implemented by the Kullback Leibler divergence. For more details on quantitative model evaluation please refer to [2].

B. Measures for Identifying Poorly Mapped Representatives

For identifying those representatives in the World Model that can only be poorly mapped the concepts in the Background Knowledge (denoted as poorly mapped representatives, *PMR*), several measures have been proposed [3]. Two local measures based on the current set of representatives $\mathcal{R}$ and the current set of concepts were presented, namely an association-based measure and a correspondence-based measure. The *association-based measure* uses the association probability $p(C|R)$ for rating how well a representative is mapped to a concept, implemented as the relative entropy difference $M(p(C|R)) = \frac{\log(|\mathcal{C}|) - H(p(C|R))}{H(p(C|R))}$. Alternatively, the concept correspondence (4) can be employed to measure the mapping quality. In both cases, thresholds have to be defined for deciding if a PMR is given.

IV. Managing Poorly Mapped Representatives

As described previously, quantitative adaptive knowledge management defines measures for rating model quality and identifying PMRs. Those measures form the basis for adaptively managing the concepts contained in Background Knowledge in response to current and historic observations of domain entities. The next important step is to detail the management process, denoted as *PMR management*, that controls and leads to the adaptation of the domain model.

A. PMR Management

In order to allow for sophisticated decisions regarding model adaptation, the knowledge management process must take into account not only the situation occurring at the present moment of time, but also situations that have occurred in the (more or less) recent past. Thus, the historic states of the World Model (and also the Background Knowledge) have to be considered for knowledge management. Following this rationale, it is useful to perform a kind of accounting on PMRs, i.e., storing those representatives that at their time of observation have been identified as PMRs in a timed list $\mathcal{P}_\mathcal{R}$ of PMRs. This PMR list constitutes the basic information structure for the PMR management approach.

For modeling systems operating in complex, only partially known environments, the occurrence of previously unknown entities might be quite probable. In analogy to human cognitive processing, yet, not all of the observed or referenced unknown entities will be of relevance for the system in order to perform its tasks. Furthermore, some of the domain entities might occur less often than others, some perhaps just once. In order to prevent to system from learning and adapting to only temporally or partially relevant concepts, the PMR management approach is designed to perform some estimates of relevance for detected PMRs, based on some of the measures defined previously.

PMR management follows the general scheme presented in Algorithm 1 and can be seen as a background service being performed continuously during OOWM operations. The first step in PMR management is the identification and accounting of PMRs, using the measures presented in Sec. III-B. Besides identifying PMRs, the overall quality of the domain model is continuously evaluated using the quality measure (2). If a degrading quality is detected, i.e., if

$$Q(\mathcal{R}_{t_n}, \mathcal{C}_{t_n}) \; - \; Q(\mathcal{R}_{t_l}, \mathcal{C}_{t_l}) \; \geq \Phi_Q \, , \;\; 0 < l < n \, ,$$

the current model quality $Q(\mathcal{R}_{t_n}, \mathcal{C}_{t_n})$ exceeds the previous model quality $Q(\mathcal{R}_{t_l}, \mathcal{C}_{t_l})$, given at the last point of time t_l at which a model adaptation has occurred, by more than the threshold value Φ_Q, a model adaptation procedure is triggered. Since model quality is rated based on MDL, higher values indicate a greater description length and, thus, a worsening quality. The adaptation procedure begins with ranking the currently stored PMRs according to their relevance and utility for improving the domain model - evaluated over a time horizon including past states.

B. PMR Ranking

For ranking the representatives contained in the timed PMR list $\mathcal{P}_\mathcal{R}$, different approaches based on the concept correspondence $L(R|C)$ (4) can be chosen. Both approaches presented in the following exploit the fact that representatives and concepts are both defined as sets of attributes (Sec. II) and can thus be used interchangeably as the arguments of the correspondence measure (4). Both approaches therefore use PMRs as concept proxies for ranking PMRs.

The first approach is based on the rationale that only the most relevant PMRs should be incorporated into the domain model. The approach rates this relevance by trying

Algorithm 1 PMR Management

```
▷ Input: World Model R, Bg. Knowledge C
P_R := {};   P_R^x := {};
while OOWM is operating at current time t_n do
    ▷ Detect PMRs and add to timed list
    P_R = P_R ∪ Detect_PMRs(R_{t_n}, C_{t_n});
    ▷ Trigger adaptation if quality degrades
    while Check_Model_Quality(R_{t_n}, C_{t_n}) do
        ▷ Rank current PMRs
        P_R^* = PMR_Ranking(P_R \ P_R^x);
        ▷ Find positive samples f. adaptation
        P_R^* = Find_Samples(P_R^*, P_R);
        ▷ Perform adaptation
        C^* = Model_Adaptation(C_{t_n}, P_R^*);
        ▷ Re-evaluate model quality
        if Check_Model_Quality(R, C(C_{t_n} ← C^*)) then
            ▷ Accept model adaptation
            C_{t_n} = C^*;
            ▷ Update PMR list
            P_R = Detect_PMRs(P_R, C^*);
        else
            ▷ Mark samples as being processed
            P_R^x = P_R^x ∪ P_R^*;
        end if
        ▷ Do further adaptation if necessary
    end while
end while
```

to count how often instances of that class of domain entity the considered PMR belongs to have occurred in the past, i.e., the timed PMR list $\mathcal{P}_\mathcal{R}$. For evaluating this count, all the PMRs representing the same domain entity as the considered PMR as well as all the PMRs representing domain entities belonging to the same class have to be identified. This is necessary, since the relevance count actually must be associated which the class of domain entity rather than single instances. For identifying so related PMRs, the concept correspondence (4) is used as measure of choice, since it reflects how well a PMR as able to describe other PMRs. Thus, for each PMR $P_R^t \in \mathcal{P}_\mathcal{R}$ the count $S(P_R^t)$ is calculated according to

$$S\left(P_R^t\right) \;=\; \left|\{\, P_R \in \mathcal{P}_\mathcal{R} \;\mid\; L(P_R \,|\, P_R^t) \;<\; \Phi_S^L \,\}\right| \;,$$

using the threshold Φ_S^L for determining class affiliation.

The alternate approach calculates the potential utility of a PMR P_R^t for improving the current and historic model quality when using this PMR as a new concept. This utility is calculated as the difference in model correspondence (3) with versus without the new concept – e.g., when regarding only the current model quality, according to

$$\Delta L\left(P_R^t\right) \;=\; L(\mathcal{R}_{t_n} \,|\, \mathcal{C}_{t_n}) \;-\; L(\mathcal{R}_{t_n} \,|\, \mathcal{C}_{t_n} \cup \{P_R^t\}) \;.$$

The results of both approaches can be used for ranking the current PMRs, choosing that PMR P_R^* as adaptation candidate with the highest rating, i.e., either the highest number of similar PMRs or the highest predicted utility.

C. Finding Learning Samples - PMR Clustering

As the result of PMR ranking, the PMR P_R^* most promising for model adaptation is determined. The next step in PMR management is to find the set of candidate PMRs ${\mathcal{P}_\mathcal{R}}^*$ that are to be passed to the model adaptation sub-component of PMR management for performing the actual adaptation. For determining this set of candidate PMRs, the association probability $p(C|R)$ (1) is used as a measure of similarity between PMRs. As the case for concept correspondence, it is also possible to instantiate the concept parameter of the association probability using a representative as argument. In consequence, the set of candidate PMRs is calculated based on the threshold Φ_S^A according to

$${\mathcal{P}_\mathcal{R}}^* \;=\; \left\{\, P_R \mid\; p(P_R^* \,|\, P_R) \;<\; \Phi_S^A \,\right\} \;.$$

D. Applying Model Change

Based on the set of candidate PMRs, Background Knowledge $\mathcal{C}$ can be adapted. Details on this adaptation process are presented in Sec. V. As the result of this model adaptation, a changed domain model $\mathcal{C}^*$ is returned. For this model, the overall quality has to be re-evaluated - this time, accumulating over all the previous steps of time. If the model quality has improved, the changed domain model is accepted and used as new Background Knowledge. In addition, the timed PMR list is updated based on the new domain model, removing e.g. those representatives which are no longer poorly mapped. If the new model did not lead to an improved overall quality (e.g., due to being overly complex), it will be discarded. In addition, the candidate PMRs are marked as being processed (using the set ${\mathcal{P}_\mathcal{R}}^x$).

After this step, the current model quality will be re-evaluated, and if it is still below the threshold Φ_Q, the adaptation loop will be re-executed. If not, the current time t_n is marked as the time of last model change, and the current model quality $Q(\mathcal{R}_{t_n}, \mathcal{C}_{t_n})$ is used as a3 new base value for triggering model adaptation.

V. Acquiring Concept Definitions from PMRs

The goal of PMR management is to allow the OOWM to adapt its Background Knowledge to newly observed, relevant domain entities. Therefore, PMR management keeps track of the representatives that cannot be mapped to any of the concepts contained in Background Knowledge. When an adaptation of the model seems necessary, indicated by a decreasing quality, PMR management selects the PMRs most promising for model improvement and provides these

candidate PMRs to a *concept acquisition* component. The tasks of concept acquisition in adaptive knowledge modeling are presented in the following and an approach for implementing these tasks based on methods for mixture model reduction is proposed.

A. General Model Adaptation

The general course of action for model adaptation in adaptive world modeling has been presented in [3]. Both main kinds of adaptation, i.e., acquiring and adding a new concept definition to the model as well as extending an existing concept, can be handled by a *general approach*. In this general approach, the probability distribution representing a single new or adapted concept attribute is obtained by combining all given positive samples, i.e., the values of all candidate PMRs for this attribute, into a single distribution – this is done for each given attribute. In case of creating a new concept, all positive samples are treated with equal importance. For extending an existing concept, the attributes of candidate PMRs have to be weighted against the attribute of the existing concept. This can be done by assigning each PMR attribute a weight of 1, while the concept attribute is weighted with the number of representative attributes originally employed for defining the concept.

Combining attribute distributions can be performed as the normalized weighted sum over all the distributions to be combined. For *discrete attributes*, the mean $\sum w_i \cdot p^d_{A_i}(a)$ can be calculated over the distributions $p^d_{A_i}(a)$ of all candidate PMRs in $\mathcal{P}_\mathcal{R}$ and, when being the case, the concept C_e to be extended. Beyond averaging the attribute distributions, generalization steps can be adequate, depending on the type of quality being represented by the attribute. For discrete attributes, generalization should only be applied if it is possible to define some underlying structure for the attribute domain like neighborhood relations. For ordinally scaled attributes generalization steps like performing a kind of morphological dilation operation (in analogy to image processing) might be beneficial for increasing the generalization ability of the acquired description. A kind of morphological erosion operation, removing small components from a discrete distribution, might even be beneficial for nominally scaled attributes in some cases, e.g., for increasing the attribute specificity (and thus reducing model complexity) in case of noisy measurements. Figure 2 illustrates the combination and erosion operations. Specific generalization operations have to be defined for each quality when applying the OOWM to a given domain. Such specific operations are subject to further elaboration in future work.

B. Generalization by Mixture Reduction

For combining the *continuous attributes* A^c_i belonging to the candidate PMRs in $\mathcal{P}_\mathcal{R}$ (or to C_e), a mixture distribution can be calculated. This will result in creating a Gaussian mixture model in the OOWM. For creating more general

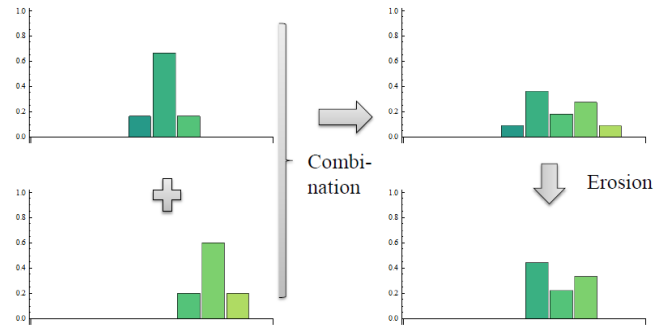


Figure 2. Combination of two discrete attribute distributions and erosion operation on resulting distribution.

concept distributions, a generalization can be performed. Since the attribute distributions of a given set of candidate PMRs represent the results of an uncertain measurement process for similar entities, it can be assumed that, to a good part, the considered distributions describe similar values (e.g., distributions that are partly overlapping). For generalizing from such observation distributions (as well as for arriving at a computationally more efficient representation), approaches to reducing the components of the combined mixture model can be applied. For *Gaussian mixture reduction*, many different approaches exists in literature, including [10], [11] or [12]. In [10], West defines a basic algorithm for merging mixture components into a single Gaussian distribution, retaining lower order moments. Runnalls [11] proposed the idea of employing the Kullback-Leibler (*KL*) divergence as a measure for selecting the components to be merged, based on the resulting approximation quality of the reduction. This approach is further elaborated by [12], allowing the specification of an error threshold as abort criterion for component reduction instead of specifying the desired number of resulting components.

For reducing the number of continuous attribute distributions during *OOWM concept acquisition*, a combined approach was chosen, using the basic algorithm from West and the merging approach proposed by Runnalls. For selecting the components to be merged in each step, the Mahalanobis distance is used (similar to [12]), ensuring that close by components (which represent similar values) get merged first. In this way, clusters of similar values are formed prior to merging these clusters into larger ones during later steps.

As a result of this approach, the approximation error of each merging step, as measured by the KL divergence, increases steadily. For controlling the reduction process, the KL divergence between the original mixture and the reduced mixture is employed. If an appropriate threshold is chosen as abort criterion, this approach allows to abort the reduction process at the time when different clusters are about to being merged. This situation can occur when observed entities, though belonging to the same class, are

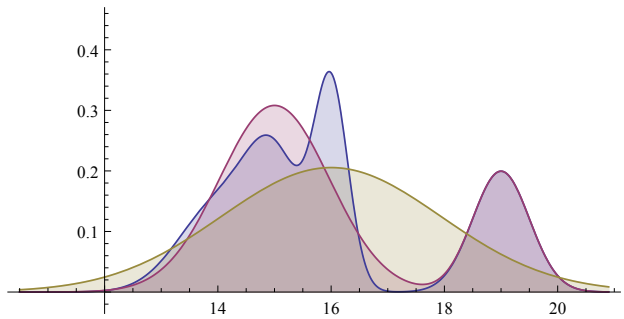
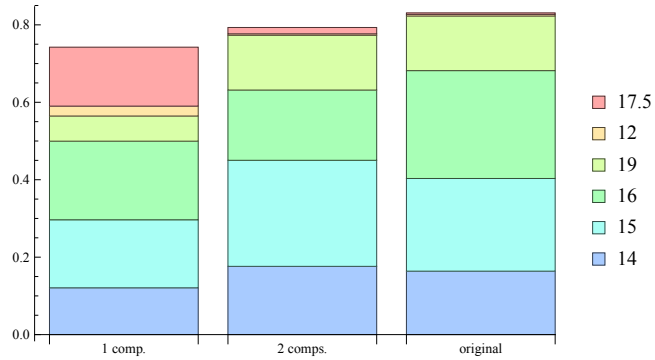


Figure 3. Gaussian mixture reduction for a mixture of 4 components with means at 14, 15, 16 and 18 (cm). Resulting reduction to a single Gaussian distribution (yellow) and reduction aborted according the approximation error threshold (purple), resulting in 2 components.

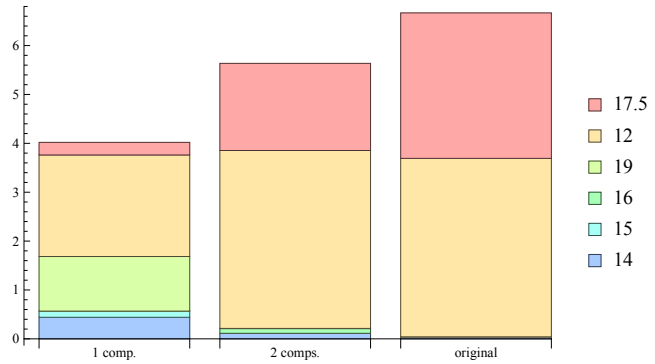
defined by discontinuous typical ranges of attribute values, like the size of different kinds of spoons, cup, etc. In such cases, generalizing beyond these intervals would not well reflect the actual circumstances, but lead to a kind of over-generalization – which should be prevented by aborting.

The approach presented above is able to prevent such over-generalizations, given the appropriate thresholds. Since the operations performed during mixture reduction are mainly linear, the approach is scale invariant. This means that a change in the units of measurement used for representing the attribute distributions does not influence the approximation error measured by KL divergence – a fact which significantly simplifies the task of choosing appropriate threshold values. Still, exactly defining when the merging of two clusters will result into over-generalization remains specific to the type of attribute being processed. So, differing thresholds might have to be chosen for different attributes. As a possible improvement, approaches similar to the morphological operations described for discrete attributes could be performed on the initial Gaussian mixture for further automating the reduction process – e.g., a combination of dilation and erosion operations like closing.

Figure 3 illustrates the concept acquisition approach for a single continuous attribute using a one-dimensional Gaussian mixture. The mixture initially contains four components, representing learning samples e.g. extracted from given candidate PMRs. Two reduction results are displayed: a reduction to a single Gaussian distribution and a reduction aborted prior to cluster merging, resulting in two separate components. As a proof of concept and utility for the presented concept acquisition approach, the example mixture distributions displayed in Fig. 3 have been used as sample concept attributes. Based on these distributions, association probability and attribute correspondence values have been computed for different sample representative attributes, including the four components of the initial mixture. The resulting values are displayed in Figure 4. As can be seen, the proposed approach performs as expected, e.g., by generating



(a) Association probabilities.



(b) Attribute correspondences.

Figure 4. Association probabilities (a) and attribute correspondences (b) for each of the four components of the original Gaussian mixture, used as representatives (blue to green), to the two different mixture reduction results consisting of 1 or 2 components, respectively, as well as to the original mixture, used as concept distributions. Additionally, the values for two Gaussian distributions with means at 12 cm (outside the original mixture, yellow) and at 17.5 cm (in between the two peaks of the original mixture, red) are shown as a proof of concept for the desired generalization abilities of the different reduction results, e.g., containing or excluding the in-between distribution.

higher association values for the initial components than for the close-by but outlying sample, or by including the in-between sample into the one-component concept distribution and excluding it from the two-component distribution.

VI. Proof of Concept and Initial Evaluation

As proof of concept and initial evaluation, the presented PMR management approach was implemented and integrated into an OOWM system demonstrator. A simulated scenario based on a kitchen scene was used for evaluation, extending the scenario defined as in [3]. In this scenario, different fruits serve as domain entities. As attributes, a discrete color attribute and a continuous length attribute are used for defining concepts and entity observations (cf. [3]).

Initially, the Background Knowledge consists of the concept set $\mathcal{C} = \{Apple, Banana, Coconut\}$. Over 11 steps of discrete time, domain entities are observed, starting at $t_n = 2$: three apples (at $t_n \in \{2, 4, 10\}$), a pear ($t_n = 3$), a

banana ($t_n = 8$), a coconut ($t_n = 6$), a strawberry ($t_n = 9$) and three oranges ($t_n \in \{5, 7, 11\}$). The resulting model quality $Q(\mathcal{R}, \mathcal{C})$ (2) is depicted as the blue line in Fig. 5, with $t_n = 1$ depicting the pure model complexity. Significant increases in model quality can be seen at the time steps $t_n \in \{3, 5, 7, 9, 11\}$. For the same time steps, PMRs have been identified using the association-based PMR measure, as depicted by the purple dots denoting binary detection results.

At time step $t_n = 11$, the threshold Φ_Q for model quality is reached, causing all identified PMRs to be ranked and the first orange ($t_n = 5$) to be chosen as PMR candidate. As the result of PMR clustering, a set containing all three oranges is passed to model adaptation, which generates and adds a new orange concept to Background Knowledge in $t_n = 12$. This re-improves model quality as desired. A more complex and realistic scenario for evaluating the proposed PMR management approach is subject to future work.

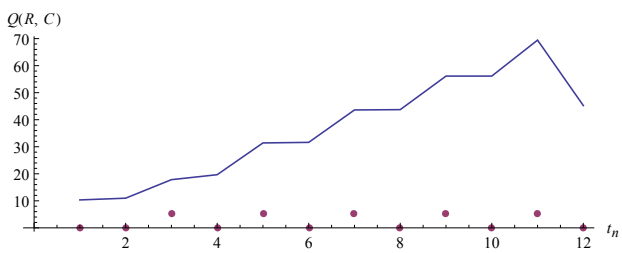


Figure 5. Evolution of model quality in a sample scenario containing unforeseen entities. In the last time step, a new concept has been learned.

VII. Conclusion

In this contribution, the current work on adaptive knowledge management for Object-Oriented World Modeling was presented, focusing on the management and handling of domain entities that are only poorly represented by the current domain model used in a world modeling systems. Based on different measures for rating model goodness, i.e., the ability to represent entities observed in an application domain, as well as on measures for identifying poorly represented entities, a holistic approach was proposed and detailed, which uses such entities as training examples for dynamically adapting the domain model. Besides an algorithm describing the holistic approach to managing unforeseen entity observations and controlling the model adaptation, details were presented on how to realize the acquisition of new concepts, i.e., entity class descriptions, using specifically adapted methods for Gaussian mixture reduction. Future work on adaptive world modeling contains the following tasks:

- Evaluating alternative mechanisms for triggering model changes based on events rather than on model quality
- Defining additional domain model management operations, e.g., the deletion of no longer required concepts for reducing model complexity
- Examining if and how PMR management and concept acquisition can be formulated as a combined optimization problem allowing for a more integrated quantitative approach to model adaptation
- Extending the design of Background Knowledge from set-based to hierarchical domain models.

References

[1] A. Belkin, A. Kuwertz, Y. Fischer, and J. Beyerer, "World Modeling for Autonomous Systems," in *Innovative Information Systems Modelling Techniques*, C. Kalloniatis, Ed. InTech, 2012.

[2] A. Kuwertz and J. Beyerer, "Knowledge Model Quantitative Evaluation for Adaptive World Modeling," in *Proceedings of the IEEE Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA 2013), 2013*, San Diego, USA, Feb. 2013.

[3] ——, "Quantitative Measures for Adaptive Object-Oriented World Modeling," in *Proceedings of 4th Workshop on Dynamics of Knowledge and Belief at the 36th Annual German Conference on Artificial Intelligence (KI-2013)*, Koblenz, Germany, Sep. 2013, pp. 89–104.

[4] T. Belpaeme, J. Looveren, and L. Steels, "The Construction and Acquisition of Visual Categories," in *Learning Robots*, ser. Lecture Notes in Computer Science, A. Birk and J. Demiris, Eds. Springer Berlin Heidelberg, 1998, vol. 1545, pp. 1–12.

[5] M. Daoutis, "Knowledge Based Perceptual Anchoring," *KI - Künstliche Intelligenz*, vol. 27, no. 2, pp. 179–182, 2013.

[6] D. H. Fisher, "Knowledge Acquisition Via Incremental Conceptual Clustering," *Machine Learning*, vol. 2, no. 2, pp. 139–172, Sep. 1987.

[7] W. Iba and P. Langley, "Unsupervised Learning of Probabilistic Concept Hierarchies," in *Machine Learning and Its Applications, Advanced Lectures*. Springer-Verlag, 2001, pp. 39–70.

[8] K. Pedersen, J. K. Allen, and F. Mistree, "Numerical Taxonomy: A Systematic Approach to Identifying Potential Product Platforms," in *Proceedings on the International Conference on Engineering Design (ICED'01)*. WDK Publications, 2001, pp. 515–522.

[9] P. D. Grünwald, *The Minimum Description Length Principle*, ser. MIT Press Books. The MIT Press, 2007, vol. 1.

[10] M. West, "Approximating Posterior Distributions by Mixtures," *Journal of the Royal Statistical Society: Series B*, vol. 55, no. 2, pp. 409–422, 1993.

[11] A. R. Runnalls, "A Kullback-Leibler approach to Gaussian mixture reduction," *IEEE Trans. Aerosp. Electron. Syst*, pp. 989–999, 2007.

[12] M. F. Huber and U. D. Hanebeck, "Progressive Gaussian Mixture Reduction," in *Proceedings of the 11th International Conference on Information Fusion (Fusion 2008)*, Cologne, Germany, Jul. 2008, pp. 1–8.