Scene Interpretation as a Configuration Task

Lothar Hotz, Bernd Neumann

Technical Report: FBI-HH-B-262/05

Universität Hamburg, Fachbereich Informatik,
Arbeitsbereich Kognitive Systeme, HITeC e.V.

März, 2005

Zusammenfassung

Bisherige Forschung zeigte, dass wissensbasierte Szeneninterpretation und wissensbasierte Konfigurierung als logische Modellkonstruktion angesehen werden kann. In diesem Bericht zeigen wir, dass auch aus der Anwendungssicht beide Aufgaben ähnlich sind und bestehende Konfigurierungstechnologien benutzt werden können, um generische Szeneninterpretationssysteme zu implementieren. Nutzbringende Merkmale solcher Systeme sind ausdrucksstarke Wissensrepräsentation, flexible Kontrolle, geleitete Erzeugung von Hypothesen und Constraint-Verwaltung. Wir beschreiben, wie eine Videoaufnahme einer Tischdeckszene mithilfe des Konfigurierungssystems KONWERK, welches Teil unseres Szeneninterpretationssystems SCENIC ist, interpretiert werden kann.

Scene Interpretation as a Configuration Task

Lothar Hotz and Bernd Neumann

Abstract

From past research it is known that both knowledge-based scene interpretation and knowledge-based configuration can be conceived as logical model construction. In this report we show that also from an application-oriented point of view, both tasks are very similar and existing configuration technology can be used to implement a generic scene interpretation system with highly useful features, in particular expressive knowledge representation, flexible control, knowledge-guided hypothesis generation and constraint management. We describe an experiment where a table-laying scene-in-progress is interpreted using the configuration system KONWERK as part of our scene interpretation system SCENIC.

1. Introduction

This paper is about knowledge-based interpretation of real-life dynamic scenes. Typical example tasks are traffic scene interpretation, soccer team-play analysis, criminal-act recognition, or understanding indoor activities, such as table-laying, in a smart-room environment. The goals of scene interpretation go beyond single-object recognition as several objects may contribute to the meaning of a scene, and common-sense knowledge about meaningful occurrences and purposeful behaviour of agents may play a part.

Scene interpretations typically involve inferred facts, expectations and predictions. Consider the example of a table-laying scene observed by a smart-room camera. As plates, cutlery and other objects are placed on the table, it is natural to come up with an interpretation such as "the table is laid for a dinner-for-two" before the table is completely laid and in spite of partial occlusions. In fact, given context knowledge in terms of daytime and dinner habits, the interpretation could be inferred almost without visual evidence, for example by the clatter of dishes. In general, one can say that scene interpretations consist of educated guesses or hypotheses rather than deductions, or, as Max Clowes (1971) has put it, scene interpretations are "controlled hallucinations". It is the purpose of this paper to shed light on this space of feasible hallucinations and on a particular way, inspired by configuration technology, to determine a scene interpretation.

Reiter and Mackworth were the first to analyse the space of possible interpretations in a formal knowledge-representation framework. They showed that scene interpretation is formally equivalent to logical model construction [Reiter & Mackworth 87]. Roughly, viewed as model construction, an interpretation can be seen as an instantiation of a conceptual knowledge base consistent with evidence, i.e. with information about the scene delivered by sensors and low-level image analysis. It is well known that, in general, evidence about a scene may permit multiple scene interpretations. The important insight of this formalisation is that, in a knowledge-based framework, the space of possible interpretations can be narrowed down by logical consistency rather than relying solely on cost functions or preference measures as, for example, in probabilistic approaches [Rimey 93].

On the other hand, as pointed out in [Neumann & Weiss 03], the space of consistent interpretations may still be huge (note that we use "interpretation" both for a logical model and for the corresponding scene description in terms of instantiated concepts). A scene interpretation may contain arbitrary propositions, for example about objects outside the field of view, as long as they do not contradict axiomatic knowledge and evidence. Hence further criteria are required to narrow down the interpretation space and select a "best" interpretation.

Nevertheless, a system which allows to construct interpretations consistent with a conceptual knowledge base and with concrete evidence may provide a useful framework for scene interpretation. This led us to examine existing model-construction systems for possible use in scene interpretation tasks. Description Logics (DLs) were investigated in [Neumann & Möller 04], in particular the DL system RACER [Haarslev & Möller 01]. It turned out that the model-construction procedure of RACER which is at the heart of consistency checking, could not be used for generating "possible interpretations" as it is optimised to prove or disprove the existence of models, but not to generate task-dependent models. However, the RACER query language provides powerful retrieval mechanisms [Haarslev et al. 04], which can be used for constructing scene interpretations, along with other inference processes offered by RACER.

In this paper we examine configuration technology for a possible employment for scene interpretation. Configuration systems have been developed in support of tasks where parts (usually technical components) have to be configured to form a system, which meets given specifications. A typical configuration task is to configure a computer according to customer wishes. It may seem far-fetched to look at technical configuration tasks in connection with real-life scene interpretation, but it has been shown [Buchheit et al. 95] that the logics of configuration are equivalent to model construction and hence essentially the same as the logics of scene interpretation. Furthermore, configuration technology is well understood after two decades of research and development, and there exist many implemented configuration systems.

In the folllowing section we take a closer look at model construction, which is the common logical basis for configuration and scene interpretation. We then show correspondences and differences between configuration and scene interpretation tasks, and propose how a configuration process can in principle be used for scene imterpretation.

In Section 3 we describe the concrete scene interpretation system SCENIC which has been implemented using the configuration system KONWERK [Günter 95], and present a concrete interpretation experiment to demonstrate the potential of the approach.

We conclude that the object-oriented knowledge-representation facilities of the configuration system KONWERK, in particular its constraint system, provide a very useful basis for conceptual modelling and flexible interpretation strategies.

2. Conceptual Framework for Scene Interpretation and Configuration

In this section we first present the rationale for modelling scene interpretation as logical model construction. We then show that configuration tasks have basically the same structure, and how scene interpretation can be modelled as a configuration process.

The work of Reiter and Mackworth mentioned above shows that, under certain assumptions, scene interpretation can be formulated as a finite model construction task and implemented as constraint satisfaction. Model construction essentially applies to the symbolic processing after primitive symbols (representing evidence) have been determined by low-level image analysis. In general, to construct a logical model means to construct a mapping from constant symbols and predicates of a symbolic language into the corresponding entities of a domain such that all predicates become true. In scene interpretation tasks, the domain is usually the real world, the constant symbols denote scene elements, objects and higher-level entities determined by a vision system, and the predicates express class membership and relations for such entities. Part of the mapping is determined by low-level scene analysis, which connects symbols to real-world scene entities, the remaining part is constructed in terms of hypotheses about the scene and represented by the corresponding symbols as place-holders.

The finiteness assumption underlying the analysis of Reiter and Mackworth is unrealistic for real-world tasks, and scene interpretation must be defined as a *partial* model construction if the knowledge representation language permits infinite models [Schröder 99]. Interpretations in terms of partial models are also natural for focussed tasks, such as avoiding a moving obstacle or answering a query, where complete model construction is not required. Hence, in general, it is appropriate to describe the logical basis of scene interpretation as partial model construction. Accordingly, from a logical point of view, scene interpretation is the construction of a symbolic description consistent with conceptual knowledge about the world and concrete knowledge about the scene, the latter consisting of sensor-based evidence and context information.

The conceptual knowledge for scene interpretation is commonly modelled in terms of taxonomical and compositional hierarchies. It has been shown in [Neumann and Weiss 03] that constructing a scene interpretation is essentially a search problem which can be viewed as "navigating" in the space of possible interpretations defined by the taxonomical and compositional relations and by incrementally instantiating concepts while maintaining consistency. Four kinds of interpretation steps suffice to construct any scene interpretation consistent with conceptual knowledge, evidence and context:

- aggregate instantiation (moving up a compositional hierarchy)
- aggregate expansion (moving down a compositional hierarchy)
- instance specialisation (moving down a taxonomical hierarchy)
- instance merging (unifying instances obtained separately)

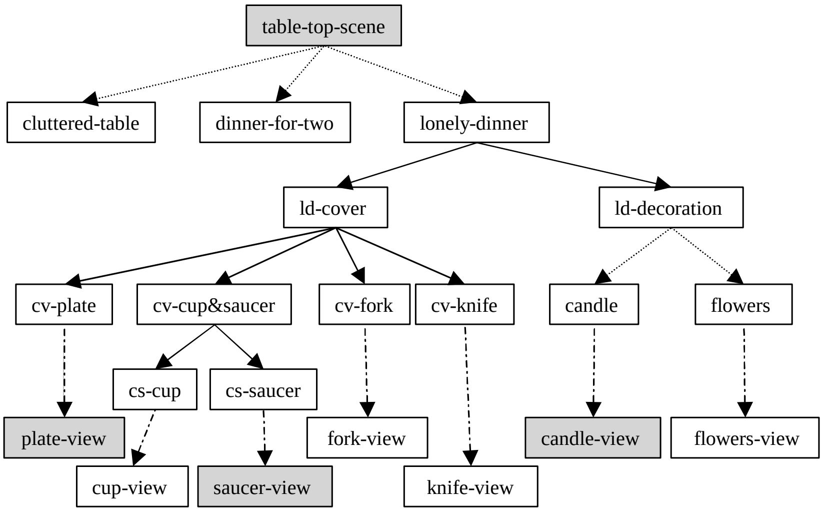

Fig. 1: Illustration of knowledge-based scene interpretation.
Solid edges denote has-part relations, dotted edges has-specialisation relations, dot-dashed edges connect evidence to scene objects. Shaded boxes indicate concepts, which are initially instantiated in example task.

As a concrete example, consider scene interpretation based on the (extremely simplified) knowledge structure shown in Fig. 1. Note that evidence (such as a plate-view), although connected by special "has-evidence" edges, is considered as part of the corresponding scene objects and treated accordingly by the interpretation steps. Prior knowledge of a table-top scene and initial evidence in terms of a plate-view, a saucer-view, and a candle-view are assumed to be given and marked as instantiated concepts (shaded boxes). Starting with evidence, aggregate-instantiation steps may lead to the instantiation of higher-level concepts such as "ld-cover" and "lonely-dinner", an aggregate-expansion step may lead to "decoration" as part of a "lonely-dinner", an instance-specialisation step may generate "candle", and after expansion,"candle-view" may then be merged with the candle evidence, etc.

We now turn to configuration tasks, which also obey the logical model-construction paradigm, as pointed out above. What are the correspondences and differences between scene interpretation and configuration? We restrict our comparison to structure-based configuration, which can be considered the prevailing configuration methodology. Structure-based configuration is also the underlying method for several implemented systems, in particular PLAKON [Cunis et al. 89] and KONWERK [Günter 95], which originated in the research group of the authors.

We can use Fig. 1 again to illustrate configuration. Simply interpret the conceptual structure as a representation of allowed table-top configurations and consider the task of laying the table according to some specific requirements, e.g. "lonely-dinner with candle". In structure-based configuration, such tasks are solved stepwise with essentially the same kinds of steps as used for interpretation. Requirements constitute the initial instantiations, typically including a high-level aggregate and constraints on parts. The final configuration may be reached by a mix of top-down and bottom-up steps.

Comparing the configuration process with scene interpretation in detail, we note several correspondences:

(i) A conceptual knowledge base for scene interpretation uses essentially the same structural relations (aggregation and generalisation) as a conceptual knowledge base for configuration.

(ii) Evidence and context information in interpretation tasks correspond to task requirements in configuration tasks.

(iii) The four kinds of interpretation steps listed above also occur as configuration steps.

However, there are also differences, which will be discussed in the following.

Partiality
It had been argued that a scene interpretation is a partial model in the sense that only a task-dependent subset of all instantiations inferrable from the conceptual knowledge base must be included in an interpretation. Hence, depending on the task on hand, a specific scene may be interpreted in diverse ways, for example including details in one interpretation and omitting details in another. A configuration, on the other hand, typically constitutes a complete model for an aggregate specified by the configuration task.

Non-monotonicity
When interpreting dynamic scenes, one has to deal with time-dependency and changes. For example, moving objects may enter or leave the scene, hence interpretations at one instance may no longer be true at another. On first glance, this seems to correspond to a configuration task with changing requirements, which would be outside the scope of existing configuration technology. However, by relating propositions about a dynamic scene to the time intervals for which the propositions hold, evidence about a dynamic scene need not be withdrawn.

Incrementality
In many applications, it is desirable to perform incremental scene interpretation. This means that a partial model must be constructed, possibly involving predictions, before all evidence is available. In configuration, this would correspond to selecting components among alternatives before receiving all requirements. It is obvious that new requirements may be in conflict with premature configuration decisions and, similarly, new evidence may be in conflict with premature interpretations, hence backtracking is in order in such cases. While in most

configuration tasks incremental requirements can easily be avoided, real-time scene interpretation, and in particular robot vision, must deal with incremental evidence.

Fortunately, existing configuration technology supports the additional requirements posed by incremental processing to a large extent. First, backtracking mechanisms are available to undo decisions which have led into a conflict. Second, configuration systems often offer control mechanisms, which allow to focus on specific parts of the evolving configuration. In real-time scene interpretation, this can be used to focus on interpreting the past before the future. This would require, of course, that the advancing real-time is known to the system and can be exploited for interpretation control.

Uncertainty
Different from typical configuration tasks, scene interpretation usually involves uncertain information of several kinds. For one, it is well-known that evidence provided by sensor signals about physical objects is probabilistic by nature because of many unknown influencing factors. Hence any piece of evidence may be attributed to possibly many objects, causing a potentially large interpretation space from which the most likely interpretation has to be chosen. Similarly, concepts at any representation level may be part of many aggregates at higher levels, from which to choose in a stepwise interpretation process. As shown in detail in [Neumann & Möller 04], such choice points are characteristic for model construction in a formal knowledge-representation framework, and it is highly desirable to provide a preference measure in order to guide local decisions. For the interpretation of natural scenes, statistics about the variability of scenes (or estimates thereof) provide a natural source for probabilistic guidance and, as a rule, such information should be brought to bear.

Another source of uncertainty is the fuzziness of high-level concepts in scene interpretation, for example of spatial relations corresponding to natural-language prepositions such as "behind" or "near". When transforming quantitative results of low-level image-analysis into high-level predicates, it may be useful to represent the grade of applicability of a predicate by a fuzzy value. Epistemically, the degree of applicability is clearly different from a measure of likelihood, so this is a distinct challenge for representation formalisms.

Both kinds of uncertainty are not relevant for typical configuration problems, and configuration systems are not designed to support uncertainty management of this sort. But it is conceivable that the constraint propagation framework of structural configuration systems can also harbour probabilistic inferences, hence configuration technology remains a good candidate for scene interpretation.

3. Interpreting Table-laying Scenes with KONWERK

In this section we describe the experimental system SCENIC (SCENe Interpretation as Configuration) which utilises configuration technology for concrete scene interpretation experiments. The purpose is twofold, to show that the formal correspondences between scene interpretation and configuration described above can in fact be exploited for vision system development, and second, to provide additional detail about knowledge representation requirements and control issues which arise in incremental scene interpretation.

The example scenes are taken from the table-laying scenario mentioned earlier. A camera is installed above the table and observes a table-top. Human agents, sometimes acting in parallel, place dishes and other objects onto the table, for example, covers as customary for a dinner-for-two. It is the task of the scene interpretation system to generate high-level interpretations such as "place-cover" or "lay-dinner-table-for-two". Occurrences of this kind are complex enough to involve several interesting aspects of high-level scene interpretation such as temporally and spatially constrained multiple-object motion, a knowledge base with compositional structure, and the need for mixed bottom-up and top-down interpretation steps.

In the following we first give an overview of the modelling and inferencing techniques provided by the configuration system KONWERK, which performs the symbolic interpretation subtask in SCENIC. In Section 3.2 we describe SCENIC and the knowledge base, which is used for interpreting table-laying scenes. An experiment with this knowledge base is described in Section 3.3.

3.1 Overview of the Configuration System KONWERK

The configuration system KONWERK used for the experiments is a prototypical implementation of a generic configuration system [Günter 95] designed to support the configuration of aggregates based on component descriptions in a knowledge-base. The relevant knowledge is organised in four separate modules:

Concept Hierarchy. Object classes (concepts) are described using a highly expressive object description language, and embedded in a taxonomical hierarchy. Object properties are specified by parameters with restricted value ranges or sets of values. A compositional hierarchy is induced by the special structural relation part-of. Objects selected for a concrete configuration are instantiations of these object classes.

Constraints. Constraints pertaining to properties (parameters) of more than one object are administered by a constraint net. Conceptual constraints are formulated as part of the conceptual knowledge base and instantiated as the corresponding objects are instantiated. Constraints are multi-directional, i.e. propagated regardless of the order in which constraint variables are instantiated. At any given time, the remaining possible values of a constraint variable are given as ranges or value sets.

Task Description. A configuration task is specified in terms of an aggregate which must be configured (the goal) and possibly additional restrictions such as choices of parts, prescribed properties, etc. Typically, the goal is the root node of the compositional hierarchy, as in the example shown in Fig. 1.

Procedural Knowledge. Configuration strategies can be specified in a declarative manner. For example, it is possible to prescribe phases of bottom-up or top-down processing conditioned on certain features of the evolving configuration.

The KONWERK executive system performs stepwise configuration according to the following basic algorithm:

```
Repeat
        Check for goal completion
        Determine current strategy
        Determine possible configuration steps
        Select from agenda and execute one of
              { aggregate instantiation,
                aggregate expansion,
                instance specialisation,
                parameterisation,
                instance merging }
        Propagate constraints
        Check for conflict
```

Comparing with the configuration (and interpretation) steps discussed in Section 2, the KONWERK executive cycle features all the steps mentioned there, but also includes parameterisation as an additional operation. Parameterisation means that a component property such as size or position is specified or constrained. This can be considered as a substep of specialisation, refining the description of a component.

In KONWERK, an aggregate is completely configured if all properties of the aggregate have been parameterised, all its required parts have been completely configured, and all constraints are satisfied. As noted earlier, this completeness requirement is at odds with the notion of scene interpretation as a partial model, where details may be missing or the scope of an interpretation may be limited depending on the task. However, KONWERK offers several means for automatic parameterisation, for example by using default values, which can be used to hide configuration steps not required for a scene interpretation.

A conflict is encountered when the constraint net cannot be satisfied with the current partial configuration. In this case, automatic backtracking occurs. Backtracking can be controlled by procedural knowledge to achieve "intelligent backtracking" and avoid unnecessary repetition of configuration steps.

3.2 The Scene Interpretation System SCENIC

To perform scene interpretation, KONWERK has been combined with image analysis modules as shown in Fig. 2.

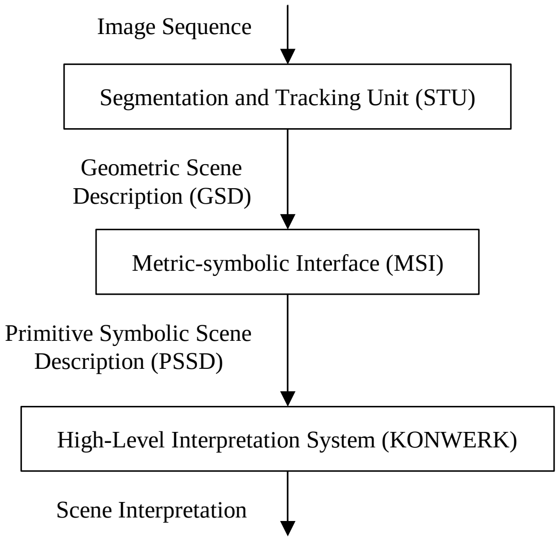
Image Sequence
Segmentation and Tracking Unit (STU)
Geometric Scene Description (GSD)
Metric-symbolic Interface (MSI)
Primitive Symbolic Scene Description (PSSD)
High-Level Interpretation System (KONWERK)
Scene Interpretation

Fig. 2: The scene interpretation system SCENIC consists of three modules: the segmentation and tracking unit STU, the metric-symbolic interface MSI, and the high-level interpretation module realised by the configuration system KONWERK.

The segmentation and tracking unit STU is tailored to meet the needs of our table-laying scenes. Important static objects (such as the table) are segmented manually and entered into the factual knowledge base off-line. Moving objects are detected by comparing successive image frames and by region growing around seed points determined from change areas. The shapes of moving objects are classified into view types for each frame, in our scenario restricted to view-types corresponding to table-top objects such as plate-view, saucer-view, fork-view, etc. Objects are tracked throughout the image sequence, and the successive positions are recorded as object trajectories. The trajectory and view-type data up to a point of time constitute the evolving Geometric Scene Description (GSD), which is the output of the STU. The view type of an object may be ambiguous regarding the correct physical object class, or change along the trajectory, e.g. because of occlusion. To be able to disambiguate such low-level classifications is an important requirement for the high-level interpretation component.

The task of generating symbolic entities from the GSD is performed by the metric-symbolic interface MSI. Following the approach presented in [Neumann 02], symbolic entities are assigned to interesting perceptual primitives for time intervals where a qualitative constancy can be observed. In our scenario, interesting perceptual primitives are location, speed and orientation of moving objects, distances between objects, and angles between reference orientations, as well as temporal derivatives thereof. Interesting qualitative constancies are

- moving / stationary
- increasing / decreasing distance
- increasing / decreasing angle
- disjoint / touching / overlapping / within

The high-level interpretation system realised by the configuration system KONWERK performs interpretations based on a conceptual knowledge base for table-laying scenes, see Fig. 3. The Upper Ontology of the knowledge base - i.e. the domain-independent part - consists of concepts related to real-world scenes and to the evidence obtained by sensory equipment. A real-world scene is composed of subscenes which may be decomposed into further subscenes, thus forming a compositional hierarchy. A subscene concept describes an aggregate of objects (or activities) which constitute a meaningful entity by themselves. In our domain, typical subscenes are `cover` or `laying-a-dinner-for-two` activities.

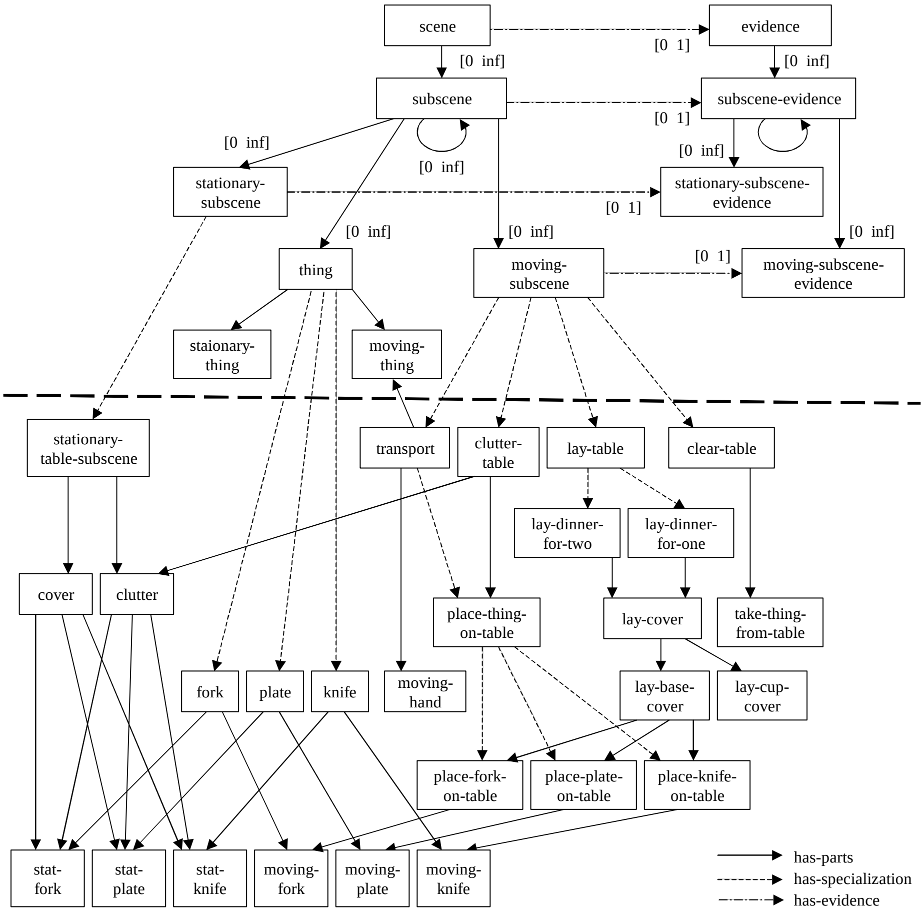

Fig. 3: Structure of conceptual knowledge base of SCENIC. Upper part above dashed line shows domain-independent concepts (Upper Ontology), lower part illustrates some domain-specific concepts of the table-laying scenario (not all relations shown).

The root concept scene represents all possible symbolic scene descriptions. One can think of scene as a concept both for real scene descriptions and for (possibly hypothetical) descriptions to be constructed by the interpretation process. Scene interpretations will always contain an instance of scene, which may therefore be used as a starting point for top-down processing.

The numbers in brackets are bounds on the number of instances of a relation. For example, a specific scene (i.e. an instance of the concept scene) may consist of any number of subscene instances.

Evidence concepts describe the evidence provided by the MSI. In our scenario, evidence is only modelled for primitive objects, and STU and MSI attempt to provide evidence for the behaviour of primitive objects only. In general, however, a knowledge base may also include evidence concepts directly associated with scene and subscene aggregates. This can be useful if an aggregate view is much different from the composition of the individual views of its parts, or evidence pertains to a scene as a whole (e.g. daytime).

In dynamic scenes, object descriptions extend over time, and the corresponding concepts are designed to specify time-dependent properties - such as position - over some time interval. In our knowledge-representation framework, time is represented by discrete time points corresponding to the image frame rate provided by a video camera. It is useful to distinguish between time intervals of motion and non-motion; hence the Upper Ontology includes moving-subscene and stationary-subscene concepts which partition the behaviour of the corresponding parent concept into motion and non-motion subintervals. Consecutive subintervals of an object are related by the temporal relation meets (not shown in the figure).

The domain-specific part of the conceptual knowledge base (below heavy dashed line) describes concepts of our table-laying scenario, in particular composite action descriptions for laying a dinner for a specific number of persons, clearing the table, placing individual items onto the table, etc. There are also concepts describing static configurations such as various kinds of covers. At the most specific level, the domain-specific knowledge base contains primitive object concepts such as moving-plate associated with concepts for corresponding evidence via the relation has-evidence.

The knowledge base also encompasses constraints between objects, e.g. between the components of an aggregate or between a scene object and its associated evidence. Constraints are not shown in Fig. 3, but play a significant role in defining the geometry of table-top concepts. For example, the geometry of the aggregate basic-cover consisting of plate, knife, fork, spoon and table edge, is defined in terms of distance ranges between the bounding-boxes of all objects of the aggregate.

The constraint system of the configuration framework plays an important part as it represents spatial and temporal coherence between the components of an aggregate and allows propagating evidence through the constraint net. It is interesting to note that the constraint net plays a role comparable to a Bayesian Networks in probabilistic scene interpretation (e.g. [Rimey 93]). Constraints on value ranges can be interpreted as flat distributions, and constraint propagation as a special form of belief propagation. It seems feasible to extend the configuration approach by integrating probability distributions instead of constraints. This would also provide the much needed preference measure for the space of logically possible interpretations.

3.3 A Scene-Interpretation Experiment

To demonstrate the effectiveness of the configuration approach for scene interpretation, we have set up SCENIC to interpret an evolving scene where a dinner-for-two is laid by two human agents. Initially, a context is defined by creating instances of `scene` and `table` (in view of the camera). Then, by one agent, a plate and a saucer are laid for the left cover, and simultaneously by another agent, a saucer and a cup are laid for the second cover. The corresponding evidence is supplied to KONWERK in one bulk at frame 300 in terms of tracked regions, classified correctly by the STU as `hand-view, plate-view, etc.` except of the cup which was sometimes mistaken as a saucer and hence classified ambiguously as `dish-view` (the parent concept of `cup-view` and `saucer-view`). In addition, KONWERK receives as input instances of the topological predicate `touch-view,` which is true if two regions touch while having the same motion state.

High-level interpretation begins bottom-up by instantiating the physical objects corresponding to the evidence, including `dish-2` for the ambiguously classified region, and `touch` occurrences for the `touch-view` evidence (see Fig. 4). Furthermore, based on a list of interesting bottom-up predicates, `transport` occurrences are determined by specialising `touch` occurrences involving a `hand`.

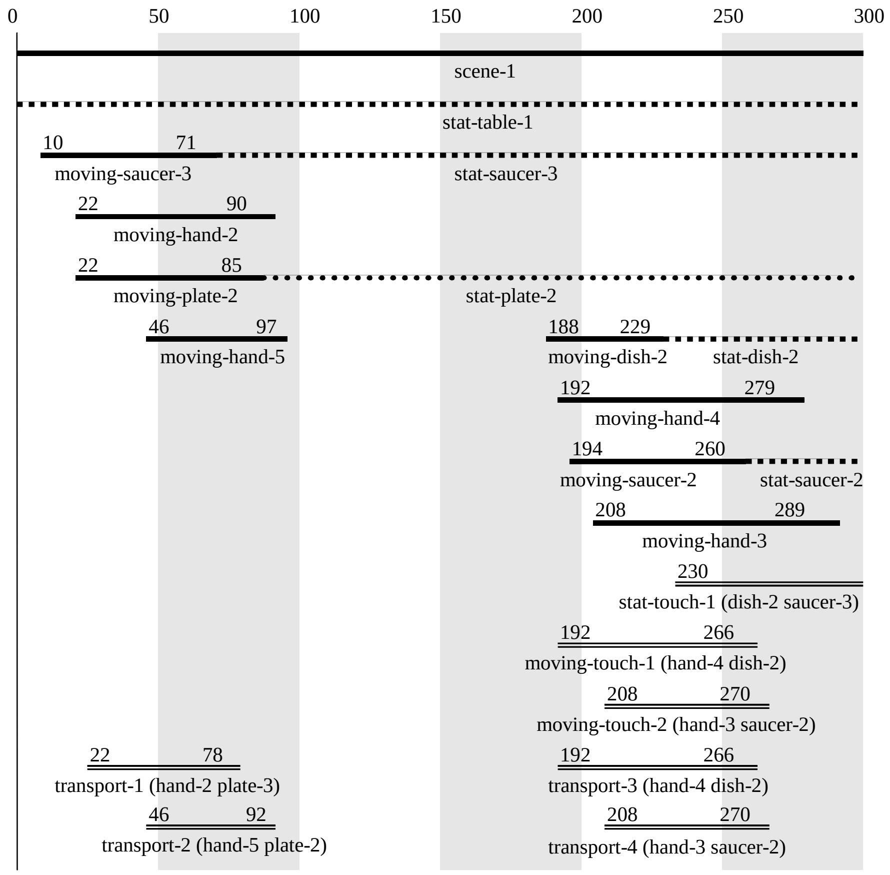

Fig. 4: Instantiated concepts during initial bottom-up phase of scene interpretation. Instances of primitive motion concepts are marked as solid lines, of the corresponding stationary concepts as dotted lines, of aggregates as double lines. Components of aggregate instances are shown in parentheses.

At this point, the control strategy of KONWERK has exhausted its bottom-up repertory and invokes top-down interpretation steps by expanding `scene-1,` which was created as an initial context. One may think of this phase as an exploration of high-level concepts, which might be responsible for the objects and occurrences observed so far. The ensuing specialisation steps require that choices are made, for example between `lay-dinner-for-one` or `lay-dinner-for-two` or `clutter-table`. We have used KONWERKs option to specify preferred values in terms of defaults to first guess `lay-dinner-for-two`, hence two instances of `cover` are hypothesised with locations constrained by `table-1`. Next, both covers are expanded into a `basic-cover` and a `cup-cover` (an aggregate composed of cup and saucer), and continuing top-down hypothesis generation, the left `basic-cover` is expanded into plate, knife, fork, and spoon.

At this point, the plate is immediately merged with `plate-2` generated from the evidence, and the well-defined location of `plate-2` is propagated through the constraint net generating restricted locations for all other hypothesised objects. Similarly, the right `basic-cover` and `cup-cover` are expanded. The saucer component of the `cup-cover` is instantly merged with `saucer-3` and the cup component with `dish-2`, exploiting the high-level knowledge of the cup-cover hypothesis to specialise the dish as a cup. This demonstrates that low-level ambiguities can indeed be resolved by top-down expectations.

The state of interpretation at frame 300 is illustrated in Fig. 5. Objects supported by evidence are shown in natural colours. Hypotheses are shown in artificial colours together with equally coloured boxes delineating possible positions according to current constraints.

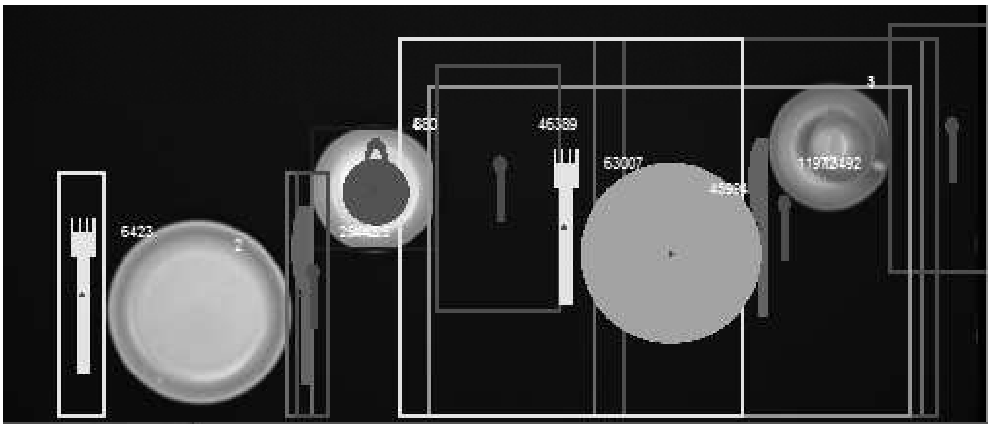

Fig. 5: Intermediate scene interpretation as an instance of `lay-dinner-for-two`. Objects in natural colours are supported by evidence, objects in artificial colours are hypotheses based on high-level conceptual knowledge. Hypotheses are shown at the centre of boxes, which represent possible locations.

It is important to note that this interpretation is not unique. In fact, by setting the default choice of the top-down hypothesis-generation phase to `lay-dinner-for-one`, an alternative interpretation is generated at frame 300, shown in Fig. 6. Here cup and saucer on the right are treated as unconstrained components of a `clutter-table` occurrence. As pointed out in Section 2, model construction allows for all interpretations consistent with conceptual knowledge and evidence.

51 interpretation steps were needed to obtain the first intermediate scene interpretation, using 90 sec of CPU time (1.8 GHz PC). Backtracking and additional 8 interpretation steps were needed to arrive at the alternative intermediate interpretation, using additional 45 sec of CPU time.

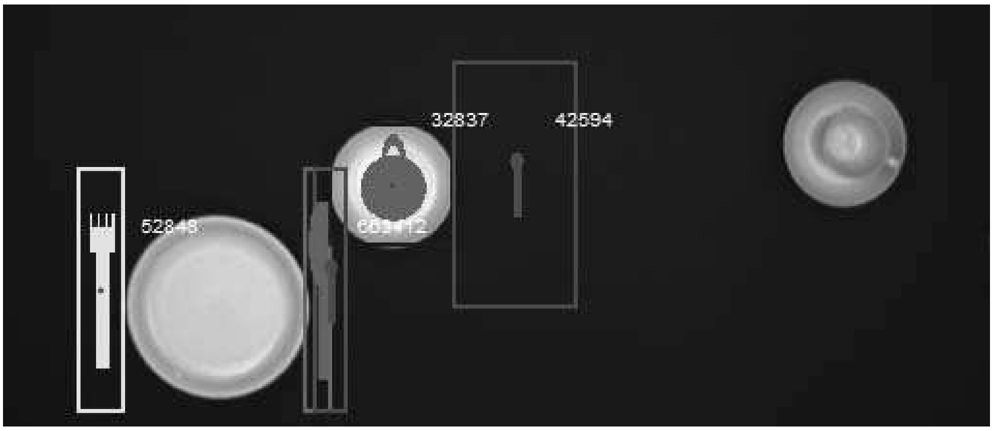

Fig. 6: Alternative intermediate scene interpretation generated for the same scene in terms of instances for `lay-dinner-for-one` and `clutter-table`.

4. Conclusions

Starting from the observation that scene interpretation and configuration are both logical model-construction tasks, we have shown that scene interpretation can in fact be implemented within the framework of a configuration system. Several desirable features of a scene interpretation system can be realised:

- The framework is generic and allows to construct interpretations of a scene which are consistent with conceptual knowledge, evidence and context information.

- By utilising the flexible control facilities of structure-based configuration, a mix of bottom-up and top-down processing is possible which allows to hypothesise high-level aggregates from partial evidence and thus predict the spatial and temporal evolution of a scene.

- High-level knowledge may be brought to bear to resolve ambiguities arising from low-level image analysis or even guide low-level processing.

The configuration approach can be seen as a framework which allows to navigate a (possibly large) space of logically consistent interpretations., but does not provide guidance as to which interpretation is more likely. However, there are well-defined places where guiding knowledge in terms of probability distributions or other preference measures can be introduced without jeopardising logical consistency. This is a topic of ongoing work of the authors.

References

Buchheit, M., Klein, R., Nutt, W.: Constructive Problem Solving: A Model Construction Approach towards Configuration. DFKI Technical Memo TM-95-01, Deutsches Forschungszentrum für Künstliche Intelligenz, Saarbrücken, January 1995.

R. Cunis, R., Günter, A., Syska, I., Peters, H., Bode, H.: PLAKON - An approach to domain-independent construction. In Proc. 2. IEA/AIE, Tennesse, USA, ACM-Press, 1989, 866-874.

Günter, A.: KONWERK - ein modulares Konfigurierungswerkzeug. In F. Maurer, M.M. Richter (eds.), Expertensysteme '95, infix Verlag, St. Augustin, 1995, 1-18.

Haarslev, V., Möller, R., Wessel, M.: Querying the Semantic Web with Racer + nRQL. In Proc. KI-04 Workshop on Applications of Description Logics (ADL-2004), Ulm, Germany, September 20-21, 2004, available online as CEUR Workshop Proceedings 115

Haarslev, V., Möller, R.: RACER System Description. In Proc. Int. Joint Conf. on Automated Reasoning (IJCAR 2001), LNAI Vol. 2083, Springer, 2001, 701-705.

Neumann, B.: A Conceptual Framework for High-Level Vision. FBI-HH-B245/02, FB Informatik, Universität Hamburg, 2002.

Neumann, B., Möller, R.: On Scene Interpretation with Description Logics. FBI-B-257/04, Fachbereich Informatik, Universität Hamburg, 2004. To be published in Cognitive Vision Systems, H.-H. Nagel and H. Christensen, eds., Springer.

Neumann, B., Weiss, T.: Navigating through logic-based scene models for high-level scene interpretations. In Proc. 3rd Int. Conf. on Computer Vision Systems (ICVS-2003), Springer, 2003, 212-222.

Reiter, R., Mackworth, A.: The Logic of Depiction. TR 87-23, Dept. Computer Science, Univ. of British Columbia, Vancouver, Canada, 1987.

Rimey, R.D.: Control of Selective Perception Using Bayes Nets and Decision Theory. Dissertation, Univ. of Rochester, TR 468, 1993

Schröder, C., Möller, R., Lutz, C.: A Partial Logical Reconstruction of PLAKON/KONWERK. In Workshop Reports KI'96, Dresden, September 1996, also published in DFKI-Memo D-96-04, Proc. Workshop on Knowledge Representation and Configuration WRKP'96, 55-64.

Schröder, C.: Bildinterpretation durch Modellkonstruktion: Eine Theorie zur rechnergestützten Analyse von Bildern. Dissertation (in German), DISKI 196, infix, 1999.