Knowledge Modelling and Management

Part B (5)

Yun-Heh Chen-Burger

http://www.aiai.ed.ac.uk/~jessicac/project/KMM

Knowledge Model

A CommonKADS Approach

An Overview for the Knowledge Model

- A knowledge model includes three types of knowledge (knowledge category):
  - Domain knowledge (inc. Domain Schema and Knowledge Base);
  - Inference knowledge;
  - Task knowledge.
- Typically a KM includes the following items:
  - A diagram of the full domain schema: e.g. UML class diagram, Ontology, ER data model – domain model.
  - An inference-structure diagram.
  - A list of knowledge roles.
  - Textual and graphical specifications of the tasks and task methods.
- Compare with other paradigm:
  - OO methods, e.g. UML class diagram
  - Process modelling methods, e.g. IDEF3, OWL-S, BPEL4WS, BPML.
  - ER methods, e.g. Relational diagram

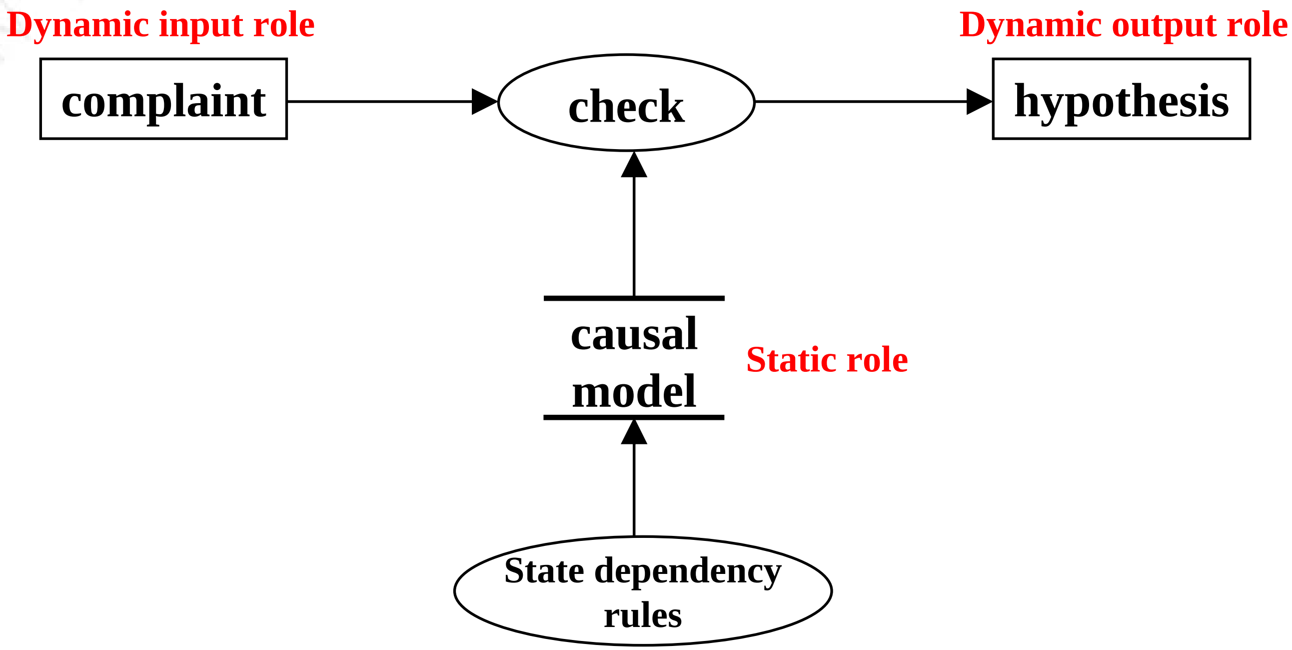
Inference and Knowledge Roles

- **Knowledge role: describes the functional role of data on which the inference operates and describes;**
- **Inference I/Os are described in terms of functional roles - abstract names of data objects that indicate their role in the reasoning process.**
- **For example, a typical knowledge role is a "hypothesis": it plays the role of a "candidate solution" in an abductive inference.**

Knowledge roles in CommonKADS

- Two types of knowledge roles:
  - Dynamic role: data objects that are instantiated at run-time – they are the input and output of an inference:
    - » Dynamic input role:
      - E.g. for the inference task "check", "Complaint" plays an dynamic input role;
    - » Dynamic output role:
      - in the same example, "Hypothesis" plays an dynamic output role in "check“;
  - Static role: data objects that are stable over time:
    - » Knowledge base
      - Rules
        - E.g. state-dependency rules, V&V rules, inference rules.
      - Facts
        - E.g. concepts, properties, constraints and relationships.

Example Knowledge Roles



Creating a Knowledge Model

A Step-by-Step Guide

Knowledge model construction
A task-oriented approach - 1

- **Stage I: Knowledge identification:**
  - A lexicon/glossary of domain terms;
  - Survey of existing components (task template, domain schemas, component reuse);
  - Source:
    - » Knowledge items in the organisation model, characterization of knowledge items and tasks
    - » ontology, entity relational data model, UML class diagrams UML activity diagram, business process model.

Knowledge model construction A task-oriented approach - 2

- **Stage II: Knowledge specification:**
  - Choose a task (template);
  - Construct a domain schema with representative instances;
  - Two approaches:
    - » Inference → domain and task knowledge (middle-out);
    - » Domain and task knowledge → inference (middle-in).
- **Stage III: Knowledge refinement:**
  - Populate knowledge schema;
  - Validation by scenarios: paper-based, small prototype.
- **Iterative feedback loops.**

Stage I: Knowledge Identification

- **Select a <u>knowledge intensive task</u> and main knowledge items related to this task.**
- **This task may be (already) classified in the task library: e.g. assessment, configuration, monitoring, etc, to assist reuse.**
- **Goal: survey knowledge items and prepare them for knowledge model specification in stage II.**
- **Two aspects of tasks:**
  - **Data related aspects: explore and structure information sources for the task - create a lexicon or glossary for terms used – domain familiarization.**
  - **Task related aspects: study, check and revise tasks in more details; list potential reusable knowledge-model components.**

Data Related Aspects – 1

Domain familiarization

- Identify the source of expertise: books, manuals, single expert, multi-experts, any conflicting sources?
- Identify Key words, texts and concepts: text marking; hold (informal, un-structured) interviews to get an overview of the domain;
- Challenge: find a balance between learning about the domain <u>without becoming a full domain expert.</u>

Data Related Aspects – 2
Domain familiarization

- Talk to knowledge user – to get key features of problem solving process.
- Avoid too much details – focus on the type or function of a knowledge item that may play in your system.
- Construct typical scenarios – to understand the domain and may be used for verification and validation.
- Recommended time: maximum: 2 person-weeks.
- Results of domain familiarization stage:
  - List of domain knowledge sources and their characterizations;
  - List of summary of selected key texts;
  - Descriptions of scenarios;
  - Most importantly - your understanding of the domain.

Task-related Aspects:
List potential model components - 1

- **Goal:** to enable the reuse of model components.
  - Similar principles are used in SE (software pattern reuse), and KE's ontology reuse.
- **Task dimension:**
  - Establish task types for the application tasks;
  - Select and reuse the correct task templates [1+] from the CommonKADS library:
    - » E.g. Analytic task types:
      - classification, diagnosis, assessment, monitoring, prediction.
    - » E.g. Synthetic task types:
      - design, configuration of design, assignment and matching, planning, scheduling, modelling.

Task-related Aspects:
List potential model components - 2

- **Domain dimension:**
  - **Establish the types of the domain:**
    - » **Is this a technical domain?**
    - » **Where can one find resources?**
    - » **Is the knowledge mainly heuristics-based?**
  - **Any standardized descriptions?**
    - » **E.g. Art and Architecture Thesaurus (AAT), Medical subject headings (MeSH), product model libraries, etc.**

Task-related Aspects:
List potential model components - 3

- **Pointers:**
  - Regarding mapping between application tasks and generic task types.
    - » This may be difficult sometimes, as application tasks are often more complex – therefore it is often a combination of several generic (smaller) task types from the task library.
  - Names given to application tasks do not necessarily map to generic task-type names:
    - » E.g. the travelling planning task may not necessarily be a planning task in a KS.
- **Results:**
  - List of potential task templates (inference processes);
  - Mapping between application tasks and generic tasks.

Stage II: Knowledge Specification

- **Goal:** to get a complete specification of the knowledge model – inc. some example knowledge instances.
- **Three main activities:**
  1. Choose a task (template) – to start with;
     - So it is goal-oriented;
  2. Construct an initial domain schema – for this task;
  3. Specify the three knowledge categories within this application domain:
     - » Domain knowledge;
     - » Inference knowledge;
     - » Task knowledge.

1. Choose a task template: reusing past design patterns

- Prefer a task template that has been used before – empirical evidence.
  - A task template is a partial generic knowledge model where inference and task knowledge are specified.
  - CommonKADS provides a catalogue of such task templates.
- When a match between an application and a generic task is found – annotate the links between them [1+].
- When no template is found – questions whether it is a "reasoning" task.
- A bad template is better than no template.

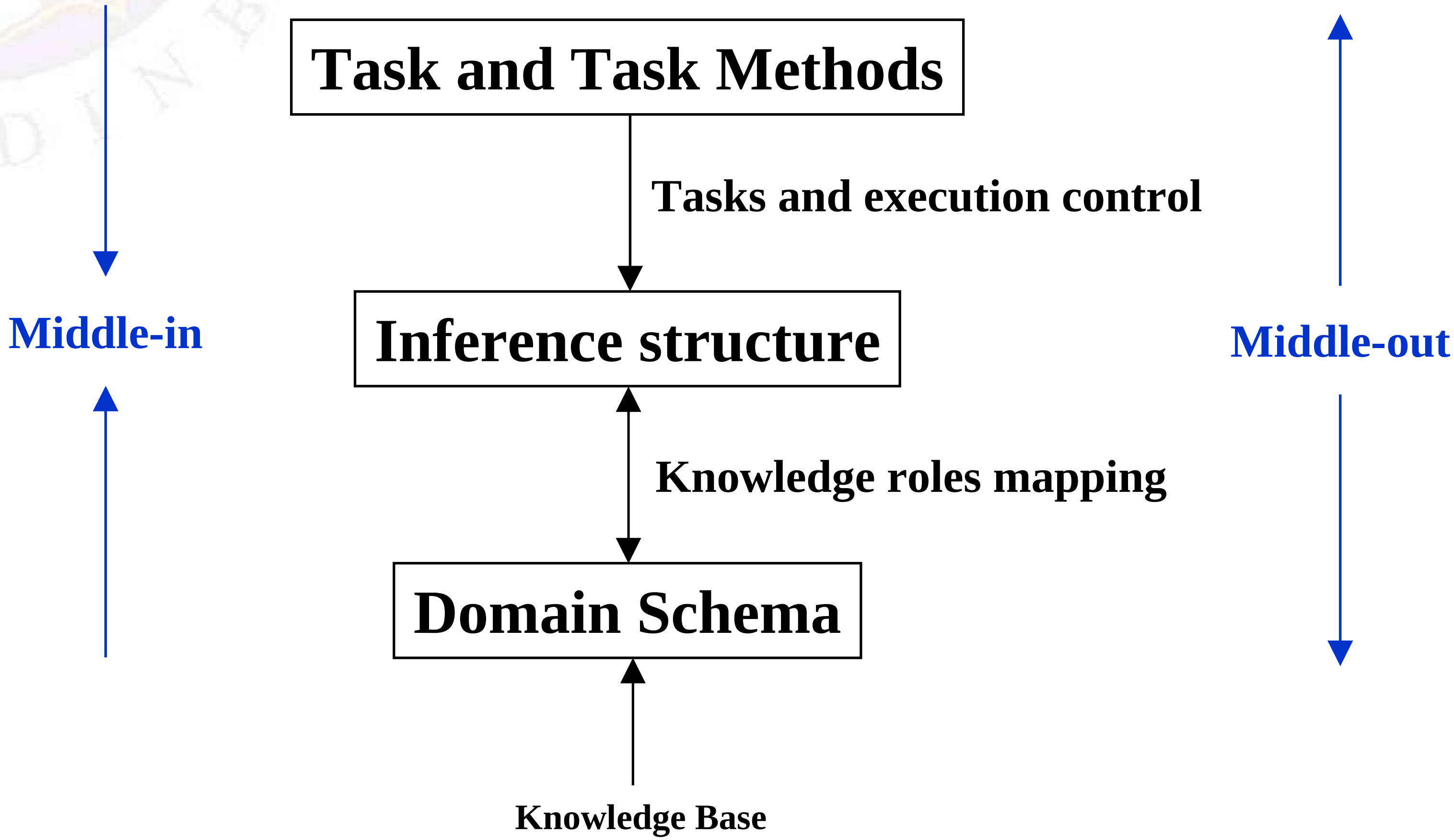
2. Construct an initial domain schema

- **Two types of domain knowledge included:**
  - **Domain-specification concepts: data types.**
  - **Method-specific concepts: e.g. axioms/rules.**
- **Guidelines:**
  - **Adhere to terminologies in existing data model - helps communication between experts, software systems and personnel.**
  - **Limit use of notations, e.g. concepts (classes), subtypes and relations.**
  - **Construct the domain-specific data model independently from task model – so to ensure it be free from implementation considerations.**

3. Complete specification of the knowledge model

- **Given a task template and an initial domain schema.**
- **Mapping approaches:**
  - **Middle-out: work from inference knowledge outward - to complete and match with domain and task knowledge (model); the approach is preferred, but it requires the task template chosen provide a task decomposition that is detailed enough to act as a good approximation of the inference structure.**
  - **Middle-in: work from task and domain knowledge (model) inward – by decomposing tasks through consecutive applications of methods as well as refining the domain knowledge - so that the two ends meet through matching with components in the inference structure.**
- **Similar terms used in the ontology community for concept identification and ontology construction and not to be confused with:**
  - **Middle-out, Button-up, Top-down.**

Mapping approaches of Middle-out and Middle-in



Tasks in CommonKADS

- **Two main task knowledge-types:**
  - **Task;**
  - **Task method.**
- **Tasks may be decomposed into sub-functions.**
- **A sub-function may be one of following types:**
  - **Another task;**
  - **Inference – leaf node;**
  - **Transfer function – leaf node;**
- **The method used for decomposing a task is described in a "task method" – including "control structure" of sub-functions (execution orders).**
- **Composite and Primitive tasks:**
  - **a primitive task has (still) one layer of decomposition, i.e. it may be further decomposed into several leaf nodes;**
  - **this definition is different from other major process-oriented methods, e.g. UML 2.0, IDEF3, PSL, OWL-S 1.1 (Atomic).**
- **Page 112– 114 [1].**

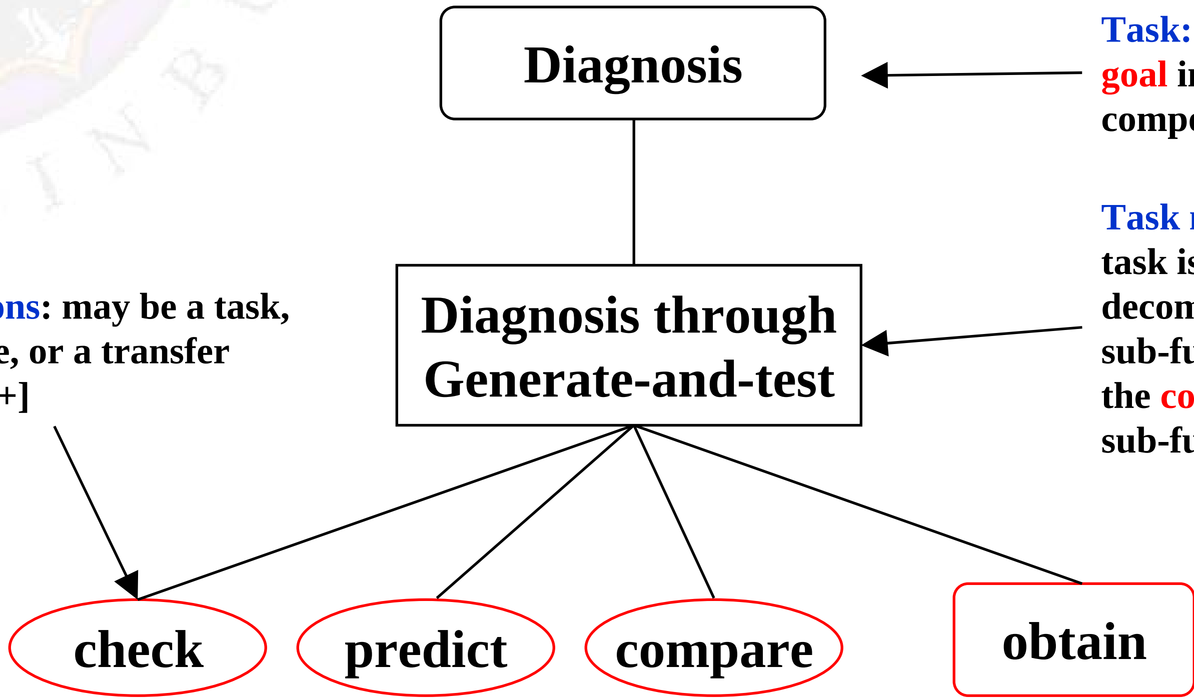
Car fault diagnosis example

Task and task methods from CommonKADS

Task: defining the **reasoning goal** in terms of I/O. This is a composite task.

Task method: describes **how** a task is realised through decomposition into sub-functions. It describes the **control structure** of sub-functions.

Sub-functions: may be a task, an inference, or a transfer function. [1+]



Transfer function

Task

Task method

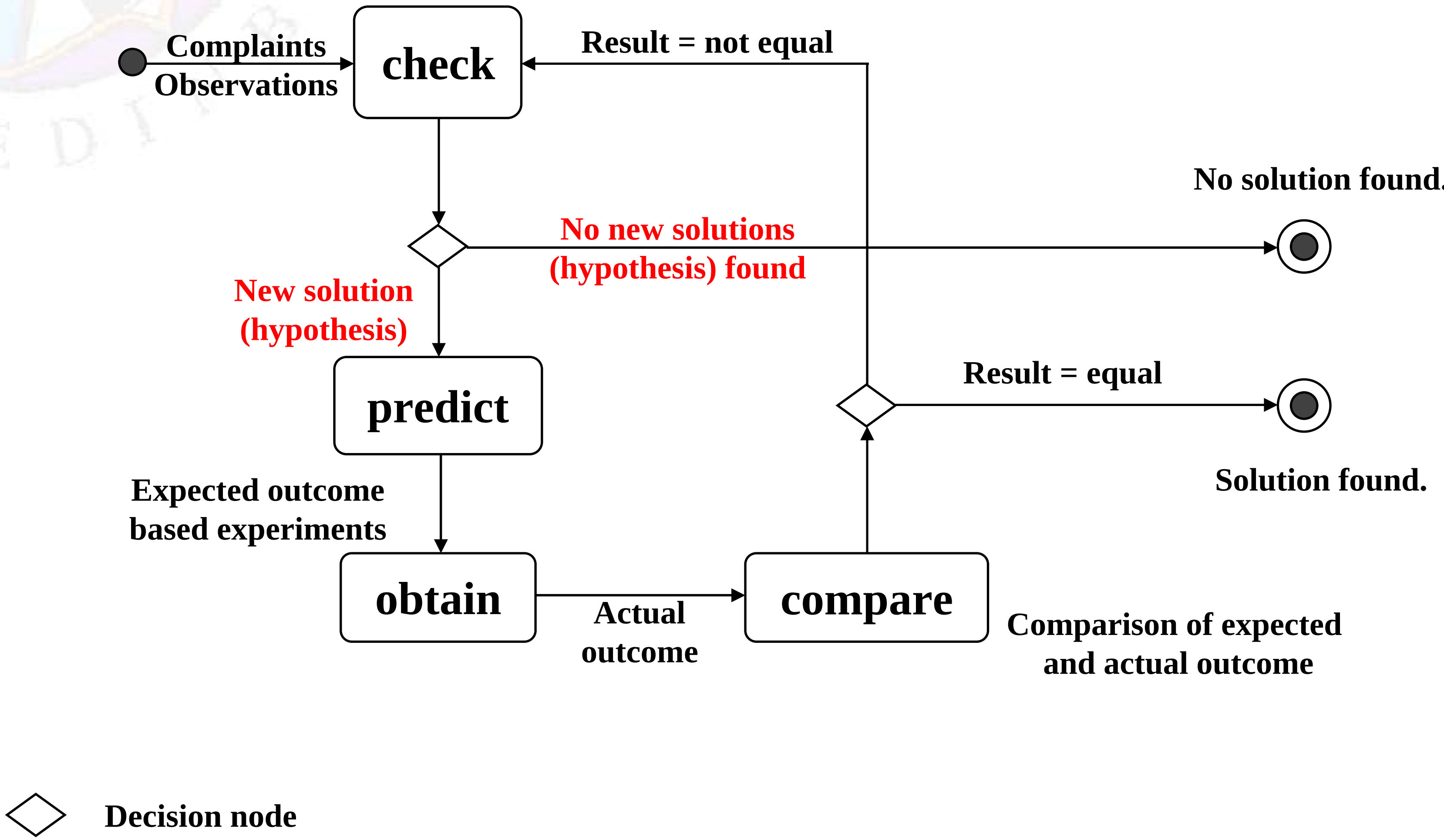
Inference

- **Sub-functions** can be another **task**, an **inference**, or a **transfer function**.
- **Inferences and transfer functions are Leaf Node functions.**
- **Transfer (I/O) functions are obtain (upon request), receive, present, provide (upon request).**
- **The sub-function "check" is originally called "cover" in [1].**

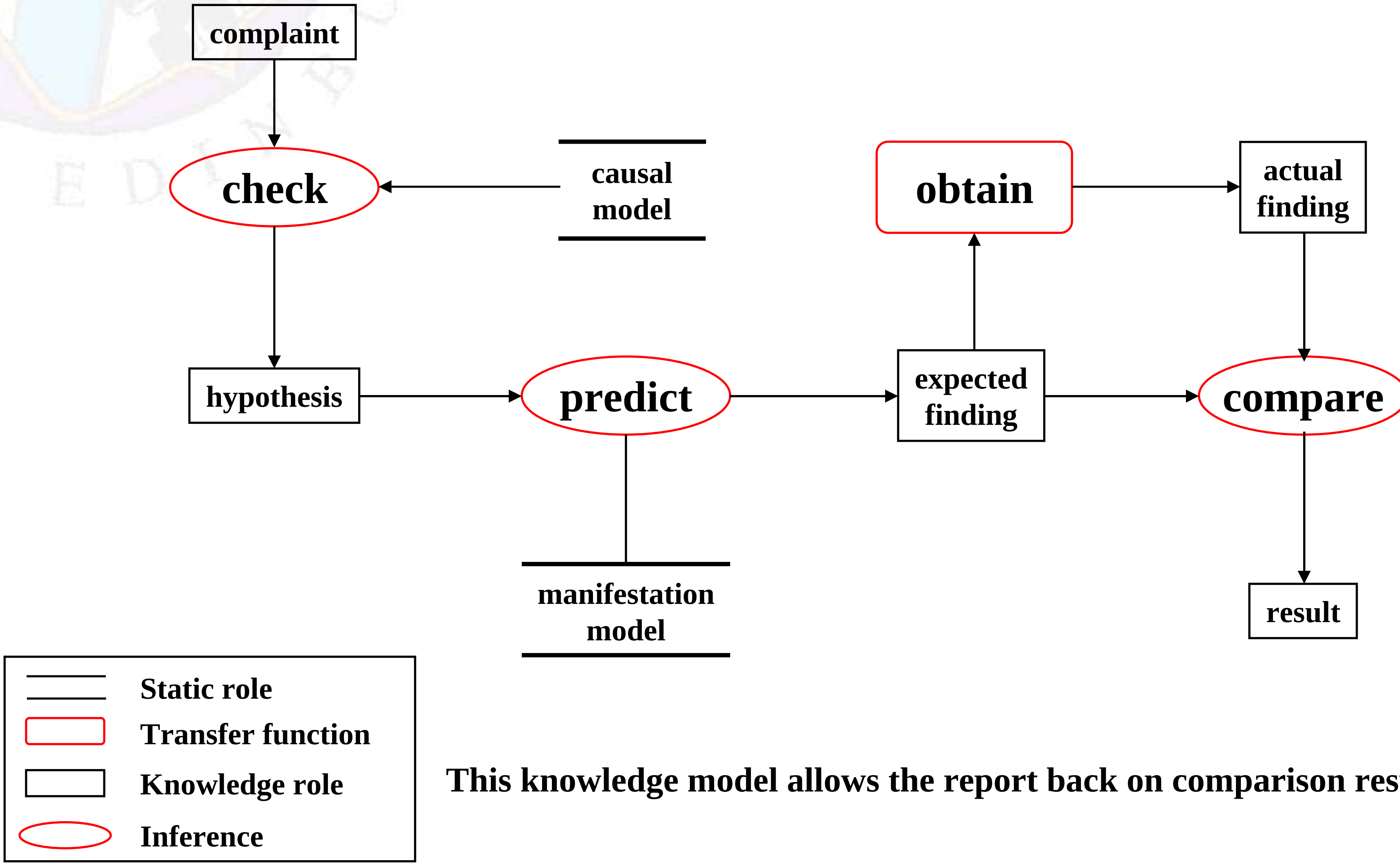
Car fault diagnosis example

UML Activity Diagram

Start diagnosis Through Generate-and-test



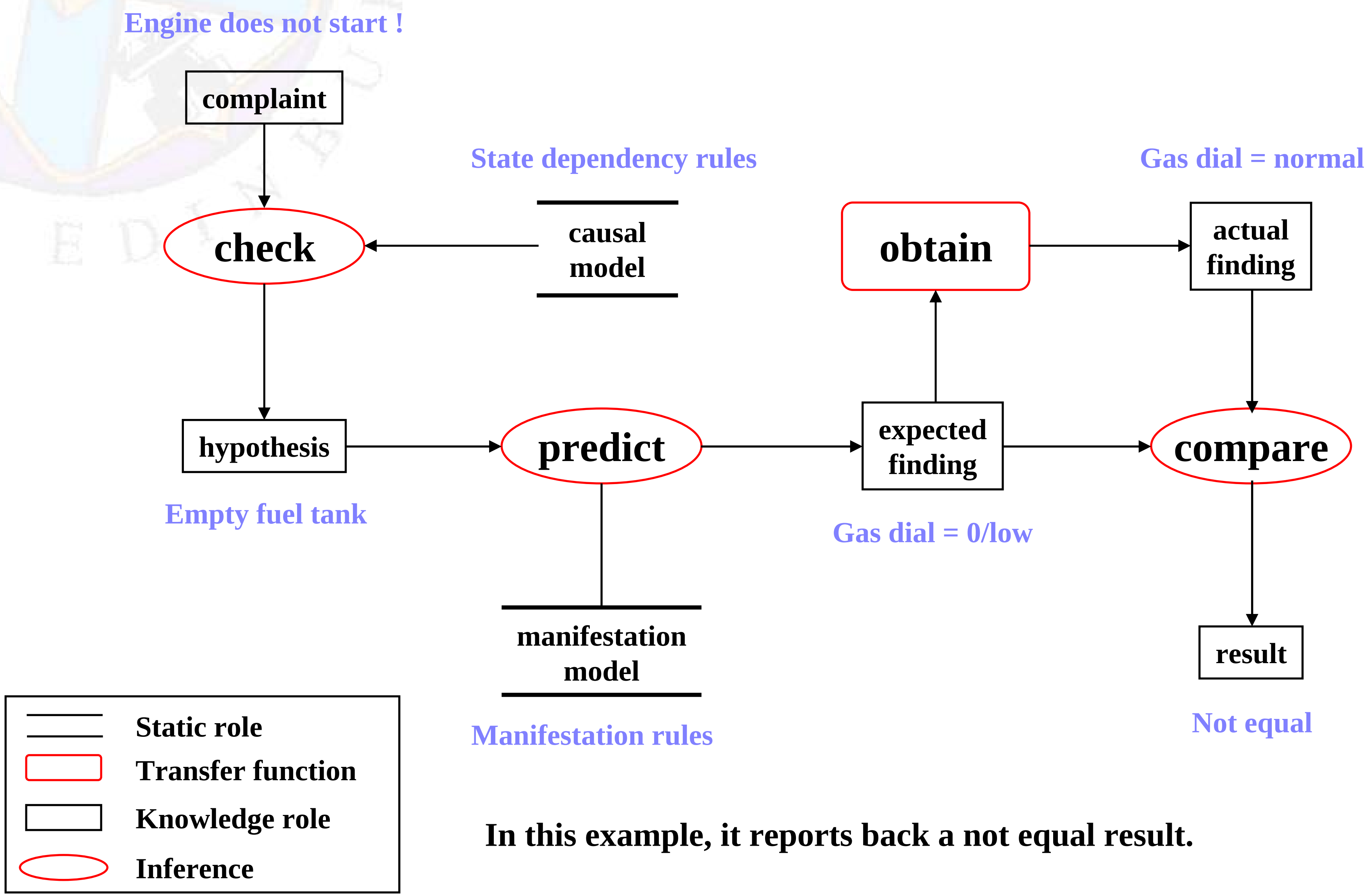
Car fault diagnosis example
CommonKADS inference structure



This knowledge model allows the report back on comparison results.

Car fault diagnosis example

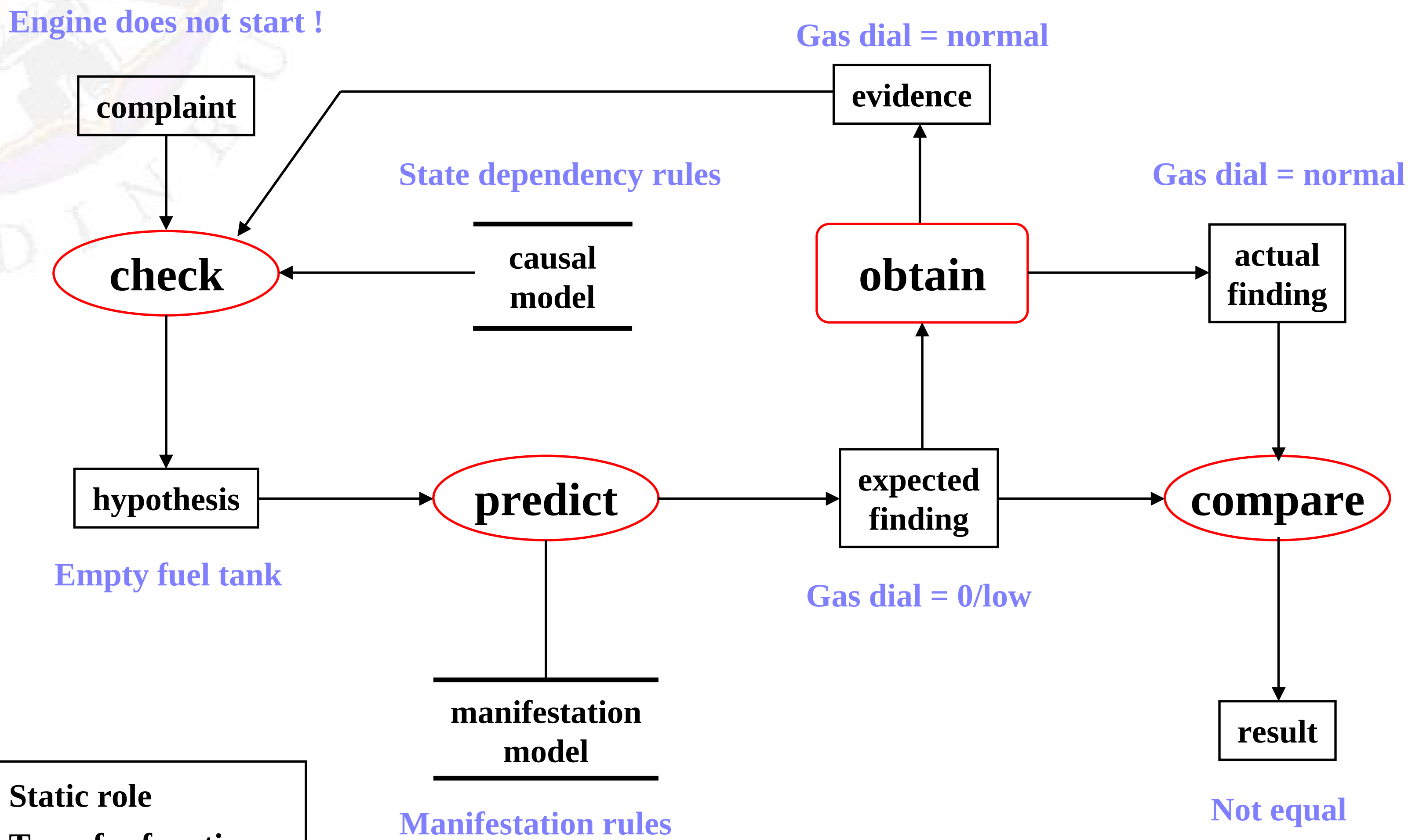
CommonKADS inference structure

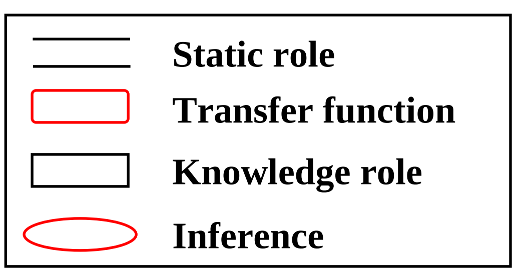


In this example, it reports back a not equal result.

Car fault diagnosis example

CommonKADS inference structure



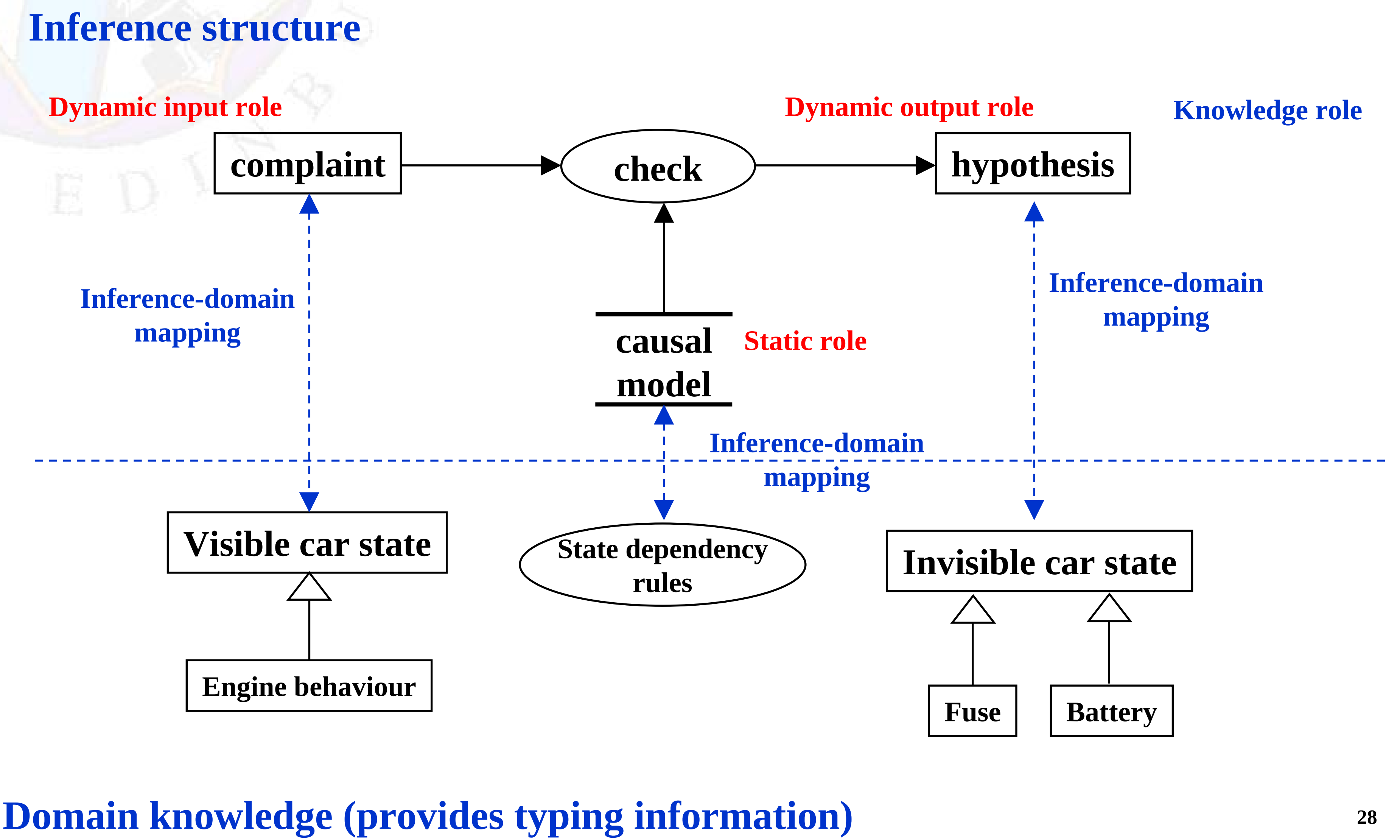


This knowledge model also allows <u>feedback of comparison results</u> so the dynamic knowledge of the case is changed.

Discussion

- Knowledge model provides an understanding of the use and role of knowledge in the reasoning process.
- It does not necessarily provide the procedural logic of a knowledge system, although control flow information may be derived from it.
- The knowledge model is not intended to provide an overview of all of the control flow
- Whether there is a branching control flow in a knowledge system it may depend on the knowledge available to the system, e.g.
  - Having the knowledge of a rule "If a=T then b=T", this may indicate a particular action for the system;
  - However, if another related rule is discovered later on, e.g. if a = F then b = F, then additional actions may be needed, thus change the control flow of the system.
  - In a way, we treat knowledge as data that may be updated and expanded; while at the same time, it may affect the control flow of our system.
- An UML activity diagram is intended to define the system control flow as clearly as possible, so that it facility programming tasks and communication of its programming logic to its users/ other developers, etc.

Mapping knowledge roles to domain knowledge (types)



Mapping tasks to the inference structure

- Note how all leaf node sub-functions are presented in the inference structure – and with the appropriate types:
  - Check;
  - Predict;
  - Obtain;
  - Compare.
- Analysis:
  - Hypothesis and solutions are not necessarily always the same.
  - In addition, the task “obtain” may be further decomposed into two sub-functions “Give testing instructions” and “Obtain results”.

Knowledge Model Worksheet KM-1

Knowledge Model	Full knowledge-model specification in text plus selected figures.
Information sources used	List all the information sources about the application domain that were consulted. Produced during the identification stage.
Glossary	List of terms with definitions, may be with hyperlinks.
Components considered	List of potentially reusable components (task templates, ontologies, knowledge bases) and rationale of (not) reused.
Scenarios	List of problem-solving scenarios.
Validation results	Descriptions of paper or computer-based (prototyping) validation results.
Elicitation material	Including source material, e.g. interview transcripts.

Stage III: Knowledge Refinement

- Two activities are carried out:
  1. **Validate** the knowledge model, usually using a simulation technique.
  2. **Complete** the knowledge base by adding domain-knowledge instances.

1. Validate knowledge model

- **Verification – is the model right?**
  - Checking consistency:
    - » Automatic and Manual;
  - Completeness/coverage of application domain;
  - Syntactic checking based on methods used (tool support);
  - Semantics checking across different models - e.g. using an ontological-based checking.
- **Validation – is this the right model ?**
  - Manual walk-through of typical scenarios;
  - Automate a simulation for the model behaviours by using (typical) scenarios;
  - This may be used to compare with results from a manual walk-through and/or expected behaviours of the system – chapter 12 gives more details.

Typical V&V Questions

- How well does the model fit our application domain and towards achieving project goals?
- Is there any difference exhibit between the models and the scenario? Are they prescribed in such a way that is on purpose?
- Should the model be adapted?
- If so, where should the model be adapted? Give justifications.

2. Complete the knowledge base

- **Knowledge instances derived from examples are normally not included.**
- **Knowledge instances derived from transcripts of interviews, protocols, etc, are included – called knowledge types.**
- **A knowledge base is unlikely to be complete at first, and needs to be maintained throughout its life time – the cold-start syndrome.**
- **Alternatively, automation techniques may be used to learn/infer new instances for the knowledge base.**
- **(In fact, some techniques have now been used to automatically construct part of domain models, e.g. self-learning ontology and information extraction techniques, in research).**

Summary and Overview

- Main aspects include in a knowledge model:
  - Domain knowledge (knowledge items) in the application area: what are them, where do they come from? Described in an UML class diagram, ontology, ER data model – conceptual models. Includes types, rules, example knowledge instances (in scenarios) as well as knowledge base itself.
  - Task knowledge: may be described using CommonKADS task model, UML activity model, process model – stay at schema level.
  - Inference knowledge: describe in CommonKADS inference structure, including tasks, data dependency and knowledge roles.
  - The mapping between them – towards an integration.
- Verification and Validation of the KM
- Output: KM-1 (Knowledge Model worksheet 1)

Main Reference

- [1] (section 6.1, Chapter 7) in Knowledge Engineering and Management: The CommonKADS Methodology. Guus Schreiber, Robert de Hoog, Hans Akkermans, Anjo Anjewierden, Nigel Shadbolt, Walter Van de Velde. http://www.amazon.co.uk/exec/obidos/ASIN/0262193000/qid=1091803195/sr=1-1/ref=sr_1_2_1/026-4023131-7023627.

Additional Reference

- [1+] (chapter 5 and 6) in Knowledge Engineering and Management: The CommonKADS Methodology. Guus Schreiber, Robert de Hoog, Hans Akkermans, Anjo Anjewierden, Nigel Shadbolt, Walter Van de Velde. http://www.amazon.co.uk/exec/obidos/ASIN/0262193000/qid=1091803195/sr=1-1/ref=sr_1_2_1/026-4023131-7023627.