



Knowledge Engineering for Hybrid Intelligence

Ilaria Tiddi

Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
i.tiddi@vu.nl

Stefan Schlobach

Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
k.s.schlobach@vu.nl

Victor de Boer

Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
v.de.boer@vu.nl

André Meyer-Vitali

DFKI
Saarbrücken, Germany
andre.meyer-vitali@dfki.de

ABSTRACT

Hybrid Intelligence (HI) is a rapidly growing field aiming at creating collaborative systems where humans and intelligent machines cooperate in mixed teams towards shared goals. A clear characterization of the tasks and knowledge exchanged by the agents in HI applications is still missing, hampering both standardization and reuse when designing new HI systems. Knowledge Engineering (KE) methods have been used to solve such issue through the formalization of tasks and roles in knowledge-intensive processes. We investigate whether KE methods can be applied to HI scenarios, and specifically whether common, reusable elements such as knowledge roles, tasks and subtasks can be identified in contexts where symbolic, subsymbolic and human-in-the-loop components are involved. We first adapt the well-known CommonKADS methodology to HI, and then use it to analyze several HI projects and identify common tasks. The results are (i) a high-level ontology of HI knowledge roles, (ii) a set of novel, HI-specific tasks and (iii) an open repository to store scenarios¹ – allowing reuse, validation and design of existing and new HI applications.

CCS CONCEPTS

• **Computing methodologies** → **Ontology engineering**; • **Social and professional topics** → *Systems analysis and design*.

KEYWORDS

Knowledge Engineering, Hybrid Intelligence, CommonKADS

ACM Reference Format:

Ilaria Tiddi, Victor de Boer, Stefan Schlobach, and André Meyer-Vitali. 2023. Knowledge Engineering for Hybrid Intelligence. In *Knowledge Capture Conference 2023 (K-CAP '23)*, December 05–07, 2023, Pensacola, FL, USA. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3587259.3627541>

¹Materials at <https://github.com/kmitd/HI-CommonKADS>



This work is licensed under a Creative Commons Attribution International 4.0 License.

K-CAP '23, December 05–07, 2023, Pensacola, FL, USA

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0141-2/23/12.

<https://doi.org/10.1145/3587259.3627541>

1 INTRODUCTION

The field of Hybrid Intelligence (HI) envisions of creating collaborative systems, where humans and intelligent machines operate in mixed teams collaboratively, synergistically, and proactively to achieve shared goals [1]. In a typical HI scenario, humans and artificial agents complement each other's limitations. For example, human limitations such as stereotypes, error proneness, in-group favoritism or short memory are mitigated by machine-driven decision-making, while the human feedback ensures the machines' fairness, sample efficiency and task generalizability. This team collaboration requires advanced interaction of heterogeneous actors (i.e. humans and artificial agents), which are aware of and can adapt to each others' tasks and knowledge within the organization they operate. HI being in its infancy, a clear characterization of such a complex interaction in terms of the knowledge and the tasks the mixed teams are involved in is currently missing. This prevents to efficiently develop HI scenarios and, more importantly, to compare and reuse concepts and design patterns across HI applications². In other words, Hybrid Intelligence currently lacks formal, systematic representation of the interaction between a hybrid team of humans and artificial agents, and a method to design both the knowledge and the tasks involved in these interactions is therefore needed.

A well-known method to describe tasks and roles in knowledge-intensive processes as the HI scenarios is using Knowledge Engineering (KE) [21]. Methods such as CommonKADS [20] have been extensively used since the 1980s in order to define the basic organization of a knowledge-based expert system, through modeling and implementing design components and their relationships. The main insight behind CommonKADS was that it should not only be used to develop an expert system supporting organizations in their tasks, but also to engineer the organization's knowledge and modeling requirements. To do this, the CommonKADS methodology defined and used a typology of generic tasks decomposable into primitive inference steps, that engineers could use to design and implement knowledge-based systems. By moving the implementation to the more abstract "knowledge level", CommonKADS ultimately allowed to standardize and reuse task-types, problem solving methods and elementary inference steps across applications [24].

In this paper, we bring forward the idea that Knowledge Engineering methods such as CommonKADS can be used to help us formally describe Hybrid Intelligence applications, i.e. modeling

²Henceforth, "application" and "scenario" are used interchangeably to indicate a hybrid team of humans and artificial agents jointly working to solve a given problem.

the tasks and knowledge types involved in it. This allows us to identify and standardize tasks and knowledge across HI scenarios, support future researchers in designing HI systems by providing them with reusable and interoperable design blocks, and ultimately contribute to better characterizing a rapidly growing field in the AI area. The main questions we investigate therefore are: (1) *can the classical Knowledge Engineering methods help us characterizing modern, Hybrid Intelligence applications?* And, (2) *can we identify common components that are typical to HI scenarios?*

In order to answer our questions, we first adapt the well-known CommonKADS methodology to Hybrid Intelligence, namely by following the *knowledge analysis* steps in order to characterize knowledge roles, tasks and subtasks composing the Knowledge Model of an HI system. We then validate such Knowledge Model by describing and finding common knowledge structures, tasks and inference steps in 7 Hybrid Intelligence scenarios gathered from the community. These include research projects both in the Dutch Hybrid Intelligence consortium³ and similar European initiatives. The results are: (i) a high-level ontology of knowledge roles for HI applications, (ii) a preliminary set of HI-specific tasks and (iii) an open repository where HI applications and their Knowledge Models can be recorded. These can ultimately serve for purposes such as reuse (i.e. designing new HI scenarios), quality control (comparing to existing scenarios) and evaluation (validation of new scenarios).

2 RELATED WORK AND MOTIVATION

Hybrid Intelligence (HI). Hybrid Intelligence is described as the combination of human and machine intelligence, where humans and machines collaborate in mixed teams synergistically, proactively, and with the purpose to achieve shared goals [1]. In HI scenarios, goals that were unreachable by either humans or machines can be achieved through complementing each others' strengths and limitations [1, 4]. Several projects are born since the seminal papers have appeared, e.g. the Dutch Hybrid Intelligence Centre, the European Network for Human-Centred Artificial Intelligence⁴, the Human-Centred AI research group⁵, and events such as the Human-centred AI workshop series⁶ and the first editions of the Hybrid Human-Artificial Intelligence conference (HHAI)⁷.

Some work defined the core concepts around Hybrid Intelligence. A research agenda for HI is presented in [1], which includes the use of psychological theories of cooperation and theory of mind to improve synergetic work (Collaborative HI), reinforcement learning for machines to adapt to humans and their environment (Adaptive HI), societal and personal value-aware methodologies to design systems (Responsible HI), ontologies and knowledge graphs as background knowledge to share and explain awareness, goals and strategies (Explainable HI). [4] attempts to characterize what is HI, starting from the different dimensions of the term "intelligence" (e.g. social, logical, spatial, musical). The authors compare a hybrid intelligence with human, collaborative and artificial intelligence and identify HI as a type of intelligence that intersects the human and the artificial one. Other characterizations are more limited in

scope, e.g. [11] describes the ways humans support AI systems (task design, quality control, etc.), while taxonomies and standardizations for human-machine teams are presented in [5, 13, 15].

Works attempting to formally define specific HI tasks and systems from a Knowledge Engineering perspective are e.g. modular design patterns for hybrid systems combining symbolic and subsymbolic techniques [22, 23], or formal languages used to describe the dynamic allocation of task in hybrid, human-agents teams [25, 26]. In [3], a set of business applications operating in different domains are analyzed to derive a hierarchical set of dimensions relevant to design HI systems: these include the task to perform, the learning actor (human or machine) and the type of interaction (human-to-AI or AI-to-human). We use a similar bottom-up approach to identify knowledge type and tasks in HI scenarios.

Knowledge Engineering for Application Design. Knowledge Engineering (KE) was conceived to turn the construction of knowledge-based systems into an engineering process, following Software Engineering approaches [21]. Different modeling frameworks address various KE aspects, i.e. CommonKADS [20] describes models, MIKE [2] formalizes the executable specification of the model, PRO-TÉGÉ [7] allows collaborative knowledge acquisition and reasoning.

The CommonKADS methodology has been applied in a variety of scenarios, including e-governance [28], smart grid management [29], and robot control [9]. Some works have extended CommonKADS to multi-agent systems scenarios for e.g. supply chain management and traffic simulations [16, 18]. These scenarios have been mostly dealing with single user, stand-alone knowledge-based systems to support the domain expert in complex, but well-defined tasks (e.g. diagnosis, assessment, planning) [27]. With the rise of the Semantic Web and its open-ended tasks (e.g. semantic search, information retrieval, knowledge representation at scale), the KE methodologies evolved into open semantic patterns, e.g. Ontology Design Patterns, boxologies and scripts for informed machine learning-based language-driven applications [6, 23]. Similarly to us, [24] uses KE to describe tasks in Semantic Web applications.

Motivation and Challenges ahead. Hybrid Intelligence is an emerging field and initial efforts have been made in characterizing the type of intelligence involved in these scenarios (complementary, collaborative, adaptive etc.). What is missing is a clear analysis of the knowledge and tasks that the mixed teams need to exchange, in order to support the design of HI systems. Knowledge Engineering methods such as CommonKADS have been used to support the engineers in clarifying the structure of knowledge-intensive systems in the past. CommonKADS' Knowledge Model in particular has helped specifying the data and knowledge structures required for the application. We therefore **hypothesize that KE can help us in the task of describing Hybrid Intelligence scenarios**, i.e. understanding the different knowledge types, engineer the system design and support reusability across applications. The advantage of this method over classical ontology engineering is that it allows us to capture in parallel multiple aspects involved in a HI scenario, namely knowledge about the organization, the agents, their interaction, and tasks they need to perform. Assuming this, three challenges arise: (i) CommonKADS was used mostly for single-user applications, while we deal with complex scenarios with multiple actors and complementary expertise; (ii) CommonKADS has been

³<https://www.hybrid-intelligence-centre.nl/>

⁴<https://www.humane-ai.eu/>

⁵<https://hcai.site/>

⁶<https://sites.google.com/view/hcai-human-centered-ai-neurips/home>

⁷hhai-conference.org

successful with applications using small-scale domain ontologies, but have not been applied in applications relying on large repositories of less structured knowledge (i.e. modern Knowledge Graphs); (iii) CommonKADS dealt with a closed number of well-defined tasks, that might not be easily applied in HI scenarios that involve symbolic, subsymbolic and human-in-the-loop components. Understanding whether Hybrid Intelligence applications can rely on such tasks or different ones is crucial. In order to tackle these challenges, in the next sections we will first adapt the CommonKADS methodology to HI, particularly extending its Knowledge Model with an ontology of knowledge roles for HI applications, and then use it to characterize several HI projects, to ultimately identify a set of common, reusable HI-specific tasks.

3 METHODOLOGY: COMMONKADS FOR HI

3.1 Background on CommonKADS

The Knowledge Acquisition and Documentation Structuring method (KADS) and its successor CommonKADS are well-known approaches in Knowledge Engineering to develop knowledge-based systems [19].

The idea is that developing an intelligent system involves engineering “templates” to describe both the system behavior in the application domain, and the concrete organization behind it. The knowledge engineer would therefore model the domain expert knowledge as well as how this is used within the organization [20].

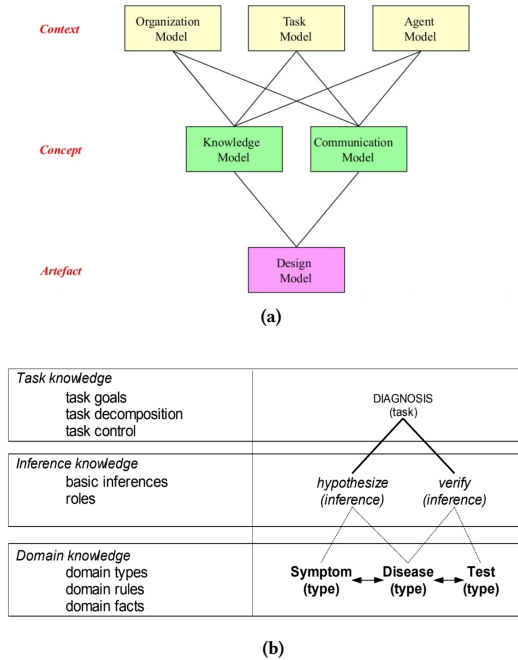


Figure 1: (a) CommonKADS models and (b) a Knowledge Model of a medical application [19].

To represent knowledge at different levels, CommonKADS offers several models serving as requirement specification for a KB-system to be developed (cf. Figure 1a). The *Organization Model* describes the structural and administrative features of the organization under observation. The *Task Model* represents the tasks that must be

performed inside the organization in a structured, hierarchical way. The *Agent Model* describes the (human or artificial) actors participating in the task execution, along with their main attributes. The *Communication Model* specifies the relation between the various agents and what type of data they share. The *Knowledge Model* represents the types and structures of the knowledge that is used for the different tasks. The role of different knowledge components in decision-making is also provided in a human-readable way. The *Design Model* provides the technical specifications, i.e. software architecture and implementation tools.

Let us imagine the case of a HI system for robotic surgery, where a robotic microscope and a surgeon complement each other’s skills on the level of micrometer surgery. The robotic agent has to learn how best to align its angle and zoom of vision with the activities of the human surgeon. This requires a mutual understanding of each upcoming surgical procedure, which is to be acquired during practice sessions. In this scenario, the **Organization Model** would allow for analysis of the hospital setting where the two agents operate, i.e. discovering problems and opportunities for the application of an HI system, establishing its feasibility, and assessing the impact on the hospital if succeeding. The **Task Model** would describe the global layout of the micrometer surgery task, including inputs/preconditions (e.g. patient history, experience gained through the practice sessions), outputs/postconditions (e.g. performance criteria) and necessary resources (e.g. surgery or monitoring tools). The **Agent Model** would describe the characteristics of the robot and the human surgeon, namely their competences (zooming, manipulating, communicating, etc.), authority and legibility. These 3 models together would allow to first analyze the hospital setting and the success factors for the surgeon HI system. The Knowledge and Communication models would then allow to obtain a conceptual description of the functions and data handled and delivered by the HI system. The **Knowledge Model** would include the types of knowledge used in performing the surgery (e.g. knowledge of the human body, diagnosis of the patient, status of the surgery tools, established procedure) including their role in the specific steps of the surgery (e.g. a patient’s history is needed before inducing a given anaesthesia). The **Communication Model** would represent the communicative transactions between the robot microscope and the surgeon, including e.g. human-robot dialogues, communication of the shared mental models or planning solutions. The **Design Model** would ultimately convert all the data, functions and communication links into a technical specification (architecture, implementation platform, software modules) for an HI system to be implemented. While all of these models are relevant to HI applications, the goal of *this paper* is to identify HI recurrent knowledge roles and tasks. The current work is therefore focused on CommonKADS’ *Knowledge Model*, as this helps clarifying the role that different knowledge components play in the HI processes in human-understandable way. A structured analysis of Hybrid Intelligence scenarios following the full CommonKADS methodology is left for future work.

The Knowledge Model specifies the vocabulary used, i.e. the main classes of the domain (e.g. patients, diseases, tools) and the reasoning task(s) that need to be performed (e.g. assessment, configuration, diagnosis). This knowledge is divided in 3 layers (cf. Figure 1b with an example of a medical diagnosis application):

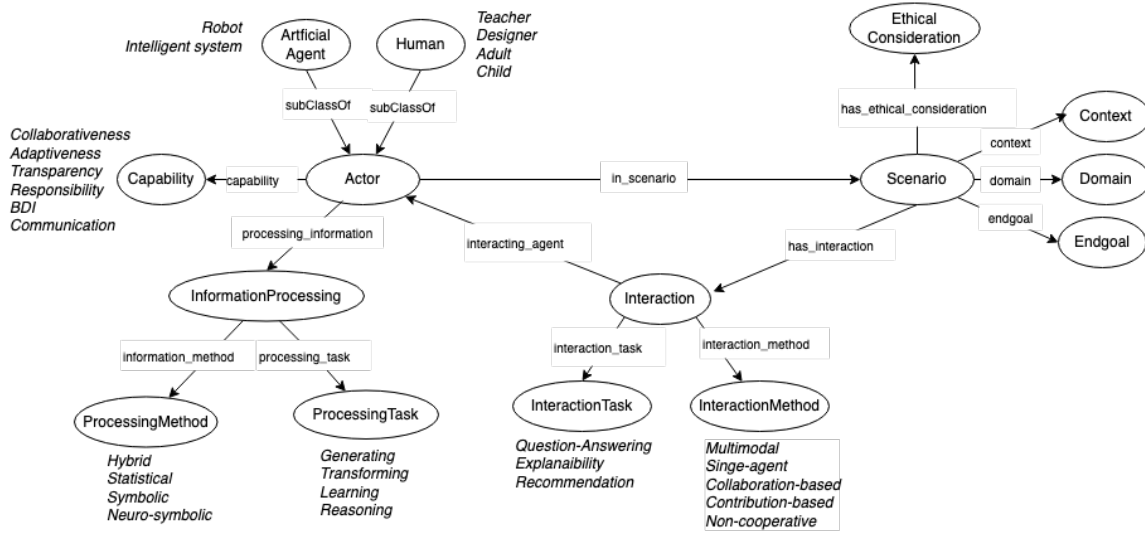


Figure 2: Knowledge roles for Hybrid Intelligence scenarios. *Italic terms are examples of instances.*

1. the *task layer* describes what problem-solving subtasks need to be carried out, and how (namely, specification of input, output and goal of every subtask);
2. the *inference layer* includes the set of primitive reasoning steps, that are combined in subtasks. It is the lowest level of decomposition of the task layer subtasks;
3. the *domain knowledge layer* provides the domain knowledge required to execute the inference steps (classes and relationships).

The Knowledge Model is built in a spiral way: first, a domain ontology with an initial terminology is created (*identification*). The terms should be functional, as they serve as inputs and outputs of the primitive inference steps. These inputs can be variant (dynamic) or invariant (static). Next, the tasks and primitive inference steps are defined graphically (*specification*). The inference steps should belong to a standard set as much as possible. One can either choose a task template and decompose the application according to the inference steps, ultimately linking them to the domain ontology (*middle-in strategy*), or first define the inference steps based on the domain ontology, then identifying a generic task method (*middle-out strategy*). Ultimately, the model is tested with instances from the ontology (*refinement*).

3.2 A Knowledge Model for Hybrid Intelligence

Following the steps just described, we first define a high-level ontology to represent roles in an HI application, then establish a set of HI tasks, subtasks and primitive inference steps.

Domain Knowledge Layer. This layer describes the knowledge involved in a HI scenario in the forms of concepts, relations and facts needed for reasoning. This knowledge will vary from application to application (e.g. a personalized health assistant will need knowledge about nutrition, personal values and sport activities, while a trash-picking robot will need knowledge about objects, affordances and bin locations), so defining a specific terminology is not useful here. However, a set of abstract classes and relationships can be defined, that indicate their **knowledge role** in the reasoning process and can be reused across scenarios.

The high-level ontology of HI knowledge roles is presented in Figure 2. An OWL version is available in the online repository. In every HI application, 3 main components are considered: (1) the **participating actors**, described in terms of characteristics, capabilities they have and information they are able to process; (2) the **interaction** between them, including the performed task and interaction modality; (3) the **scenario** and characteristics such as its domain, context, end goal and potential ethical considerations.

In a practical example, a simple HI scenario (class *Scenario*) could be aiming at supporting scholars (class *Endgoal*) in the Digital Humanities (*Domain*) using a transparent (*EthicalConsideration*) assistant for analyzing historical biographies (*Context*). Two instances of the class *Actor* are here involved, i.e. a human practitioner called Matthias and the MyQASystem Q&A system, which is able to answer, explain and communicate (class *Capability*) using Argumentation Theory and Pattern Matching (*ProcessingMethod*) to reason over the data (*ProcessingTask*). The actors interact collaboratively (*InteractionMethod*) to build explanations for the human (*InteractionTask*). An additional domain-specific ontology would then describe biographies from a digital humanities perspective.

Note that this high-level ontology serves as a foundation to align and compare design components across Hybrid Intelligence scenarios, but is non-exhaustive, nor extensively validated through competency questions. Developing a complete HI ontology is outside the scope of this paper, but planned future work, along with alignments to existing ontologies and vocabularies.

Task Layer. CommonKADS tasks and subtasks are described in a hierarchical fashion. Top-level tasks are decomposed into smaller subtasks, which in turn split into even smaller units (the *task decomposition* process). The leaves at the lowest level of the task structure are inference steps that are linked in the inference layer.

Similar to the domain knowledge layer, a natural question to ask is whether a set of generic tasks and inferences specific to HI scenarios are needed. The original CommonKADS tasks (e.g. analytic tasks such as diagnosis, monitoring, classification or synthetic tasks such as scheduling, planning, designing) were thought for rule-based

Table 1: Non-exhaustive HI task and subtask decomposition.

Recognition	Training, Classification		
Prediction	Regression	Training, Predicting	
Reasoning	Inference	Induction, Abduction, Deduction	
Action	Cognitive	Attention	Monitor, Orienting, Sustained
		Memory	Recollect, Short-term, Sensory
		Perception	Vestibular, Olfaction, Audition, Touch, Vision
		Decision Making	Individual, Societal, Debiasing
	Knowledge-aid	Mental Models, Team Role Allocation, Creativity	
	Physical	Movement	Body Part Mvt., Translation

systems and single-agents, which might be limiting for complex applications relying on different techniques and actors (symbolic and subsymbolic reasoning, human-in-the-loop, communication, team coordination). Using notions from the literature [3, 14, 22], we instead create a non-exhaustive hierarchy of tasks and subtasks achievable by both humans and machines (cf. Table 1), resulting in 4 top-level tasks: (i) *Recognition*: i.e. recognizing instances of a class, e.g. autonomous driving or smart assistants tasks; (ii) *Prediction*: i.e. predicting future events based on past data, e.g. stock prices or weather forecast; (iii) *Reasoning*: i.e. understanding data to derive complex rules deductively, inductively or abductively; (iv) *Action*: i.e. conducting a certain action, be it a cognitive or a physical one. While not complete, this structure offers a primitive set of top-level processes that can be used to formally describe HI scenarios.

Inference Layer. In the inference layer, the lowest units of the task decomposition are connected to each other using knowledge roles as inputs and outputs. These units are either *inferences* carrying out a primitive reasoning step (e.g. a “predict” inference step takes a hypothesis as input and uses a manifestation model to output an expected outcome), or *transfer functions* in charge of communicating with other agents. While a few abstract inferences steps for HI scenarios can be defined, we can also keep the standard CommonKADS transfer functions (*obtain*, *receive*, *present*, *provide* a given input) as they allow to describe the communication within the HI teams. We also maintain the distinction between *static* inputs (e.g. a knowledge graph, a trained model, a societal values scheme) and *dynamic* inputs (e.g. explanations, actions, recommendations).

4 VALIDATION

We validate the HI Knowledge Model to answer our research questions, namely to understand if KE can help characterize HI applications (Section 4.1) and identify HI-specific tasks (Section 4.2).

4.1 Characterizing HI Scenarios

To answer our first research question, we use the HI Knowledge Model of Section 3.2 to characterize different HI scenarios: 2 PhD projects of the Dutch Hybrid Intelligence consortium (S1, S2), 2 research projects of the European Humane-AI network (S3, S4), and 3 applications presented at the Hybrid Human-Artificial Intelligence conferences (S5, S6, S7). For each of them, we describe their scenario and mark the terms that are mapped to the knowledge roles in the ontology of Figure 2. We then provide their task decomposition (terminology, inference steps and performed tasks) in both semi-structure text and a flowchart. Due to space restrictions, we only graphically report on 4 of these scenarios in this paper, while

the complete material can be found online⁸. As mentioned, the repository serves as a basis to store newly-designed applications, fostering reuse, quality control and evaluation of HI scenarios.

Following CommonKADS, we use a UML-based notation with semi-formal semantics: the subtask leaves are represented as ovals (inferences) or rounded rectangles (transfer functions); rectangles are input/output knowledge roles, which can be either dynamic (full squares) or static (horizontal lines); arrows starting with a dot mean the input/output is a list; dashed boxes merge inference steps into more general subtasks.

(S1) Detecting conflicting, non-cooperative Smart Assistants.

S1 involves smart assistants (class *Scenario*) for improving personal health (*Endgoal*). An artificial agent (*Actor*) provides recommendations to users (*Actor*) based on their daily diet and exercise schedule. The aim is to detect and explain (*Capability*) deceptive behaviors such as lies about one’s own activity from data errors due to external conditions (e.g. available resources, varying environmental conditions). Argumentation Theory (*ProcessingMethod*) is used to detect errors and conflicts through reasoning (*ProcessingTask*) over the users’ prior knowledge in the form of a knowledge graph.

TERMINOLOGY: (dynamic) user questions, behaviors, errors, causes, historical user data. (static) societal health values, external background knowledge.

INFERENCE STEPS:

1. *receive* question from a user;
2. *classify* a user behavior based on questions, historical data, and societal values;
3. *assess* the user behaviors;
4. *classify* a deceptive behavior as an error using Argumentation Theory;
5. *induce* the causes for an error;
6. *rank* the classified behaviors ;
7. *provide* the recommended behavior based on the detected cause.

TASKS: Recognition (1-2), Monitoring (3), Explaining (5), Recommendation (4-7).

(S2) Learning Explainable Sequential Behaviors.

An embodied mobile agent and a human collaborator (both *Actor* instances) perform manipulation tasks involving a complex sequence of steps (*Scenario*), e.g. unlocking and opening a cabinet before retrieving a tool. Actions such as lifting a heavy object can be performed only by the robot, while others such as inserting a key into a small lock only by humans. Using Reinforcement Learning (*ProcessingMethod*), the agent learns the optimal sequential behavior without the exact problem description as in classical planning. Knowledge Graphs are used as prior knowledge to learn low-level policies (*EndGoal*) in the abstract transition graph, in order to track the agent’s state and explain (*Capability*) dangerous states to the collaborator [10].

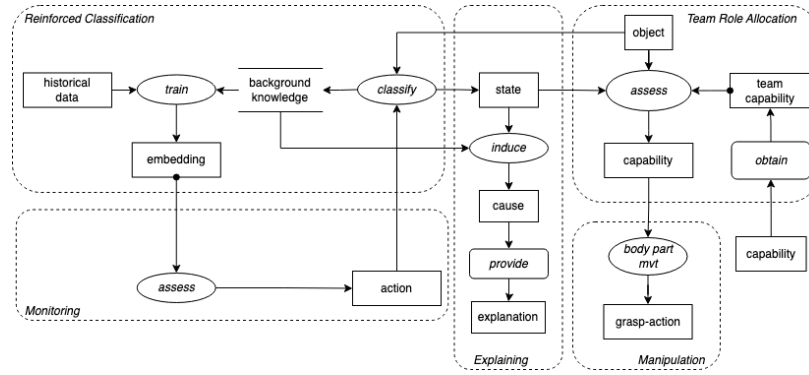
TERMINOLOGY: (dynamic) object, capabilities, actions, causes, explanations, states. (static) background knowledge (Knowledge Graph).

INFERENCE STEPS:

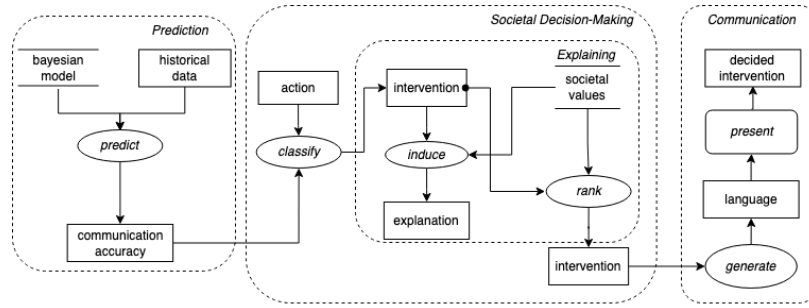
1. *train* a model based on the background and some historical action data;
2. *assess* the agent’s actions to identify the optimal one;
3. *classify* the agent’s state as dangerous or not;
4. *induce* the dangerous states and
5. *provide* them to the users;
6. *obtain* the other agents’ capabilities;
7. *assess* the other agent’s capabilities;
8. *move body part* to perform the grasping action accordingly;

TASKS: Classif. (1-3), Explaining (4-5), Team Role Allocation (6-7), Manip. (8).

⁸<https://github.com/kmitd/HI-CommonKADS>



(a) S2: Learning Explainable Sequential Behaviors.



(b) S3: Collective Decisions in Law and Economics.

Figure 3: Task Decomposition for Hybrid Intelligence Scenario S2 (a) and Scenario S3 (b).

(S3) Collective Decisions in Law and Economics. An artificial agent (*Actor*) needs to intervene to resolve a dilemma (*Scenario*) in a group decision, e.g. a jury (*Actor*). Bayesian logic (*ProcessingMethod*) is used as a meta-analytical tool to formalize the main criteria (i.e. individual and group accuracy in communication) for assessing when an intervention is (ir)responsible (*EthicalConsideration*), and to explain (*Capability*) the conditions (*EndGoal*) under which the autonomous agent should take the responsibility to act.

TERMINOLOGY: (dynamic) historical data, communication accuracy, action, intervention, explanation, language, decided intervention. (static) Bayesian model, societal values.

INFERENCE STEPS:

1. *predict* the communication accuracy using Bayesian modelling and past data;
2. *classify* interventions as (ir)responsible based on an input action;
3. *generate* explanations for these based on known societal values;
4. *rank* the interventions based on societal values,
5. *generate* language-based interventions;
6. *present* the intervention to the group.

TASKS: Predict. (1), Societal Decision-Making (2-4), Explaining (3), Comm. (5-6).

(S4) Educational Recommenders with Narratives. An autonomous artificial tutor (*Actor*) reconstructs the educational paths of individual learners (*Actor*) as personalized narratives, to recommend their educational resources based on their history. The agent uses Bayesian theory (*ProcessingMethod*) to model interests, prior knowledge of the learner and their semantic relatedness with the education topics, based on which it improves the performance of the recommendations. Public KGs such as WordNet and WikiData are used

to perform high-level reasoning (*ProcessingTask*), ultimately fulfilling the learners' learning goals more effectively (*EndGoal*) [17].

TERMINOLOGY: (dynamic) learning goal, historical data, personal narrative, recommended resource, educational resource, accuracy, structured knowledge. (static) Bayesian Theory, external structured knowledge (KGs).

INFERENCE STEPS:

1. *generate* personal narratives based on learning goals and historical data;
2. *engineer* topic models using Bayesian theories and
3. *infer* their accuracy;
4. *rank* educational resources based on personal narratives and computed accuracy;
5. *filter* resources based on external KGs;
6. *adapt* the recommendation based on the learner changing goals.

TASKS: Creation (1), Reasoning (2-3,5), Recommendation (4), Adaptation (6).

(S5) Machine Intelligence with Knowledge Graphs. A hybrid system generates diagnoses (*Scenario*) based on the patient's symptoms and medical history stored in a Knowledge Graph [8]. A human operator and an embeddings-based (*ProcessingMethod*) system (both *Actors*) with complementary scientific expertise (*Capability*) perform the prediction and justify their propositions to each other (*InteractionTask*) in order to come up with a final decision (*EndGoal*).

TERMINOLOGY: (dynamic) symptoms, embeddings, historical medical events, diagnoses, justifications, decisions. (static) human memory, medical KG.

INFERENCE STEPS:

- 1-2. (human) *recollect* similar events and *induce* the diagnosis;
- 3-4. (machine) *train* and *classify* diagnoses;

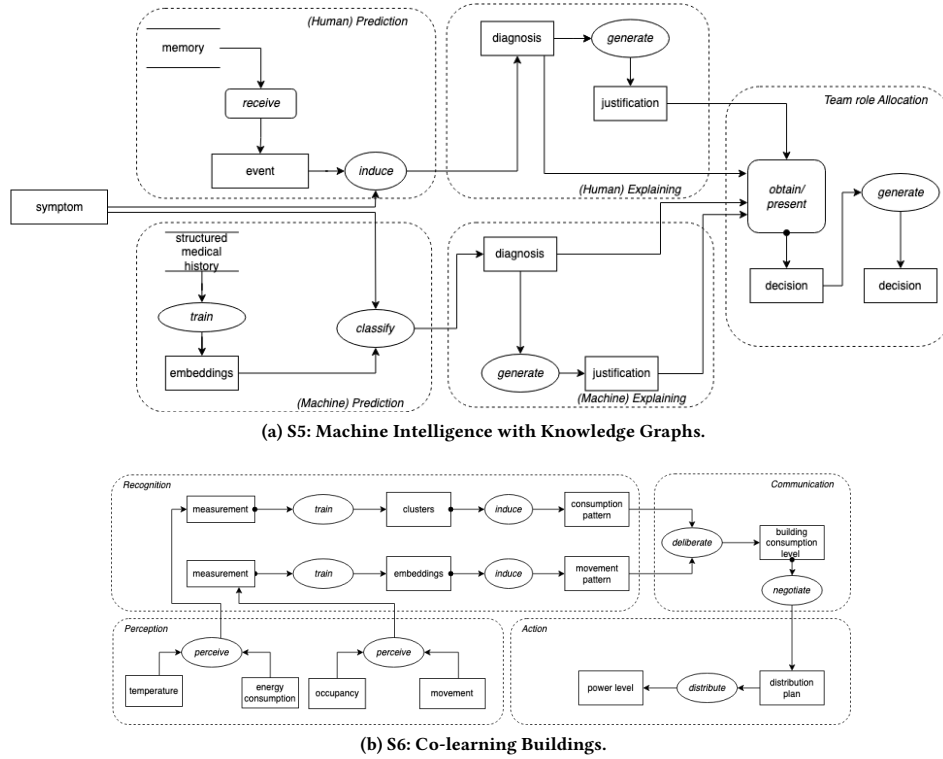


Figure 4: Task Decomposition for Hybrid Intelligence Scenario S5 (a) and Scenario S6 (b).

5-6. generate a justification for the diagnosis and
 7-8. present it to the other agent that can
 9. obtain the information;
 10. ultimately, generate a combined decision.
 TASKS: Prediction (1-4), Explaining (5-6), Team Role Allocation (7-10).

(S6) Co-learning Buildings. A campus (*Context*) involves AI-based buildings (*Actor*) negotiating on power level for the heating system (*EndGoal*). They optimize their own energy consumption, take into account the energy needs of others when moving between different buildings, and also interact with human grid operators accounting for the preferences of building owners. The agents learn (*ProcessingMethod*) occupancy-movement patterns and building energy consumption behavior in a collaborative co-learning setting (*InteractionMethod*), where the learned models are negotiated in order to deliberate on a power distribution plan [15].

TERMINOLOGY: (dynamic) data, measurements, clusters, embeddings, movement and consumption patterns, consumption models, distribution plans.

INFERENCE STEPS:

- 1-2 perceive data from sensors (energy, temperature, occupancy, movement);
- 3-6. train&induce energy consumption and occupancy-movement patterns based on the measurements;
7. deliberate the building consumption model to the other agents;
8. negotiate on a distribution plan;
9. distribute the power level to the campus.

TASKS: Perception (1-2), Recognition (3-6), Communication (7-8), Action (9).

(S7) KG-based Guide for Virtual Heritage Exhibitions. An agent (*Actor*) with Q&A *Capability* offers personalized guidance to visitors of a virtual museum (*Scenario*) based on their interests (*EndGoal*). Virtual reality (*InteractionTask*) is used to predict

multi-modal user inputs and reason (both *ProcessingTask*) over their profiles and the interactions with the agent. A KG is used to answer user questions about the exhibition and suggest related items computed using embeddings (*ProcessingMethod*) [12].

TERMINOLOGY: (dynamic) questions, events, actions, profiles, embeddings, museum items. (static) museum KG, historical data.

INFERENCE STEPS:

1. recollect events;
2. perceive user actions based on the Virtual Reality action;
3. train a model to compute KG embeddings;
4. rank items to recommend to the user;
5. finally, present the item to the user.

TASKS: Recollection (1), Profiling (2), Recommendation (3-4), Communication (5).

4.2 Identifying Common HI Tasks

To answer our second research question, we use the task decompositions to compare HI scenarios and identify tasks that are HI-specific. Our analysis reveals that classical CommonKADS tasks such as Monitoring, Prediction or Classification occur among many scenarios (S1, S2, S3, S5, S6). This was expected, as autonomous agents based on (subsymbolic) reasoning are involved. This also happens in more open-ended tasks such as Recommendation (S1, S4, S7), not directly envisioned by CommonKADS but common in modern applications relying on large knowledge graphs. Interestingly, structured knowledge is often used as input for subsymbolic learning (S1, S2, S4, S5, S7), suggesting that KGs are widely accepted by HI to provide insightful background information and boost the learning performance.

Additionally, we remark the emergence of previously unseen tasks, e.g. Explaining (S1, S2, S3, S5), Communication (S3, S6, S7) and negotiation (S3, S6) as part of it. These tasks can be considered as knowledge-aid subtasks involving Creation and/or Language, as identified in Table 1. Other interesting subtasks are those involving awareness of an interaction happening in a team with certain capabilities and/or societal values. Tasks such as Societal Decision-Making (S3), Team Role Allocation (S2, S5), Goal/Action Adaptation (S2, S4) indicate that the classical CommonKADS method needs to be extended to better represent situations involving hybrid collaboration. We find also tasks involving multimodal knowledge and data, e.g. capturing human actions through Virtual Reality (S6) or sensing and grasping actions performed by artificial agents (S2, S6), also suggesting that the task design has to be extended from classical KE ones. These tasks not only show that more Hybrid Intelligence-specific tasks exist, but also conveniently reflect the core HI concepts of adaptability, collaborativeness, explainability and responsibility identified by [1, 4]. The task decompositions are examples serving to design new HI scenarios, e.g. S5 can help when designing a *collaborative* and *explainable* HI application, S2 can guide the creation of *adaptive* and *explainable* scenarios, S3 and S6 are exemplary for *collaborative* and *responsible* scenarios, etc.

While the tasks we identify are not complete, our results show how an adaptation of classical Knowledge Engineering methods helps identifying common design components for HI scenarios. This is also demonstrated by the high-level ontology of HI knowledge roles that we built for the domain knowledge layer. All inputs/outputs of the inferences are instances in the ontology, allowing us to derive a HI structure composed of mixed actors, capabilities, interactions and processing types in every scenario. The ontology can now be further extended to function as a standard vocabulary for Hybrid Intelligence, while the HI-specific tasks can be integrated into reusable task templates. These steps will ultimately serve as bases to formally characterize Hybrid Intelligence, i.e. fostering reuse, comparison, system design and evaluation.

5 CONCLUSIONS

Hybrid Intelligence, where humans and machines cooperate synergistically to achieve shared goals, is an active field of research. We used Knowledge Engineering to formally characterize HI applications, providing an ontology of HI knowledge roles, a set of HI-specific tasks, and an open repository serving as a basis for standardization, reuse and validation. Future work will focus on publishing the HI ontology and on identifying HI task templates.

REFERENCES

- [1] Zeynep Akata, Dan Balliet, Maarten De Rijke, Frank Dignum, Virginia Dignum, Gusztá Eiben, Antske Fokkens, Davide Grossi, Koen Hindriks, Holger Hoos, et al. 2020. A Research Agenda for Hybrid Intelligence: Augmenting Human Intellect with Collaborative, Adaptive, Responsible, and Explainable Artificial Intelligence. *Computer* 53, 08 (2020), 18–28.
- [2] Jürgen Angele, Dieter Fensel, Dieter Landes, and Rudi Studer. 1998. Developing KB-systems with MIKE. In *Domain Modelling for Interactive Systems Design*. Springer, 9–38.
- [3] Dominik Dellermann, Adrian Calma, Nikolaus Lipusch, Thorsten Weber, Sascha Weigel, and Philipp Ebel. 2019. The Future of Human-AI Collaboration: a Taxonomy of Design Knowledge for Hybrid Intelligence Systems. In *Hawaii international conference on system sciences (HICSS)*.
- [4] Dominik Dellermann, Philipp Ebel, Matthias Söllner, and Jan Marco Leimeister. 2019. Hybrid intelligence. *Business & Information Systems Engineering* 61, 5 (2019), 637–643.
- [5] Alpana Dubey, Kumar Abhinav, Sakshi Jain, Venu Arora, and Asha Puttaveerana. 2020. HACO: a framework for developing human-AI teaming. In *Proceedings of the 13th Innovations in Software Engineering Conference on Formerly known as India Software Engineering Conference*. 1–9.
- [6] Aldo Gangemi and Valentina Presutti. 2009. Ontology Design Patterns. In *Handbook on ontologies*. Springer, 221–243.
- [7] John H Gennari, Mark A Musen, Ray W Fergerson, William E Grosso, Monica Crubézy, Henrik Eriksson, Natalya F Noy, and Samson W Tu. 2003. The Evolution of Protégé: an Environment for Knowledge-Based Systems Development. *International Journal of Human-computer studies* 58, 1 (2003), 89–123.
- [8] Christophe Guéret. 2022. Knowledge Graphs in support of Human-Machine Intelligence. In *International Conference on Hybrid Intelligence*. IOSPress.
- [9] M Henao, Jose Soler, and V Botti. 2001. Developing a Mobile Robot Control Application with CommonKADS-RT. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 651.
- [10] Niklas Höpner, Ilaria Tiddi, and Herke van Hoof. 2022. Leveraging Class Abstraction for Commonsense Reinforcement Learning via Residual Policy Gradient Methods. *IJCAI-ECAI2022* (2022).
- [11] Ece Kamar. 2016. Directions in Hybrid Intelligence: Complementing AI Systems with Human Intelligence.. In *IJCAI*. 4070–4073.
- [12] Dou Liu, Claudia Libbi, Delaram Javdani Rikhtehgar, and Shenghui Wang. 2022. What would you like to visit next? - Using a Knowledge-Graph Driven Museum Guide in a Virtual Exhibition. In *HAI2022*. IOSPress.
- [13] Azad M Madni and Carla C Madni. 2018. Architectural framework for exploring adaptive human-machine teaming options in simulated dynamic environments. *Systems* 6, 4 (2018), 44.
- [14] Anne Collins McLaughlin and Vicky E Byrne. 2020. A Fundamental Cognitive Taxonomy for Cognition Aids. *Human Factors* 62, 6 (2020), 865–873.
- [15] André Meyer-Vitali and Wico Mulder. 2022. Trustworthy Hybrid Team Decision-Support. In *Proceedings of the First International Conference on Hybrid Human-Machine Intelligence. Workshop on the Representation, Sharing and Evaluation of Multimodal Agent Interaction, befindet sich International Conference on Hybrid Human-Artificial Intelligence (HHAI), June 13-17, Amsterdam, Netherlands (Frontiers of AI)*. IOS Press.
- [16] Johannes Nguyen, Thomas Farrenkopf, Michael Guckert, Simon T Powers, and Neil Urquhart. 2021. Using Semantic Technology To Model Persona For Adaptable Agents. In *ECMS*.
- [17] Maria Perez-Ortiz, Claire Dormann, Yvonne Rogers, Sahan Bulathwela, Stefan Kreitmayer, Emine Yilmaz, Richard Noss, and John Shawe-Taylor. 2021. X5learn: A Personalised Learning Companion at the Intersection of AI and HCI. In *26th International Conference on Intelligent User Interfaces-Companion*. 70–74.
- [18] Mohammad Sabri and Marjan Fatemi Garakani. 2012. Modeling an Agent-mediated Supply Chain Management System using MAS-CommonKADS Methodology. *International Journal of Mechatronics, Electrical and Computer Technology* 2, 3 (2012), 58–75.
- [19] Guus Schreiber, Hans Akkermans, Anjo Anjewierden, Nigel Shadbolt, Robert de Hoog, Walter Van de Velde, Bob Wielinga, R Nigel, et al. 2000. *Knowledge Engineering and Management: the CommonKADS Methodology*. MIT press.
- [20] Guus Schreiber, Bob Wielinga, Robert de Hoog, Hans Akkermans, and Walter Van de Velde. 1994. CommonKADS: A Comprehensive Methodology for KBS-development. *IEEE expert* 9, 6 (1994), 28–37.
- [21] Rudi Studer, V Richard Benjamins, and Dieter Fensel. 1998. Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering* 25, 1-2 (1998), 161–197.
- [22] Michael van Bakkum, Maaike de Boer, Frank van Harmelen, André Meyer-Vitali, and Annette ten Teije. 2021. Modular Design Patterns for Hybrid Learning and Reasoning Systems. *Applied Intelligence* 51, 9 (2021), 6528–6546.
- [23] Frank van Harmelen and Annette ten Teije. 2019. A Boxology of Design Patterns for Hybrid Learning and Reasoning Systems. *Journal of Web Engineering* (2019).
- [24] Frank van Harmelen, Annette ten Teije, and Holger Wache. 2009. Knowledge Engineering Rediscovered: towards Reasoning Patterns for the Semantic Web. In *Proceedings of the fifth international conference on Knowledge capture*. 81–88.
- [25] Jip J van Stijn, Mark A Neerincx, Annette ten Teije, and Steven Vethman. 2021. Team Design Patterns for Moral Decisions in Hybrid Intelligent systems: A Case Study of Bias Mitigation. In *2021 AAAI Spring Symposium on Combining Machine Learning and Knowledge Engineering, AAAI-MAKE 2021*. CEUR-WS, 1–12.
- [26] Jasper van der Waa, Jurriaan van Diggelen, Luciano Cavalcante Siebert, Mark Neerincx, and Catholijn Jonker. 2020. Allocation of Moral Decision-Making in Human-Agent Teams: a Pattern Approach. In *Engineering Psychology and Cognitive Ergonomics. Cognition and Design*. Springer, Cham, 203–220.
- [27] Bob J Wielinga. 2013. Reflections on 25+ years of Knowledge Acquisition. *International journal of human-computer studies* 71, 2 (2013), 211–215.
- [28] Dong Yang, Lixin Tong, Yan Ye, and Hongwei Wu. 2006. Applying CommonKADS and Semantic Web Technologies to Ontology-based e-government Knowledge Systems. In *ASWC*. Springer, 336–342.
- [29] Ming Zhou, Jianwen Ren, Jianxun Qi, Dongxiao Niu, and Gengyin Li. 2007. CommonKADS Methodology for Developing Power Grid Switching Orders Systems. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 87.