Efficient Reasoning with Ambient Trees for Space Exploration

Emil Vassev and Mike Hinchey

Lero—the Irish Software Engineering Research Centre,
University of Limerick, Limerick, Ireland

{Emil.Vassev, Mike.Hinchey}@lero.ie

Abstract. Modern reasoning is based on inference techniques such as induction, deduction, abduction, subsumption, classification and recognition. These inference techniques are very inefficient when applied to large amounts of knowledge such as ones employed by contemporary unmanned spacecraft. For efficient reasoning, we aim at knowledge representation based on special ambient trees determining special knowledge contexts to help such spacecraft retrieve context-relevant knowledge and perform deductive reasoning, which would not be otherwise highlighted. Contexts via their ambient trees provide a sort of a condensed and explicit symbolic representation of the world. This representation is cleaned from the overwhelming information that is non-relevant to the context and thus, it provides for efficient models of situations to reason about.

Keywords: reasoning, knowledge representation, space exploration, autonomous spacecraft

1 Introduction

Modern unmanned spacecraft have *onboard intelligence* that helps them reason about situations in space where autonomous decision making is required [1]. The basic compound in the reasoning process is *knowledge*. Smart spacecraft employ appropriately structured knowledge that is used by onboard inferential engines. The knowledge is integrated in the spacecraft system via *knowledge representation* techniques to build a computational model of the space-mission domain of interest in which symbols serve as *knowledge surrogates* for real world artifacts, such as spacecraft components and functions, mission details, space objects such as planets, satellites, asteroids, etc. The domain of interest can cover any part of the real world or any hypothetical system about which one desires to represent knowledge for computational purposes. Modern reasoning is based on inference techniques such as *induction*, *deduction*, *abduction*, *subsumption*, *classification* and *recognition*. Although efficient on small knowledge models, those inference techniques are very inefficient when applied to large amounts of knowledge such as ones employed by modern unmanned spacecraft. Therefore, for efficient reasoning, it should be possible to reason by emphasizing on relevant knowledge by ignoring selected parts that are not relevant to the particular situation of interest. In our approach, special ambient trees define *knowledge contexts* that help spacecraft retrieve context-relevant knowledge and perform deductive reasoning, which would not be otherwise highlighted. Contexts via their ambient trees provide a sort of a condensed and explicit symbolic representation of the world. This knowledge representation is *cleaned* from the overwhelming information that is non-relevant to the knowledge context and thus, it provides efficient models of situations to reason about.

2 Theoretical Model for Efficient Reasoning with Ambient Trees

The most prominent and powerful knowledge-representation technique is ontology [2]. A space-mission ontology gives a *formal* and *declarative representation* of the knowledge domain in terms of explicitly described domain concepts, individuals and the relationships between those concepts/individuals. The Theoretical Model for Efficient Reasoning with Ambient Trees (for short Efficient Reasoning with Ambient Trees (ERAT)), relies on the ontology technique to represent knowledge in smart spacecraft. According to ERAT, the knowledge $\boldsymbol{K}$ of smart spacecraft, capable of reasoning and decision making, can be formally presented as a tuple of three main knowledge components (knowledge models):

$$\boldsymbol{K} = \langle \boldsymbol{K_I}, \boldsymbol{K_C}, \boldsymbol{K_E} \rangle \qquad (1)$$

where $\boldsymbol{K_I}$ is *internal knowledge*, $\boldsymbol{K_C}$ is *control knowledge*, and $\boldsymbol{K_E}$ is *external knowledge*. The internal knowledge $\boldsymbol{K_I}$ carries information about the internal structure and capabilities of the system. The control knowledge $\boldsymbol{K_C}$ gives the system knowledge about its control parameters and mission. Finally, the external knowledge $\boldsymbol{K_E}$ is to provide the spacecraft with information about the surrounding environment, e.g., solar system, solar storms, planetary systems, asteroids, gravity force of the near space objects, etc. Each one of the $\boldsymbol{K}$ components is presented by an ontology $\boldsymbol{O}$ and a set of special contexts $\boldsymbol{C_X}$.

$$\boldsymbol{K_X} = \langle \boldsymbol{O}, \boldsymbol{C_X} \rangle \qquad (2)$$

This way of structuring knowledge is a sort of *context-oriented knowledge structuring* and it helps us to decompose complicated intelligent behavior into many "simple" and context-dependent behavior modules. Furthermore, an ontology is composed of hierarchically organized sets of *meta-concepts* $\boldsymbol{C_M}$, special *concept trees* $\boldsymbol{C_T}$, *object trees* $\boldsymbol{O_T}$, *relations* $\boldsymbol{R}$, predicates $\boldsymbol{V}$ and facts $\boldsymbol{F}$.

$$\boldsymbol{O} = \langle \boldsymbol{C_M}, \boldsymbol{C_T}, \boldsymbol{O_T}, \boldsymbol{R}, \boldsymbol{V}, \boldsymbol{F} \rangle \qquad (3)$$

Meta-concepts $\boldsymbol{C_M}$ provide a context-oriented *interpretation* of concepts. Concept trees $\boldsymbol{C_T}$ consist of semantically related *concepts* $\boldsymbol{C}$. Every *concept tree* $\boldsymbol{c_T} \in \boldsymbol{C_T}$ has a *root concept* $\boldsymbol{r_C}$, because the architecture ultimately must reference a single concept that is the connection point to concepts that are outside the concept tree. A root concept may *optionally* inherit a meta-concept, which is denoted $[r_C \rightarrow c_M]$ (see Formula 5). Every concept c has a set of *properties* P and optional sets of functionalities A, *parent concepts* C_P and *children concepts* C_H. Figure 1 depicts a sample *concept tree* structuring the capabilities of a space robot (e.g., automatic probe).

$$C_T = \langle\, r_C ,\, C \,\rangle \qquad (4)$$

$$r_C \in C,\ [r_C \rightarrow c_M] \qquad (5)$$

$$c_M \in C_M$$

$$c = \langle\, P ,\ [A] ,\ [C_P] ,\ [C_H] \,\rangle \qquad (6)$$

$$C_P \subset C,\ C_H \subset C$$

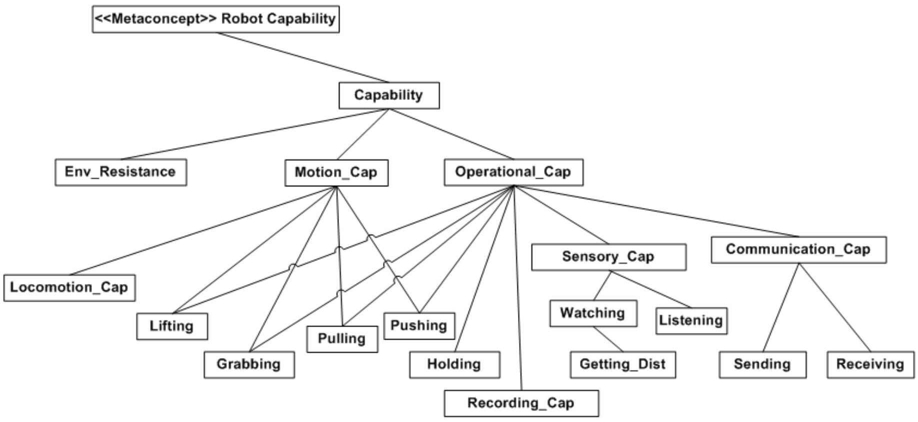

Fig. 1. Concept Tree: Robot Capability

Object trees O_T are conceptualization of how objects existing in the world of interest are related to each other. The relationships are based on the principle that objects are concepts realization and have properties where sometimes the value of a property is another object, which in turn also has properties. Such properties are termed as *object properties* P_O. An object tree consists of a root object r_O and an optional set of nested object trees O_T' formed by those object properties. An *object* is an instance of a concept.

$$o_T = \langle\, r_O ,\ [O_T'] \,\rangle \qquad (7)$$

Relations R connect two concepts/objects (only binary relations are considered), and facts F define true statements in an ontology O. Predicates V are special knowledge structures that specify distinct *interstate relations* or schemes for evaluation of complex states (e.g., context states). For example, we can specify a predicate that verifies whether the system is fully functional or to verify whether there is an upcoming high-energy solar storm. A predicate might be used whether a specific object, the environment or the entire system is in a specific state. Thus, a predicate v formally can be presented as a tuple of predicate concepts C_V, predicate states S_V and a Boolean expression over ontology ($bf(O)$) that determines what conditions must be held to conclude that the predicate states are active (see Formula 8). Note that a state (including a predicate state s_V) can be presented as a Boolean expression over ontology that holds (see Formula 9). A Boolean expression determining a state may involve *concept properties*, *object properties* and *concept functionalities*. Predicates might be used to determine if a specific context is present based on state evaluation.

$$v = \langle C_V , S_V , bf(O) \rangle \qquad (8)$$
$$v \in V, C_V \subset C, S_V \subset S$$
$$s_V = \langle bf(O) \rangle \qquad (9)$$
$$s_V \in S_V , S_V \subset S \ (S_V \text{ - predicate states})$$

Here, a state-determination function N shall determine for each state $s \in S$ the Boolean expression that must hold in order to consider a state "active".

$$N: S \times O \rightarrow \langle bf(O) , S \rangle \qquad (10)$$

Contexts C_X are intended to extract the relevant knowledge from an ontology O. Moreover, contexts carry interpretation for some of the meta-concepts (see Formula (11)), which may lead to new interpretation of the *descendant concepts* (derived from a meta-concept – see Formula 5). We consider a very broad notion of context, e.g., the space environment in a fraction of time or a generic situation such as currently-ongoing important system function, such as observing, listening, etc. Thus, a context must emphasize the key concepts in an ontology O, which helps the inference mechanism narrow the domain knowledge by exploring the concept trees down to the emphasized key concepts only. Thus, depending on the context, some low-level concepts might be subsumed by their upper-level parent concepts, just because the former are not relevant for that very context. For example, the planet Mars can be considered as a natural space object with gravity and spin orbit or as an important exploration target, which requires knowledge in great details about Mars. As a result, context interpretation of knowledge will help the system deal with "clean" knowledge and the reasoning shall be more efficient.

A context $c_X \in C_X$ consists of:

- special *ambient trees* A_T
- a set of context states S_{CX} determining when a context is present;
- optional *context interpretations* I_{CX}.

An ambient tree $a_T \in A_T$ consists of a *concept tree* c_T, a set of *ambient concepts* C_A, part of the concept tree c_T, and optional *context interpretation* i.

$$C_X = \langle A_T , S_{CX} , [I_{CX}] \rangle \qquad (11)$$
$$A_T = \langle c_T , C_A , [i] \rangle \qquad (12)$$
$$c_T \in C_T, C_A \subset C, \ i \in I_{CX}$$
$$s_{CX} = \langle bf(O) \rangle \qquad (13)$$
$$s_{CX} \in S_{CX} , S_{CX} \subset S \ (S_{CX} \text{ - context states})$$

The ambient concepts C_A are concepts, which explicitly determine a new level of deepness for their concept tree, i.e., ambient concepts subsume all of their child concepts (if any). As result, when the spacecraft reasons about a particular context (expressed with the ambient trees), the reasoning process does not consider those child concepts, but their ambient parents, which are far more generic, and thus less detailed. This technique reduces the size of the relevant knowledge, by temporarily removing from the concept trees all the ambient concepts' children (child concepts). We may think about ambient trees as filters the system applies at runtime to reduce the visibility (the amount of concepts) of a concept tree.

A context state s_{CX} determines when the context is present and thus, the ambient trees might be applied to hide the irrelevant knowledge. A context can be associated with multiple states where each state is a sufficient guarantee that the context is present. For example, we may have a state "*the spaceship is currently receiving signals from Earth*" that may uniquely identify that the *message-receiving context* is present.

Here, a context-aware function $\boldsymbol{E}$ is required to determine at runtime the current context. This function should operate on both predicates $\boldsymbol{V}$ (see Definition 8) and contexts $\boldsymbol{C_X}$ to determine the current context via the evaluation of the current system state, which eventually is a context state as well ($\boldsymbol{s_X}$) (see Definition 14).

$$\boldsymbol{E}: \boldsymbol{V} \times \boldsymbol{C_X} \xrightarrow{s_X} \boldsymbol{c_X} \quad (14)$$

Finally, we need a function $\boldsymbol{F}$ that will apply the discovered context over the ontology and force the system use the reduced in size context ontology $\boldsymbol{O_X}$.

$$\boldsymbol{F}: \boldsymbol{c_X} \times \boldsymbol{O} \rightarrow \boldsymbol{O_X} \quad (15)$$

3 Context Awareness

In general, autonomous spacecraft engage in interactions with the ground base on Earth and with the operational environment (space). When interacting with the environment, such spacecraft also perceive important structural and dynamic aspects of the same [3]. To become interaction-aware, such a system needs to be aware of its physical environment and whereabouts and its current internal status. This ability is defined as *awareness* and it helps intelligent computerized systems to sense, draw inferences for their own behavior and react. The notion of awareness should be generally related to perception, recognition, thinking and eventually prediction [4]. Recall that this approach requires relevant knowledge (see Definitions 1) that helps the system autonomously determine contexts (see Definition 14). This can be achieved via a mechanism called "Pyramid of Awareness" [4] where a complex chain of functions shall be implemented to control the *context awareness* process via monitoring, recognition, assessment, and learning.

4 Example of Efficient Reasoning with Ambient Trees

This technique could be successfully used to refining spacecraft knowledge that is relevant to a specific context. For example, let us suppose a spacecraft is using its radio to listen to its communication channels and to space sounds and noise. In such a case, we have a "listening" context, i.e., only knowledge relevant to speech, sounds and noise should be considered. The expressed with the ontologies spacecraft knowledge has concepts related to "sound" knowledge, e.g., concepts like "vocal commands", "speech", "sound", "noise", etc. The listening context defines all the ambient concepts, which should be used to subsume parts of their concept trees that are not relevant to sound knowledge. Figure 2, shows the application of an ambient tree from the listening context to a concept tree. The resulted concept tree is smaller in terms of concepts, which leads to faster and more efficient reasoning about situations in the listening context.

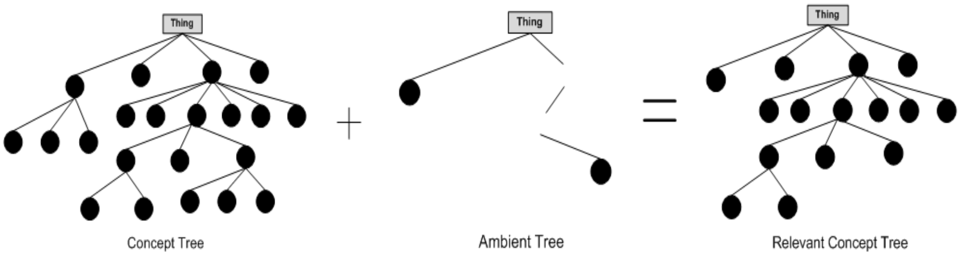

Fig. 2. Reducing knowledge size through ambient trees

5 Conclusion

In this paper, we outlined an approach to a knowledge representation technique helping unmanned spacecraft perform efficient reasoning by emphasizing context-relevant knowledge. The approach is based on special ambient trees intended to hide irrelevant knowledge by reducing the size of the ontology trees used to represent that knowledge. A special context-aware function is required to determine at runtime the current context and another function is required to automatically apply that context. Finally, the approach requires a context-awareness mechanism to keep the knowledge relevant and up-to-date by taking into consideration internal, control and environmental factors. The proposed awareness mechanism is the so-called "Pyramid of Awareness", which we are currently developing for another project of ours.

Currently, the approach is under development within the mandate of the ASCENS FP7 Project [5].

6 Acknowledgment

This work was supported by Science Foundation Ireland grant 03/CE2/I303_1 to Lero—the Irish Software Engineering Research Centre at University of Limerick, Ireland and by the European Union FP7 Integrated Project Autonomic Service-Component Ensembles (ASCENS).

References

1. Vassev, E., Sterritt, R., Rouff, C., Hinchey, M.: Swarm Technology at NASA: Building Resilient Systems. IT Professional, vol. 14 (2), pp. 36-42 (2012)
2. Vassev, E., Hinchey, M.: Knowledge Representation and Reasoning for Intelligent Software Systems. IEEE Computer, vol. 44 (8), pp. 96-99 (2011)
3 Vassev, E., Hinchey, M.: The Challenge of Developing Autonomic Systems. IEEE Computer, vol. 43 (12), pp. 93-96 (2010)
4 Vassev, E.: Building the pyramid of awareness. Awareness Magazine - Self-awareness in Autonomic Systems, July (2012). DOI: 10.2417/3201207.004320
5 ASCENS – Autonomic Service-Component Ensembles. 2012. http://www.ascens-ist.eu/.