Combining Planning Contexts

Saša Buvač and John McCarthy
Computer Science Department
Stanford University
Stanford
California 94305-9020
`{buvac, jmc}@cs.stanford.edu`
`http://www-formal.stanford.edu`

Abstract

It is often necessary to combine objects that were not designed to work together. These objects may be databases of facts, programs, hardware or plans. Even if the objects were intended to be used together, maintenance of adherence to their specifications and ensuring the consistency of terminology through time is often difficult. Each object was developed in a *context*, and these contexts differ, either slightly or greatly. The terminology within each subcontext is likely to be specialized, and making them work together requires some generalization.

We describe an approach based on our formal theory of context, the basic papers about which were written starting in 1993. An umbrella context refers to the subcontexts as *first class objects*, and the basic relations are $ist(c,p)$, meaning that the *proposition* p is *true* in the context c and also $value(c,e)$, designating the *value* of the *term* e in the context c. Besides these there are *lifting formulas* that relate the propositions and terms in subcontexts to possibly more general propositions and terms in the outer context. Subcontexts are often specialized with regard to time, place and terminology.

The goal is that no matter what corners the specialists paint themselves into, what they do can be *lifted* out and used in a more general context.

Introduction

In this paper we show how the formal theory of context, as developed in declarative AI, can be used to combine plans which were produced by different agents and were not originally intended to be used together.

Formal theory of context is based on the notion that an umbrella context can refer to subcontexts as *first class objects*. The basic relations are $ist(c,p)$, meaning that the *proposition* p is *true* in the context c and also $value(c,e)$, designating the *value* of the *term* e in the context c. This enables us to write formulas which relate the propositions and terms in subcontexts to possibly more general propositions and terms in the outer context.

Since it was first introduced in declarative AI as a possible solution to the problem of *generality* , (McCarthy 1987), formal theory of context has found a large number of uses in various areas of AI. R. V. Guha's doctoral dissertation, (Guha 1991), argues that without using contexts it would have been virtually impossible to create and successfully use a knowledge base the size of Cyc (Guha & Lenat 1990; Lenat 1995). The knowledge sharing community has also recognized the need for explicating context when transferring information from one agent to another (Patil *et al.* 1992; Finin & Mayfield 1995; Knoblock *et al.* 1995). Currently, proposals for introducing contexts into agent communication languages, including the Knowledge Interchange Format or KIF (Genesereth & Fikes 1992), are being considered.

Combining objects that were not originally designed to work together is a common application for theory of context (McCarthy & Buvač 1994; Farquhar *et al.* 1995). These objects may be databases of facts, programs or hardware. Even if the objects were intended to be used together, maintenance of specifications and terminology is often incomplete. Each object was developed in a *context*, and these contexts differ, whether slightly or greatly. The terminology within each subcontext is likely to be specialized, and making them work together requires some generalization.

The main contribution of this paper is showing that the existing methods for combining entities can be simply modified to allow the combining of plans produced by different planners using different languages. The paper is organized as follows. We first give an overview of the formal theory of context and list some established applications of this theory in AI. This is followed by an exposition of how the theory of context can be used to combine plans which were not intended to be used together. Finally, we give conclusions and an extensive list of references to relevant work on formalizing context.

Features of Context Formalisms

The original proposals for context, (McCarthy 1987; 1993), have influenced a number of researchers to develop logics of context; these include (Shoham 1991; Guha 1991; Buvač & Mason 1993; Giunchiglia *et al.* 1993; Nayak 1994; Attardi & Simi 1995; Buvač, Buvač, & Mason 1995). Although the logics differ in various formal points, here is a list of some features which are shared by most formalizations of context.

1. Contexts are treated as formal objects, i.e. objects in the semantics which can be denoted by constants in the language and over which variables can range. Treating contexts as formal objects allows us to state relations between contexts in the same way we state relations between any other objects in the logic. For example, we can define a relation $specializes(c1, c2)$ which holds when context $c2$ is a specialization of context $c1$.

2. We introduce a new modality *ist* which is pronounced "is true". Formulas of the form $ist(c, p)$ are to be taken as assertions that the proposition p is true in context c. This allows us to express that, for example, John McCarthy is a professor in the context of Stanford University: $ist(Stanford, professor(jmc))$. Note that sentences of the form $ist(c, p)$ can themselves be true in contexts, e.g. we can have $ist(c1, ist(c, p))$. The *ist* modality also allows us to state *lifting axioms*–axioms relating what is true in one context based on what is true in another context. Lifting axioms are central for reasoning with multiple contexts; we will give examples of lifting axioms throughout the following sections.

3. All formulas are stated in some context, so we write $c : \phi$ when formula ϕ is given in context c. There is no outermost context; it is always possible to *transcend* the outermost context so far referred to. Transcendence is discussed in (McCarthy 1993) but not used in this paper.

4. In order to capture the intuitive patterns of contextual reasoning, the formal system needs to contain rules for entering and exiting a context. For example, suppose we have the formula $c0 : ist(c, p)$. We can then *enter* the context c and infer the formula $c : p$. Conversely, if we have the formula $c : p$ we can infer $c0 : ist(c, p)$ by *exiting* the context c. We don't always want to be explicit about the sequence of all the contexts that were entered, but the logic needs to be such that the system always exits into the context it was in before entering. The enter and exit operations can be thought of as being analogous to the push and pop operations on a stack. In the logic presented in (Buvač, Buvač, & Mason 1995), the sequence of contexts that has been entered is always explicitly stated.

5. Besides propositions being true in contexts, we also need the ability to talk about a value of a term t in context c, which we will write as $value(c, t)$. For example, we may need $value(c, time)$, when c is a context that has a time, e.g. a context usable for making assertions about a particular situation. In (McCarthy & Buvač 1994), we show how formulas containing the *value* function can be treated as abbreviations of formulas containing *ist*.

The examples in this paper do not commit us to any particular logic of context, but assume that the language has the above features.

Applications in AI

In this section we summarize the key examples from (McCarthy & Buvač 1994) which motivate the use of contexts as formal mathematical objects in AI.

1. Conventional linguistic applications like the referents of pronouns and anaphora are readily treated by contexts. For example, we need to relate the surgeon's "Scalpel" to the sentence "Please hand me a number 3 scalpel", cf. (Buvač 1996).

2. Defining a theory in a way that permits it to be lifted from a narrow context to a richer outer context. (McCarthy 1993) discusses lifting a simple theory of $above(x, y)$ as the transitive closure of $on(x, y)$ to an outer situation calculus context that uses $on(x, y, s)$ and $above(x, y, s)$. A key formula of that paper is

$$c: \quad (\forall xys)(on(x, y, s) \equiv \qquad (1)$$

$$ist(context\text{-}of\text{-}situation(s), on(x, y))),$$

which relates the three argument situation calculus predicate $on(x, y, s)$ and the two element predicate $on(x, y)$ of the specialized theory of *on* and *above*. The use of contexts to implement "microtheories" in Cyc is described in (Guha 1991). The point is to allow people entering knowledge about some phenomenon to do it in a limited context, but leave open the ability to use the knowledge in a larger context.

3. Defining a narrow context for a problem and importing facts that permit the problem to be solved by considering only the set of salient features. For example, in formulating the well known missionaries and cannibals problem a person or program must take a number of common sense facts into account, but ends up with a 32 state search space because all that is relevant in this context is the number of missionaries, cannibals and boats on each bank of the river.

4. Relating databases with different conventions. Imagine that the Airforce and the General Electric Company have databases both of which include prices for the jet engines that the company sells the Airforce. However, suppose the databases don't agree on what the price covers, e.g. spare parts. We can use one context for the Air Force database, another for the GE database, and a third context that needs to relate information from both. Our theory allows formulas that relate what is true in the 3 contexts involved.

Combining Planning Languages

It is often necessary to combine objects that were not originally designed to work together. Even if the objects were intended to be used together, maintenance of adherence to their specifications and ensuring the consistency of terminology through time is often difficult. Intuitively, our goal is to ensure that no matter what corner the designer paints himself into, what he did can be *lifted* out of it. It has previously been argued, cf. (McCarthy & Buvač 1994; Farquhar *et al.* 1995), that the formal theory of context is useful for performing such tasks when the objects which need to be combined are databases of facts or logical formulas. We proceed to show how the methodology extends to combining subplans.

The basic idea is to represent each subplan as though it was developed in a *context*. These contexts will differ, either slightly or greatly. The terminology within each subcontext is likely to be specialized, and making the plans work together requires some generalization. For this purpose we use *lifting formulas*, i.e. formulas that relate the propositions and terms in subcontexts to propositions and terms in an outer context.

Here is an example. Assume that a route planner, like the route optimization program of the TRAINS project (Allen *et al.* 1995), and a supply planner, like the transportation scheduler developed at Kestrel (Smith, Parra, & Westfold 1995), have been developed independently by different groups. Given a source and a destination, the route planner will find the best route between these places. It however, has no notion of which supplies need to be transported and no notion of time. The supply planner keeps track of the supplies of some economic system and informs us which supplies need to be moved at any given time. We assume that the supply planner has no knowledge about the routes that the supplies need to travel to reach their destination.

To fill in a work order we need to integrate the information produced by the supply planner with that of the route planner. Assume that the supply planner produces

$$\mathit{supply\_planner}: \quad \mathit{transport}(\mathit{equipment1}, \mathit{Rome}, 11/6/95, \mathit{Frankfurt}, 1/20/96) \tag{2}$$

informing us that equipment1 needs to be transported from Rome Air Force base in New York on 11/6/95, to Frankfurt on 1/20/96. The context constant *supply_planner* denotes the context in which supply planner operates and reports its results. Now assume that the route planner tells us that the best route from Rome to Frankfurt is via New York City (NYC). This is represented in the context of the route planner, *route_planner*, by stating

$$\mathit{route\_planner}: \quad [\mathit{Rome}, \mathit{NYC}, \mathit{Frankfurt}] = \mathit{route}(\mathit{Rome}, \mathit{Frankfurt}). \tag{3}$$

Note that *route* is a function returning a list which encodes the best route. Integrating this information inside the problem solving context, *ps*, we get

$$\mathit{ps}: \quad \mathit{transport}(\mathit{equipment1}, [\mathit{Rome}, \mathit{NYC}, \mathit{Frankfurt}], 11/6/95, 1/20/96) \tag{4}$$

stating that equipment1 needs to be transported by the route Rome–NYC–Frankfurt departing on 11/6/95 and arriving on 1/20/96. This information can now be entered into the work order. Note that the same predicate symbol, *transport*, is used in different ways in two different contexts: its arity and its arguments are different in the *supply_planner* context and in the *ps* context.

The context formalism enables us to capture this style of reasoning in logic. We write *lifting axioms* which describe how the information from different contexts can be integrated. In the above example the lifting formula is

$$\mathit{ps}: \quad (\forall x)(\forall l1)(\forall l2)(\forall d1)(\forall d2) \; \mathit{ist}(\mathit{supply\_planner}, \mathit{transport}(x, l1, d1, l2, d2)) \rightarrow \mathit{transport}(x, \mathit{value}(\mathit{route\_planner}, \mathit{route}(l1, l2)), d1, d2). \tag{5}$$

If the formula $\mathit{transport}(x, l1, d1, l2, d2)$ is true in the context of the supply planner, then the formula $\mathit{transport}(x, \mathit{value}(\mathit{route\_planner}, \mathit{route}(l1, l2)), d1, d2)$ holds in the problem solving context. Intuitively, formula 5 expresses that if the supply planner states that some items x need to be transported leaving $l1$ on $d1$ and arriving to $l2$ on $d2$, and if

$\mathit{value}(\mathit{route\_planner}, \mathit{route}(l1, l2))$ is reported by the route planner as the best route from $l1$ to $l2$, then the information which can be entered into the work order is $\mathit{transport}(x, \mathit{value}(\mathit{route\_planner}, \mathit{route}(l1, l2)), d1, d2)$. In other words, formula 5 specifies the integrating of a plan which involves the notions of time and supplies produced by the supply planner with the details involving a route produced the route planner. The lifting axiom 5 allows us to derive the plan given by formula 4 in the problem solving context *ps* from the plans given by formulas 2 and 3 in the contexts of their corresponding planners.

A term with a definite meaning in one context often needs translation when used in another context. Thus *Rome* may mean Rome NY in a data base of US Air Force bases but needs translation when a formula is lifted to a context of worldwide geography. Lifting formulas similar to 5 can be used to do this type of translation, cf. (McCarthy & Buvač 1994; Buvač & Fikes 1995).

Discharging Kindness Assumptions

Any plan produced from the lifting axiom 5 makes numerous assumptions. For example, it assumes that the shortest path will always get the cargo to its destination on time. Although this assumption is usually valid, we can imagine a scenario in which an urgent delivery will need to take a longer route in order to get to its destination on time. We thus need to consider the timeliness of a path in scenarios which involve urgent deliveries.

In robotics, assumptions of this sort are commonly called *kindness assumptions*, cf. (Nourbakhsh & Genesereth 1996), because they amount to assuming that the world is kind, i.e. that things will turn out in our favor most of the time. Kindness assumptions are a useful tool and are commonly made both when constructing and integrating plans. They allow us to focus on the aspects of a plan that seem to be relevant to the problem at hand and to disregard details which we assume will hold for that particular problem class. However, whenever kindness assumptions are made it is important to have a mechanism which enables us to discharge such assumptions and reason about their validity in cases when it is unclear whether they hold. The context formalisms enables us to do this in the framework of logic.

Assume that after deriving the plan in formula 4 (by integrating the plans from the route and supply planning contexts) we realize that the delivery is needed urgently. At this point our goal is to discharge the timeliness assumption and take the proposed path through NYC only if it gets equipment1 to Frankfurt on time. The desired plan, which is given in an urgent problem solving context *ps_urgent*, is thus

$$\begin{aligned} \mathit{ps\_urgent}: \quad & \mathit{timely\_route}(11/6/95, 1/20/96, \\ & [\mathit{Rome}, \mathit{NYC}, \mathit{Frankfurt}]) \rightarrow \\ & \mathit{transport}(\mathit{equipment1}, \\ & [\mathit{Rome}, \mathit{NYC}, \mathit{Frankfurt}], 11/6/95, 1/20/96) \end{aligned} \tag{6}$$

where deciding whether *timely_route* holds will involve looking up airplane schedules and local delivery facilities in some data base. We are assuming that conditional plans, like formula 6, can be represented by the system. In the general case, formula 6 follows from formula 4 and the lifting axiom

$$\begin{aligned} \mathit{ps\_urgent}: \quad & (\forall x)(\forall r)(\forall d1)(\forall d2) \\ & \mathit{ist}(\mathit{ps}, \mathit{transport}(x, r, d1, d2)) \rightarrow \\ & \mathit{timely\_route}(d1, d2, r) \rightarrow \mathit{transport}(x, r, d1, d2). \end{aligned} \tag{7}$$

In some planning instances we will want to consider the timeliness issues at the very outset. We can avoid using the original problem solving context *ps* by inferring a lifting theorem which integrates a plan from the route planner and a plan from the supply planner to directly produce a plan in *ps_urgent*

$$\begin{aligned} \mathit{ps\_urgent}: \quad & (\forall x)(\forall r)(\forall l1)(\forall l2)(\forall d1)(\forall d2) \\ & (\mathit{ist}(\mathit{supply\_planner}, \mathit{transport}(x, l1, l2, d1, d2)) \wedge \\ & \mathit{timely\_route}(d1, d2, \mathit{value}(\mathit{route\_planner}, \mathit{route}(l1, l2)))) \rightarrow \\ & \mathit{transport}(x, \mathit{value}(\mathit{route\_planner}, \mathit{route}(l1, l2)), d1. d2) \end{aligned} \tag{8}$$

Formula 8 logically follows from the lifting axioms given in formula 5 and formula 7.

Conclusion

Integrating plans which were not originally designed to be used together is a task that frequently comes up in real world applications. However, this task is typically performed by humans. Contexts enable us to formalize this style of reasoning, thus providing a logical basis for developing computer programs which will be able to mechanically integrate plans produced by different systems.

Acknowledgments

The authors would like to thank Eyal Amir, Tom Costello and Jeaneah Paik for their valuable comments.

This research is supported in part by the ARPA/ONR grant number N00014-94-1-0775.

References

Allen, J. F.; Schubert, L. K.; Ferguson, G.; Heeman, P.; Hwang, C. H.; Kato, T.; Light, M.; Martin, N. G.; Miller, B. W.; Poesio, M.; and Traum, D. R. 1995. A formalization of viewpoints. *Journal of Experimental and Theoretical AI* 7:7–48.

Attardi, G., and Simi, M. 1995. A formalization of viewpoints. *Fundamenta Informaticae* 23(3).

Buvač, S., and Fikes, R. 1995. A declarative formalization of knowledge translation. In *Proceedings of the ACM CIKM: The 4th International Conference on Information and Knowledge Management.*

Buvač, S., and Mason, I. A. 1993. Propositional logic of context. In *Proceedings of the Eleventh National Conference on Artificial Intelligence.*

Buvač, S.; Buvač, V.; and Mason, I. A. 1995. Metamathematics of contexts. *Fundamenta Informaticae* 23(3).

Buvač, S. 1996. Resolving lexical ambiguity using a formal theory of context. In van Deemter, K., and Peters, S., eds., *Semantic Ambiguity and Underspecification.* CSLI Lecture Notes. To appear.

Farquhar, A.; Dappert, A.; Fikes, R.; and Pratt, W. 1995. Integrating information sources using context logic. Technical Report KSL-95-12, Knowledge Systems Laboratory, Stanford University. Also appears in the 1995 AAAI Spring Symposium Series on Information Gathering from Distributed, Heterogeneous Environments.

Finin, T., and Mayfield, J. 1995. *Workshop on Intelligent Information Agents.* Held in conjunction with the Fourth International Conference on Intelligent Information Agents.

Genesereth, M. R., and Fikes, R. E. 1992. Knowledge interchange format, version 3.0, reference manual. Technical Report Logic Group Report Logic-92-1, Stanford University.

Giunchiglia, F.; Serafini, L.; Giunchiglia, E.; and Frixione, M. 1993. Non-omniscient belief as context-based reasoning. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence.*

Guha, R. V., and Lenat, D. B. 1990. Cyc: A midterm report. *AI Magazine* 11(3):32 59.

Guha, R. V. 1991. *Contexts: A Formalization and Some Applications.* Ph.D. Dissertation, Stanford University. Also technical report STAN-CS-91-1399-Thesis, and MCC Technical Report Number ACT-CYC-423-91.

Knoblock, C.; Levy, A.; Chen, S.-S.; and Wiederhold, G. 1995. Information gathering form heterogeneous, distributed environments. Technical report, AAAI Spring Symposium.

Lenat, D. B. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the Association for Computing Machinery* 38 (11).

McCarthy, J., and Buvač, S. 1994. Formalizing Context (Expanded Notes). Technical Note STAN-CS-TN-94-13, Stanford University.

McCarthy, J. 1987. Generality in artificial intelligence. *Comm. of ACM* 30(12):1030–1035. Also in *ACM Turing Award Lectures, The First Twenty Years*, ACM Press, 1987; and reprinted in (McCarthy 1990).

McCarthy, J. 1990. *Formalizing Common Sense: Papers by John McCarthy.* 355 Chestnut Street, Norwood, NJ 07648: Ablex Publishing Corporation.

McCarthy, J. 1993. Notes on formalizing context. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence.*

Nayak, P. P. 1994. Representing multiple theories. In *Proceedings of the Twelfth National Conference on Artificial Intelligence.*

Nourbakhsh, I. R., and Genesereth, M. R. 1996. Making assumptions to simplify planning. *Autonomous Robots* 1(3).

Patil, R. S.; Fikes, R. E.; Patel-Schneider, P. F.; Mckay, D.; Finin, T.; Gruber, T.; and Neches, R. 1992. The DARPA knowledge sharing effort: Progress report. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning.*

Shoham, Y. 1991. Varieties of context. In Lifschitz, V., ed., *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy.* Academic Press.

Smith, D. R.; Parra, E. A.; and Westfold, S. J. 1995. Synthesis of High-Performance Transportation Schedulers. Technical Report KES.U.95.1, Kestrel Institute. http://kestrel.edu/www/publications.html.