Making Hybrid Plans More Clear to Human Users

-

A Formal Approach for Generating Sound Explanations

Bastian Seegebarth, Felix Müller,
Bernd Schattenberg, and Susanne Biundo

Institute of Artificial Intelligence

June 28th, 2012

ulm university universität uulm

Motivation – Transparent Decision Support

- Applications of planning technology:
  - Emergency planning
  - Assistance of cognitively impaired and elderly people
  - Support in daily activities
- Planning systems provide recommendations when and how to act to subjects who are competent themselves
- Humans might scrutinize the systems' suggestions

To prevent this a planning system must be able to:

- Give reasons for decisions
- Present them in a comprehensible manner
- Build trust in the system's competence

Our Contribution

- Provide a formal framework for the generation of (raw) *plan explanations*
- Raw Explanations can be used as input to a text generation system to produce the actual explanation
- Implement the explanation for the existence of actions and orderings on actions in a plan in the framework, i.e., the system can answer the questions:
  - Why does the user have to execute a given action from the plan?
  - Why does the user have to execute two actions from the plan in the order given in the plan?
- Implement a prototype explanation system

Hybrid Planning – Problem Formalization

Hybrid planning framework combines classical and hierarchical planning

$\pi = \langle \mathcal{T}, \mathcal{M}, P_{init} \rangle$ is a hybrid planning problem with

- $\mathcal{T}$ is a set of task schemata of the form $\langle t(\bar{v}), pre, eff \rangle$, $\bar{v}$ is the parameter list of t
- $\mathcal{M}$ is a set of decomposition methods of the form $\langle t(\bar{v}), P \rangle$, P is a partial plan
- P_{init} is the initial partial plan that needs to be decomposed
  - plan step *init* has the initial state as effects
  - plan step *goal* has the goals as preconditions

Hybrid Planning – Partial Plans and Solutions

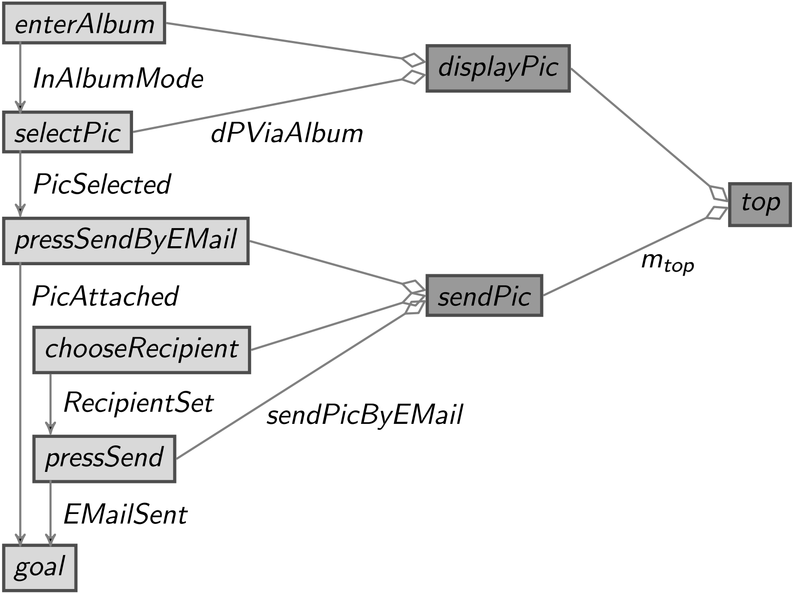
$P = \langle PS, \prec, V, C \rangle$ is a partial plan with

- PS is a set of plan steps of the form $s{:}t(\bar{v})$, s is a unique label and $t(\bar{v})$ is a (partially) instantiated task
- $\prec$ is a partial order on PS
- V is a set of constraints on the variables appearing in PS
- C is a set of causal links of the form $\langle s \rightarrow_p s' \rangle$

A partial plan is a solution to a planning problem if

- Every precondition is established by a causal link
- No causal links are threatened
- P contains only primitive tasks and can be obtained from P_{init} by decomposition of abstract tasks and the insertion of plan steps, causal links, ordering, and variable constraints

Running Example



Basic Framework

- Formalize information about the plan, its construction process, and basic arguments as a first-order logic axiomatic system
- Construct an explanation by finding a proof for a formula that represents the requested aspect
- Elements of axiomatic system are based on the problem specification, the construction process that led to the plan, and the underlying planning formalism
- Explanations are provably correct w.r.t. the underlying planning system

Causal and Decomposition Structure

Construct axiomatic system Σ from plan $P = (PS, \prec, V, C)$.

Causal Structure:

- Add $CR(s, p, s')$ to Σ for every causal link $\langle s \rightarrow_p s' \rangle \in C$

Decomposition Structure:

- Add $DR(s, m, s')$ to Σ if s was introduced by the decomposition of s' via method m

Examples

Examples:

- $CR(enterAlbum, InAlbumMode, selectPic)$
- $CR(selectPic, PicSelected, pressSendByEMail)$
- $DR(pressSendByEMail, sendPicByEMail, sendPic)$
- $DR(sendPic, mTop, top)$

Basic Explanations

$Nec(s)$ denotes that s is necessary for the plan to be a solution (this does not mean that there cannot be a plan without s)

A plan step is necessary if it establishes a goal:

- $\forall s.[[\exists g.CR(s, g, goal)] \Rightarrow Nec(s)]$

...or if it establishes a precondition of a plan step that is necessary:

- $\forall s.[[\exists s', p.[CR(s, p, s') \wedge Nec(s')]] \Rightarrow Nec(s)]$

A plan step is necessary if it is a step of the initial partial plan:

- $\forall s.[DR(s, m_{top}, top) \Rightarrow Nec(s)]$

...or if it is a sub step of a step from the initial partial plan:

- $\forall s.[[\exists s', m.[DR(s, m, s') \wedge Nec(s')]] \Rightarrow Nec(s)]$

A First Explanation

- $\forall s.[[\exists g.CR(s, g, goal)] \Rightarrow Nec(s)]$
- $\forall s.[[\exists s', p.[CR(s, p, s') \wedge Nec(s')]] \Rightarrow Nec(s)]$
- $\forall s.[DR(s, m_{top}, top) \Rightarrow Nec(s)]$
- $\forall s.[[\exists s', m.[DR(s, m, s') \wedge Nec(s')]] \Rightarrow Nec(s)]$

To explain why executing *enterAlbum* is necessary:

1. $Nec(enterAlbum)$
2. $CR(enterAlbum, InAlbumMode, selectPic)$
3. $Nec(selectPic)$
4. $CR(selectPic, PicSelected, pressSendByEMail)$
5. $Nec(pressSendByEMail)$
6. $DR(pressSendByEMail, sendPicByEMail, sendPic)$
7. $Nec(sendPic)$
8. $DR(sendPic, m_{top}, top)$

Translation to Natural Language

1. $Nec(enterAlbum)$
2. $CR(enterAlbum, InAlbumMode, selectPic)$
3. $Nec(selectPic)$
4. $CR(selectPic, PicSelected, pressSendByEMail)$
5. $Nec(pressSendByEMail)$
6. $DR(pressSendByEMail, sendPicByEMail, sendPic)$
7. $Nec(sendPic)$
8. $DR(sendPic, m_{top}, top)$

In natural language (Future Work):
"Entering the album is necessary to select the picture. You must select the picture in order to use the *send by EMail...*-function. That is a necessary sub step of sending the picture which is part of your initial problem specification."

Levels of Abstraction for Explanations

- Explaining only on primitive level leads to overly long and detailed explanations
- Level of detail should be variable for different parts of an explanation
  - Explain on primitive levels for parts of the plan that the user is not familiar with
  - Skip over other parts by explaining on high level of abstraction
- Through decomposition relations the explanation can be moved to higher level of abstraction

Problem: the Causal Structure of a plan is usually given only in terms of primitive plan steps

Therefore, abstract plan steps can only be explained via decomposition relations

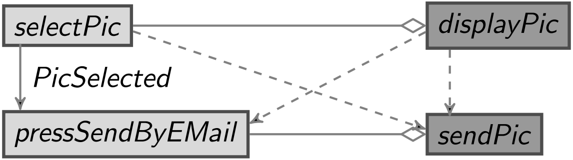
Causal Relations for Abstract Plan Steps

What is the causality produced and consumed by an abstract plan step?

Let plan steps inherit causal relations from their sub steps:

- $\forall s, p, s'.[[\exists m, s''.[DR(s'', m, s) \wedge CR(s'', p, s')]] \Rightarrow CR(s, p, s')]$
- $\forall s, p, s'.[[\exists m, s''.[DR(s'', m, s') \wedge CR(s, p, s'')]] \Rightarrow CR(s, p, s')]$

Example:



Causal Relations over Abstract State Features

Sometimes inherited causal relations do not seem reasonable:

- $CR(sendPic, EMailSent, goal)$
- $CR(sendPic, PicAttached, goal)$

Introduce set of decomposition axioms:

- $PicTransferred \Leftrightarrow [PicAttached \wedge EMailSent] \vee [PicPrinted \wedge FaxSent]$

Derive causal relations over abstract state features:

- $\forall s, s', p, p'.[[CR(s, p, s') \wedge CR(s, p', s')] \Rightarrow CR(s, p \text{ and } p', s')]$
- $\forall s, s', a.[[\exists da, d.[AbsL(a, da) \wedge FDec(d, da) \wedge CR(s, d, s')]] \Rightarrow CR(s, a, s')]$

Example:

- From the above causal relations and axioms we can derive: $CR(sendPic, PicTransferred, goal)$

Experimental System & Discussion

- We have implemented a prototype system to generate explanations as specified by the formal framework
- Thousands of explanations can be found within a few seconds
- How to select among the abundance of possible explanations?
  - Size of explanation
  - Type of arguments
  - Means for presentation (text, graphics, speech)
  - Existing user knowledge

Summary

- Communication of plans is crucial for the acceptance of planning technology
- We presented a general framework for the generation of explanations
- Instantiation of the framework for the explanation of plan steps and the ordering of plans
- Future work has to deal with the presentation and selection of raw explanations