Supporting Interleaved Plans in Learning Hierarchical Plan Libraries for Plan Recognition

Martín Marchetta
mmarchetta@fing.uncu.edu.ar

Raymundo Forradellas (PhD)
kike@uncu.edu.ar

Universidad Nacional de Cuyo
Mendoza - Argentina

Centro de Estudios y Aplicaciones Logísticas

Plan Recognition

- What is it?
  - Inferring an observed agent's intentions, sufficient to explain its behaviour
- Useful to:
  - Provide intelligent assistance to a human user
  - Infer knowledge-rich intentions in knowledge-rich domains

Plan Recognition

- Most plan recognition techniques rely on a plan library, and assume that this plan library is complete
- Manual creation of plan libraries is difficult, time consuming and error prone
- Plan libraries can require much maintenance, specially when the user can pursue plans not considered during the plan library creation

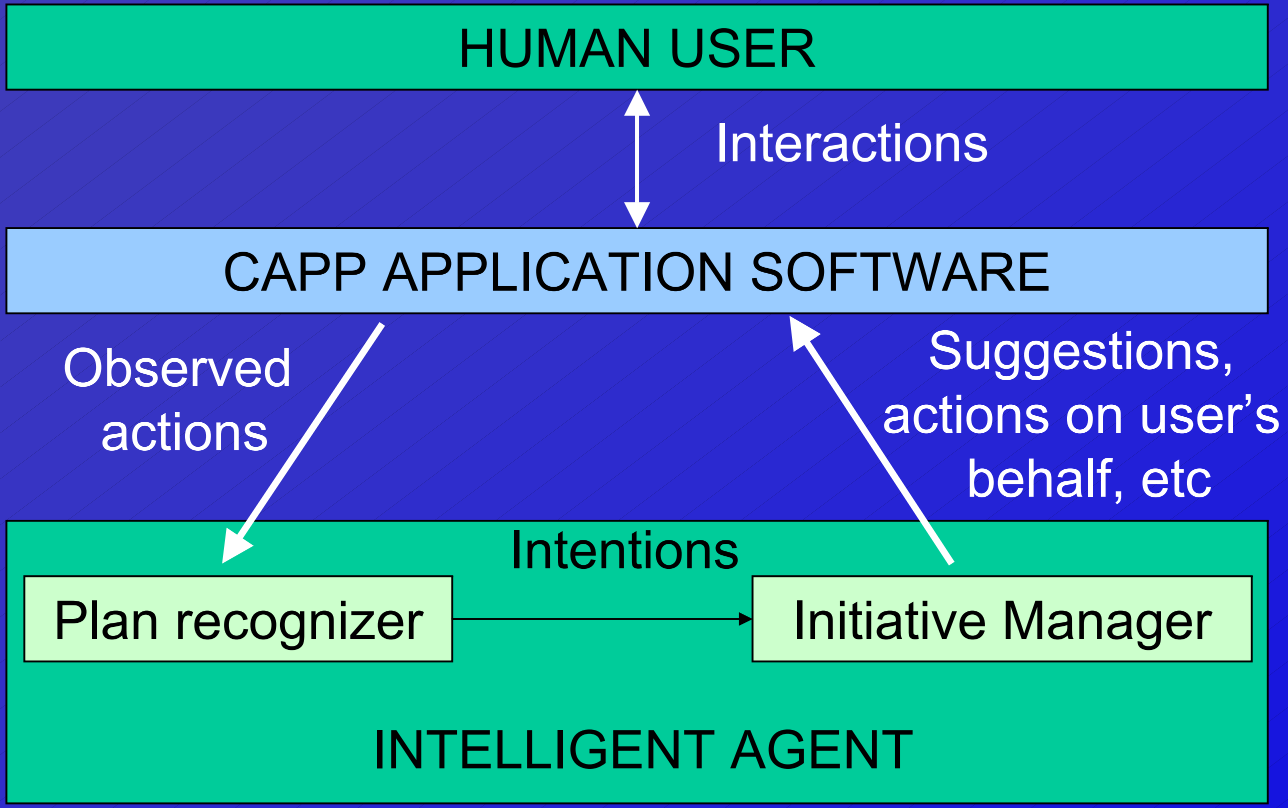
Our application domain

- Computer Aided Process Planning
  - Process planning:
    - Is the selection of production methods (machines, machining operations, etc)
  - Process planning includes selection of:
    - Machining operations sequence
    - Machine type
    - Tools and fixturing
    - Assembly sequence

Computer Aided Process Planning

- Current approaches
  - Variant: Retrieve and adapt plan templates from a database
  - Generative: Create new plans based on machining features
- Current issues
  - Variant:
    - Is almost manual (creation and retrieval of plan templates)
    - Requires much user knowledge, and maintenance
  - Generative:
    - Requires complete features information
    - Requires much knowledge representation (production rules)
    - It's hard for the human planner to contribute with his experience

Our approach to CAPP



Plan Libraries Generation Issues

- Some works address the automatic generation of plan libraries. These approaches have some of the following drawbacks:
  - They learn little expressive libraries
  - They require an explicit training phase
  - They use supervised learning
- Our learning algorithm:
  - Learns event decompositions
  - Supports interleaved plans
  - Uses unsupervised learning

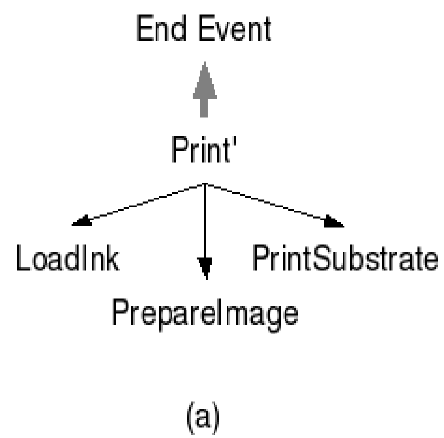
Learning of event decompositions

- Event decomposition: a set of simpler actions that must be done in order to accomplish it

- Our learning mechanism identifies unknown sequences, as well as known complete and partial decompositions in the observations
  - A **partial decomposition** of a non-primitive event is a set containing only a part of the decomposition of that event
  - A **complete decomposition** of a non-primitive event is a set containing its full decomposition (i.e, all its parts)

Learning of event decompositions

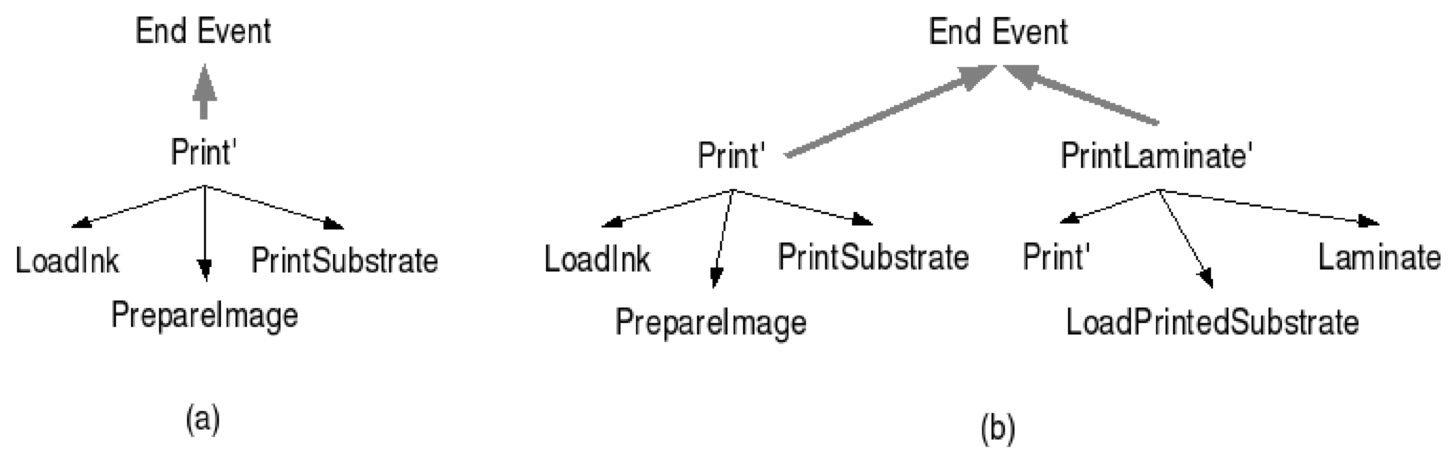
- Each time an unknown set of actions is processed, a new event is created in the plan library



(a) AS_1 = {LoadInk, PrepareImage, PrintSubstrate}

Learning of event decompositions

- When complete decompositions are identified, they are replaced by their corresponding non-primitive event in the sequence

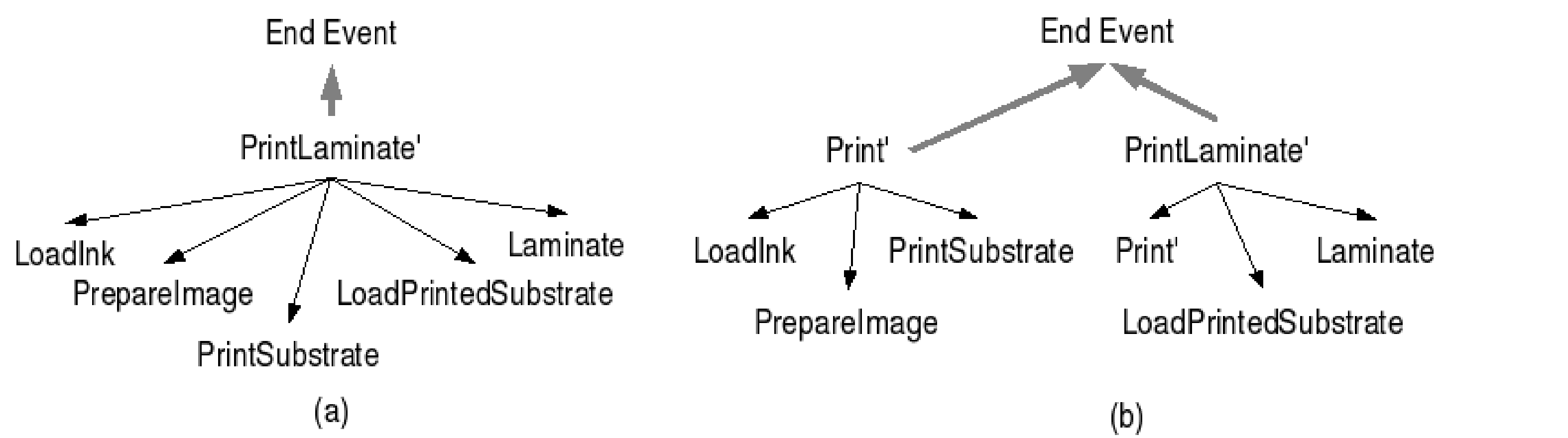


(a) AS_1 = {LoadInk, PrepareImage, PrintSubstrate}

(b) AS_2 = {LoadInk, PrepareImage, PrintSubstrate, LoadPrintedSubstrate, Laminate}

Learning of event decompositions

- When partial decompositions are identified, they are considered a new event, and are inserted in the plan library as needed

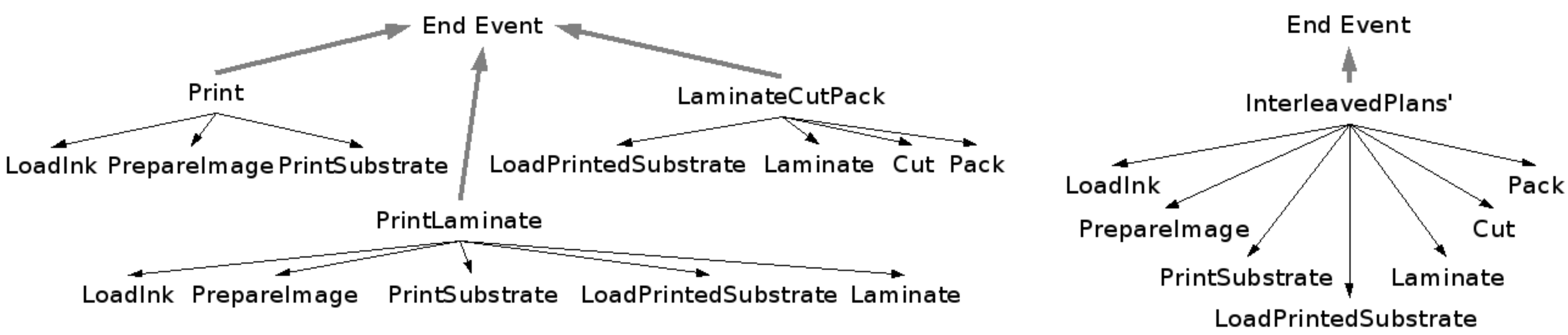


(a) AS_1 = {LoadInk, PrepareImage, PrintSubstrate, LoadPrintedSubstrate, Laminate}

(b) AS_2 = {LoadInk, PrepareImage, PrintSubstrate}

Interleaved plans

- Humans usually execute interleaved plans
- Learning from observations that may contain interleaved plans is difficult. For example:



(a) Real library (b) Learned library

AS_1 = {LoadInk, PrepareImage, PrintSubstrate, LoadPrintedSubstrate, Laminate, Cut, Pack}

Interleaved plans

- Our approach: Detect and correct interleaved plans

$$N(A_T) = N(A_{\neg int}) + N(A_{int})$$

$$N(A_{int}) = N(A_{1e}) + N(A_{2e}) + \ldots + N(A_{ne})$$

$$N(A_T) = \underbrace{N(A)}_{N(A_{\neg int})} + \underbrace{N(A_{1e}) + N(A_{2e}) + \ldots + N(A_{ne})}_{N(A_{int})}$$

$$P(A_{int}) = \frac{N(A_{int})}{N(A_T)} = \frac{N(A_{1e}) + N(A_{2e}) + \ldots + N(A_{ne})}{N(A_T)}$$

Where:

$P(A_{int})$	Probability that A is in fact an interleaving of plans
$N(A_{int})$	Approximation of the number of observations where A is an interleaving of plans
$N(A_T)$	Number of observations where A appears (completely or partially)
A_i	Event that is part of A
$N(A_{ie})$	Number of observations where the i-th part of A appears as end event

Interleaved plans

- $P(A_{int})$ can be used in several ways to detect interleaved plans:
  - Comparing $P(A_{int})$ with $P(A_{-int})$
  - Using a threshold for $P(A_{int})$ beyond which A is considered and interleaving of plans rather than a real single plan
  - Using a threshold for the rate $P(A_{int})/P(A_{-int})$

Conclusions

- Plan recognition techniques can provide useful capabilities to intelligent assistants in knowledge rich domains (such as Computer Aided Process Planning)
- Hand-coding plan libraries is tedious an error prone
- An unsupervised learning mechanism of hierarchical plan libraries was presented, that supports the detection and correction of interleaved plans
- In the future, more features of the plan library need to be learned (ordering and causal restrictions, actions parameters, abstractions, errors detection and correction)

Questions and opinions