Validating Procedural Knowledge in the Open Virtual Collaboration Environment

Gerhard Wickler
AIAI, University of Edinburgh, UK
`g.wickler@ed.ac.uk`

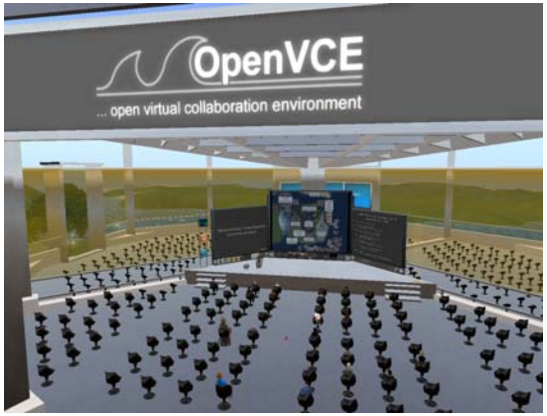
OpenVCE
... open virtual collaboration environment

Intelligent Systems Track
@ ISCRAM 2013

Introduction and Overview

- crisis response (planning) is a collaborative effort
  - idea: use new media technologies to support task-centric collaboration (use procedural knowledge)
  - problem: development and validation of procedures

- Overview
  - Procedural Knowledge and OpenVCE
  - OpenVCE Workflow
  - Validating Procedural Knowledge
  - Evaluation and Future Work
  - Conclusions



Procedural Knowledge

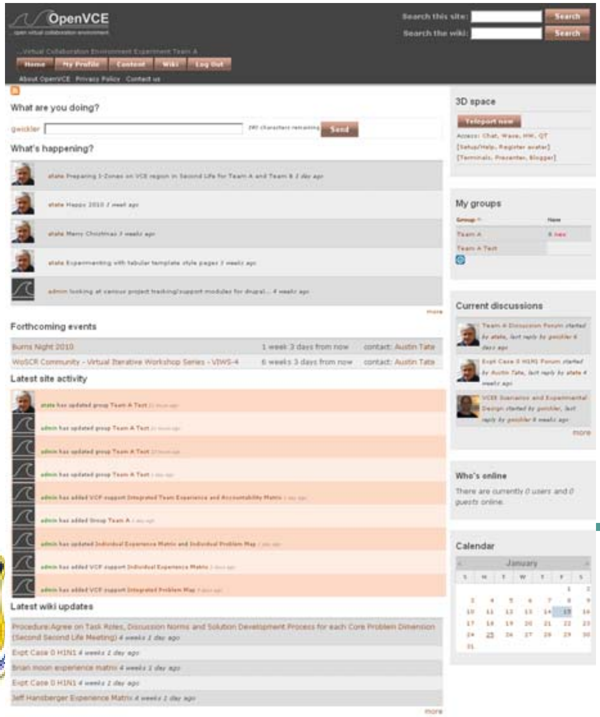
- SOPs: manual describing courses of action
  - represent best-practise knowledge; authored by experts
  - mostly available in form of books; used for teaching and training

- problems (with use in emergencies):
  - access time: finding best procedure in (large) body of text takes time, when time is short
  - structure: procedure described in free text form; must read all to find specific information
  - updating: procedure changes over time; old knowledge tends to persist, especially in people

- structure: hierarchical task networks

OpenVCE

- OpenVCE: supports collaboration in virtual spaces
  - website based on open source content management system (Drupal + MediaWiki) for asynchronous collaboration
  - 3D virtual space for synchronous collaboration (Second Life)
  - virtual collaboration protocol: procedure for using OpenVCE



OpenVCE

open virtual collaboration environment

Search this site: Search

Search the wiki: Search

...Virtual Collaboration Environment Experiment Team A

Home | My Profile | Content | Wiki | Log Out

About OpenVCE | Privacy Policy | Contact us

What are you doing?

gwickler [ ] 140 characters remaining Send

What's happening?

- atate Preparing I-Zones on VCE region in Second Life for Team A and Team B 1 day ago
- atate Happy 2010 1 week ago
- atate Merry Christmas 3 weeks ago
- atate Experimenting with tabular template style pages 3 weeks ago
- admin looking at various project tracking/support modules for drupal... 4 weeks ago

more

Forthcoming events

Burns Night 2010	1 week 3 days from now	contact: Austin Tate
WoSCR Community - Virtual Iterative Workshop Series - VIWS-4	6 weeks 3 days from now	contact: Austin Tate

Latest site activity

- atate has updated group Team A Test
- admin has updated group Team A Test
- admin has updated group Team A Test
- admin has updated group Team A Test
- admin has added VCP support Integrated Team Experience and Accountability Matrix
- admin has added Group Team A
- admin has updated Individual Experience Matrix and Individual Problem Map
- admin has added VCP support Individual Experience Matrix
- admin has added VCP support Integrated Problem Map

Latest wiki updates

- Procedure:Agree on Task Roles, Discussion Norms and Solution Development Process for each Core Problem Dimension (Second Second Life Meeting) 4 weeks 1 day ago
- Expt Case 0 H1N1 4 weeks 1 day ago
- Brian moon experience matrix 4 weeks 1 day ago
- Expt Case 0 H1N1 4 weeks 1 day ago
- Jeff Hansberger Experience Matrix 4 weeks 1 day ago

more

3D space

Teleport now

Access: Chat, Wave, HW, QT

[Setup/Help, Register avatar]

[Terminals, Presenter, Blogger]

My groups

Group	New
Team A	6 new
Team A Test	

Current discussions

- Team A Discussion Forum started by atate, last reply by gwickler 6 days ago
- Expt Case 0 H1N1 Forum started by Austin Tate, last reply by atate 4 weeks ago
- VCEE Scenarios and Experimental Design started by gwickler, last reply by gwickler 6 weeks ago

more

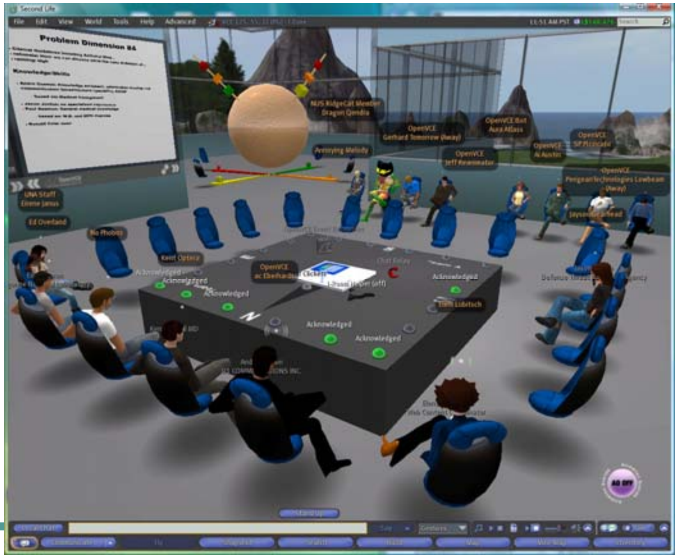
Who's online

There are currently 0 users and 0 guests online.

Calendar

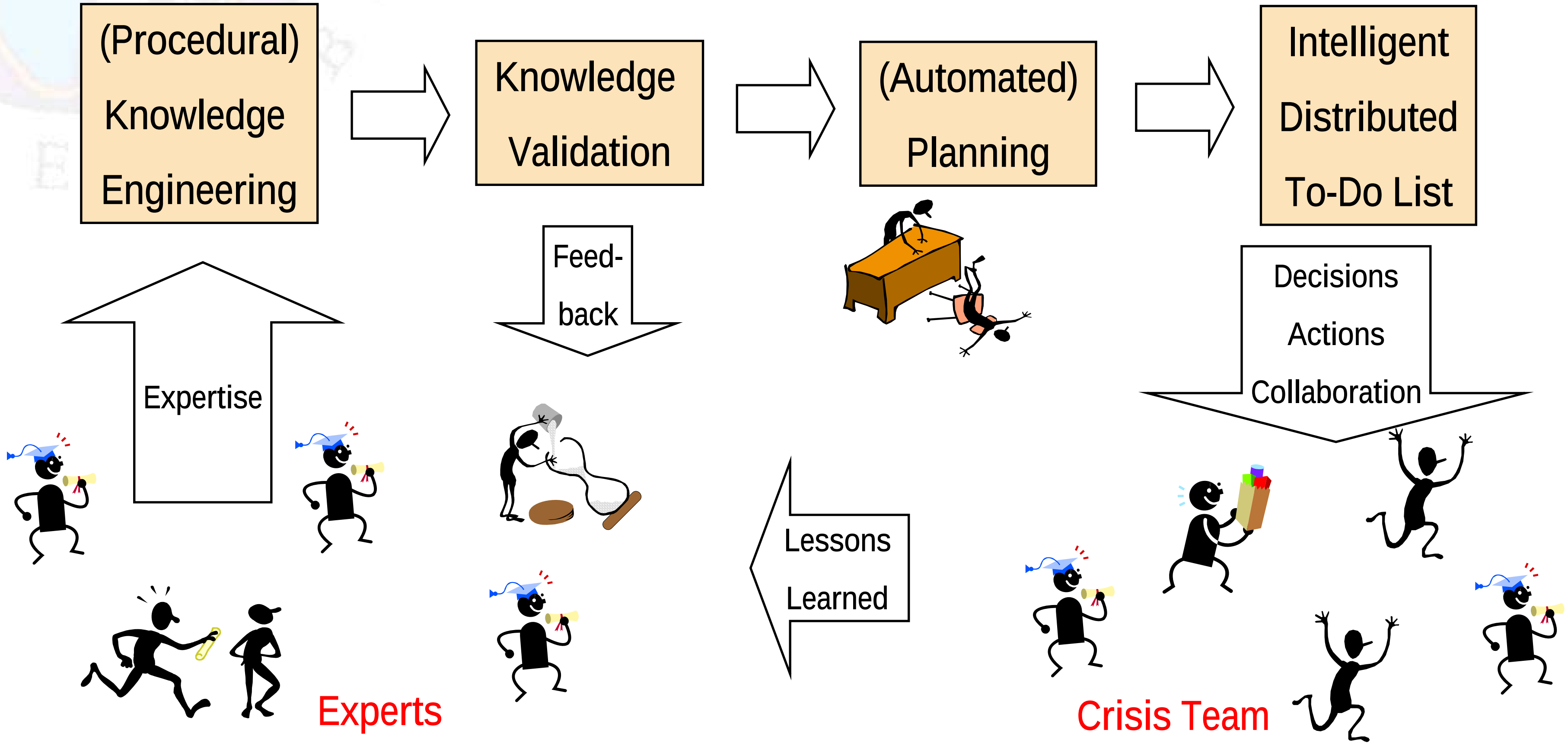
January

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						



Intelligent Systems Track
@ ISCRAM 2013

Using Procedural Knowledge in OpenVCE



(Procedural) Knowledge Engineering

Knowledge Validation

(Automated) Planning

Intelligent Distributed To-Do List

Expertise

Feed-back

Decisions

Actions

Collaboration

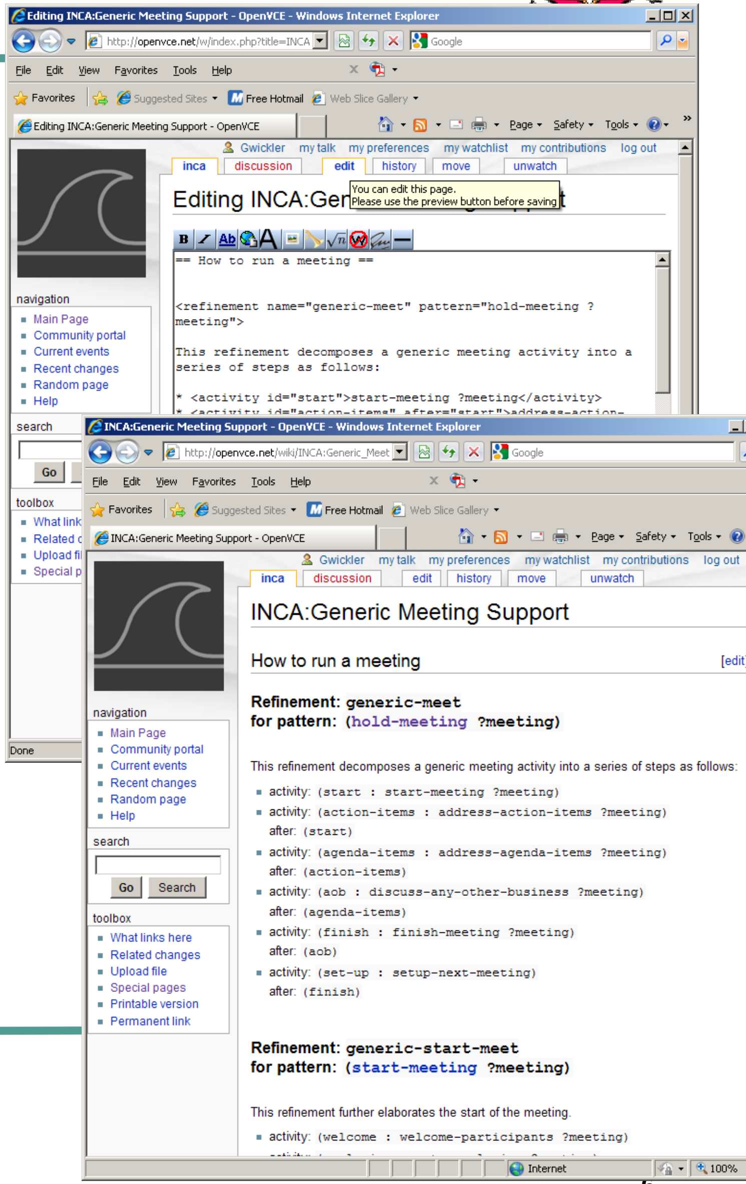
Lessons Learned

Experts

Crisis Team

Collaborative SOP Development

- idea: use wiki as tool for collaborative SOP development
  - MediaWiki: open source, scalable, robust tool
  - SOP/<I-N-C-A> extension: supports structured procedural knowledge
- creating an SOP on the wiki:
  - write the unstructured procedure as free text
  - divide into articles; one method/refinement per article
  - mark up articles using extension tags
- validating procedural knowledge:
  - automatic analysis of formal aspects
  - test for consistency
  - revise procedural knowledge


Editing INCA:Generic Meeting Support - OpenVCE - Windows Internet Explorer
http://openvce.net/w/index.php?title=INCA
Google
File Edit View Favorites Tools Help
Favorites
Suggested Sites
Free Hotmail
Web Slice Gallery
Editing INCA:Generic Meeting Support - OpenVCE
Page Safety Tools
Gwickler my talk my preferences my watchlist my contributions log out
inca discussion edit history move unwatch
You can edit this page.
Please use the preview button before saving
navigation
Main Page
Community portal
Current events
Recent changes
Random page
Help
search
Go
toolbox
What link
Related c
Upload fi
Special p
Done
INCA:Generic Meeting Support - OpenVCE - Windows Internet Explorer
http://openvce.net/wiki/INCA:Generic_Meet
Google
File Edit View Favorites Tools Help
Favorites
Suggested Sites
Free Hotmail
Web Slice Gallery
INCA:Generic Meeting Support - OpenVCE
Page Safety Tools
Gwickler my talk my preferences my watchlist my contributions log out
inca discussion edit history move unwatch
navigation
Main Page
Community portal
Current events
Recent changes
Random page
Help
search
Go Search
toolbox
What links here
Related changes
Upload file
Special pages
Printable version
Permanent link
[edit
Internet
100%

Editing INCA:Ger

```
== How to run a meeting ==


<refinement name="generic-meet" pattern="hold-meeting ?
meeting">

This refinement decomposes a generic meeting activity into a
series of steps as follows:

* <activity id="start">start-meeting ?meeting</activity>
```

INCA:Generic Meeting Support

How to run a meeting

Refinement: `generic-meet`
for pattern: `(hold-meeting ?meeting)`

This refinement decomposes a generic meeting activity into a series of steps as follows:

- activity: `(start : start-meeting ?meeting)`
- activity: `(action-items : address-action-items ?meeting)`
  after: `(start)`
- activity: `(agenda-items : address-agenda-items ?meeting)`
  after: `(action-items)`
- activity: `(aob : discuss-any-other-business ?meeting)`
  after: `(agenda-items)`
- activity: `(finish : finish-meeting ?meeting)`
  after: `(aob)`
- activity: `(set-up : setup-next-meeting)`
  after: `(finish)`

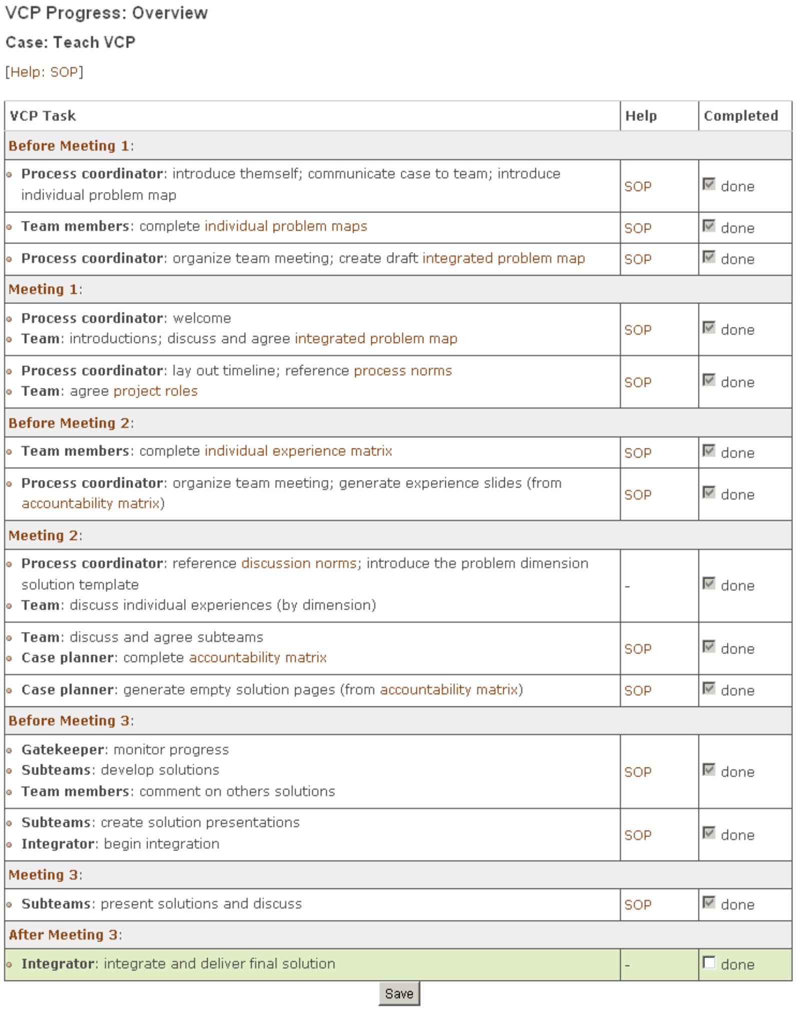
Refinement: `generic-start-meet`
for pattern: `(start-meeting ?meeting)`

This refinement further elaborates the start of the meeting.

- activity: `(welcome : welcome-participants ?meeting)`

Distributed Use of Procedures

- export (formal) procedural knowledge to planning tool
- result: hierarchical task network
  - integrated into OpenVCE website
  - linked to procedural knowledge in wiki
  - linked to terminology in wiki
- plan execution:
  - people execute tasks
  - use capability model of agents involved to distribute tasks
  - link tasks to code for automatic execution (e.g. tasks related to 3D virtual world)



VCP Progress: Overview

Case: Teach VCP

[Help: SOP]

VCP Task	Help	Completed
Before Meeting 1:		
Process coordinator: introduce themself; communicate case to team; introduce individual problem map	SOP	☑ done
Team members: complete individual problem maps	SOP	☑ done
Process coordinator: organize team meeting; create draft integrated problem map	SOP	☑ done
Meeting 1:		
Process coordinator: welcome **Team**: introductions; discuss and agree integrated problem map	SOP	☑ done
Process coordinator: lay out timeline; reference process norms **Team**: agree project roles	SOP	☑ done
Before Meeting 2:		
Team members: complete individual experience matrix	SOP	☑ done
Process coordinator: organize team meeting; generate experience slides (from accountability matrix)	SOP	☑ done
Meeting 2:		
Process coordinator: reference discussion norms; introduce the problem dimension solution template **Team**: discuss individual experiences (by dimension)	-	☑ done
Team: discuss and agree subteams **Case planner**: complete accountability matrix	SOP	☑ done
Case planner: generate empty solution pages (from accountability matrix)	SOP	☑ done
Before Meeting 3:		
Gatekeeper: monitor progress **Subteams**: develop solutions **Team members**: comment on others solutions	SOP	☑ done
Subteams: create solution presentations **Integrator**: begin integration	SOP	☑ done
Meeting 3:		
Subteams: present solutions and discuss	SOP	☑ done
After Meeting 3:		
Integrator: integrate and deliver final solution	-	□ done

Planning Domain Validation

- problem formulation: planning domain + problem define search space; essential for efficient planning
- idea:
  - explicitly represent (redundant) domain features
  - automatically (and efficiently) extract same features
  - ensure consistency of the representation
- overview (domain features):
  - static vs. fluent relations
  - domain types
  - reversible actions
  - inconsistent effects
- PDDL: contains types (optional); other features not supported

Static vs. Fluent Relations

- **example: DWR domain**

```
(:predicates
      :static (adjacent ?l1  ?l2 - location)
      :fluent (at ?r - robot ?l - location)
```

etc.

```
(:action move
      :parameters (?r - robot ?from ?to - location)
      :precondition (and
               (adjacent ?from ?to) (at ?r ?from)
               (not (occupied ?to)))
      :effect (and
               (at ?r ?to) (not (occupied ?from))
               (occupied ?to) (not (at ?r ?from)) ))
```

- **domain validation:**
  - **static relations: must not appear in effects**
  - **fluent relations: should appear somewhere in effects**

Domain Types: Example

- **example:**

```
(:predicates
      :static (adjacent ?l1  ?l2 - location)
      :fluent (at ?r - robot ?l - location)
```

etc.

```
(:action move
      :parameters (?r - robot ?from ?to - location)
      :precondition (and
              (adjacent ?from ?to) (at ?r ?from)
              (not (occupied ?to)))
      :effect (and
              (at ?r ?to) (not (occupied ?from))
              (occupied ?to) (not (at ?r ?from)) ))
```

- **domain validation:**
  - **derive type system from operator specification**
  - **compare declared types to derived types**

Reversible Actions: Example

- **example: DWR domain**

```
(:action move
     :parameters (?r - robot ?from ?to - location)
     :precondition (and
                (adjacent ?from ?to) (at ?r ?from)
                (not (occupied ?to)))
     :effect (and
                (at ?r ?to) (not (occupied ?from))
                (occupied ?to) (not (at ?r ?from))
     :reverses (move ?r ?to ?from) ))
```

- **domain validation:**
  - **compare operator pairs: test whether one reverses the effects of the other**
  - **compare to "reverses" declaration in definition**

Inconsistent Effects: Example

- example: DWR domain

```
(:action move
    :parameters (?r - robot ?from ?to - location)
    :precondition (and
            (adjacent ?from ?to) (at ?r ?from)
            (not (occupied ?to)))
    :effect (and
            (at ?r ?to) (not (occupied ?from))
            (not (at ?r ?from)) (occupied ?to) ))
```

- domain validation:
  – identify potential inconsistencies in effects
  – add (implicit) inequalities to prevent instances

Evaluation

- small number of planning domains (from IPC)
  - domains were authored independent form this work
  - domains were authored by experts
- feature extraction algorithms were applied to all domains
  - runs in negligible time
  - provides feature values for all features in all domains
- check consistency (manually)
  - no feature values in domain specifications
- results:
  - automatically extracted features appear sensible
  - conceptual flaw in one of the domains was highlighted
  - also: minor issue in planner

Conclusions

- OpenVCE:
  - wiki extension supports collaborative development and validation of (semi-formal) procedural knowledge
  - virtual collaboration supported by procedural knowledge
- features:
  - static vs. dynamic: trivial, but helpful
  - types:
    - derived type system is most specific (of its kind)
    - flat; not hierarchical taxonomy/ontology
  - inconsistent effects: useful for planning
  - reversible actions: necessary criterion only

The Future

- Hedlamp: Machine Learning and Adaptation of Domain Models to Support Real Time Planning in Autonomous Systems
  - automatically acquiring procedural knowledge (machine learning)
  - domain analysis: towards a more human-like understanding of procedural knowledge (used in planning context)
  - use domain analysis to automatically improve learned model
- see: http://www.aiai.ed.ac.uk/project/hedlamp/
- or go there: http://virtual.aiai.ed.ac.uk:8002/ Hedland/
  - use in Firestorm OpenSim browser