

Miscellaneous Issues

- *Frame Functions*
- *Saving and Restoring*
- *CLOS Classes*
- *Mix and Match Inferencing*



Frame Functions: Concepts

add-type <instance> <concept>

"assert that <instance> ISA <concept> "

get-types <instance>

"retrieve all concepts satisfied by <instance>"



Frame Functions: Setting Role Fillers

add-value <instance> <role> <filler>

***"add <filler> to the set of fillers of role <role>
on instance <instance>"***

set-value <instance> <role> <filler>

***"set <filler> to be the only filler of role <role>
on instance <instance>"***

only recommended for single-valued roles

set-values <instance> <role>

<list-of-fillers>

***"make each filler in <list-of-fillers> be a filler of
role <role> on instance <instance>"***

***Always requires a list, even for single valued
roles.***



Frame functions

`get-value <instance> <role>`

"retrieve the single filler of role <role> on instance <instance>"

Error if there is more than one value.

`get-values <instance> <role>`

"return the set of fillers of role <role> on instance <instance>"

Always returns a list of fillers.



Saving and Restoring

- *Knowledge Bases and contexts can be saved to files*

```
(save-kb [<kbName>]  
         :pathname <filename>)
```

```
(save-context contextName  
              :pathname <filename>)
```

- *The save files can be compiled and loaded into Loom images to restore the state of the knowledge base.*



CLOS Classes

Execute the following:

```
(creation-policy :clos-instance)
(defconcept Ship
  :roles ((name :type String)
         length))
(create nil 'Ship)
```

Side-effects:

```
(defrelation name)
(defrelation length)
(eval '(defclass Ship (THING)
      ((name :initform nil)
       (length :initform nil))))
(make-instance 'Ship)
```



Mix and Match Inferencing

- **INSTANCE-IN-CONTEXT**

- at creation time, adds an instance to the concept-instance index*

- **INSTANCE-WITH-CONCEPTS**

- permits more than one type to be asserted on an instance*

- **INSTANCE-WITH-INVERSES**

- automatically adds and removes inverse links in response to slot updates*

- **INSTANCE-WITH-DYNAMIC-SLOTS**

- non-preallocated slots use alist storage on an instance*



Mix and Match Inferencing (cont.)

■ **INSTANCE-WITH-NEGATION**

supports negated type and negated role filler assertions

■ **INSTANCE-WITH-HISTORIES**

record (in a differential history) the prior states of an instance

■ **INSTANCE-IN-MATCH-NETWORK**

instance participates in matches that trigger production rules

■ **INSTANCE-WITH-TIME**

supports temporal assertions

