FRAMES (SCHEMAS) for “stereotyped” packets of knowledge (Minsky 1975)

When you open the door...

What do you expect??

(Rochester, D&C, 1996)
FRAMES

"Packets" of information about a certain thing or kind of thing

- slots & values
- types of values
- default values
- type hierarchies
- frame systems... eg. views of an object

[Betty is a person with
  mother ANN
  father CARL
  citizenship CANADA
  age 29
  occupation DENTIST
  car (a HONDA)
  hobbies (a-set (one-of PAINTING, SKY-DIVING))
]

[ES is a EAT with
  agent JOHN
  theme (a PIZZA) or theme PIZZA3
  location LIVING-ROOM
  day 10/15/09
  time 12:00am
  manner GREEDY
  enjoyment +10 ]

Logically:

PERSON (BETTY) ∧
  mother (BETTY) = ANN ∧
  father (BETTY) = CARL ∧
  citizenship (BETTY) = CANADA ∧
  age (BETTY) = 29 ∧ individual
  occupation (BETTY) = DENTIST ∧
  HONDA (CAR (BETTY)) ∧
  hobbies (BETTY) = set-of (PAINTING, SKY-DIVING)

But the idea of default values (later) is important
Indefinite frame types

(a PIZZA) \( \lambda x \) PIZZA\((x)\)

(a PIZZA with topping PEPPERONI)
\( \lambda x \) [PIZZA\((x)\) \& topping\((x)\) = PEPPERONI]

(a PIZZA with topping (a HEAT-TOPPING))
\( \lambda x \) [PIZZA\((x)\) \& HEAT-TOPPING\((\text{topping}(x))\)]

(a PIZZA with topping (one-of PEPPERONI, NEATBALLS))
\( \lambda x \) [PIZZA\((x)\) \& (topping\((x)\) = PEPPERONI \lor topping\((x)\) = NEATBALLS)]

(a PIZZA with topping (a set (a topping) with cardinality 2)))
\( \lambda x \) [PIZZA\((x)\) \& \( \exists s. (\text{set}(x, s) \Rightarrow \text{topping}(s)) \) \&
\( \text{topping}(x) = s \) \&
\( \text{cardinality}(s) = 2 \)]

(a PIZZA with topping (a set (one-of PEPPERONI, NEATBALLS) with cardinality 2))

DEFINITIONS - partial & full

(Define PERSON inherits-from
(a CREATURE with name (a STRING)
home-address (a ADDRESS)
age (a NUMERIC))

\( \forall x \). PERSON\((x) \Rightarrow \) CREATURE\((x) \) \&
STRING\((\text{name}(x))\) \&
ADDRESS\((\text{home-address}(x))\)
NUMERIC\((\text{age}(x))\)

(Define BACHELOR as
(a PERSON with sex MALE
marital-status SINGLE))

\( \forall x \). BACHELOR\((x) \Rightarrow \) PERSON\((x) \) \&
sex\((x) = \text{MALE} \) \&
marital-status\((x) = \text{SINGLE} \)
**DEFAULT VALUES**

As, by default, the color of a canary is YELLOW;
by default, a car has 4 wheels, 4 doors, ... 
by default, a flight booked by a customer in Rochester starts in Rochester

Define BOOKING-RECORD as
(a RECORD with flight-date (a DATE)
source (a CITY with default ROCHESTER)
dest (a CITY)
airline (a AIRLINE)
fare (a NUMBER))

∀x. BOOKING-RECORD (x) ⊃
   RECORD (x) ⊃
   DATE (flight-date (x)) ⊃
   CITY (source (x)) ⊃ default (source (x)) = ROCHESTER
   ... etc

In particular cases,
use this to assume
source (x) = ROCHESTER
if no information to the contrary is received

**PROCEDURAL ATTACHMENT**

Suppose BOOKING-RECORD allows multiple travellers...

source ... 
dest ... 
airline ... 
fare ... 
travellers (α-set PERSON) 
total-fare [(α (cardin (travellers SELF))) (fare SELF 1.08)]

B&L call this type of procedure "IF-NEEDED".
They also allow "IF-ADDED" procedures that do some extra work when a slot value is supplied. E.g.,

source ... 
dest ... 
next-leg (a BOOKING-RECORD)
[IF-ADDED
   (self (previous-leg (next-leg SELF))
     SELF)]

(i.e., make the next leg point back to this leg;
B&L's access specification actually looks like SELF:next-leg:previous-leg ← SELF)