Intelligent Systems: Reasoning and Recognition

James L. Crowley

ENSIMAG 2 Second Semester 2017/2018
Lesson 14 30 March 2017

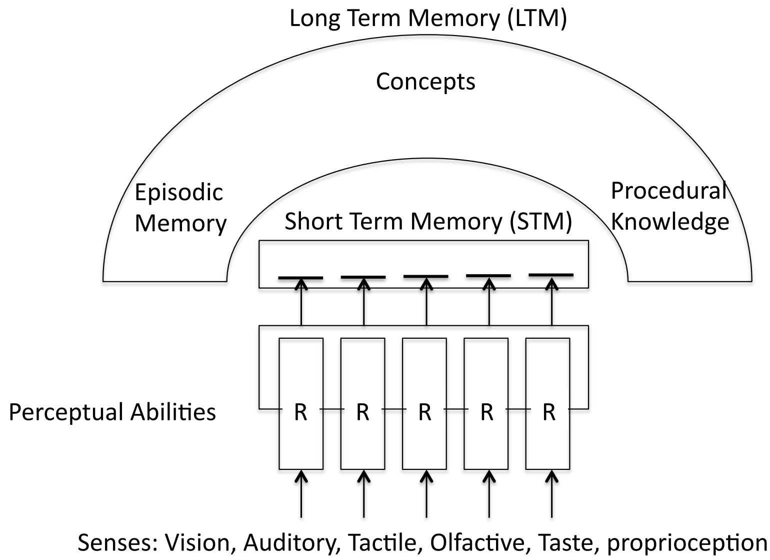
Structured Knowledge Representations: Concepts, Schema, Frames and Scripts

Bibliography:

1) W. Kintsch, *Comprehension: A paradigm for cognition*, Cambridge university press, 1998
2) G.A. Miller, The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological review, 63(2), p.81.1956
3) J. R. Anderson, A spreading activation theory of memory, Journal of Verbal Learning and Verbal Behavior, Volume 22, Issue 3, Pages 261-295, June 1983
4) M. Minsky, A Framework for Representing Knowledge, in: Patrick Henry Winston (ed.), The Psychology of Computer Vision. McGraw-Hill, New York (U.S.A.), 1975.
5) P. N Johnson-Laird, Mental models, MIT Press Cambridge, MA, USA, 1989.

Background from Cognitive Science

Short Term and Long Term Memory



(inspired by Rasmussen, Card-Moran-Newell, Anderson, Kintsch, many others)

Most models of human cognitive models share a number of common elements:

- Perception: Transforms and combines sensory stimuli to Phenomena
- Perceptual Memory: Temporary buffer holding recent stimuli
- Short Term Memory: 7+/-2 memory slots (perceived or remembered)
- Long Term Memory

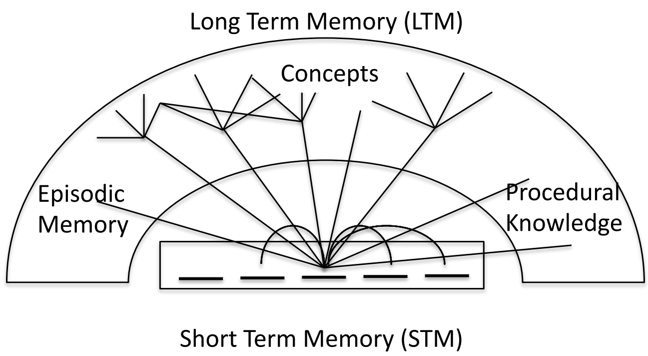
Long-term memory (LTM) refers to memory structures used in several different cognitive abilities:

- Episodic Memories: recordings of significant sensory experiences
- Semantic Memory: Abstract representations for sensory experiences
- Procedural Memory: Sequences of operations to accomplish goals
- Spatial memory (e.g. network of places in the hippocampus)

Spreading Activation

Most theories posit some form of "spreading activation" (Anderson 83) in which activation energy propagates through a network of cognitive "units".

Spreading activation is mechanism for associating cognitive units and controlling the contents of the limited Short-term memory.



Activation energy spreads from short-term memory to other elements of short-term memory and to long-term memory including concept memory, episodic memory, procedural knowledge, etc. Activated units then spread their energy to other units where it can arrive from multiple paths and accumulate. At the same time the energy decays with time.

Units that receive energy from several other units can become "activated" and can replace one of the 7+/- 2 active units in short term memory. (Miller 56)

Theories differ in describing how activation energy propagates and how this propagation can be controlled by emotions and physiological state.

The limited size of short-term memory is the primary bottleneck for cognition.
This limit is NOT because of the cost of memory. The limitation is the algorithm complexity caused by spreading activation. $O((7b)^d)$ where b is the average branching factor (number of associated units) and d is the average depth.

Chunking

Chunking is a process of grouping individual cognitive units into larger composed units. Chunking allows multiple cognitive units to be held in short term memory at the same time, overcoming the limits to short-term memory. However, associations to LTM and STM are with the chunk and not its individual elements.

To say more, we need to define what we mean by a cognitive "unit".

Conceptual Knowledge

Concepts

A concept are mental constructs that represent abstract or generic ideas generalized from particular instances. Concepts are basic elements of cognition. Concepts provide abstractions for reasoning and communications.

In some theories, concepts are referred to as "units" (E.g KRL: Knowkedge Representation Language) or as "entities" (Johnson-Laird's Situation models).

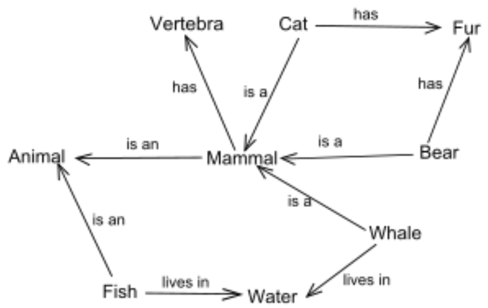
Concepts arise as abstractions or generalizations from experience or from the transformation of existing concepts. Concepts can be learned from experience, or by communication. Concepts can represent words, actions, perceived phenomena, experiences, feelings, etc.

A concept is instantiated (reified) by all of its memories of actual or imagined phenomena. These memories may be images, sounds, image sequences, feelings, or any other perceived phenomena (e.g. taste, smell, etc).

In cognitive psychology, Concepts are studied as the units of human cognition. In philosophy, concepts are studied in the field of ontology, as part of the question of what entities may be said to exist and how such entities may be identified and associated. The study of ontologies has been carried over to informatics and semantic web where an ontology defines the elements of a domain.

Semantic Networks

A semantic network, or frame network, is a network that represents semantic relations between concepts. This is often used as a form of knowledge representation. It is a directed or undirected graph in which nodes represent concepts and arcs represent relations between concepts.



Common relations include class hierarchies (ISA, AKO) and part hierarchies (PART-OF and COMPOSED-OF), Spatial relations (left-of, right-of, above, below), termporal relations (Before, after, during, etc).

ISA represents class hierarchy for entities. Is-a enable for reasoning about classes and categories. Has associates a concept with components. (Fish Has eyes).

Schema

Schema are declarative structures for representing concepts. The term Schema and was originally proposed by Emmanual Kant in "Critique of Pure Reason (1781). Schema have been used in philosophy and cognition psychology since the 19th centurey to represent concepts for reasoning, perception, problem solving and natural language interaction.

A schema (plural schemata or schemas) describes a pattern of thought that organizes information. A key property of Schema is the association of concepts with procedures for perception, action and reasoning.

Schemas represent concepts as data structures with slots that define the properties of the concept and associate the concept with other concepts.

A typical Schema for a concept has
1) A name.
2) A definition.
3) Meanings: memories of examples of the concept
4) Roles: Operations or procedures that are enabled or prevented by the concept.
5) Relations to other concepts and other elements in LTM.
(In many schema systems, meaning and roles are part of the list of associations).

The definition of a concept can be Intensional or Extensional.

An intensional definition specifies a test (or set of tests) that determine if an entity is an instance of the concept. (The recognition function studied in the first 9 lectures of this course provides intensional definition of concepts for perception).

An extensional definition provides a list of entities that can be identified as belonging to the concept.

Structured Knowledge: Concepts, Schema, Frames and Relations

A meaning typically denotes memories that serve as examples. Meanings can be from actual examples or can be imagined. Meanings can be visual, episodic, auditory, olfactory, emotional or examples of feelings.

Roles are operations or procedures (procedural knowledge) that are enabled or prevented by the concept. Roles can also refer to uses that the concept can have.

For Example: Consider the number 5.

The number 5 has
1) a name: (five in english, cinq in french etc).
2) a definition (the name of a set of all sets with 5 elements)

This is an intentional definition that may implemented either by counting the elements (5 comes after 4) or by direct recognition.

3) Meanings: Experiences with examples of the concept 5. (visual pattern, sounds).

4) Roles: Operations such as addition, subtraction, division, etc that are made possible. (Example 5 can not be directly divided by 2).

5) Relations.

Multiple kinds of relations are possible:
ISA and AKO: Identifies the concept as a member of a larger class.
(AKO = A Kind Of). Examples: (5 ISA number) (5 ISA integer) (5 ISA odd)
Part-Of: Identifies the concept as a component of a larger concept
(5 is a part of the number 15, 5 is a part of the formula 15/3)
Order Relations : $(5 < 6)$, $(3 < 5)$, Time relations 5h is before 6h.

Frames

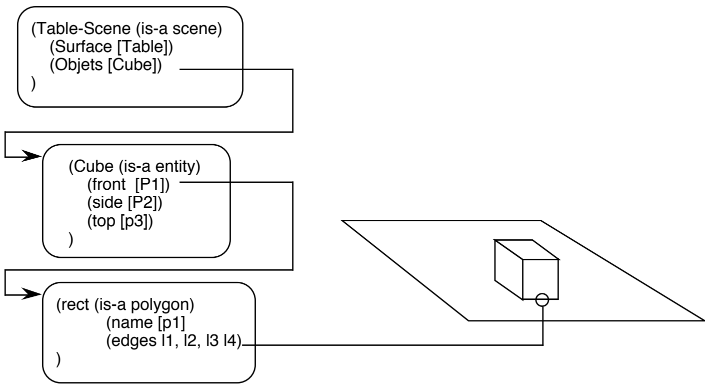
Frames are data structures that guide perception. Frames represent perceived entities as examples of concepts. Frames are used to organize perceptions in Computer Vision, Linguistics and Cognitive Systems.

In computer vision, Frames were made popular by Marvin Minsky (1976). Minsky proposed Frames as a structure to guide interpretation in a top down manner, telling a vision system where to look and what to look for. Minsky's insight was that it is much easier to see if you know what to look for.

A frame describes a perceived entity with a set of properties and relations, represented by slots, and a collection of procedures for perceiving, reasoning and acting with the concept. The key insights are:

1) To provide procedures or operations to detect entities, either as perceived entities or as other frames.
2) To provide default values for properties when perception is not possible or fails.

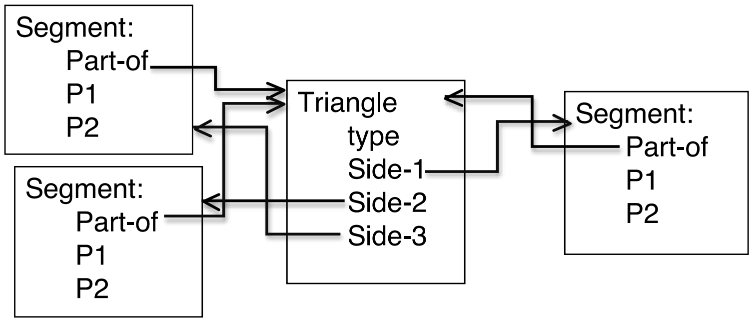
Frames represent concepts and can be composed hierarchically to describe complex entities.



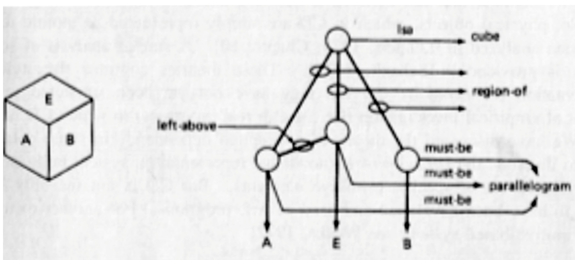
Frames provide visual context to guide scene interpretation. A Frame tells the program what to look for and where to look for it. They ask allow a system to ask questions about objects, such as what and where

Frames are composed using relations, represented by slots that contain pointers to other frames. Relations represents information about the object, such as part relations (composed of, part-of), Position relations (above, below, beside, inside, contains), Time relations (before, after, during), as well as specific properties of the entity (size, position, color, orientation).

For example, the concept Triangle has the part relation "composed of" with three segments. The triangle can also have an is-a relation (category membership) with different triangle types such as equilateral, isosceles, right angle, etc.



Ultimately, some slots point to raw perceptions (phenomena).



A Frame for a Cube
(from E. Rich "Artificial Intelligence", Fig 7-13, p231

When a slot points to an entity, the entity is said to play a "role" in the frame. Frames typically come with methods (procedures) for searching for the entities that can plays roles in the frame. Typically a slot-filling procedure will apply a set of acceptance tests to an entity to see if it satisfies the requirements for the role

Frames generally include typical examples (prototypes) that can serve as examples in reasoning, and default values that are used if no entity has been found to fill the slot. Thus frames can be used for abstract reasoning or for reasoning when perception is not possible.

The term Frames has come to represent any general representation of common sense knowledge using a slot and filler structure. Slots tell what entities to find and fillers are procedures to find entities that can fill slots.

Discovering the appropriate frame for reasoning is called "The Frame Problem".

Relations

Relations are used to organize concepts.
Examples include temporal relations, spatial relations, Part-whole relations, family relations, social relations, administrative organizations, military hierarchies, and class hierarchies.

A non-exhaustive list of relations between concepts includes:
1) Class membership (ISA, AKO) relations
2) Structural (Part-of) Relations
3) Ordinal relations (bigger-than, smaller than)
4) Spatial Relations (right-of, left-of, above, below, in-front-of, behind, etc)
5) Temporal relations (Allen's 13 relations between intervals).
6) Organizational relations (team member, leader, etc)
7) Family (parents, brothers, sisters, etc)
8) Causal (action A caused phenomena P)
This list is NON-EXHAUSTIVE. Relations can be defined as needed by a domain.

Class membership (is-a) are useful as part of the definition of a schema.
Other relations can be Over the schema:
on(knife, table), in(coffee, coffee-pot), before(cook, eat).

Predicates

Relations are formalized as Predicates (Truth functions).

A predicates is function that assigns a property to an association of arguments. Normally, predicates are assumed to be Boolean functions, but an interesting research problem is how to use probabilistic predicates to represent relations.

A predicates is a function that tells whether or not a relation is valid for a set of entities. Classically, predicates are treated as Boolean functions that can only return a value of TRUE or FALSE. As we have seen, in probabilistic reasoning, predicates represent the likelihood that the relation holds, with a value between 0 and 1.

Relations as N-Ary Predicates

The "Arity" of a relation is the number of arguments. Arity represents the number of entities associated by the relation. Relations may have an arity of 0 or more arguments.

The valence or Arity of a relation is the number of entities that it associates.

Nullary:	Friday()	;; a statement. An assertion. An axiom.
Unary:	Man(Bob)	;; Bob is a human of of gender male
Binary	Brother(Bob, Chris)	;; Bob and Chris are brothers
Ternary	Triangle(A, B, C)	;; a geometric relation associating points or lines

In some systems it is possible to have functions with variable arity. These are called polyadic functions or variadic functions. Set(A, B, C, D)

The philosopher CS Pierce (1871) demonstrated that any system of relations could be reduced to a hierarchy of binary relations by chunking. This is used in the widely used notation: (Subject Relation Object).

Implicit vs Explicit representations for Relations

Relations can be represented "implicitly" or "explicitly".

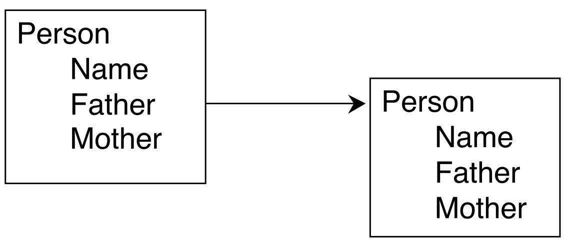
Implicit representation

Frames, and most schema systems use implicit relations.
With an <u>implicit representation</u>, the relation is represented as a pointer in a slot.

For example, with family relations:

(defclass PERSON (slot NAME) (slot FATHER) (slot MOTHER)).

The slot FATHER contains a pointer to an object of the class PERSON that represents the father of the person. The pointer is the object address.



Implicit representations are simple and more efficient in computing and memory.

However, with an implicit (slot based) representation for a concept, the set of relations is fixed and cannot change dynamically.

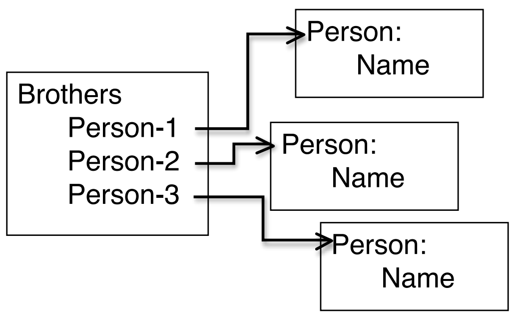
An implicit (slot based) representation for relations is not easily completed with meta information. Some forms of reasoning are much easier with an explicit representation.

Explicit representation

With an explicit representation, relations are represented by a schema whose arguments are concepts.
Explicit Representations for relations can be changed dynamically without changing the underlying concept.

(defclass PERSON (slot NAME))
(defclass Brothersp (slot person1), (slot person2), (slot person2))

A slot holds a pointer to the object that represents the relation. This object can then provide additional information about the object, such as what, where, why, when, who and how.



With an explicit representation for relations, it is possible to write a set of general procedures for acquiring (learning), reasoning, and explaining that apply to all relations.

With an implicit representation, such procedures would be specific to each class. Thus explicit relations support generalized methods acquiring (learning), reasoning, and explaining about relations.

For example, Allen's temporal reasoning is much easier to program using explicit models of relations because the set of relations between intervals changes dynamically.

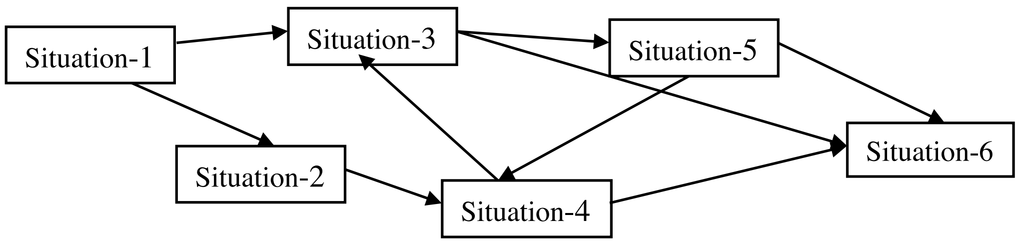
Situation Models

P. Johnson-Laird 1983 - Mental Models.

Situations models are used in cognitive psychology to express the mental models that people use to understand and reason.

Entities:	Anything that can be named or designated; People, things, etc. (entitles are defined using schema or frames)
Properties:	Descriptions of entities such as position, size, color, etc
Relations:	N-ary predicates (N=1,2,3 …) that relate entities. (relations are defined as tests on the properties of entities).
Situation:	A set of relations between entities

Situations can be organized into a state space referred to as a situation network. Each situation (or state) corresponds to a specific configuration of relations between entities. A change in relation results in a change in situation (or state).



The situation graph, along with the set of entities and relations is called a Context.

Each situation can prescribe and proscribe behaviors.
1) Behaviors: List of actions and reactions that are allowed or forbidden for each situation. Behaviors are commonly encoded as Condition-Action rules.
2) Attention: entities and relations for the system to observe, with methods to observe the entities
3) Default values: Expectations for entities, relations, and properties
4) Possible situations: Adjacent neighbors in the situation graph.

Each situation indicates:
  Transition probabilities for next situations
  The appropriateness or inappropriateness of behaviors

Behaviors include

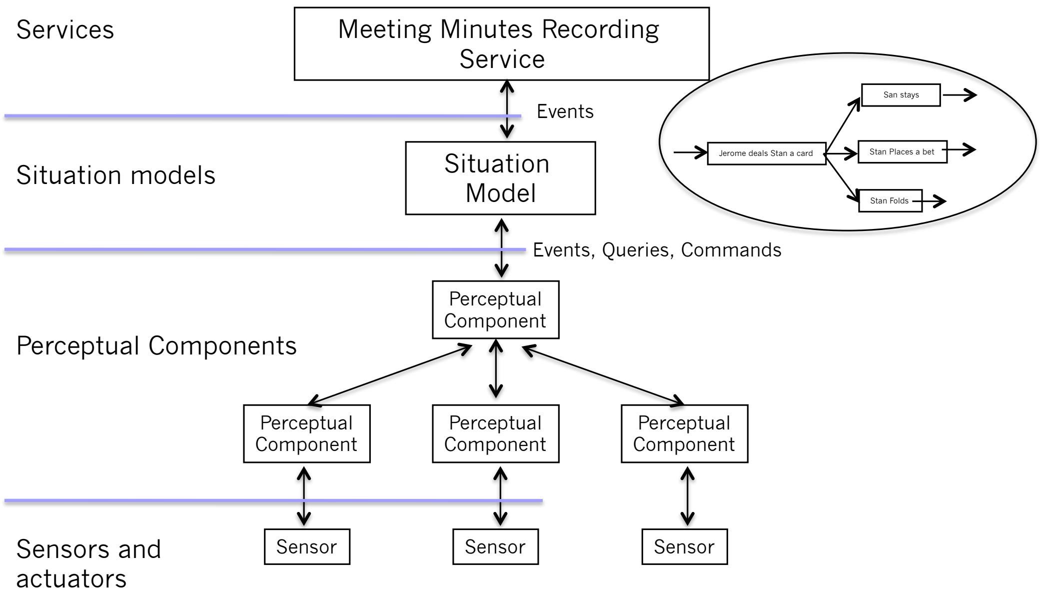
1) methods for sensing and perception, and
2) appropriateness of actions
3) changes in state in reaction to events.

The sets of entities, relations, behaviors, and situations define a "Context".

Situation models are used to construct context aware systems.

A "Context" is defined as
1) A set of entities, with their properties.
2) A set of relations between entities
3) A network of situations, such that each situation specifies
    - A list of adjacent situations, possibly with transition probabilities.
    -A list of system behaviors that are allowed or forbidden,
possibly with preferences (appropriateness) for the situation.

Example: Meeting Recording System



Entities:
Patrick, Jerome, Sonia and Stan, agenda
Roles:
Moderator, Speaker, Participant, current-agenda-item, etc
Relations:
Moderator(Patrick) speaks-to participants(…)
Participant(Jerome) talks-to Participant(Stan)
Participant(Sonia) looks-at Participant(Patrick)

Examples of uses of situation models

1) Event Recording (Startup MeanInFull - 2014)
2) Video Surveillance (Startup BlueEye Video – 2003)
3) Customer monitoring (Start up: HiLabs - 2008)
4) Actimetry and monitoring for Elderly and Handicapped
5) Socially-Aware Human-Computer Interaction
7) Context aware mobile applications (Start up: Situ8ed 2015)
8) Sociable Systems (Startup planned for 2017)

Scripts

R. C. Schank and R. P. Abelson, (1977). Scripts, plans, goals, and understanding: An inquiry into human knowledge structures. Reprinted by Psychology Press, 2013.

A script is a schema structure used to represent a stereotypical sequence of events.

Scripts are used for interpreting stories. For examples, scripts used to construct systems that interpret and extract information from Newspaper Stories. Scripts are used in natural language understanding systems to organize a knowledge base in terms of the situations that the system should understand. Scripts are also used to observe an actor and to describe (or recognize) what the actor is doing. This includes plan-recognition as well as activity description. Scripts are also be used to represent procedural knowledge for plans.

Scripts are schema much like Frames, except that the slots point to a sequence of situations.

A script is composed of

1) Scene: situation in which the script takes place
2) Props: Entities (objects) involved in the script.
3) Roles: Actors (agents) that can provoke changes in the scenes.
   Actors are typically people, but may be artificial.
4) Events (acts): A sequence of events that lead to changes in situations and make up the script.

The script can be represented as a tree or network of scenes, driven by actions of the actors.

In each scene, one or more actors perform actions. The actors act with the props. The script can be represented as a tree or network of states, driven by events.

As with Frames, scripts drive interpretation by telling the system what to look for and where to look next. The script can predict events.

Example of a script: Restaurant Script.

The classic example is the restaurant script:
Props: A restaurant with an entrance, tables, chairs, plates, eating utensils, glasses, menu, etc
Actors: The host (Maitre d'Hotel), clients, servers, chef, bus-boy, etc.
Scenes: Entry, seating, reading the menu, ordering, serving, requesting the check, paying, leaving, etc.

Scripts provide context for default reasoning.

As with Frames, scripts drive interpretation by providing procedures that tell a system what to look for and where to look for it.

Scripts also provide default knowledge for reasoning about stories or actions.
For example, for story understanding, the story will typically only provide sparse detail of what happened. The reader is expected to fill in the missing knowledge with default knowledge.

Problems with Frames.

Some Problems with Frames.
1) Top down reasoning: Frames (and most schema systems) are designed for top-down reasoning. Most human reasoning is both top-down and bottom-up, with associations flowing both ways.

2) Knowledge Acquisition: Building (by hand) a Frame system is long, tedious, and ad hoc process. There is a temptation to overload with useless information. Automatic acquisitions (learning) of frame systems for recognition and reasoning is an open research question.

3) Context Recognition (The Frame problem): Many problems are easily solved once the appropriate frame is known. Recognizing the correct context can be very difficult.

5) Semantic Alignment: Two Frame systems describing the same concepts, may not have the same relations. Meanings of similar concepts might be slightly different. However, communication and integration of conceptual knowledge from different sources requires a shared ontology.