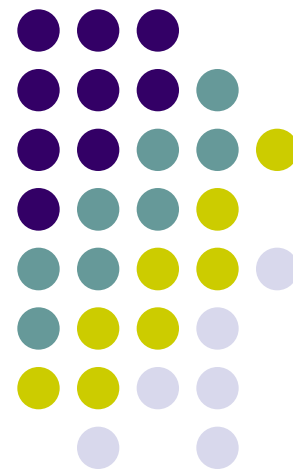
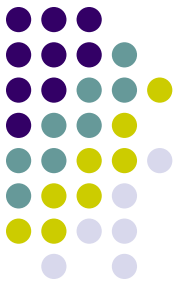


**ПРЕДСТАВЛЕНИЕ
ЗНАНИЙ:**

ФРЕЙМОВАЯ МОДЕЛЬ





СОДЕРЖАНИЕ

1. Сетевая модель и понятие фрейма
 - Концепция фрейма М.Минского
2. Виды фреймов и их структура
 - Фреймы-прототипы и экземпляры
3. Присоединенные процедуры
4. Связи фреймов и вывод в сетях фреймов
5. Фреймовые языки и системы
6. Достоинства и недостатки фреймовой модели
7. Заключение и Домашнее задание

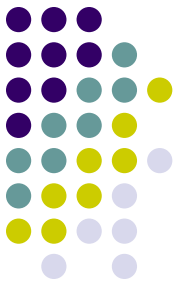
СЕТЕВАЯ МОДЕЛЬ И ФРЕЙМЫ



Фреймовый подход можно считать дальнейшим развитием сетевой модели в двух направлениях:

- Укрупнение единицы знания: информация из фрагмента сети, относящаяся к свойствам, связям некоторой сущности, хранится в одной структуре (*фрейме*), которая может обрабатываться как единое целое
{Трамвай 0815: Instance_of – Трамвай, Цвет – Красный, Маршрут – 39}
 - Все атрибуты сущности собраны воедино (во фрейме)
 - Вместо исходной СС из вершин-сущностей и связей получается сеть из вершин-фреймов.
- Возможность представления процедурного знания: концепция *процедурной привязки*: присоединения к фрейму целого набора процедур
- Истоки – когнитивная психология, теория М. Минского

ТЕОРИЯ ФРЕЙМОВ МИНСКОГО



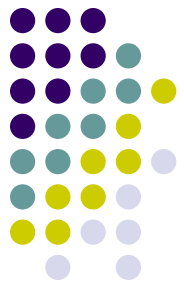
М.Минский, 1985 г. : попытка объяснить
эффективность человеческого мышления
(восприятие образов, речи, проведение аналогий) –
выявление основных структурных элементов процессов
обработки и хранения информации человеком

*“Фрейм является структурой данных для представления
стереотипной ситуации”*. Например: Комната, Экзамен

Основные положения теории:

- Фрейм – структура, образ, рамка, определенная для любого концептуального объекта в человеческой памяти...
Задаются условия заполнения *терминалов* (ячеек) фрейма
- С каждым фреймом ассоциируется обычно разнообразная,
в том числе процедурная информация: каким образом
следует использовать данный фрейм; что предположительно
может повлечь за собой его выполнение; что следует
предпринять, если эти ожидания не подтвердятся...

ТЕОРИЯ МИНСКОГО: ПРОДОЛЖЕНИЕ



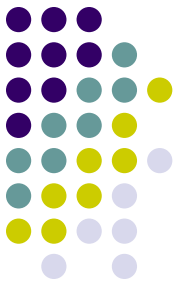
- Фреймы образуют иерархическую/ассоциативную сеть: группы семантически близких друг к другу фреймов объединены в **систему фреймов**, что позволяет
 - выразить различные зависимости (причинные, временные, структурные и др.) между фреймами системы
 - согласовывать информацию из различных источников
- Человеческое восприятие и мышление осуществляется посредством сопоставления полученной извне информации с этими фреймами-рамками, а также конкретными их элементами и значениями. Результаты существенных действий представляются в виде трансформаций между фреймами системы...
- Терминалы фрейма ... могут заполнены т.н. *“заданиями отсутствия”*... Их связь со своими терминалами не является жёсткой и неизменной... и может быть заменена на более подходящие к текущей ситуации сведения...

СЕМАНТИКА ФРЕЙМА



- Единица знания фреймовой модели ПЗ – *фрейм*
- Упрощенно: фрейм – набор атрибутов/ свойств описываемой сущности, точнее, набор *слотов* вида:
атрибут = значение атрибута (slot – ячейка)
- Примеры: *Комната (Стены, Пол, Потолок, Дверь)*
Экзамен (Экзаменатор, Экзаменуемый, Предмет)
- Фрейм иногда называют *ролевым*, поскольку названия/имена атрибутов характеризуют роли объектов (сущностей), стоящих в позициях значений атрибутов
- Важно: совокупность всех атрибутов определяет смысл понятия (объекта/явления/ситуации), описываемого фреймом (*имманентные* его свойства)
- Можно заметить, что фрейм – структура, аналогичная предикату (аргументы предиката – атрибуты фрейма, у фрейма они именованы)

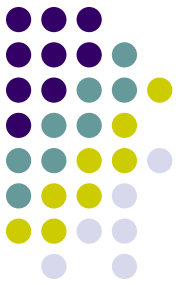
ВИДЫ ФРЕЙМОВ



Во фреймовой системе различают 2 вида фреймов:

- *Фрейм-прототип* соответствует обобщенной сущности (понятию, объекту, событию, явлению, ситуации)
 - *Фрейм-экземпляр* (фрейм-пример) соответствует конкретной/индивидуальной сущности (объекту, факту)
- ❖ Фреймы-прототипы относятся к интенциональной части БЗ, а фреймы-экземпляры – к экстенциональной части
- ❖ Общая структура обоих видов фрейма сходна:
набор слотов s_i : $[fname\ s_1\ s_2\ \dots\ s_n]$
где *fname* – уникальное имя фрейма,
а каждый слот s_i соответствует атрибуту/свойству,
упрощенно его структура имеет вид $id_i = val_i$
- ❖ Однако состав отличается: фрейм-прототип описывает всевозможные значения атрибутов/свойств всего класса объектов (конкретных сущностей)

ФРЕЙМ-ЭКЗЕМПЛЯР



- Структура фрейма-экземпляра:

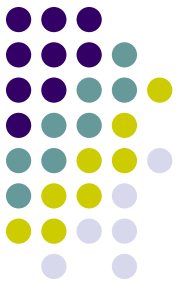
$[fname <id_1 val_1> <id_2 val_2> \dots <id_n val_n>]$

где id_k – имя соответствующего слота

val_k – значение этого слота (заполнитель слота)

- Возможные значения val_k определяются в соответствующем фрейме-прототипе
- Каждый слот – описание соответствующего атрибута конкретного объекта/факта/явления/события
- Фрейм – структура, которая при том или ином заполнении слотов значениями превращается в описание конкретной сущности (часто в ИИ и КЛ: понятие фрейма как шаблона с местами для подстановки/заполнения)
- Не все слоты фрейма-экземпляра могут быть заполнены, что соответствует не полностью определенной ситуации
- Слоты могут быть *терминальными* и *нетерминальными*

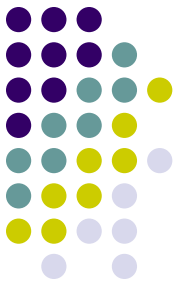
ВИДЫ СЛОТОВ ФРЕЙМА-ЭКЗЕМПЛЯРА



Во фреймовых языках/системах:

- **Терминальные слоты** обычно имеют конкретные значения любого стандартного скалярного или составного типа (*string*, *boolean*, *integer*, перечислимый тип и т.п.).
 - Терминальное значение слота соответствует сущности, которая не м. б. разложена на более простые сущности.
- **Нетерминальные слоты** имеют в качестве значения имя другого фрейма-экземпляра, представляющего содержание данного слота
 - По сути это ссылка на другой фрейм-экземпляр, и тем самым обеспечивается связь между фреймами-экземплярами
- ❖ Связь фрейма-экземпляра с прототипом: обычно есть встроенный (**системный**) слот с именем *Instance-of* значением которого есть имя прототипного фрейма (т.е. задается ссылка на прототипный фрейм)

ФРЕЙМ-ПРОТОТИП



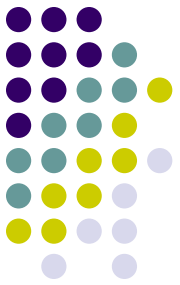
Более сложный состав слотов и фрейма в целом

- Сам фрейм включает *присоединенные процедуры*
- Слот в общем случае включает кроме имени атрибута несколько видов связанной с этим атрибутом информации – т.н. *фасеты (facets)* – аспекты атрибута

Фасеты включают:

- *Ограничения на значение (Value)* соответствующего атрибута во фрейме-экземпляре
- *Кардинальность значения (Single/Multiple)*: одно или же несколько значений указанного типа могут быть в слоте
- *Значение по умолчанию (Default)* стандартное значение, которое может быть потом переустановлено
- *Процедуры, присоединённые к слоту* (ассоциированные с ними), записанные на языке реализации фреймовой системы и выражающие процедурное знание

ФРЕЙМ-ПРОТОТИП: ОГРАНИЧЕНИЯ



- Возможные способы ограничения значения слота
 - Задание конкретного значения некоторого типа
 - Указание типа значения (*integer, real, boolean, string, date* и т.п.) или диапазона значений: (*0..120*)
 - Перечисление возможных значений
 - Логическая комбинация условий предшествующего вида
 - Указание процедуры для вычисления значения
 - Задание имени другого фрейма-прототипа (ссылка)
- Если во фрейме-экземпляре слот нетерминальный, то он нетерминальный и во фрейме-прототипе, т.е. значение этого слота – имя другого фрейма-прототипа
- Тем самым обеспечивается необходимая связь между фреймами-прототипами, что применяется при описании сложных взаимосвязанных понятий ПО

ФРЕЙМ-ПРОТОТИП: СИСТЕМНЫЕ СЛОТЫ, ЭКЗЕМПЛЯРЫ



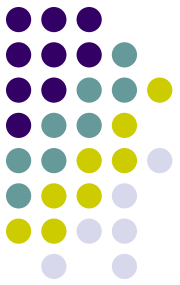
Для связей фреймов используются **встроенные слоты**:

- *is-a / a-kind-of / АКО* – слот для указания имени (имен) родового фрейма-прототипа
- *Descendents / Instances* – слот для задания «потомков» - видовых фреймов-прототипов для данного фрейма
- *Examples* – слот для задания всех фреймов-экземпляров (примеров) рассматриваемого фрейма-прототипа.

Соотношение фрейма-прототипа и экземпляра:

- ❖ По сути: фрейм-экземпляр – конкретное заполнение слотов соответствующего фрейма-прототипа
- ❖ Такой фрейм может храниться в системе ИИ в незаполненном виде, т.е. содержать не все слоты, входящие в состав фрейма-прототипа (ситуация, когда не все свойства описываемой конкретной сущности известны в настоящий момент)
- ❖ При описании конкретного объекта/ситуации получаем в общем случае набор взаимосвязанных фреймов-экземпляров

ПРИСОЕДИНЕННЫЕ ПРОЦЕДУРЫ



2 вида присоединённых процедур фрейма-прототипа:

- **Служебные**, присоединяемые к слоту фрейма или ассоциированные с фреймом-прототипом в целом
- **Демоны**, присоединяемые к слоту и запускаемые автоматически при обращении к нему и выполнении определенного условия

Служебные процедуры – обычные процедуры, вызываемые по имени для обработки слота/фрейма

- проверяют корректность значений при создании нового фрейма/слота
 - вычисляют значения слотов по данным из других слотов
 - обеспечивают необходимые для сопоставления фреймов преобразования
 - могут использоваться процедурами-демонами
- ❖ С каждым слотом можно связать произвольное число процедур (служебных и демонов)

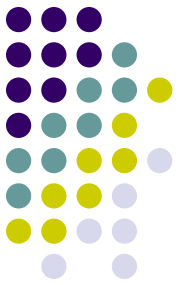
ПРИСОЕДИНЕННЫЕ ПРОЦЕДУРЫ: ДЕМОНЫ



Демоны реализуют «активные» значения

- Процедуры *if_added*: запускаются при подстановке в слот нового значения
- Процедуры *if_removed*: запускаются при удалении значения слота
- Процедуры *if_needed*: вызываются при запросе значения слота, если при обращении к слоту его значение ещё не установлено (могут реализовать *значение по умолчанию*)
- ❖ Фрейм-прототип дает возможность представления комбинации декларативных и процедурных знаний
- ❖ Присоединёнными процедуры: определяют интерпретацию слотов/атрибутов, определённых пользователем фреймовой системы (несистемных)

ПРИМЕР ФРЕЙМА: *Date*



Первая фреймовая система *GUS* (конец 70-х гг.):
Диалог на ЕЯ для бронирования авиабилета

[Date Фрейм-прототип

Month name

Day (Bouded-integer 1..31)

Year Integer (tofill assume 1995)

Weekday ((member (Sunday .. Saturday))
(whenfilled finddatefromday)
(tofill getweekday))

Summary (or (list Month Day) Weekday))

Summary – пример "обобщающего" слота, он определяет
образец задания даты (только в прототипном фрейме)

[is_a Date Month May Фрейм-экземпляр
Day 24]

По сути *whenfilled* $\approx$ *if_added* , а *tofill* ?
процедура *tofill* вызывает интерпретатор, а он – нужную процедуру

A decorative graphic in the bottom right corner consisting of a grid of colored dots in shades of purple, teal, yellow, and light blue, arranged in a pattern that tapers to the right.

Фрейм-прототип: М – множественность значений
- ссылка на фрейм, % - маркер процедуры

[Командировка

Is_a #Поездка

Examples М #К_в_Ригу

Командируемый #Сотрудник (if_needed #Сидоров)
(if_added % Проверить_возможность)

Города_посещения М String (if_added
% Забронировать_гостиницу)

Продолжительность Int 1..30 (if_needed default 7)

Дата_начала Date

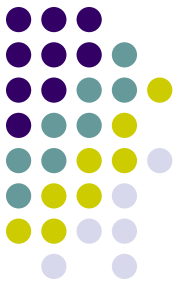
Назначение (конференция, согласование_документов)

[illegible]

Источник финансирования (РФИИ, госбюджет)

Оплата	#Оплата_командировки]
--------	-----------------------

ПРИМЕРЫ СВЯЗАННЫХ ФРЕЙМОВ



- [Оплата_командировки

Фрейм-прототип

Is_a #Оплата_расходов

Examples M #Оплата_К_в_Ригу

Аванс Int (if_added % Купить_билеты) (if_needed 0)

Транспортные_расходы M Int

Оплата_проживания Int

Суточные Int (if_needed % Рассчитать_сумму_по
срокам_поездки)

Итоговая_сумма (if_needed % Подсчитать_сумму)]

- [К_в_Ригу

Фрейм-экземпляр

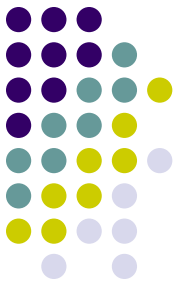
Instance_of #Командировка

Командируемый #Сидоров

Города_посещения Рига

Дата_начала 01.04.2015]

УНИВЕРСАЛЬНОСТЬ ФРЕЙМОВОЙ МОДЕЛИ



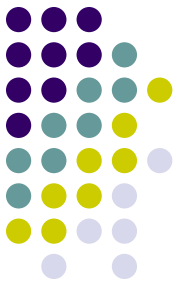
Фреймы, по глубине представляемой информации:

- Концептуальные:
 - Тематические (см. рассмотренные примеры)
 - Повествовательные: фреймы-структуры текстов (сказки, научные статьи, инструкции и т.п.)
- Лингвистические (поверхностно-семантические)
 - В системе *GUS* : *падежные фреймы* (case-frame) для обработки ЕЯ фраз:

Я хочу поехать в Сан Диего 28 мая
 - Фрейм слова-предиката *Поехать*:

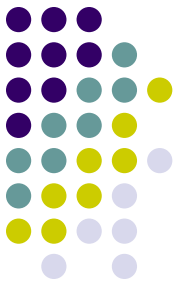
Клиент - Куда - Откуда - Время
 - Такой фрейм содержит слоты, соответствующие *семантическим падежам* Филлмора
 - Совокупность слотов – *модель управления* слова-предиката (*Поехать*), т.е. набор его валентностей

ЛИНГВИСТИЧЕСКИЙ ФРЕЙМ



- **Фреймовая семантика** в компьютерной лингвистике: смысл предложения ЕЯ может быть представлен как фрейм-экземпляр главного слова-предиката
- Филлмор (1968 г.): представление поверхно-семантической структуры предложения в виде набора падежных отношений (ролей) между глаголом-предикатом и словосочетаниями
- Основные семантические (глубинные) падежи Филлмора:
 - *Агентив* – исполнитель/инициатор ситуации/действия
 - *Объектив* – предмет, затрагиваемый действием
 - *Инструменталь* – сила/предмет, вовлеченная в действие
 - *Датив* – одуш. существо, затрагиваемое состоянием/действием
 - *Фактитив* – предмет или существо, которое возникает в результате действия/состояния
 - *Локатив* – местоположение или пространственная ориентация ситуации/действия/состояния

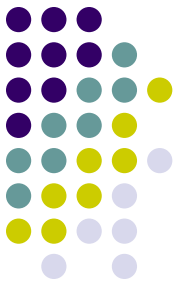
СВЯЗИ ФРЕЙМОВ: РОД–ВИД



Семант. отношение *Is-a* / Род–Вид / Класс–Подкласс
Фреймы-прототипы

- Общие слоты (свойства) родственных фреймов целесообразно представить один раз – в наиболее общих, прототипных фреймах – с тем, чтобы более частные фреймы (виды) их наследовали
- Тем самым более общее понятие-фрейм включает знание, справедливое для всех его частных случаев (видов); фреймы для более частных случаев включают лишь знания, отличающие этот частный случай от общего
- *Домен слота* – фреймы (классы), к которым привязан данный слот (свойства которых он описывает)
- Родственные фреймы образуют иерархию (дерево)
- Фреймовые системы поддерживают встроенные слоты *is-a / a-kind-of / АКО* , т.е. механизмы наследования

ФРЕЙМЫ: НАСЛЕДОВАНИЕ



Процесс наследования состоит в получении информации при движении вверх по иерархии фреймов (от фрейма видового/частного понятия – к родовым фреймам)

- ❖ Недостатки: увеличение времени поиска информации
- ❖ Преимущества: устранение дублирования информации

Виды наследования во фреймовых системах:

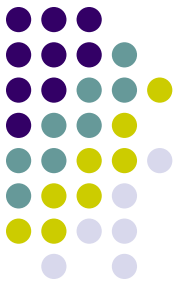
- **Строгое (жёсткое) наследование** – наследование без исключений: нижние узлы иерархического дерева всегда автоматически наследуют свойства верхних узлов, и унаследованные свойства нельзя отменить, тем самым: дедуктивный, монотонный вывод
- **Нестрогое, гибкое наследование**, которое может быть отменено при необходимости: *немонотонный вывод*
 - при обработке исключений (уникальных свойств объектов, н., свойство птиц летать для пингвина не выполняется),
 - при обработке значений по умолчанию (*Default*)

ВЫВОДЫ ПО УМОЛЧАНИЮ



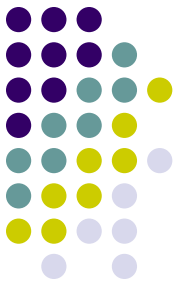
- Концепция умолчания (*Default*) : возможность приписать при необходимости отсутствующим слотам (значениям) фрейма-экземпляра стандартные, типовые значения
 - ❖ Это позволяет анализировать с помощью фреймов ситуации, в которых отсутствует упоминание о целом ряде деталей
- Присвоенные значения обычно не жёстко связаны со слотом, т.е. могут быть заменены на более точные, подходящие – если они затем обнаружены в ходе решения задачи
- Выводы, получаемые на основе значений по умолчанию, называются **выводами по умолчанию**
- Обычно *default-значение* применяется, когда значение слота не может быть вычислено другим способом, и оно не противоречит уже построенной модели конкретной ситуации
- Значение по умолчанию задается в прототипном фрейме этого уровня или выше (тем самым возможно его наследование)
- Отмена *default-значения* реализуется присоедин. процедурами

ВЗАИМОСВЯЗИ ФРЕЙМОВ



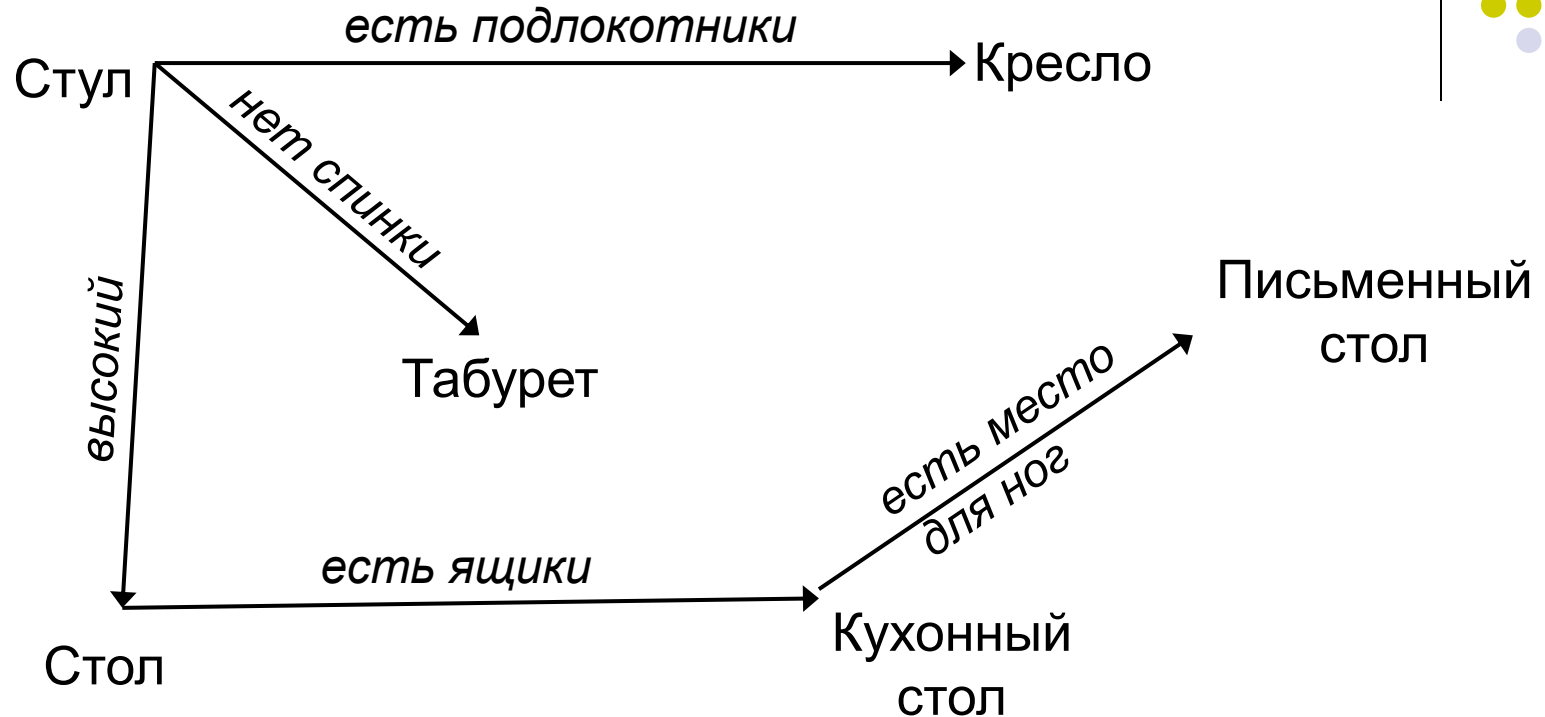
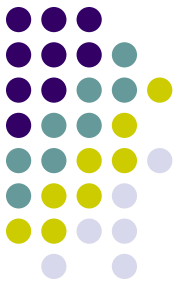
- Сложным понятиям могут соответствовать несколько взаимосвязанных фреймов системы
 - В случае зрительного образа различные фреймы могут задавать одну и ту же сцену с разных точек наблюдения
 - Несколько фреймов могут описывать объект с разных точек зрения: внутреннего и внешнего устройства, функционального назначения и т.п.
- Связи между фреймами задаются с помощью использования имени одного фрейма в качестве значения слота в другом фрейме (нетерминальное значение): этот слот именуется связью-ссылкой на фрейм
- В общем случае возникает
неоднородная **сеть фреймов**

СЕТИ ФРЕЙМОВ



- Обычно сеть фреймов-прототипов строится на основе таксономической связи (Род—Вид): у фреймов есть общие слоты, возникает иерархическая структура взаимосвязанных слотов и фреймов
- Фреймы могут быть организованы в сеть на основе других семантических отношений: часть-целое, причинных, временных, пространственных, атрибутивных, ассоциативных, отношений сходства
 - для этих отношений заводятся соответствующие слоты
 - их обработка реализуется присоединенными процедурами
- М. Минский: *"Системы фреймов связаны, в свою очередь, сетью поиска информации. Если предложенный фрейм нельзя приспособить к реальной ситуации, ... сеть поиска информации позволяет выбрать более подходящий для данной ситуации фрейм."*

СЕТЬ ПОХОЖЕСТИ ФРЕЙМОВ



- Межфреймовые сети похожести/подобия используются при распознавании образов – зрительных и речевых
- Могут быть отношения похожести между одноуровневыми (в классификации) понятиями-фреймами

ФРЕЙМЫ: ОБРАБОТКА



Операции во фреймовых системах:

- Распознавание по входной информации фрейма-прототипа и формирование фрейма-экземпляра
- Поиск нужной информации, предполагающий вывод на сети фреймов
- ❖ В различных фреймовых системах – разные условия активации фреймов
- ❖ Переход от исходного прототипного фрейма к фрейму-экземпляру может быть многошаговым, за счет постепенного уточнения слотов и их значений
- ❖ Вспомогательными операциями являются сопоставление фреймов и поиск фрейма по образцу

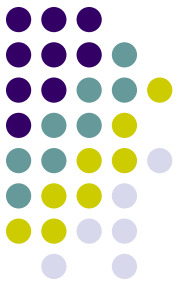
ВЫВОД ВО ФРЕЙМОВЫХ СИСТЕМАХ



Виды вывода:

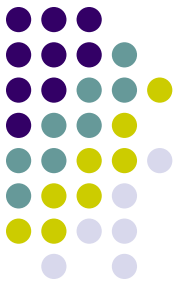
- Встроенный механизм наследования свойств-слотов
 - ❖ запускается автоматически при поиске значения нужного слота (последовательно поиск снизу-вверх в цепочке фреймов-прототипов), при этом
 - ❖ кроме значений могут наследоваться
 - значения по умолчанию
 - присоединённые к фреймам-прототипам процедуры (т.к. они представляют связанную с этим слотом информацию)
- Вывод на базе присоединенных процедур, составленных разработчиком прикладной системы

ФРЕЙМОВЫЕ ЯЗЫКИ



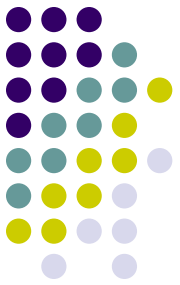
- **KRL**, 70-х гг.— сложный язык: несколько категорий фреймов, для каждого — способы их обработки (что позволяет повысить эффективность)
- **FRL**, 70-80 гг. — первый распространённый и известный
 - Изначально реализован как надстройка над Лиспом (лисповский синтаксис)
 - Многофасетность, присоединенные процедуры
 - Различные стратегии наследования
 - Широкий набор встроенных процедур обработки фреймовОтечественные реализации: на Лиспе ЕС + Рефале ЕС
- **KL-ONE**, 80-90 гг. — декларативный язык (сужение), уточнение семантики связей и их обработки с целью автоматического определения родовидовых отношений
- **PROTEGE**, 90-2000 гг. — изначально: для ПЗ в медицине
Несколько версий системы для создания онтологий ПО

ЯЗЫК *FRL*: СЛОТЫ И ФАСЕТЫ



- Различные фасеты слота, разные типы присоединенных процедур:
 - *\$value* – для него фасеты *\$if_added*, *\$if_removed*, *\$if_needed*
 - *\$default* – значение по умолчанию
 - *\$require* – служебная лисповская функция-предикат для проверки правильности значения *\$value*
- Встроенный слот *AKO* (a-kind-of) – объединение *is-a* и *instance-of* (слиты в одно)
- Слот *Classification* – для различения экземпляров прототипов (*\$value* = либо *generic*, либо *individual*)
- Может быть несколько значений слота АКО, т.е. фрейм может быть видом более чем одного фрейма

ЯЗЫК *FRL*: НАСЛЕДОВАНИЕ



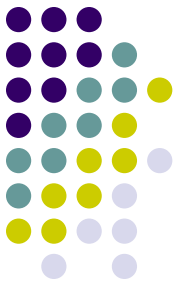
Особенности наследования:

- Множественное наследование, результат – список данных, найденных по всем АКО-путям
- Наследование с учетом фасетов
- При наследовании учитывается и фасет *\$if_added* (по сути: его наследование)

Встроенные процедуры-стратегии наследования:

- Общая стратегия множественного наследования
- *N-inheritance*: сначала проход по АКО-путям для аспекта *\$value*, а затем при необходимости (данные не найдены) проход по АКО-путям для аспекта *\$default*
- *Z-inheritance*: прежде чем идти по АКО-пути в поисках значения *\$value*, проверяется аспект *\$default* этого слота (если *\$value* пусто, а данные по умолчанию есть, то они и возвращаются в качестве значения функции)

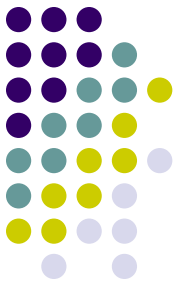
ЯЗЫК *FRL*: ФРЕЙМ-ПРОТОТИП



```
( Приказ_об_увольнении                               /FRL-запись упрощена/
  (Class generic)
  (АКО  Кадровый_приказ )
  (Увольняемый ($require #Сотрудник)
    ($if_added Вычеркнуть_из_списка_кадрового_состава))
  (Дата_зачисления ($require date))
  (Дата_увольнения ($require date))
  (Должность ($require list_of_string)
    ($if_added Пометить_должность_как_вакантную))
  (Причина_увольнения ($require list_of_string))
  (Размер_пособия_по_увольнению ($require integer)
    ($if_added Послать_приказ_в_бухгалтерию))
  (Кем_подписан_указ ($require list_of_string)
    ($default Краснов_И.И., директор))
  (Дата_заполнения ($require date)
    ($if_needed Взять_значение_из #Дата_увольнения)) )
```

ЯЗЫК *FRL*: ФРЕЙМ-ЭКЗЕМПЛЯР

/FRL-запись упрощена/



```
( U1
  (Class individual)
  (АКО Приказ_об_увольнении)
  (Увольняемый #Борисов_Б.Б.)
  (Дата_зачисления 01.03.2004)
  (Дата_увольнения 15.07.2011)
  (Размер_пособия_по_увольнению 7000)
  (Кем_подписан_указ ("Краснов_И.И."
                       "директор"))
  (Дата_заполнения 01.03.2005) )
```

ЯЗЫК *KL-ONE*: ОСОБЕННОСТИ



- Обычно при ПЗ классификационные связи Род-Вид устанавливаются человеком (инженером по знаниям)
- *KL-ONE*: Кроме встроенного наследования, возможна **Автоматическая категоризация концептов** исходя из их формального описания, для этого:
- Необходимо определение необходимых и достаточных условий для нахождения правильного места нового концепта в иерархической цепочке связей *is_a*
- Поэтому в языке потребовался
 - отказ от значений по умолчанию и присоедин. процедур
 - введение понятий **структурированного концепта, ролей**
 - раздельное описание концептов, ролей, а также **утверждений** о концептах и ролях
- ❖ Современная тенденция: отделение в ЯПЗ описания структуры родовидовых связей (*inferential structures*) от описания индивидуальных объектов (*assertional structures*)

ПРИЛОЖЕНИЯ ФРЕЙМОВЫХ СИСТЕМ



Наиболее типичные приложения в области ИИ:

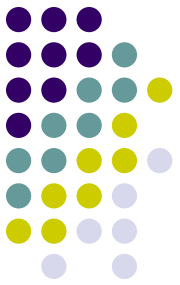
- Распознавание зрительных образов (образ – фрейм): интерпретация визуальной информации путём сопоставления с фреймами-прототипами
- Моделирование общения на ЕЯ (понимание ЕЯ-речи)
Наполнение фрейма-экземпляра происходит на основе одной или нескольких фраз
- Информационные системы: поиск данных, логический вывод – за счёт сравнения запроса (фрейма-экземпляра) с фреймами-прототипами системы
- ❖ Фреймы особо эффективны для структурного описания сложных понятий и решения задач, в которых в зависимости от ситуации желательно применять различные способы вывода.

ДОСТОИНСТВА ФРЕЙМОВОЙ МОДЕЛИ

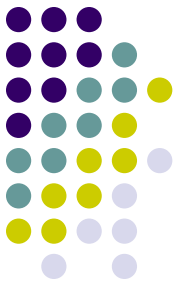


- Естественность, единообразие, интуитивность, компактность единицы ПЗ : основные свойства сущности собраны в одном месте (фрейме),
- Возможность представления знаний разного уровня:
 - На поверхностном: запись лингвистической информации (поверхностная семантика ЕЯ-фраз).
 - На глубоком: естественный и эффективный путь классификации и построения сетей фреймов ПО
- Фрейм-прототип: представление взаимосвязанных декларативных и процедурных знаний
- Присоединенные процедуры:
 - возможность вычислить значение слота
 - средство организации различных способов вывода

НЕДОСТАТКИ ФРЕЙМОВОЙ МОДЕЛИ



- Слабый теоретический базис, отсутствие стандартов на структуру/состав фреймов
- Трудоёмкость внесения изменений в родовидовую иерархию (таксономию)
- Сложность модели во всей ее полноте
- Неэффективность процедур вывода на фреймах:
 - Отслеживание интерпретатором условий вызова процедур-демонов
 - FRL: многопроходность по АКО-связям
- Трудности управления наследованием в условиях исключений (непредсказуемость поведения)
- Сложность отладки фреймовых представлений



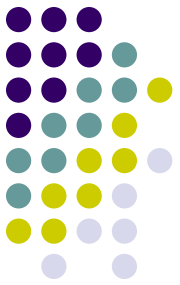
ЗАКЛЮЧЕНИЕ

- Фрейм – декларативно-процедурная структура ПЗ, обладающая большой гибкостью (сравн.: семантические сети – декларат. структура)
- Присоединенные процедуры выражают (в процедурной форме) знания, связанные с соответствующим слотом или фреймом в целом
- Фреймовая модель – Объектная модель ПЗ
 - Знание (декларативное и процедурное) организуется вокруг фрейма-объекта
 - Вызовы присоединенных процедур можно рассматривать как передачу сообщений между фреймами-экземплярами



СПАСИБО ЗА ВНИМАНИЕ!

ДОМАШНЕЕ ЗАДАНИЕ № 4



Формализация знаний ПО на базе фреймовой модели

- Для выбранной сущности (ситуации, явления, события, процесса, объекта) некоторой ПО описать связанное с ней знание в виде 1-2 взаимосвязанных фреймов-прототипов, включающих несколько присоединенных процедур. Привести соответствующие фреймы-экземпляры, дать комментарии. Синтаксис произвольный (см. слайды 16, 17, 31). (4-5 баллов)
- Срок выполнения – 2 недели (вплоть до 24 марта).
- Сдавать задание в письменном / распечатанном виде или высылать на почту nadezda.v.gruzdeva@gmail.com
- Возможные сущности/ситуации:
 - Прием у врача
 - Кредит в банке
 - Лекция / Экзамен / Коллоквиум
 - Курсовая работа
 - Прием документов у абитуриента...
 - Экскурсия
 - Аренда
 - Свадьба

и т.п.

ДОМАШНЕЕ ЗАДАНИЕ № 5

(дополнительное)



- Сравнить с предикатом:

а) фрейм-экземпляр

б) фрейм-прототип

При сравнении этих понятий указать общее и основные отличия. (4 балла)

- Срок выполнения – 3 недели
(вплоть до 31 марта).

- Сдавать задание
в письменном/распечатанном виде
или высылать на почту

nadezda.v.gruzdeva@gmail.com