



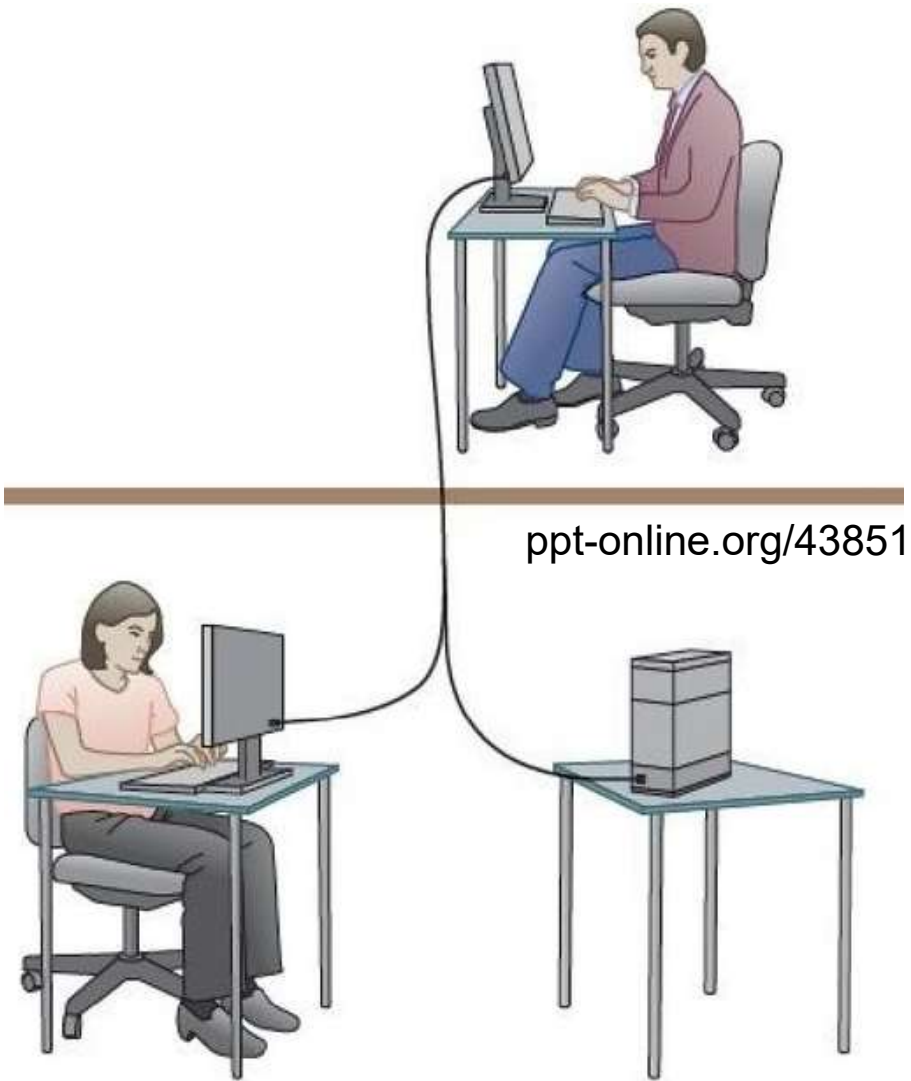
Наука будущего – наука молодых

Современные методы представления и обработки знаний

Выходец Валерий Святославович

24 августа 2022 г.

Может ли машина мыслить?



Тест Тьюринга:

«Человек взаимодействует с одним компьютером и одним человеком на некотором языке.

На основании ответов на вопросы он должен определить, с кем он разговаривает: с человеком или компьютерной программой».

Недостаток теста – отсутствие методики проверки на «умение мыслить».

Обладает ли машина знаниями?



ppt-online.org/43851



Тест на обладание знаниями:
«Человек взаимодействует с
одним собеседником на некотором
языке.

На основании ответов на вопросы
он должен определить, обладает ли
собеседник знаниями или нет».

Какова методика проверки на
«обладание знаниями»?

Что такое знание?

- **Знание** – психическое явление, которое субъективно (идеально) и не может быть непосредственно изучено.
- **Знание** – результат *познавательной деятельности*, который логически или фактически обоснован и допускает эмпирическую или практическую проверку на достоверность.

Приобретение знаний осуществляется в результате педагогического процесса, самообразования и жизненного опыта.

Применение знаний – адекватное отражение окружающего мира.

Представление знаний – отчуждение знания от его носителя во внешней (объективной) форме.

Объективация знаний – признание знаний, представленных в некоторой внешней форме объективно истинными и социально значимыми.

Каковы формы рационального познания?

- **Понятие** – форма мысли, что-либо выделяющая и называющая: субъект s , предикат p , объект o , атрибут a .
- **Суждение** – форма мысли, что-либо утверждающая: $A \sim s p o a$.
- **Умозаключение** – умственное действие, связывающее суждения-посылки с суждениями-следствиями: $A \vee B \rightarrow A$.
- **Рассуждение** – мыслительная деятельность, направленная на получение из суждений-посылок суждений-следствий.
- **Теория** – суждения-следствия, полученные в процессе рассуждения из некоторых суждений-посылок с помощью умозаключений заданного множества.
- **Знание** – результат отражения реальности, представленный в сознании субъекта понятиями и суждениями, утвержденными некоторым множеством умозаключений.

Что значит «обладать знаниями»?

- **Сужение** области знания (выделение предметной области).
- **Владение** единичными (первичными) сущностями предметной области (называть, распознавать, различать).
- **Определение** имеющихся понятий и создание новых (интенционально, экстенционально, генетически).
- **Оперирование** суждениями (фактами) о сущностях и понятиях предметной области (высказывать, запоминать, проверять).
- **Выполнение** умозаключений, присущих выбранной предметной области.
- **Получение** скрытого знания путем рассуждений на имеющихся знаниях (вывод на знаниях, запросы к знаниям).
- **Приобретение** нового знания путем обучению с учителем, с подкреплением, без учителя.

Модели представления знаний

- **Модель** представления знаний – средства формального описания знаний (язык) и методы оперирования ими.
- **Модели знаний:**
 - *логические* (язык логики): исчисление предикатов, дескрипционные логики;
 - *сетевые* (язык ориентированных графов): семантические сети, концептуальные графы, фреймы, «сущность-связь», объектные;
 - *продукционные* (язык продукций): логические, грамматические;
 - *понятийные* (понятийный язык).

Логические модели

Логика первого порядка

- **Высказывания** (φ, ψ) :
 - атомарные a, b, \dots ;
 - $(\varphi), \neg \varphi, \varphi \rightarrow \psi$;
 - $\exists a \varphi(a), \forall a \varphi(a)$.
- **Аксиомы:**
 - $(\varphi \rightarrow (\psi \rightarrow \gamma)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \gamma)),$
 - $(\varphi \rightarrow \psi) \rightarrow (\neg \psi \rightarrow \neg \varphi),$
 - $\varphi \rightarrow (\psi \rightarrow \varphi),$
 - $\varphi(a) \rightarrow \exists b(\varphi(b)),$
 - $\forall a(\varphi(a)) \rightarrow \varphi(b).$
- **Пример (утверждения и факты):**
 - $\exists x S(x), \exists x \exists y P(x, y), \exists x \exists y \exists z M(x, y, z);$
 - $\exists x \neg S(x), \exists x \exists y \neg P(x, y);$
 - $\forall x \exists y \exists z (M(x, y, z) \rightarrow z \wedge \neg P(x, y));$
 - $\forall x \forall y (z \vee S(x) \rightarrow M(x, y, z));$
 - $\exists x \forall y (S(x) \rightarrow \neg P(x, y));$
 - $S(1), S(2), P(0, A), P(1, D), M(0, E, s).$

Дескрипционная логика

- **Концепты** (C, D) и **роли** (R) :
 - атомарные концепты A, B, \dots ;
 - \top (домен), \perp (пустой концепт);
 - $\neg C, C \sqcap D, C \sqcup D$;
 - $\forall R.C, \exists R.C$.
- **Пример Tbox (утверждения):**
 - Women = Person \sqcap Female;
 - Mother = Women \sqcap \exists hasChild. \top ;
 - \forall hasChild.Person \subseteq Person;
 - Doctor \subseteq Person.
- **Пример Abox (факты):**
 - Mary: Woman \sqcap \neg Doctor;
 - Mary: \exists hasChild.Female;
 - Mary hasChild Peter;
 - Peter: Doctor \sqcap \forall hasChild. \perp .

Недостатки логических моделей

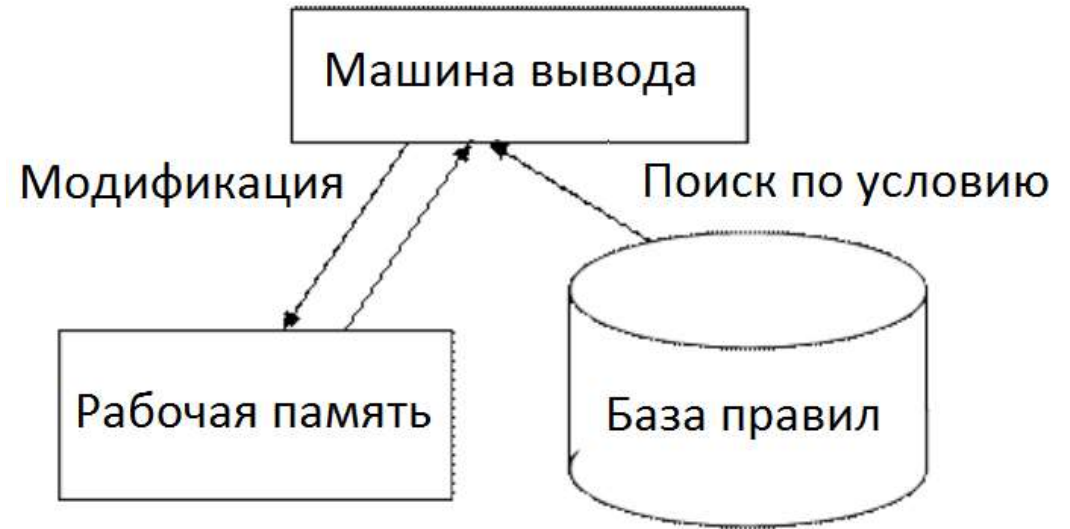
- Монотонность вывода – невозможно пересмотра промежуточных данных вывода (все факты, а не гипотезы).
- Невозможность использовать в качестве переменных предикаты.
- Неразрешимость формул логики первого порядка.
- Сложность вывода на знаниях (логика первого порядка) или низкая выразительность языка (дескрипционная логика).
- Нет средств для представления процедурных знаний.
- Обучение модели только с учителем.

Недостатки сетевых моделей

- Не содержит структуру предметной области, сложность модификации и актуализации.
- Выразительная мощность более слабая по сравнению с логическими моделями.
- Большое число отношений, результат запроса – фрагмент сети (поиск изоморфного подграфа – NP-полная задача).
- Сложность вывода, необходимость использования эвристик.
- Вывод на сетевых моделях не может гарантировать достоверность результата, не имеет логического обоснования.
- Разметка дуг задает понятия (предикаты), однако использование таких понятий требует формализации в логике второго порядка.
- Нет средств для представления процедурных знаний.
- Обучение модели только с учителем.

Продукционная модель

$P = \langle S, A \rightarrow B, Q \rangle$,
 S - предусловие;
 $A \rightarrow B$ - ядро продукции;
 Q - постусловие.



Прямой вывод: от фактов к заключению.

Обратный вывод: от заключения к фактам.

Пример

Утверждение 1: Если (отдых летом) И (человек активный), ТО (ехать в горы).

Утверждение 2: Если (любит солнце), ТО (отдых летом).

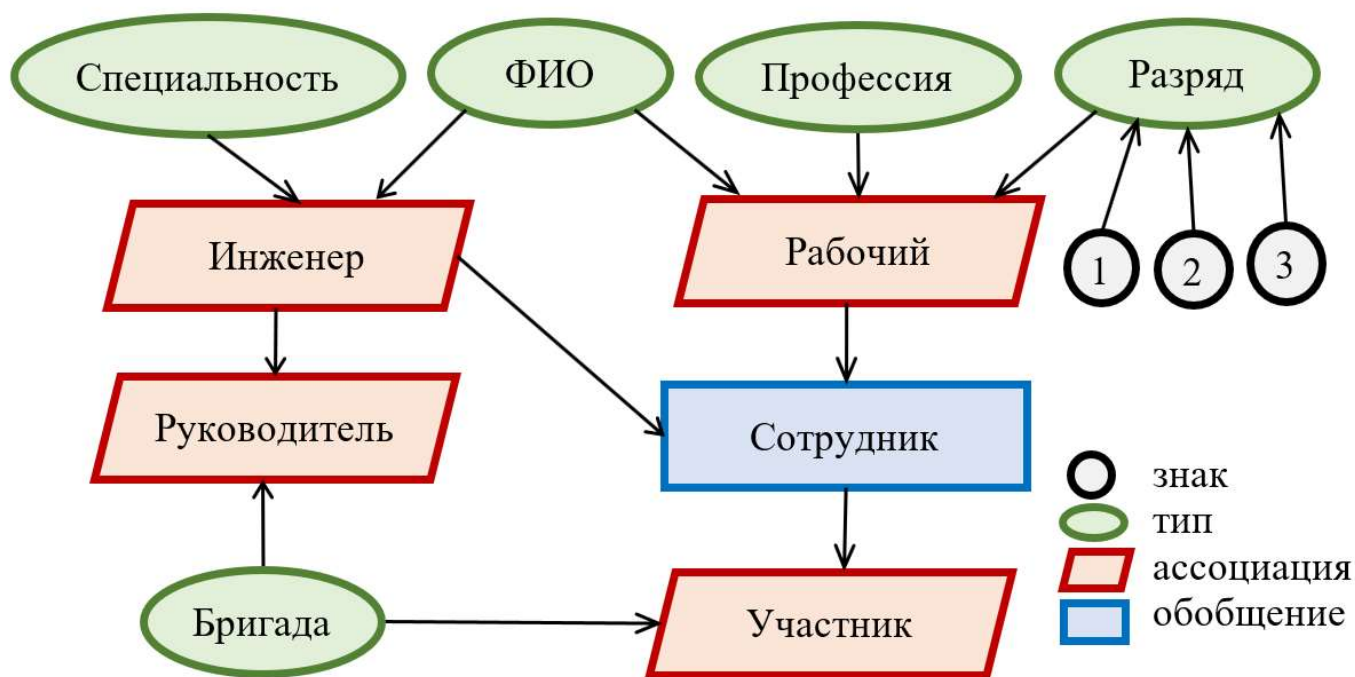
Факты: (человек активный), (любит солнце).

Недостатки продукционной модели

- Трудности проверки на непротиворечивость и полноту.
- Недетерминированность выбора из активированных продукций.
- Необходимость использования эвристик при выводе.
- Нет средств для представления процедурных знаний.
- Обучение модели только с учителем.

Понятийная модель

- **Понятийная структура** задает способы образования понятий.



- **Виды** понятий:
 - знак (элементарное);
 - тип (простое);
 - ассоциация;
 - обобщение.
- **Связи** понятия – указание понятий, использованных для его образования.

- **Понятийная модель** – понятийная структура и описание сущностей понятий предметной области (задание перечислимых или разрешимых множеств сущностей).

Конкретные понятия

○ **Элементарное понятие** (понятие-знак, понятие-значение) образуется при мысленном *выделении* в предметной области уникального представления (целостного образа) и присвоения ему имени (абстракция *отождествления*).

Примеры: Зеленый, Много, Красиво, Всегда, Движение, Любовь.


В информационных системах: 'с', 12, 1.3e-12, 18.06.2019 15:45, "abc" (значения простых типов данных).

○ **Простое понятие** (понятие-тип, понятие-признак) образуется путем *объединения* элементарных понятий, сходных в некотором смысле (абстракция *типизации*).

Примеры: Цвет (Красный, Зеленый, ...), Целое число (1, 2, 3, ...).

В информационных системах: Символ, Целое, Число с плавающей запятой, Дата и время, Строка (простые типы данных).

Абстрактные понятия

 **Понятие-ассоциация** образуется путем *соединения* нескольких понятий в одно составное (абстракция *ассоциации*).

Пример: понятие-ассоциация Погода как соединение понятий Место, Дата, Температура, Влажность, Ветер, Облачность и т.п.

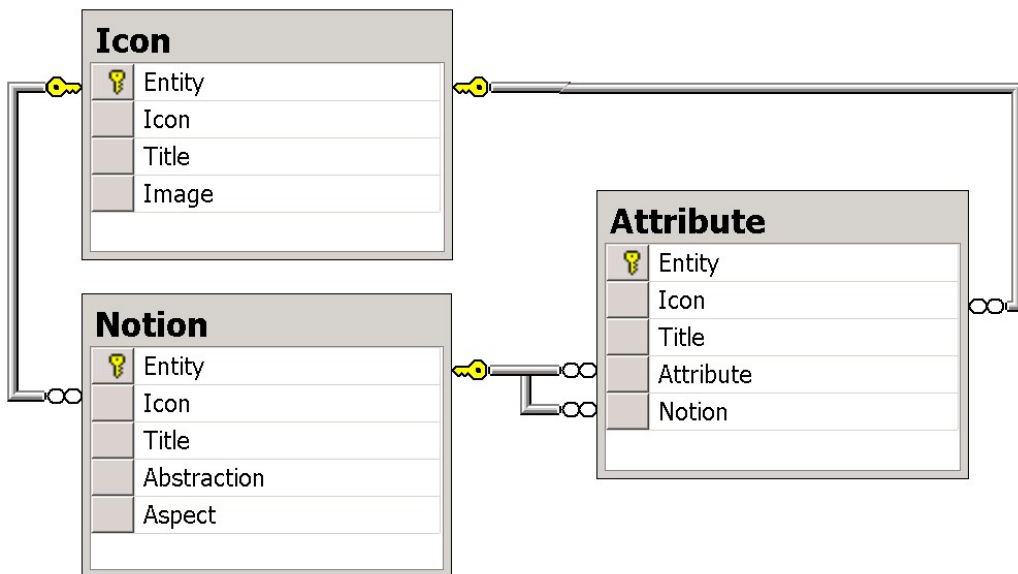
В информационных системах: массив структур, таблица базы данных.

 **Понятие-обобщение** образуются путем *объединения* нескольких понятий в одно общее (абстракция *обобщения*).

Пример: Понятие-обобщение Фрукт как объединение понятий Яблоко, Груша, Персик, Абрикос и др.

В информационных системах: массив базовых объектов, union-запрос к базе данных.

Представление знаний



Attribute

Entity	Icon	Title	Attribute	Notion
6000000003	63000000020	Notion	61000000061	61000000060
6000000004	63000000023	Attribute	61000000061	61000000060
6000000001	63000000021	Abstract	61000000064	61000000061
6000000002	63000000034	Aspect	61000000065	61000000061
6000000019	63000000019	Image	61000000045	61000000063

Notion

Entity	Icon	Title	Abstraction	Aspect
61000000060	63000000054	Attribute	A	Common
61000000061	63000000020	Notion	A	Common
61000000062	63000000018	Model	A	Common
61000000063	63000000019	Icon	A	Common
61000000064	63000000021	Abstraction	A	Common
61000000065	63000000034	Aspect	A	Common

Icon

Entity	Icon	Title	Image
63000000018	63000000018	Model	0x424D3605000000...
63000000019	63000000019	Icon	0x424D3605000000...
63000000020	63000000020	Notion	0x424D3605000000...
63000000021	63000000021	Abstract	0x424D3605000000...
63000000022	63000000022	Abstraction	0x424D3605000000...
63000000023	63000000023	Tag	0x424D3605000000...

Схема понятия

- **Абстракции** понятий $K = \{c, t, a, g\}$, где c – знак, t – тип, a – ассоциация, g – обобщение.
- **Имена** понятий $N = \{N_i | i = \overline{1, n}\}$ - понятие-знаки предметной области, обозначающие понятия.
- **Схема** понятия – набор абстрагируемых понятий (понятий-атрибутов, на которых образовано определяемое понятие).
- **Примеры:**
 - схема понятия 1 – $(c, 1)$;
 - схема понятия Разряд – $(t, 1, 2, 3)$;
 - схема понятия Рабочий – $(a, \text{Профессия}, \text{Разряд}, \text{ФИО})$;
 - схема понятия Инженер – $(a, \text{Специальность}, \text{ФИО})$;
 - схема понятия Руководитель – $(a, \text{Инженер}, \text{Бригада})$;
 - схема понятия Сотрудник – $(g, \text{Инженер}, \text{Рабочий})$;
 - схема понятия Участник – $(a, \text{Сотрудник}, \text{Бригада})$.

ФАКТЫ

- **Факты** - истинные суждения с логическими связями И (\wedge), ИЛИ (\vee), НЕ (\neg) и двумя видами элементарных выражений:

- одноместными предикатами принадлежности понятия E понятию N вида $N(E)$;

- отношениями вида $E[N] \circ V$, где $E[N]$ - функтор, возвращающий значение понятия-атрибута N сущности E , \circ - знак отношения ($=$, $<$, $>$ и т.п.), некоторая сущность V .

- **Примеры:**

- Разряд(2) – 2 есть сущность понятия-типа Разряд;
- Рабочий(X) – X есть сущность понятия-ассоциации Рабочий;
- $X[\text{Разряд}] = 3$ – Рабочий X имеет Разряд, равный 3;
- Сотрудник(Y) – Y есть сущность понятия-обобщения Сотрудник;
- $\neg Y[\text{Рабочий}] = X$ – Сотрудник Y не является Рабочим X .

Правила вывода

$$a: N_j(E) \rightarrow \bigwedge_{\forall N_i \in H_j} E[N_i]; \quad g: N_j(E) \leftrightarrow \bigvee_{\forall N_i \in H_j} N_i(E),$$

где a – правило вывода из понятия-ассоциации N_j , g – правило вывода из понятия-обобщения N_j , H_j - схема понятия N_j .

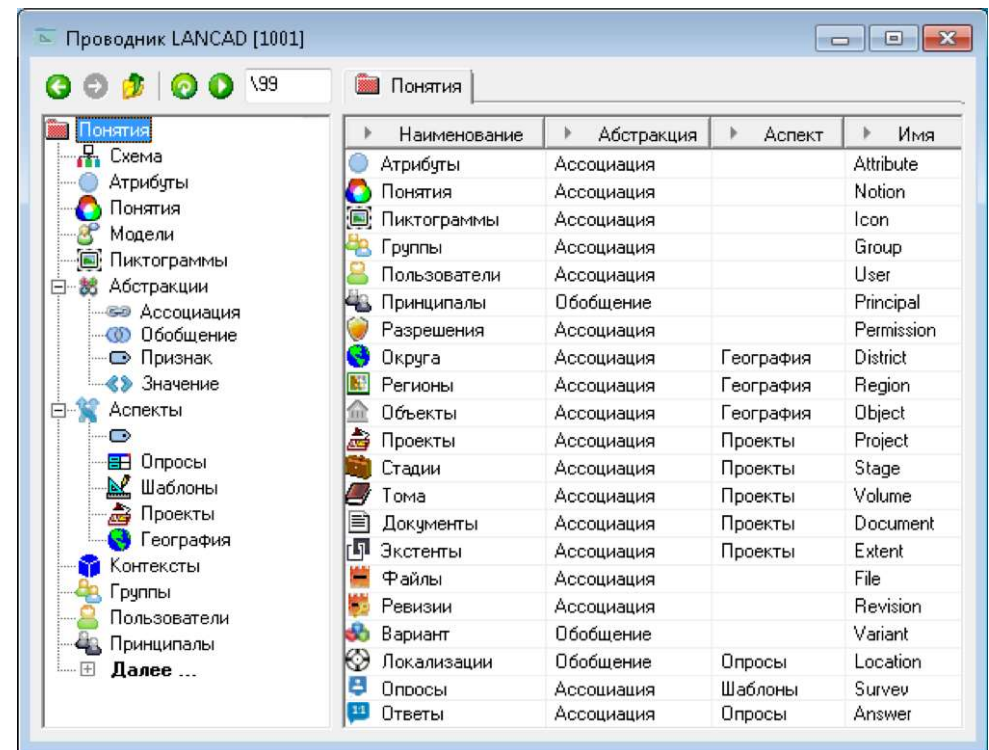
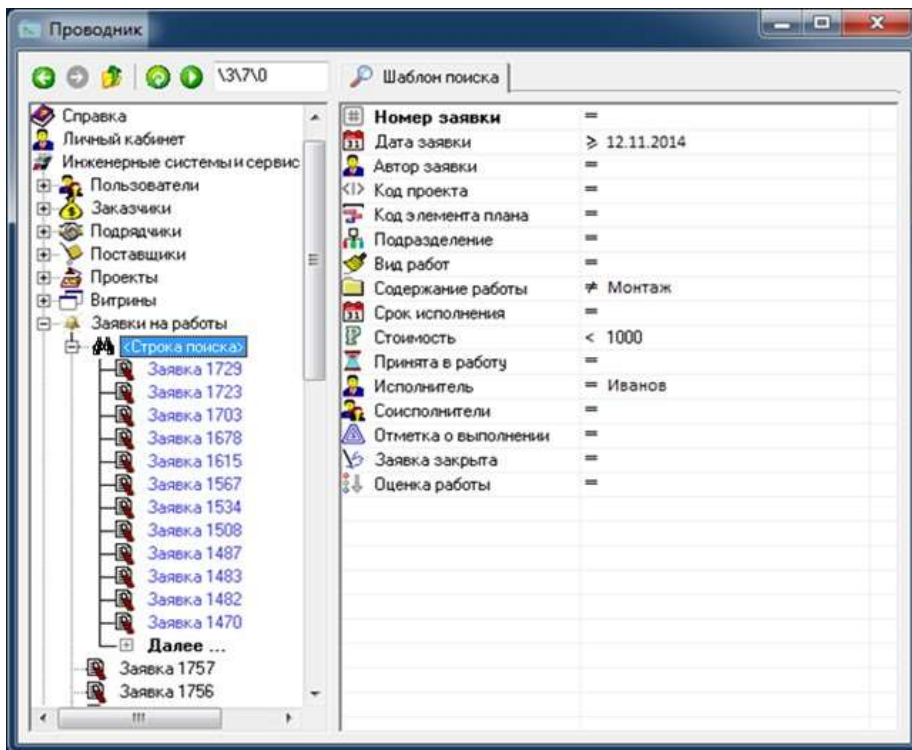
- **Примеры** правил вывода:

- $a: \text{Инженер}(X) \rightarrow X[\text{Специальность}] \wedge X[\text{ФИО}];$
- $a: \text{Рабочий}(X) \rightarrow X[\text{ФИО}] \wedge X[\text{Профессия}] \wedge X[\text{Разряд}];$
- $a: \text{Руководитель}(X) \rightarrow X[\text{Инженер}] \wedge X[\text{Бригада}];$
- $g: \text{Сотрудник}(X) \leftrightarrow \text{Инженер}(X) \vee \text{Рабочий}(X);$
- $a: \text{Участник}(X) \rightarrow X[\text{Сотрудник}] \wedge X[\text{Бригада}].$

-

Язык запросов

- **Пример** запроса: «Кто из участников бригады В не является рабочим с третьим разрядом?»
 - $Участник(X) \wedge X[Бригада] = В \wedge \forall Y (X[Сотрудник] = Y \wedge Рабочий(Y) \wedge \neg Y[Разряд] = 3).$



Процедурные знания

Для представления процедурных знаний предусмотрены процедуры и функции, которые, как и все в понятийных моделях, являются сущностями соответствующих понятий.

```
// Определение функции повышения разряда
Повышение = (Разряд) → Разряд < 3 ? 1 : 0;
// Создание индексной переменной
Индекс = 0;
// Открытие транзакции с именем Разряды
begin Разряды;
// Запрос сущностей локального понятия Рабочие
Рабочие = (X)(Участник(X) != ' ' && X[Бригада] == ' В '
&& Рабочий(X[Сотрудник]) != ' ');
```

```
// Цикл по сущностям понятия Рабочие
while (Индекс < Рабочие{ })
{
    // Создание локальной сущности Рабочий
    Рабочий = Рабочие{Индекс};
    // Увеличение атрибута Разряд
    Рабочий[Разряд] += Повышение(Рабочий[Разряд]);
    // Приращение индекса сущности понятия Рабочие
    Индекс = Индекс + 1;
}
// Обновление измененных сущностей
()Рабочие;
// Успешное завершение транзакции
commit Разряды;
```

Обучение модели

- **Обучение с учителем.** Создание, изменение и удаление понятий и сущностей непосредственно пользователем.
- **Обучение с подкреплением.** Автоматический ввод описаний предметной области от разного рода внешних источников, распознавание уже имеющихся сущностей или добавление вновь найденных.
- **Обучение без учителя.** Автоматический ввод и распознавание описаний внешних сущностей, пополнение экстенсионалов понятий, если распознанные сущности не обнаружены, создание новых понятий и наполнение их предметным содержанием.

Заключение

- **Трудности** представления и обработки знаний в логических, сетевых и продукционных моделях. Наилучшая модель для представления и обработки знаний – понятийная модель.
- **Понятийная модель** не использует языковую метафору для представления знаний (субъект+предикат+объект), в ней понятиями являются и предикаты.
- **Основа** понятийного моделирования знаний – три предельно общие операции с понятиями (отождествление, ассоциация, обобщение).
- **Семантическая инвариантность** понятийной модели – для ее интерпретации не надо знать семантику предикатов, ролей, помеченных дуг и т.п.: все, что необходимо знать о предметной области, содержится в модели.
- **Обучение с подкреплением и без учителя** – структурирование входного потока данных на основе имеющихся знаний и выделение в нем известных и неизвестных сущностей для подкрепления и пополнения понятийной модели.
- **Представление процедурных знаний** в понятийной модели.
- **Высокая эффективность** обработки знаний в формализме понятийной модели: поиск сущности в таблице понятия с n сущностями имеет асимптотику $\ln n$.

Литература

- 1. Baral C. Knowledge Representation, Reasoning and Declarative Problem Solving. – Cambridge: Cambridge University Press, 2004. – 541 p.
- 2. The description Logic Handbook: Theory, Implementation and Applications / Eds. F. Baader, D. Calvanese, D. L. McGuinness, et al. – New York: Cambridge University Press, 2003. – 573 p.
- 3. Chein M., Mugnier M.-L. Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs. – Springer, 2009. – 425 p.
- 4. Klahr D., Langley P., Neches R. Production System Models of Learning and Development. Cambridge. – Cambridge: The MIT Press, 1987. – 466 p.
- 5. Выхованец В.С. Понятийный анализ и понятийное моделирование // Управление большими системами. Вып. 92. – М.: ИПУ РАН, 2021. – С. 64-109. – <https://doi.org/10.25728/ubs.2021.92.4>.