

ФПМИ  МФТИ

Кафедра алгоритмов и технологий программирования

Понятийное моделирование знаний

Доклад на научно-исследовательском семинаре

Выхованец Валерий Святославович, д.т.н., доц.

14 мая 2024 г.

Что такое знание?

Знание – психическое образование, 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.

Структура знаний

Понятие – форма мысли, что-либо выделяющая и называющая: субъект **S**, предикат **P**, объект **O**, атрибут **A**.

Суждение – форма мысли, что-либо утверждающая: **X = S P O A**.

Умозаключение – умственное действие, связывающее суждения-посылки с суждениями-следствиями: **X ∨ Y → X**.

Рассуждение – мыслительная деятельность, направленная на получение из суждений-посылок суждений-следствий.

Теория – суждения-следствия, полученные в процессе рассуждений из некоторых суждений-посылок с помощью умозаключений заданного вида (достоверные, правдоподобные, вероятные).

Знание – объективированный результат отражения реальности, представленный в сознании субъекта понятиями и суждениями, утвержденными некоторым множеством умозаключений.

Модели знаний

Модель знаний – средства формального описания знаний (язык знаний) и методы оперирования ими (вывод на знаниях).

Модели знаний:

- *логические* (логический язык): исчисление предикатов, дескрипционные логики (дедуктивный, индуктивный и абдуктивный вывод);
- *продукционные* (язык продукций): логические, грамматические;
- *сетевые* (язык взвешенных ориентированных графов): семантические сети, концептуальные графы;
- *фреймовые* (объектный язык): фреймовые и объектные модели, модели «сущность-связь», реляционные модели;
- *понятийные* (понятийный язык).

Недостатки моделей знаний

- Невозможность использовать предикаты в качестве переменных.
- Неразрешимость формул (логика первого порядка).
- NP-полнота вывода на знаниях (логика первого порядка).
- Неестественность языков знаний (дескрипционная логика).
- Трудности проверки знаний на непротиворечивость и полноту.
- Недетерминированность процедур вывода на знаниях.
- Необходимость использования эвристик при выводе.
- Вывод на знаниях не гарантирует достоверность результата и не имеет логического объяснения (сетевые модели).
- Нет средств для представления процедурных знаний.
- Обучение моделей знаний выполняется только с учителем.

Нейронные сети и знания

Искусственные нейронные сети оказались непригодны для представления и обработки знаний:

- невозможность объяснения получаемых результатов,
- отсутствие гарантии верного (полного) решения задачи,
- непредсказуемость прогностической способности сети,
- трудности в решении задач по шагам и вычислительных задач,
- зависимость строения сети от решаемой задачи,
- длительное и ресурсоемкое обучение, риск переобучения сети;
- сложность получения обучающей выборки,
- низкое качество обучения при зашумленных (неточных) данных,
- невозможность использования в меняющихся условиях, и т. д.

Понятийная модель

Понятийная модель – это понятийная структура предметной области и описание сущностей принадлежащих ей понятий.


Понятийная структура перечисляет понятия и указывает способы их образования (абстрагирования).

Описание сущностей – это задание перечислимых и разрешимых множеств сущностей понятия (декларативные и процедурные знания).

Примеры:


- **Сотрудник {Рабочий, Инженер}**: Сотрудник – это обобщение (объединение) сущностей понятий Рабочий и Инженер;
- **Руководитель [Инженер, Проект]**: Руководитель – это ассоциация (соединение) сущностей понятий Инженер и Проекта.
- **Руководитель = {[1#Инженер, 2#Проект], [6#Инженер, 8#Проект], ...}**: 1#Руководитель = [1#Инженер, 2#Проект], ...

Конкретные понятия

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

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

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*, частный случай обобщения).

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.

Описание сущностей

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

- одноместными предикатами принадлежности сущности E понятию N вида $N\{E\}$;

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

Примеры суждений:

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

Правила вывода

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

где g (a) – правило вывода из понятий-обобщений N_j (понятий-ассоциаций N_k); H_j (H_k) – схема понятия N_j (N_k), набор образующих понятий; φ – выражение, задающее принадлежность сущности E понятию N_k .

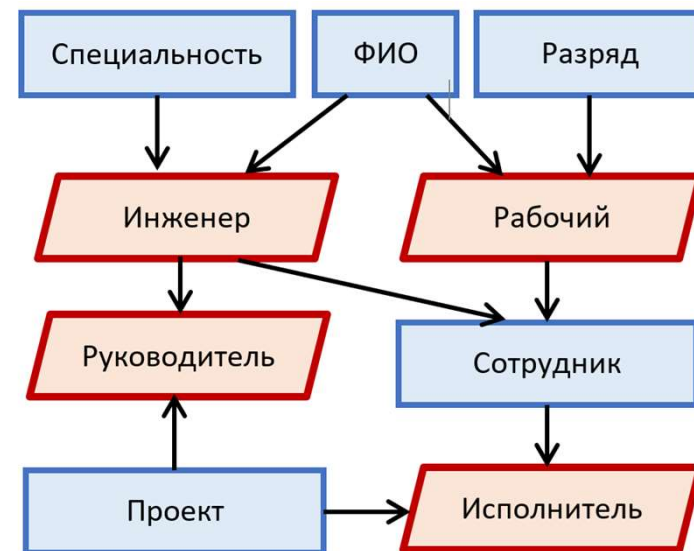
Примеры правил вывода:

g : Сотрудник $\{X\} \leftrightarrow$ Инженер $\{X\} \vee$ Рабочий $\{X\}$;

a : Рабочий $\{X\} \leftrightarrow X[\text{Разряд}] \wedge X[\text{ФИО}] \wedge$

$\wedge \varphi(X[\text{Разряд}], X[\text{ФИО}]),$

где $\varphi(X[\text{Разряд}], X[\text{ФИО}])$ – предикат, перечисляющий допустимые комбинации сущностей Разряд и ФИО в составе X .



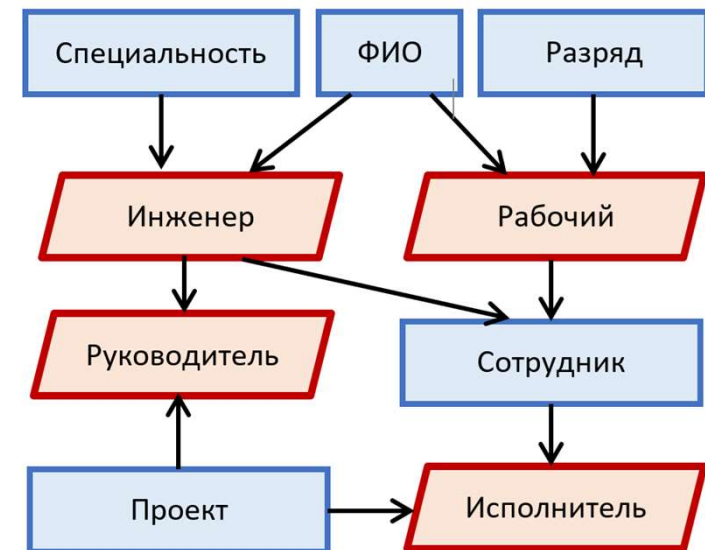
Язык запросов

Пример запроса к базе знаний: «Кто из исполнителей проекта 6 является рабочим без третьего разряда?».

$Q1 = \{X: \text{Исполнитель}\} [X] (// \{X: \text{Исполнитель}\} \leftrightarrow \forall X \in \text{Исполнитель}$
 $X[\text{Проект}] == 6 \# \text{Проект} \ \&\&$
 $\{Y: \text{Рабочий}\} (X[\text{Сотрудник}] == Y \ \&\& \ Y[\text{Разряд}] < 3)$
 $); // \text{Понятие } Q1 \text{ – это рабочие проекта 6 без 3-го разряда}$

Формулы понятийного языка – это формулы одноместного исчисления предикатов, которые

- непротиворечивы,
- разрешимы,
- полны на конечных моделях,
- эффективно вычислимы.



Процедурные знания

Для представления процедурных знаний используется понятие **Функция**:

```
define Функция [Имя: string, Аргументы: array, *Результат: notion, *Код: binary];
```

```
define Награда (a: Проект): integer {  
  // Анонимная функция повышения разряда  
  Повышение = (Разряд) -> Разряд < 3 ? 1 : 0;  
  // Создание локальной индексной переменной  
  Индекс: integer = 0;  
  // Открытие транзакции к базе знаний  
  begin "Рога и копыта"; // Строка для подключения  
  // Запрос сущностей локального понятия Рабочие  
  Рабочие = {X: Исполнитель} [X] (X[Проект] == a  
    && 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 Рабочие  
    Индекс = Индекс + 1;  
  }  
  // Обновление измененных сущностей  
  update Рабочие;  
  // Успешное завершение транзакции и отключение  
end;  
return Индекс;  
} // Пример использования - Награда(6#Проект);
```

Обучение понятийных моделей

Обучение с учителем – создание, изменение и удаление понятий и сущностей в базе знаний непосредственно пользователем.

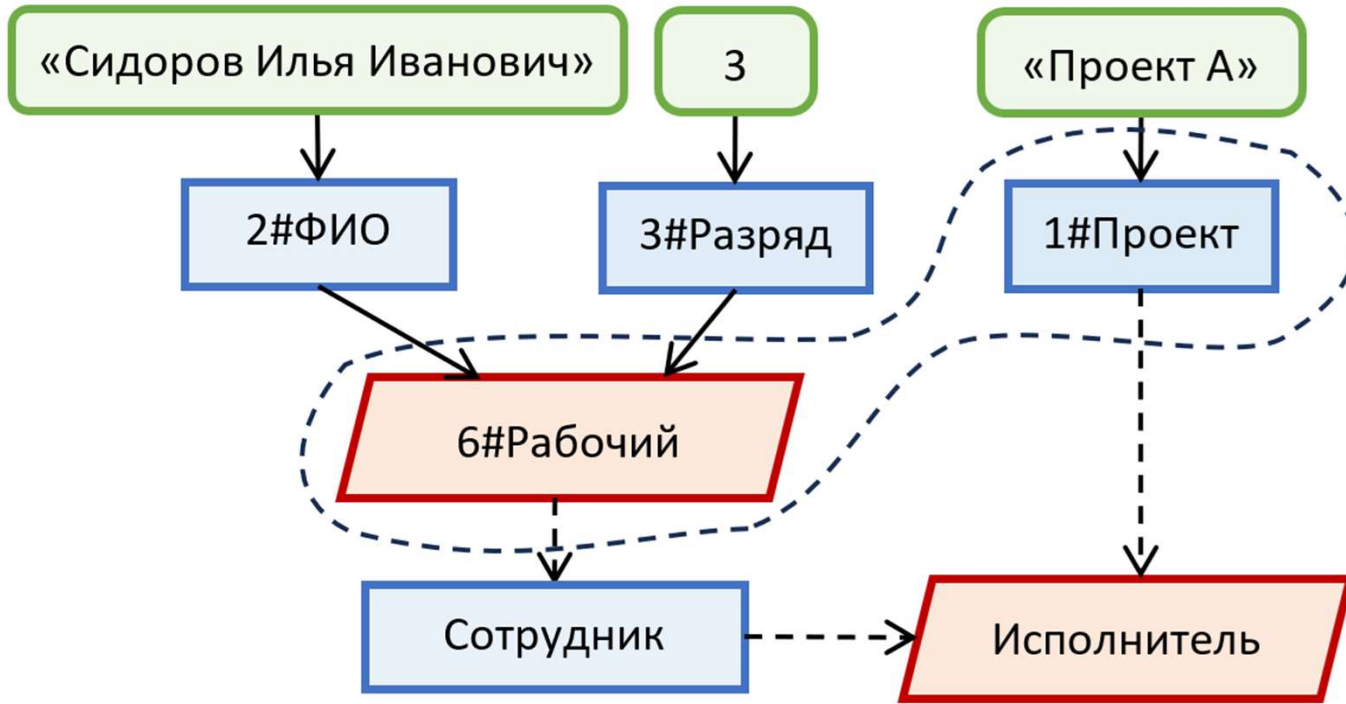
```
define Погода [Место, Дата, Температура, Влажность, ...];  
Погода ||= { ["N:55°55'46.8156", E:37°31'2.3844", '14.09.2024 20:00  
UTC+3', "12°C", "30 %", ...], ...}; // ||= – операция объединения
```

Обучение без учителя:

- автоматический ввод и распознавание поступающих сущностей;
- определение понятий этих сущностей на основе знаний, имеющихся в понятийной модели;
- пополнение имеющихся понятий новыми сущностями, если понятие поступившей сущности распознано;
- создание новых понятий из нераспознанных сущностей.

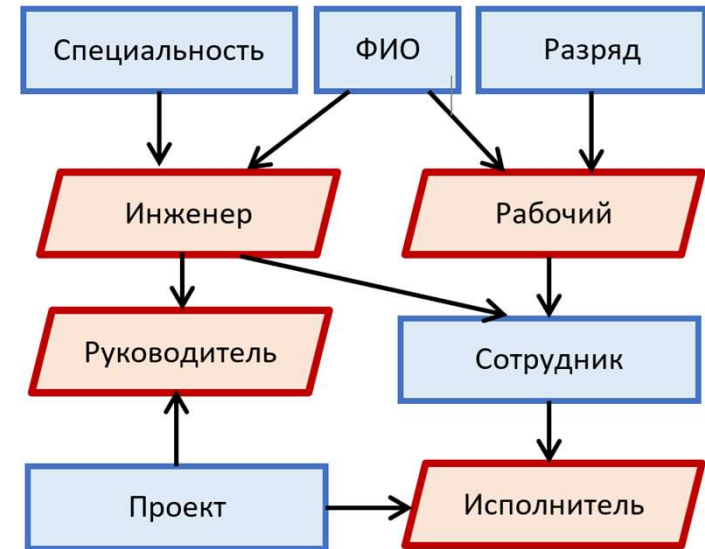
Пример обучения

«Сидоров Илья Иванович» 3 «Проект А»



Виды понятий:

- - сущность;
- ▭ - ассоциация;
- ▭ - обобщение.



Заключение

- **Трудности** представления и обработки знаний в известных моделях. Наилучшей моделью оказалась понятийная модель.
- **Понятийная модель** не использует языковую метафору для представления знаний (субъект+предикат+объект), в ней понятиями являются и предикаты.
- **Основа** понятийного моделирования – три предельно общие ментальные операции над понятиями (идентификация, ассоциация, обобщение).
- **Семантическая инвариантность** понятийной модели – для ее интерпретации не надо знать семантику предикатов, ролей, дуг и т.п.: все, что необходимо знать о предметной области, содержится в понятийной модели.
- **Представление процедурных знаний** в понятийной модели реализовано как естественный способ задания разрешимых множеств сущностей.
- **Обучение без учителя** – структурирование входного потока на основе имеющихся знаний и выделение в нем известных и неизвестных сущностей для пополнения понятийной модели новыми сущностями и понятиями.
- **Высокая эффективность** обработки знаний: поиск сущности в индексированной таблице понятия с n сущностями имеет асимптотику $\log n$.

Послесловие

- 1 **Понятийное мышление** является предельно общим и единственным механизмом рационального познания.
- 2 Не существует более общих **выразительных средств** для представления и обработки знаний, кроме как их описание в виде понятийной модели.
- 3 Имеется много задач, которые не решаются **рациональными методами**: синтез и анализ алгоритмов и программ, доказательство теорем, комбинаторные задачи, глобальная оптимизация и т. п.
- 4 Обладание знаниями и умение мыслить (осуществлять вывод на знаниях) не является достаточным для **сильного искусственного интеллекта**.
- 5 Для реализации сильного искусственного интеллекта необходимо нечто большее, например, использование не только рациональных, но и **иррациональных форм познания**, а также эмоций.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

- [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] Klahr D., Langley P., Neches R. Production System Models of Learning and Development. Cambridge. – Cambridge: The MIT Press, 1987. – 466 p.
- [4] Кесс Ю.Ю. Анализ и синтез фреймовых моделей АСУ. – М.: Энергоатомиздат, 1986. – 168 с.
- [5] Chein M., Mugnier M.-L. Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs. – Springer, 2009. – 425 p.
- [6] 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>.**