АВТОМАТИЗАЦИЯ СОЗДАНИЯ ОНТОЛОГИЙ И ИХ ПРИМЕНЕНИЕ

По материалам зарубежных публикаций

1. Средства поддержания разработки онтологии

Многослойная архитектура разработки онтологии

Язык описания онтологий:

- ***Подмножество языка логики предикатов первого порядка***
  - ***Утверждения о концептах, например: Предприятие – это утверждения обо всех предприятиях.***
- ***Иерархии (включения) = импликации***
  - ***Классификация посредством вывода:***
    “B есть разновидность A” означает:
  - ***Все объекты B являются объектами A***
    $\forall x \,.\, Bx \rightarrow Ax$,
  - ***Объект A, который не является объектом B, является противоречием (не удовлетворяющим формуле)*** $\neg \exists x \,.\, Bx \,\&\, \neg Ax$.
- ***Конструкторы***
  - ***Для формирования новых концептов из имеющихся***
    - ***and, or, not,***
    - ***some, only (‘all’), at-least, at-most***

ЧТО ТАКОЕ ОНТОЛОГИЯ НА ЯЗЫКЕ DL?

Фрейм

- ***Иерархия примитивов/элементарных концептов*** — ***да***
- ***Определения составных концептов*** — ***нет***
  - ***именовать новые концепты***
- ***Описание (аксиомы) для концептов*** — ***различны***
  - ***необходимые условия истинности концептов***
  - ***элементарные или составные***

- ***Правила/ограничения=Определение+Описание*** — ***нет***
  - ***IF что-то удовлетворяет определению, THEN оно соответствует описанию/ограничению***

Формальное определение онтологии

Основная аксиома онтологии:

$$x \in X \equiv \exists y(y \in x) \wedge \forall y,z(y \in x \wedge z \in x \rightarrow y \in z) \wedge \forall y(y \in x \rightarrow y \in X).$$

Определение 1. *Пусть* L – логический язык,

абстрактная онтология – это структура

$O: = (C, \leq_C, R, \sigma, \leq_R, IR)$, *где:*

C и R , множества концептов отношений соответственно,

$\leq c$ - частичный порядок на C, называемый иерархией концептов,

$\sigma: R \rightarrow C \times C$ - функция называемая сигнатурой,

$\leq_R$ на R- частичный порядок , где $r1 \leq R\ r2$ означает $\sigma(r1) \leq_{C \times C} \sigma(r2)$,

для $r1$, $r2 \in R$, называемый отношением иерархии.

множества IR правил вывода, выраженных на логическом языке L.

Функция $dom: R \rightarrow C$ с $dom\ (r):= \pi1(\sigma(r))$ дающая значение домена r,

Функция $range: R \rightarrow C$ с $range(r):= \pi2(\sigma(r))$ которая дает ранг отношения r.

Лексикон абстрактной онтологии

Определение 2. Лексикон для абстрактной онтологии

$O: = (C, \leq C, R, \sigma, \leq R, IR)$

$Lex:= (SC; SR; Ref\ C; Ref\ R),$

где - SC *и* SR множества лексических единиц для концептов и отношений соответственно,

$Ref\ C \subseteq SC \times C$ и $Ref\ R \subseteq SR \times R$,

- лексические референциальные присваивания для концептов и отношений соответственно.

Мы определяем для $s \in SC$,

$Ref\ C(s):=\{c \in C \mid (s,c) \in Ref\ C\}.$

Для $c \in C$ определим:

$Ref^{-1}C(c):=\{s \in SC \mid (s,c) \in Ref\ C\}.$

$Ref\ R$ и $Ref^{-1}\ R$ определяются аналогично.

Определение 3. Конкретная онтология (в узком смысле) есть пара (O, Lex), где O – абстрактная онтология и *Lex – лексикон для* O

Мереология.

Мереология представляет собою теорию частей, в отличие от теории множеств. В основе мереологии лежит рефлексивное отношение частичного порядка:

x *есть часть* x,

если x *есть часть* y *и* y *есть часть* x, *тогда* x=y,

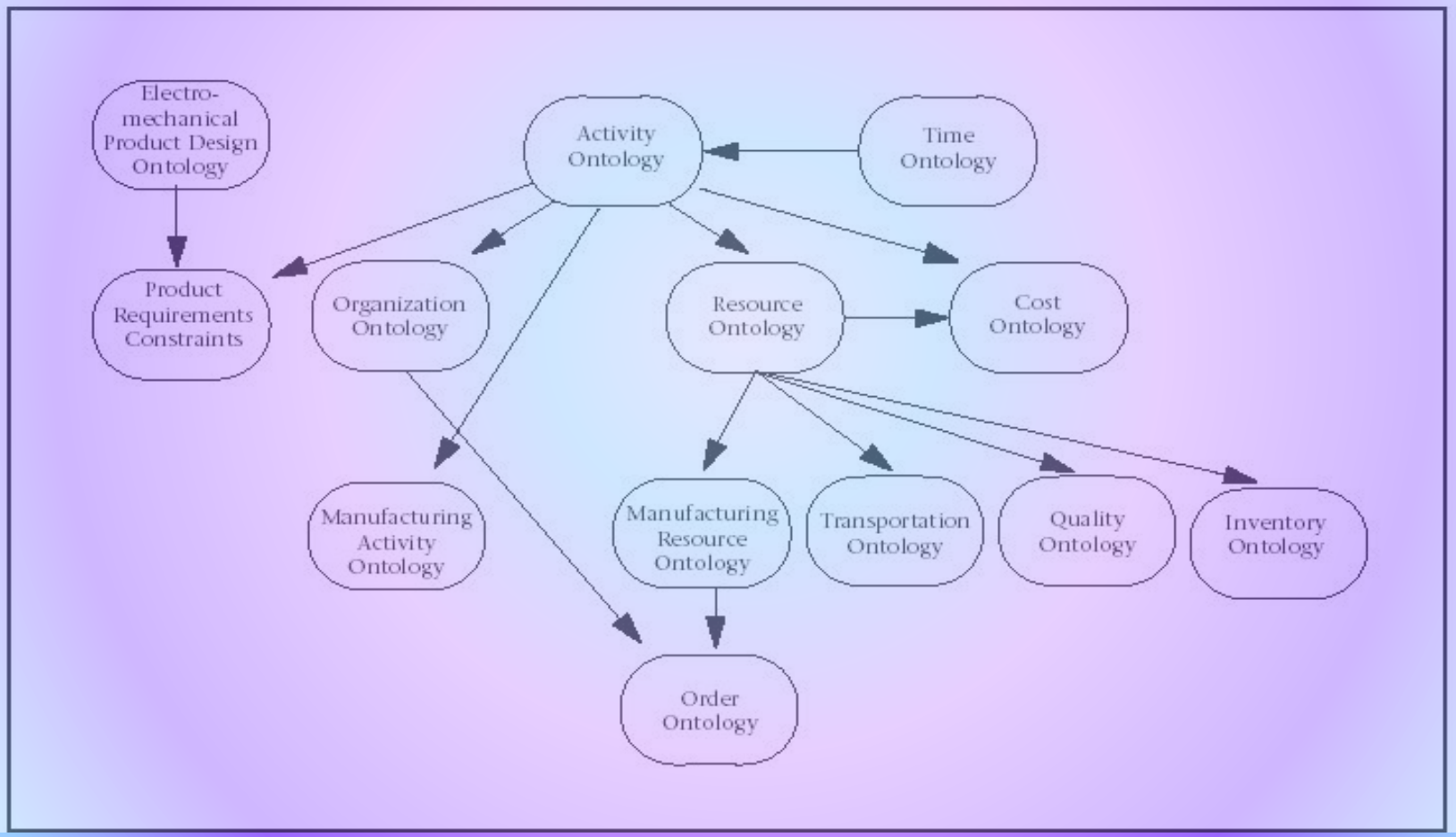
если x *есть часть* y *и* y *есть часть* z, *тогда* x *часть* z,

x *есть собственная часть* y: x *есть часть* y *и* y *не есть часть* x,

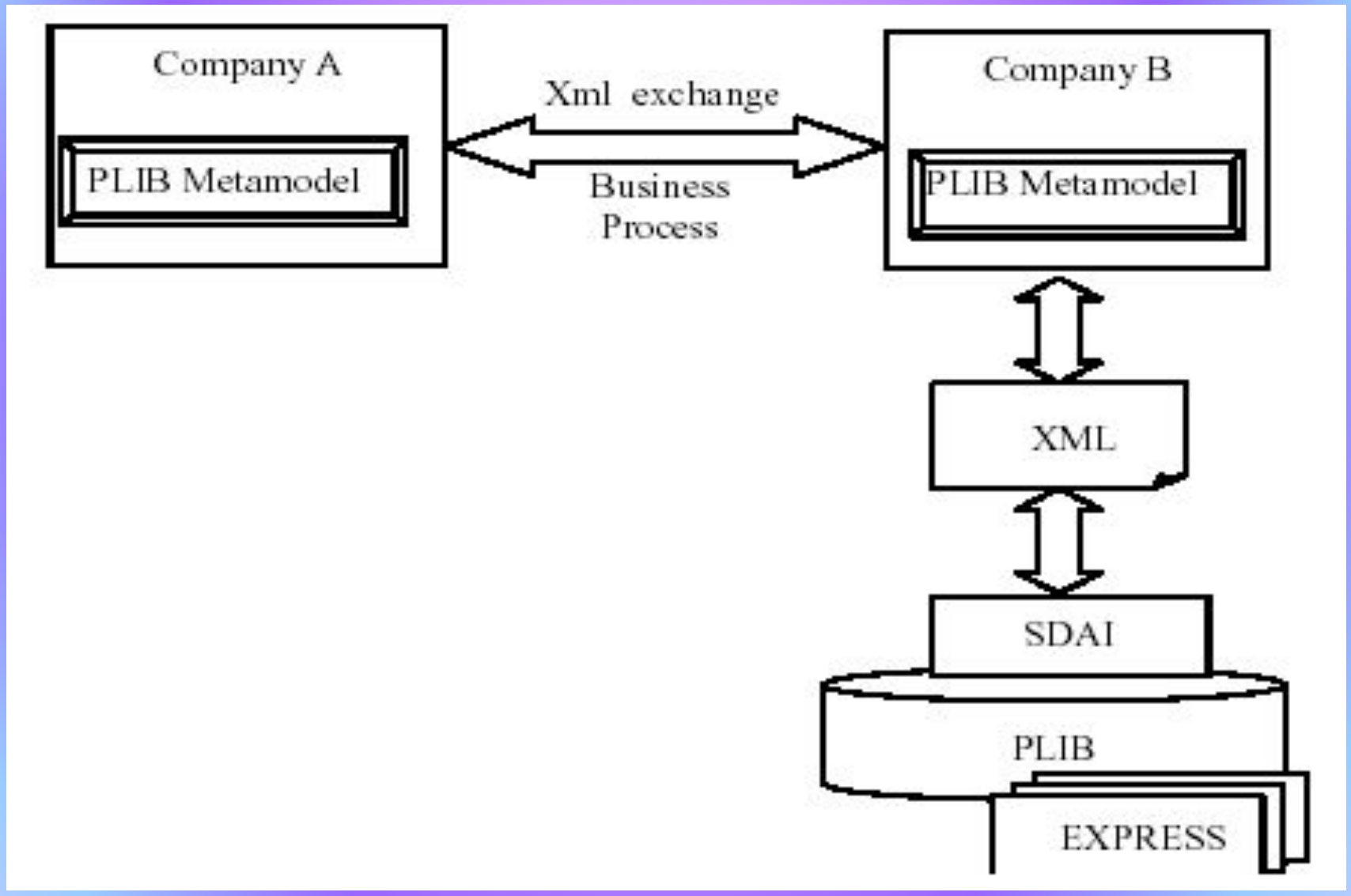
x *пересекается с* y: *существует часть* x, *которая является также частью* y,

x *и* y *не связаны:* x *и* y *не пересекаются.*

Дерево онтологий *TOVE*



Архитектура системы *PLIB*.



Определения классов в OIL и соответствующие конструкции RDF(S).

Примитивы OIL	RDFS syntax	Type
class-def	rdfs: Class	class
subclass-of	rdfs: subClassOf	property
class-expression	oil: ClassExpression	class
AND	oil: AND (subclass of ClassExpression)	class
OR	oil: OR (subclass of ClassExpression)	class
NOT	oil: NOT (subclass of ClassExpression)	class
slot-constraint	oil: Slot Constraint oil: has Slot Constraint (rdf: type of rdfs: Constraint Property) oil: NumberRestriction	class property class
has value	oil: Has Value (subclass of oil: Slot Constraint)	class
value-type	oil: Value Type (subClass of oil: Slot Constraint)	class
max-cardinality	oil: Max Cardinality (subClass of oil: NumberRestriction)	class
cardinality	oil: Cardinality (subClass of oil: NumberRestriction)	class

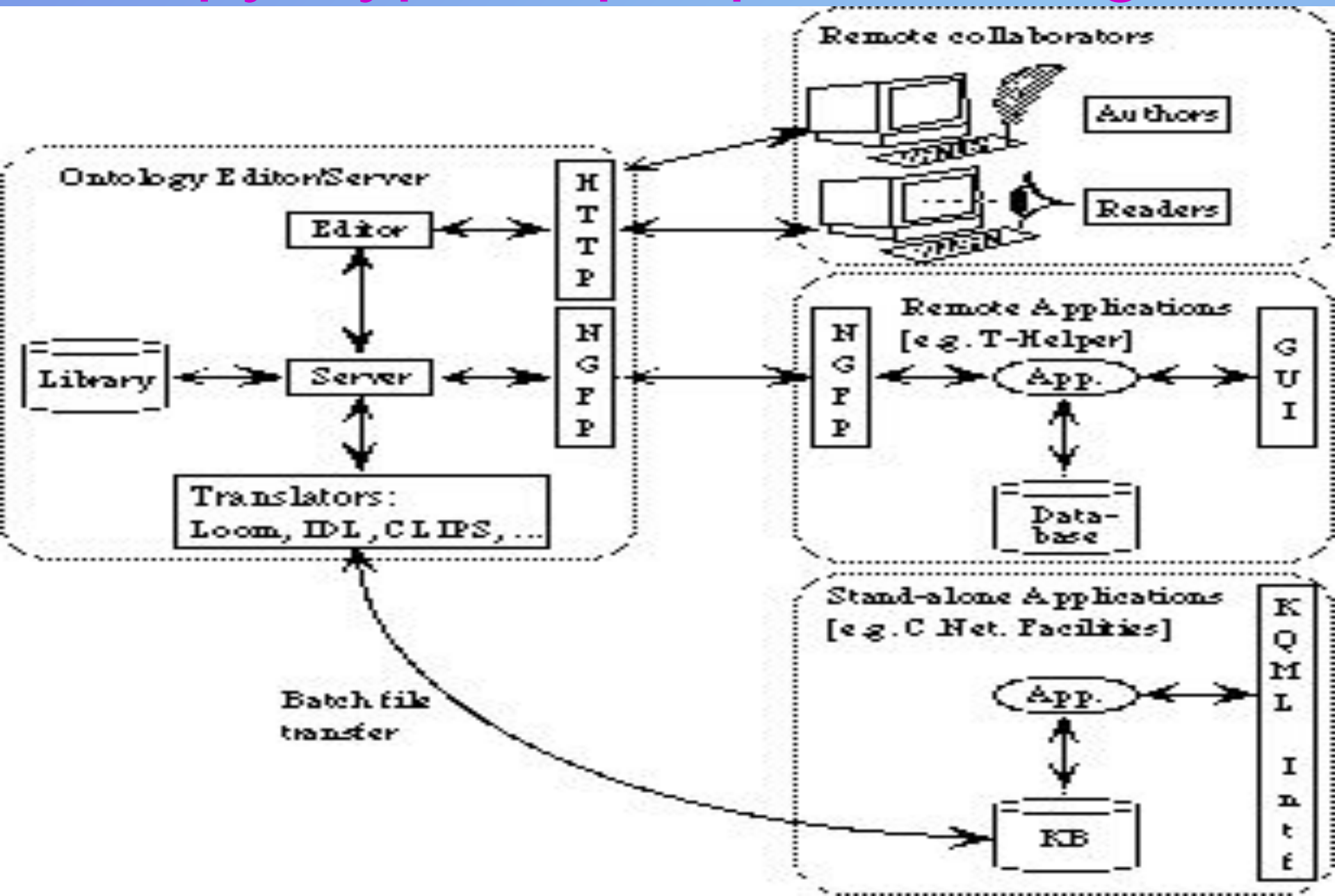
Определение спотов в OIL и соответствующие RDFS конструкции

OIL primitive	RDFS syntax	Type
slot-def	rdf: Property	class
sub slot-of	rdfs: Sub Property Of	property
domain	rdfs: domain	property
range	rdfs: range	property
inverse	oil: inverse Relation Of	property
transitive	Oil: Transitive Relation	class
symmetric	oil: Symmetric Relation	class

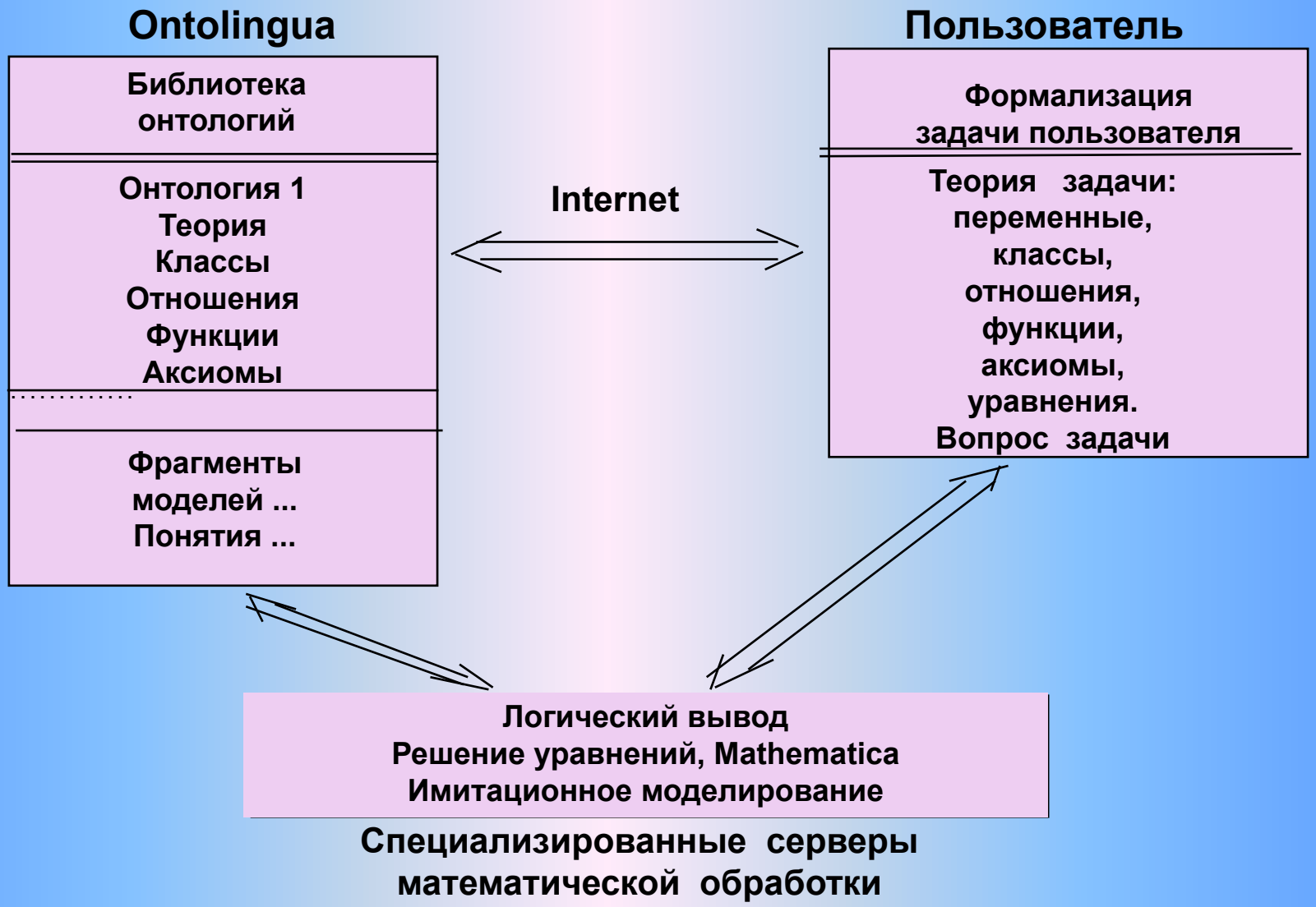
Пример онтологии.

class-def	(производственное предприятие)
class-def	(автомобиль)
class-def	(двигатель)
slot-constraint is part of has value	(автомобили)
class-def slot- constraint is part of has value	(карбюратор) (двигатель)
class-def defined	(автозавод)
subclass of	(производственное предприятие)
slot-constraint value-type	(производит автомобили)
class-def defined	(блок цилиндров)
subclass of	(автомобили)
slot-constraint value-type	(производит карбюраторный завод)
OR (slot-constraint is-part-of has value двигатель)	

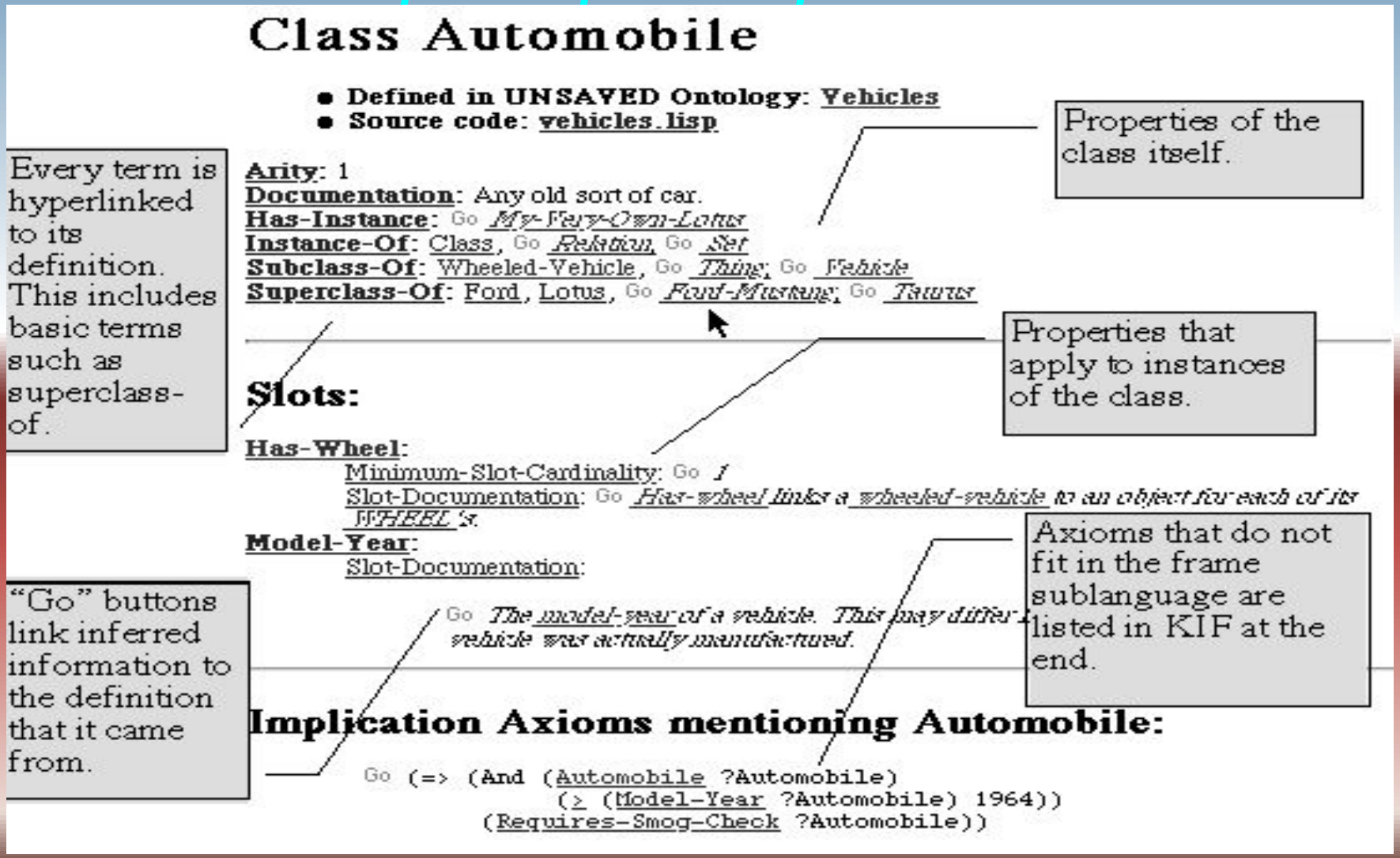
Структура сервера Ontolingua



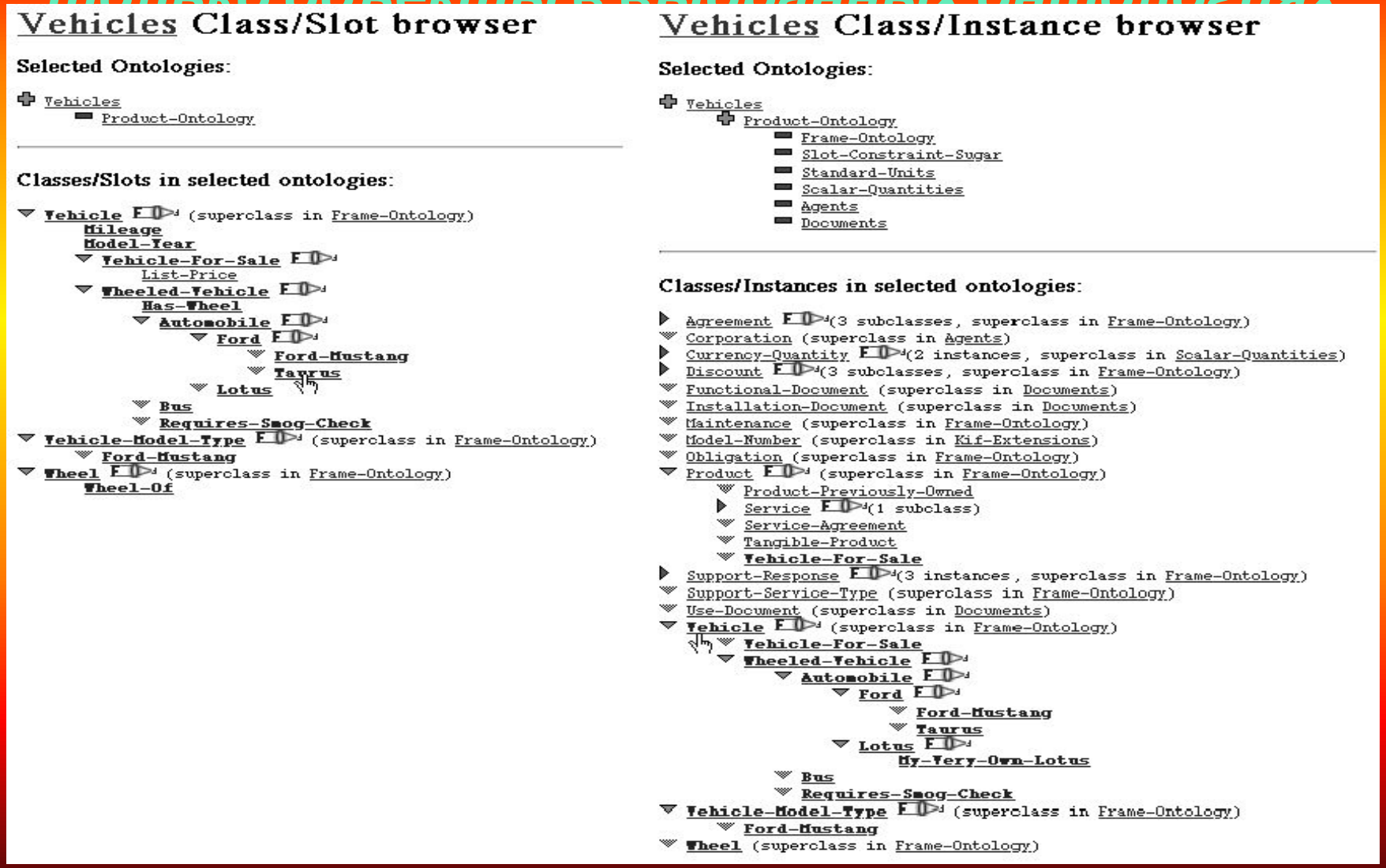
Технология работы с системой *Ontolingua.*



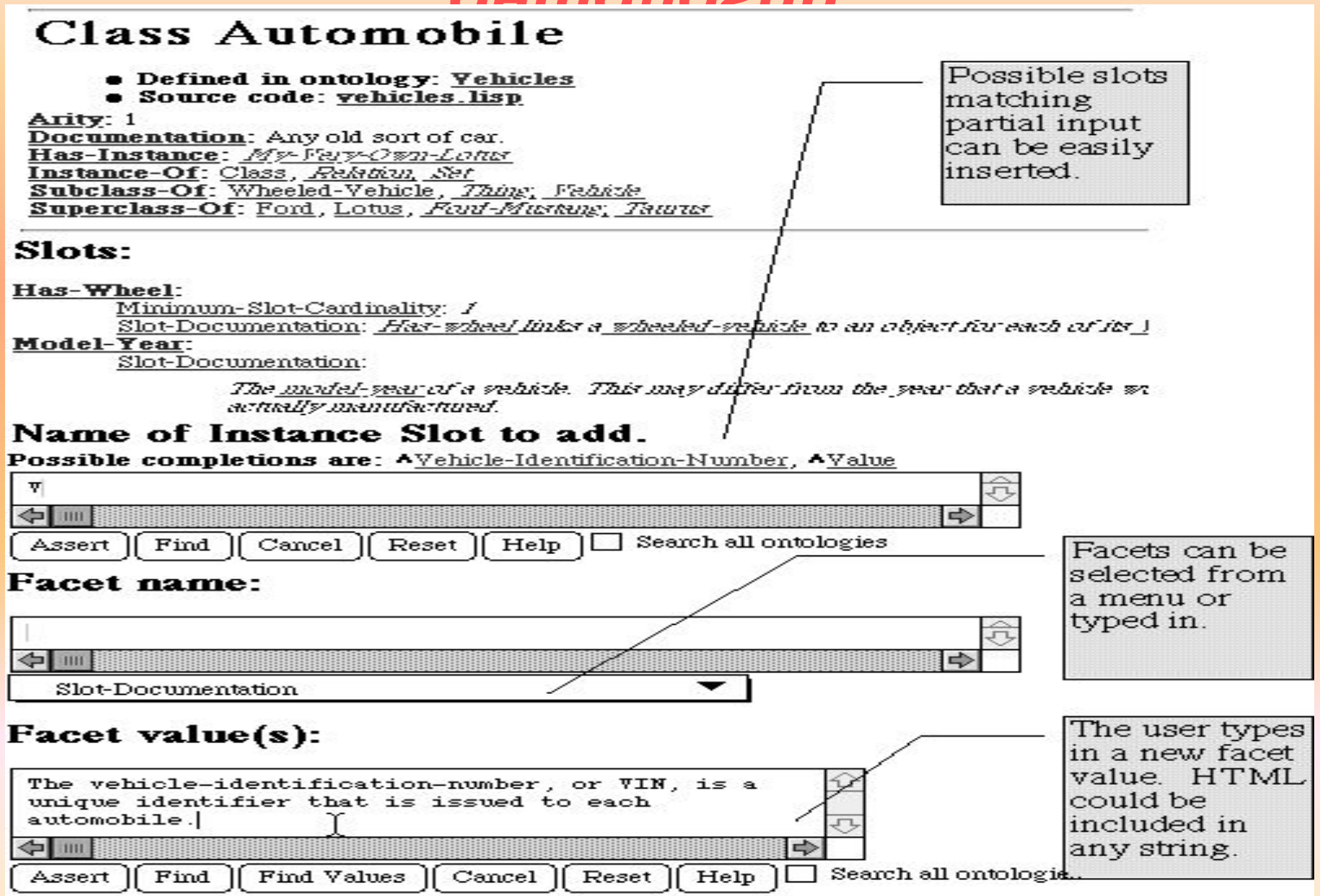
Изображение экрана просмотра информации на Сервере Онтолингва, на котором показано определение класса Автомобиль в онтологии “Транспортные средства”.



Браузер класс/слот и класс/экземпляр обеспечивает компактный просмотр множества онтологий. Отображаются только объекты в выбранных онтологиях



Интерфейс редактирования онтологий



3. Автоматизация создания онтологий: идентификация отношений

- *Сходство:* $\overline{A/B \cap B/A} = \overline{A/B} \cap \overline{B/A},$
- *Частичное совпадение:* $A \cap B$.
- *Включение:* $\overline{A} \cap B,\ A \cap \overline{B},$ *если* $\overline{A} \cap B = \varnothing,$ *то* B *- включено в* A*, если* $A \cap \overline{B} = \varnothing,$ A *- включено в* B.
- *Несходство:* $A/B \cap B/A.$

Свойства отношений

При этом могут выполняться или не выполняться какие-либо из указанных ниже отношений:

1. **Рефлексивность:** $\forall s\ (sws)$.
2. ***Симметричность:*** $\forall s\ \forall t\ (swt \supset tws)$.
3. ***Транзитивность:*** $\forall s\ \forall t\ \forall u\ (swt \wedge twu \supset swu)$.
4. ***Эвклидовость:*** $\forall s\ \forall t\ \forall u\ (swt \wedge swu \supset twu)$.
5. ***Частичная функциональность:*** $\forall s\ \forall t\ \forall u\ ((swt \wedge swu) \supset t=u)$.
6. ***Слабая связность:*** $\forall s\ \forall t\ \forall u\ (swt \wedge swu \supset twu\ vt=u\ vuwt)$.
7. ***Иррефлексивность:*** $\forall s\ \neg\ (sws)$.
8. ***Антисимметричность:*** $\forall s\ \forall t\ (swt \wedge tws \supset s=t)$.
9. ***Асимметричность:*** $(\forall s\ \forall t\ (swt \supset \neg(tws)))$.

Поиск отношений по данным c_{ij}

c_{ij} – число случаев, когда i и j связаны отношением минус число случаев, когда они не связаны этим же отношением.

Функция критерия:

$$F(Y)= \sum_i \sum_j c_{ij}' Y_{ij} \rightarrow max,$$

при ограничениях:

- Yij+Yjk-Yik≤1 ∀(I, j, k) различных – транзитивность.
- Yij-Yji=0 ∀(i≠j), симметричность.
- Yij+Yji≤1 ∀(i≠j), асимметричность.
- Yij+Yji≥1 ∀(i≠j) тотальность.

Производные отношения

Пересечение: $\forall s\ \forall t\ \ sPt \supset (sRtsQt)$, $P=RQ$.

Объединение: $\forall s\ \forall t\ \ sPt \supset (sRtsQt)$, $P=RQ$.

Произведение: $\forall s\ \forall t\ \forall v\ (sPt \supset (sRv \wedge vQt))$, $P=RQ$.

Разность: $\forall s\ \forall t\ \forall v\ sPt \supset (sRt \wedge \neg sQt)$, $P=R \wedge \neg Q$

Симметрическая разность: $\forall s \forall t\ sPt=sPt \supset (sRt \vee sQt) \wedge \neg(sRt \wedge sQt)=(sRt \wedge \neg sQt) \vee (sQt \wedge \neg sRt)$. $P=R\Delta Q=(PQ)\backslash(PQ)=(P\backslash Q)(Q\backslash P)$.

Дополнение: $\forall s\ \forall t\ \ st \supset (sEt \wedge \neg sPt)$.

Онтология «Образование»

Страница 1

<daml:Class rdf:ID="**Course**">
<rdfs:label>**Course**</rdfs:label>
<daml:sameClassAs>
<daml:Restriction>
<daml:onProperty rdf:resource="**#taughtInSemester**" />
<daml:minCardinality>**1**</daml:minCardinality>
</daml:Restriction>
</daml:sameClassAs>
<daml:sameClassAs>

Онтология «Образование»

Страница 1

```
<daml:Restriction>
 <daml:onProperty rdf:resource="#hasInstructor" />
 <daml:cardinality>1</daml:cardinality>
 </daml:Restriction>
 </daml:sameClassAs>
 </daml:Class>
 <daml:Class rdf:ID="AICourse">
 <rdfs:label>AI Course</rdfs:label>
 <rdfs:subClassOf rdf:resource="#Course" />
 </daml:Class>
```

Онтология «Африка»

Страница 1

ontology-container
title "African animals"
creator "Ian Horrocks"
    subject "animal, food, vegetarians"
    description "A didactic example ontology describing African animals"
    description.release "1.01"
    publisher "I. Horrocks"
    type "ontology"
    format "pseudo-xml"
    format "pdf"
    identifier "http://www.cs.vu.nl/~dieter/oil/TR/oil.pdf"
    source "http://www.africa.com/nature/animals.html"
    language "OIL"
    language "en-uk"
    relation.hasPart "http://www.ontosRus.com/animals/jungle.onto"

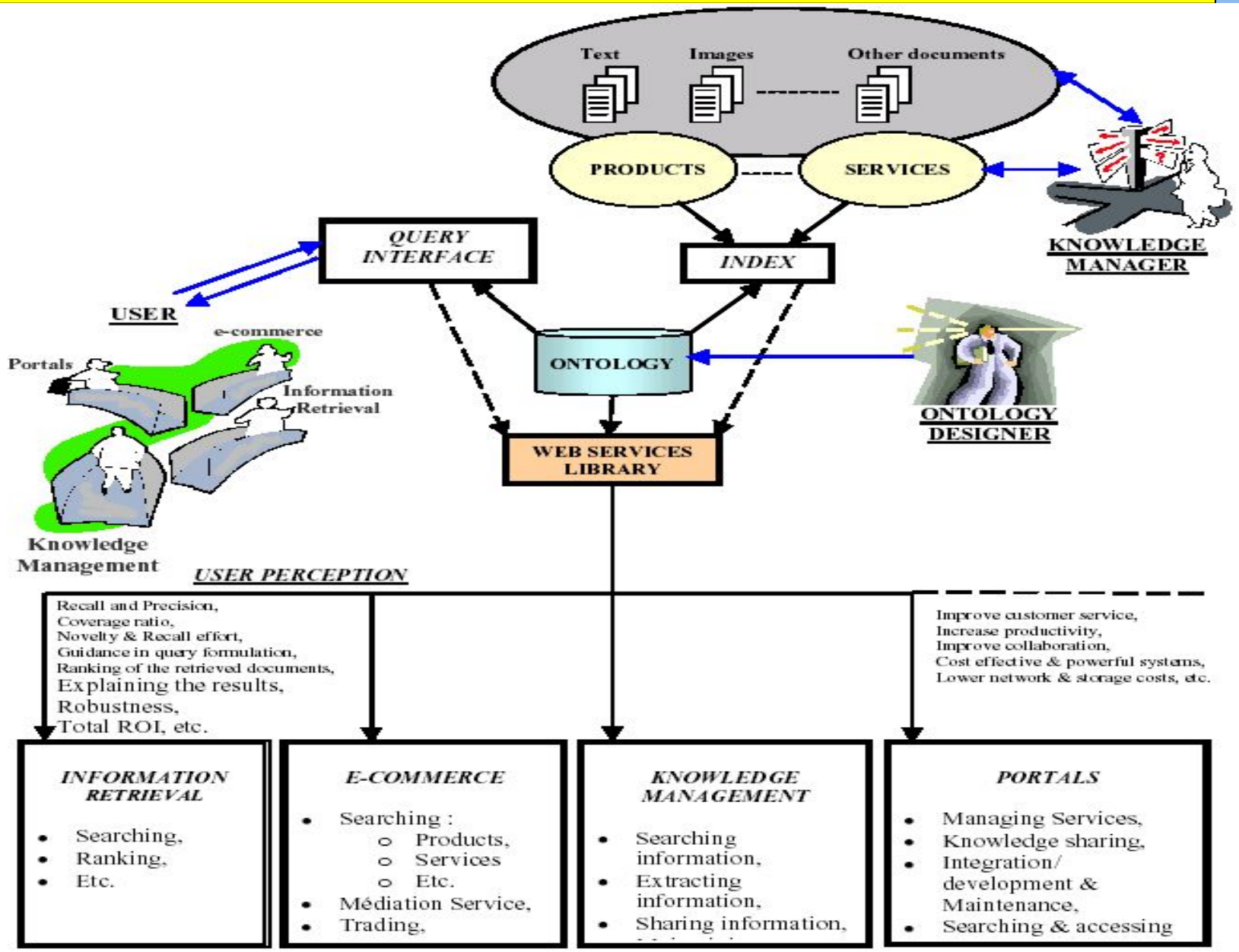
Онтология «Африка»

Страница 2

ontology-definitions
slot-def eats
inverse is-eaten-by
slot-def has-part
inverse is-part-of
    properties transitive
class-def animal
    class-def plant
subclass-of NOT animal
class-def tree
subclass-of plant
class-def branch
slot-constraint is-part-of

4. Области применения Онтологий

Применение онтологий



Онтология и e-commerce

- *Основанный в 1998, RosettaNet – это независимый, самофинансируемый, некоммерческий консорциум посвященный развитию:*
  - *XML-базирующихся стандартных электронных интерфейсов торговли,*
  - *чтобы выровнять процессы между участниками цепи поставки на глобальном основании*
- *Консорциум RosettaNet включает такие компании как IBM, Microsoft, EDS, Netscape, Oracle, SAP, Cisco systems, Compaq и Intel*

Электронный деловой Интерфейс обмена данными

Несколько примеров Предпочтительных Интернет Провайдеров (PIP)

Кластер: Информация об изделии

PIP 2A1: Распределение новой информации об изделии
PIP 2A2: Проверка информации об изделии
PIP 2A9: Проверка ЕС технической информации

Кластер: Управление заказом

PIP 3A2: Запрос о цене и доступности
PIP 3A3: Заказ на поставку
PIP 3A4: Управление заказом на поставку
PIP 3A6: Определение статуса заказа
PIP 3A7: Уведомление о принятии заказа на поставку
PIP 3B2: Уведомление об отгрузке
PIP 3B4: Проверка статуса отгрузки

Кластер: Управление инвентарем

PIP 4B1: Список инвентаря

Кластер: Маркетинг информационного управления

PIP 5C1: Определение списка изделий
PIP 5C2: Регистрация проекта запроса
PIP 5C3: Выполнение проекта запроса

RosettaNet PIPs

- Более чем 100 PIPs сгруппированы в кластеры и затем в разделы

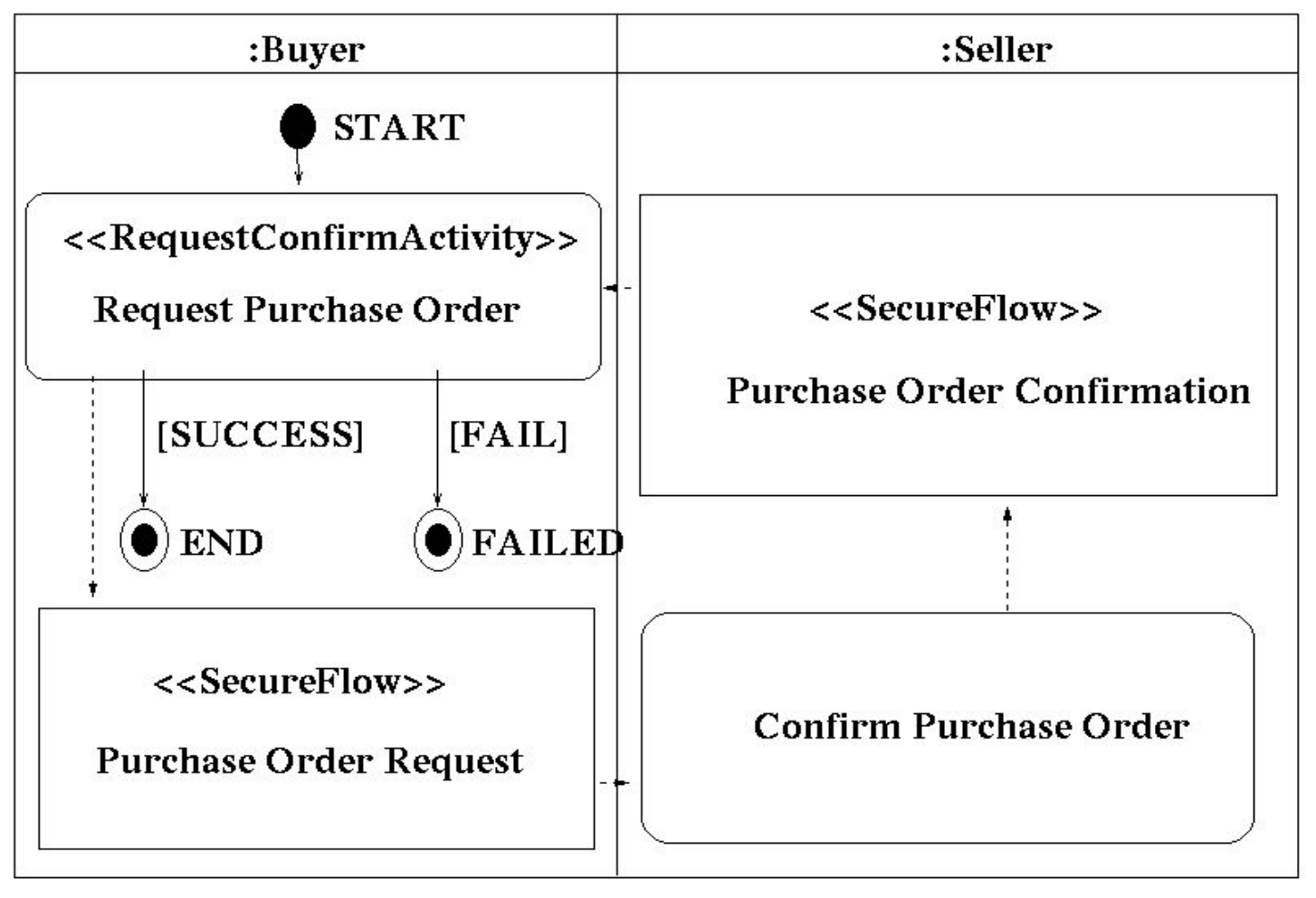
- **Например, Кластер 3 –** это **Управление запросом,** а **Раздел 3A** в этом кластере – это **Ввод квоты и заказа**

- Пример PIPs в этом разделе: **PIP3A4: Управление заказом на поставку**

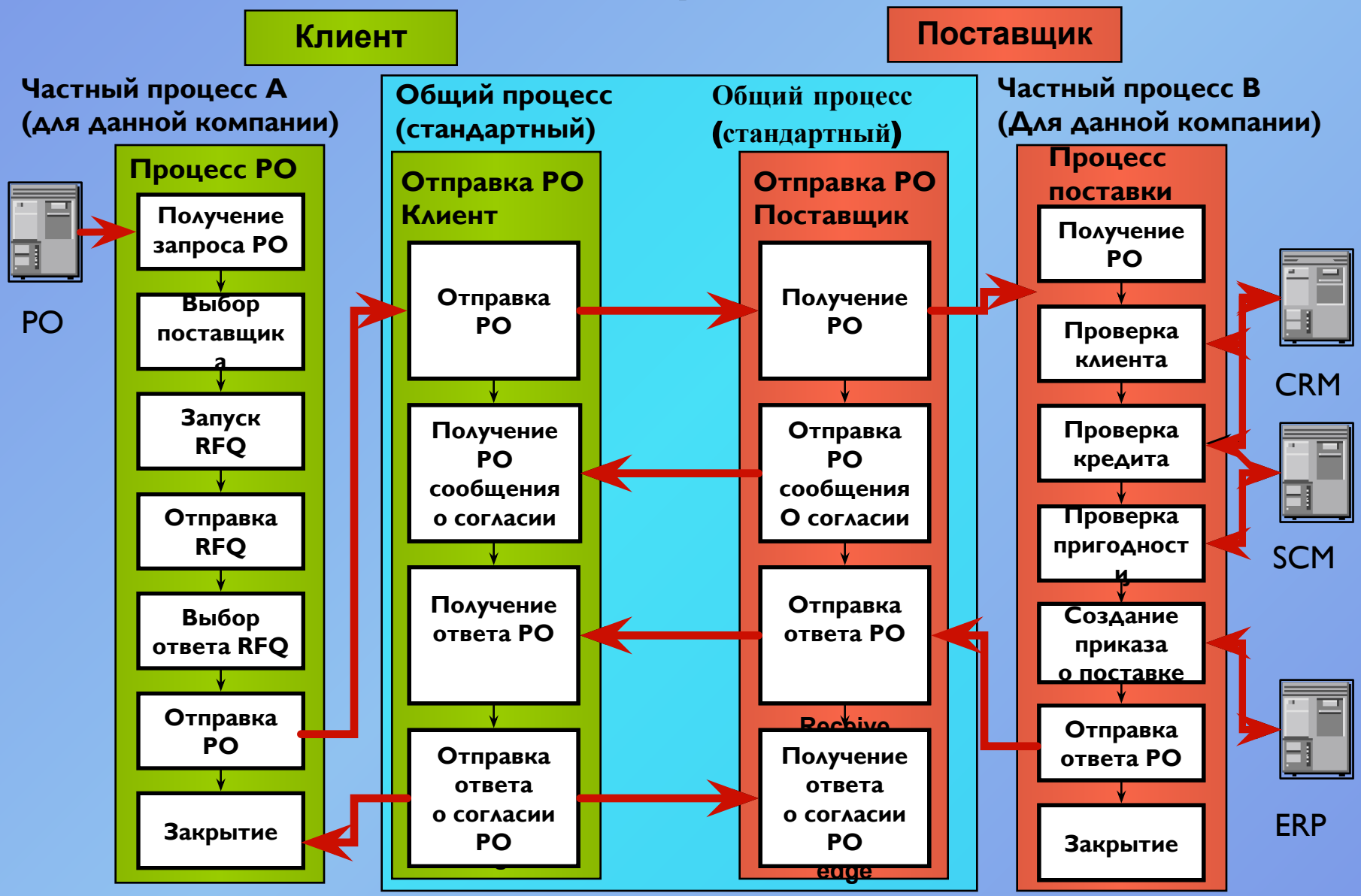
PIP 3A4: Управление заказом на поставку

- **Покупатель** создает **Заказ на поставку** и отправляет его **Продавцу**
- **Продавец** получает **Заказ на поставку** и высылает **Уведомление о принятии заказа на поставку**
- **Покупатель** принимает решение о **сотрудничестве или отказе** в зависимости от **содержания сообщения**

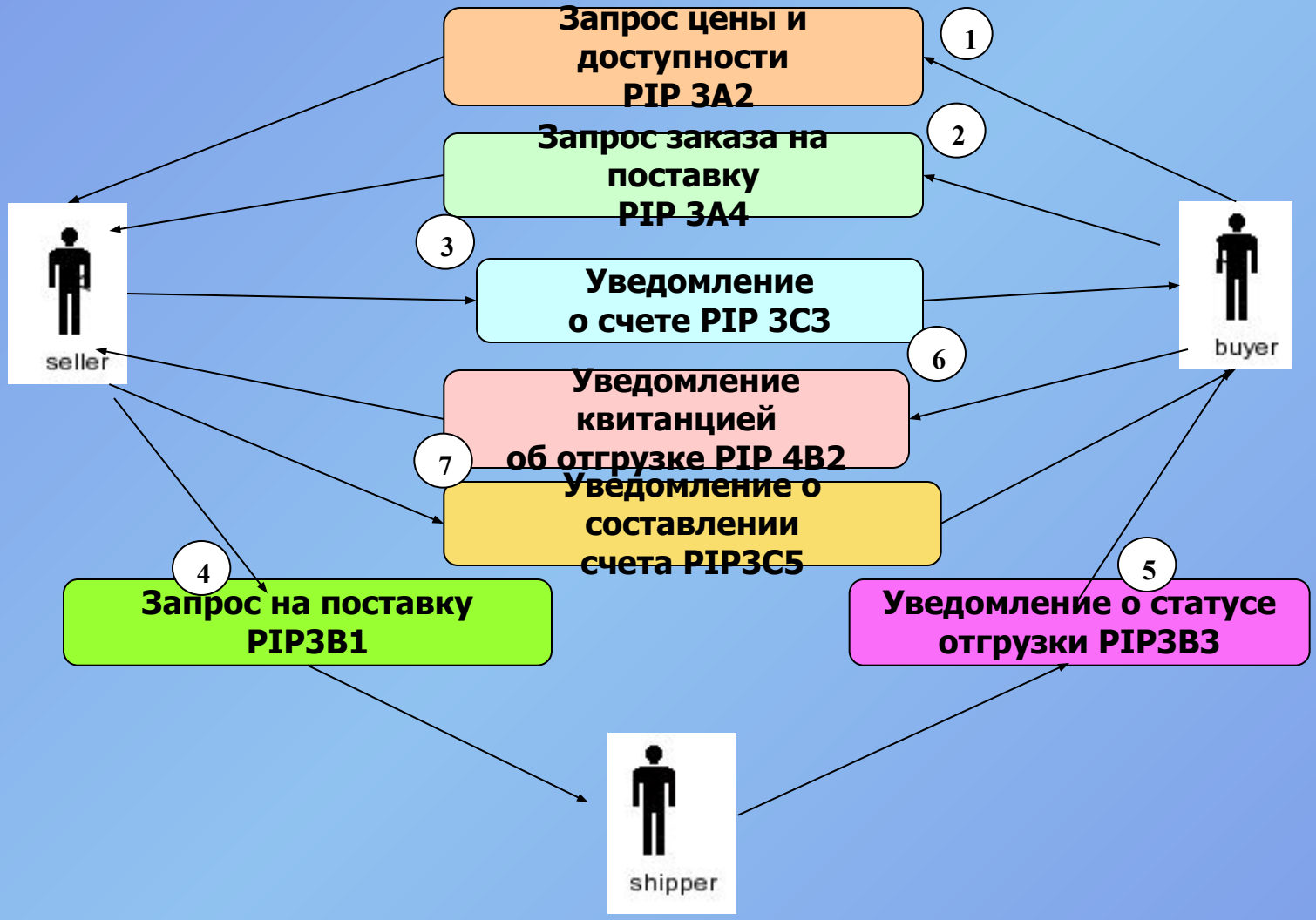
RosettaNet: Блок-схема бизнес-процесса для PIP3A4



Бизнес-процессы



Пример стандартного бизнес-процесса



Пример

- Рассмотрим, например, сценарий, где покупатель запрашивает цену и пригодность некоторых изделий от продавца (PIP3A2)
- После получения ответа покупатель отправляет Запрос Заказа на поставку (PIP3A4)
- Продавец, с другой стороны, после подтверждения Запроса Заказа на поставку, посылает уведомление (PIP3C3) в виде счета покупателю
- Продавец посылает запрос транспортировки (PIP3B1) грузоотправителю (есть третье лицо в этом

Пример

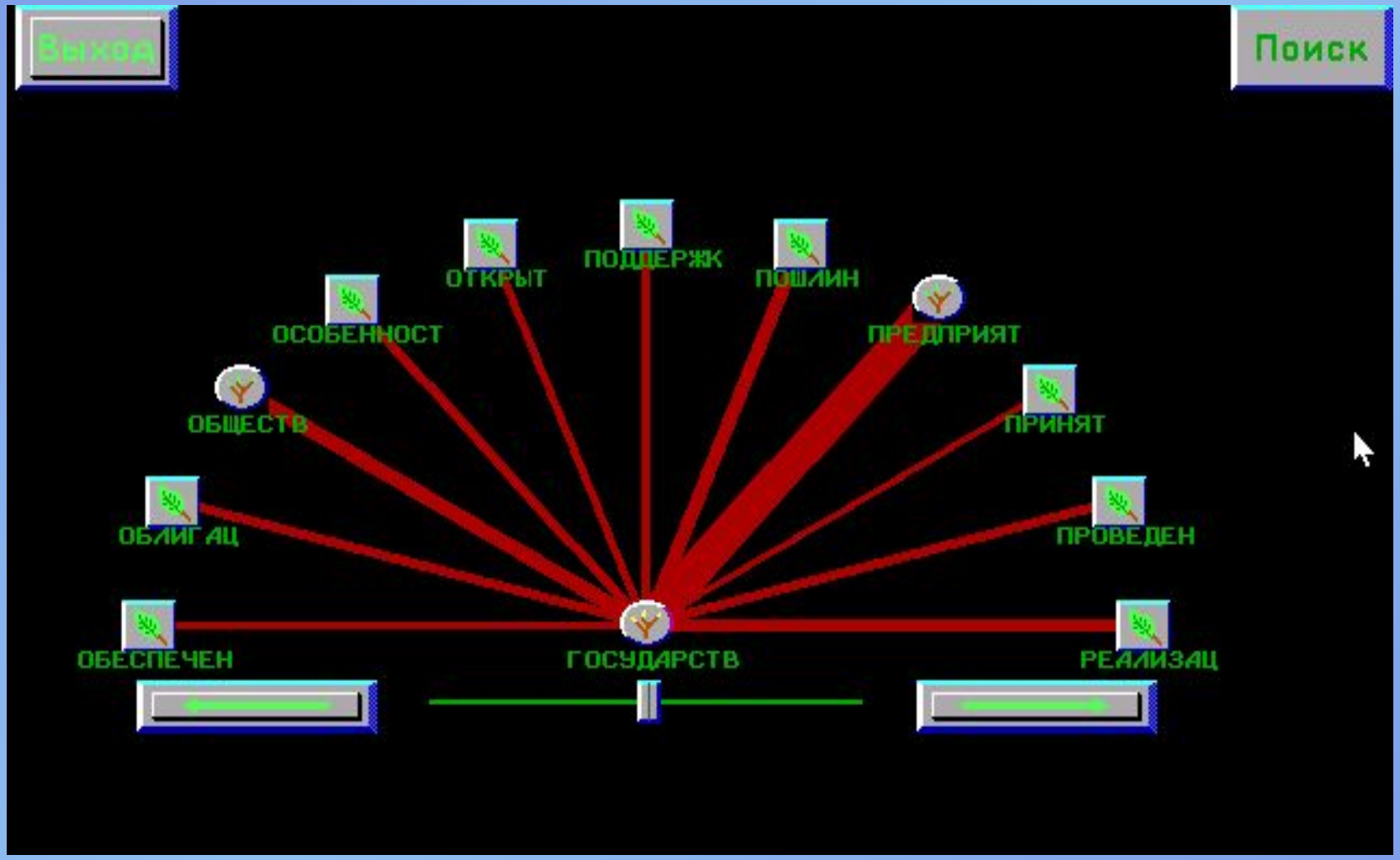
- Грузоотправитель после отгрузки товаров посылает уведомление об отгрузке (PIP3B3)
- Когда покупатель получает уведомление, то посылает (PIP4B2) квитанцию об отгрузке продавцу.
- Наконец, продавец готовит на утверждение составленные счета и уведомляет покупателя (PIP3C5)

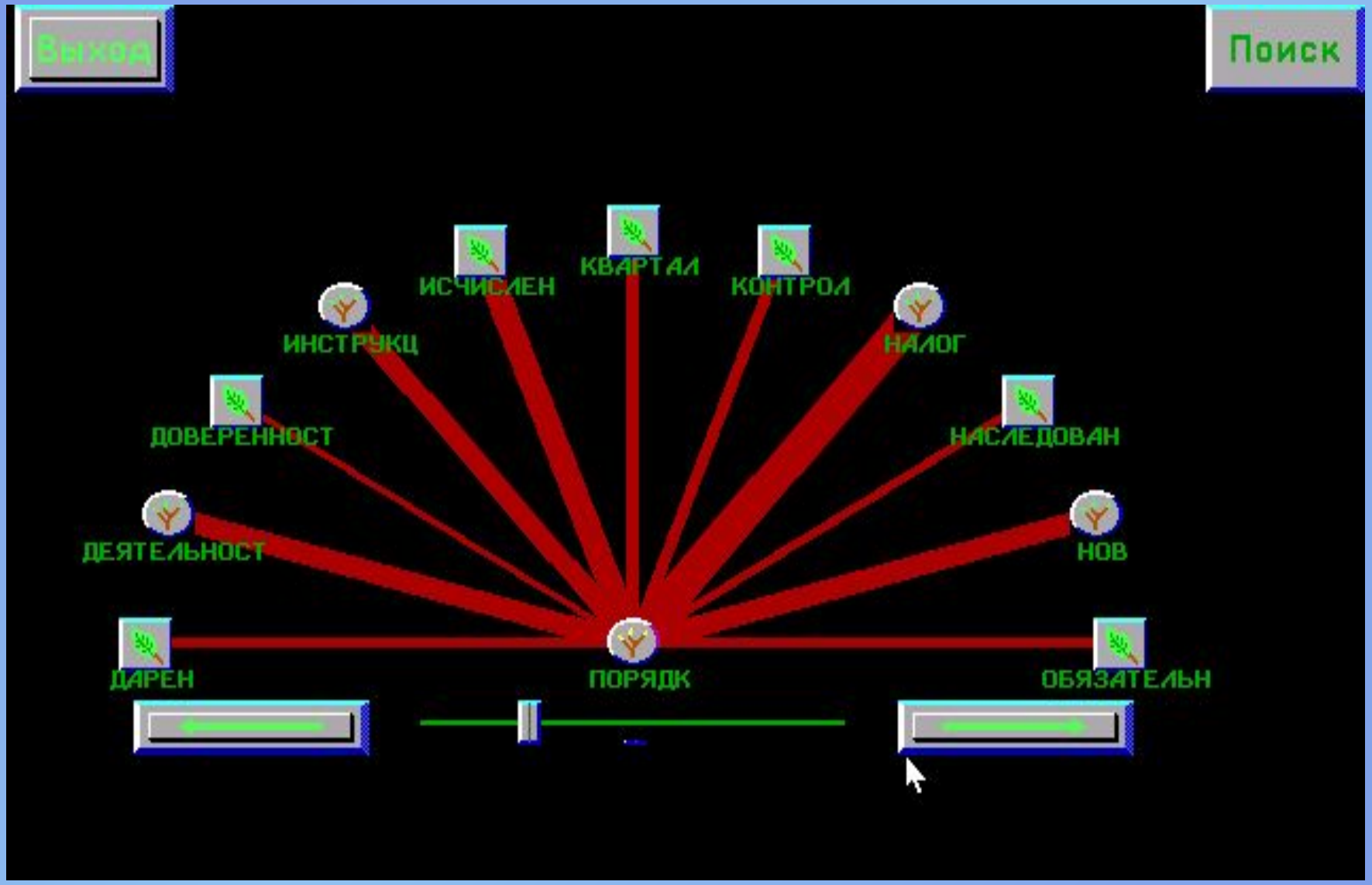
Классификация и распределение продуктов в RosettaNet

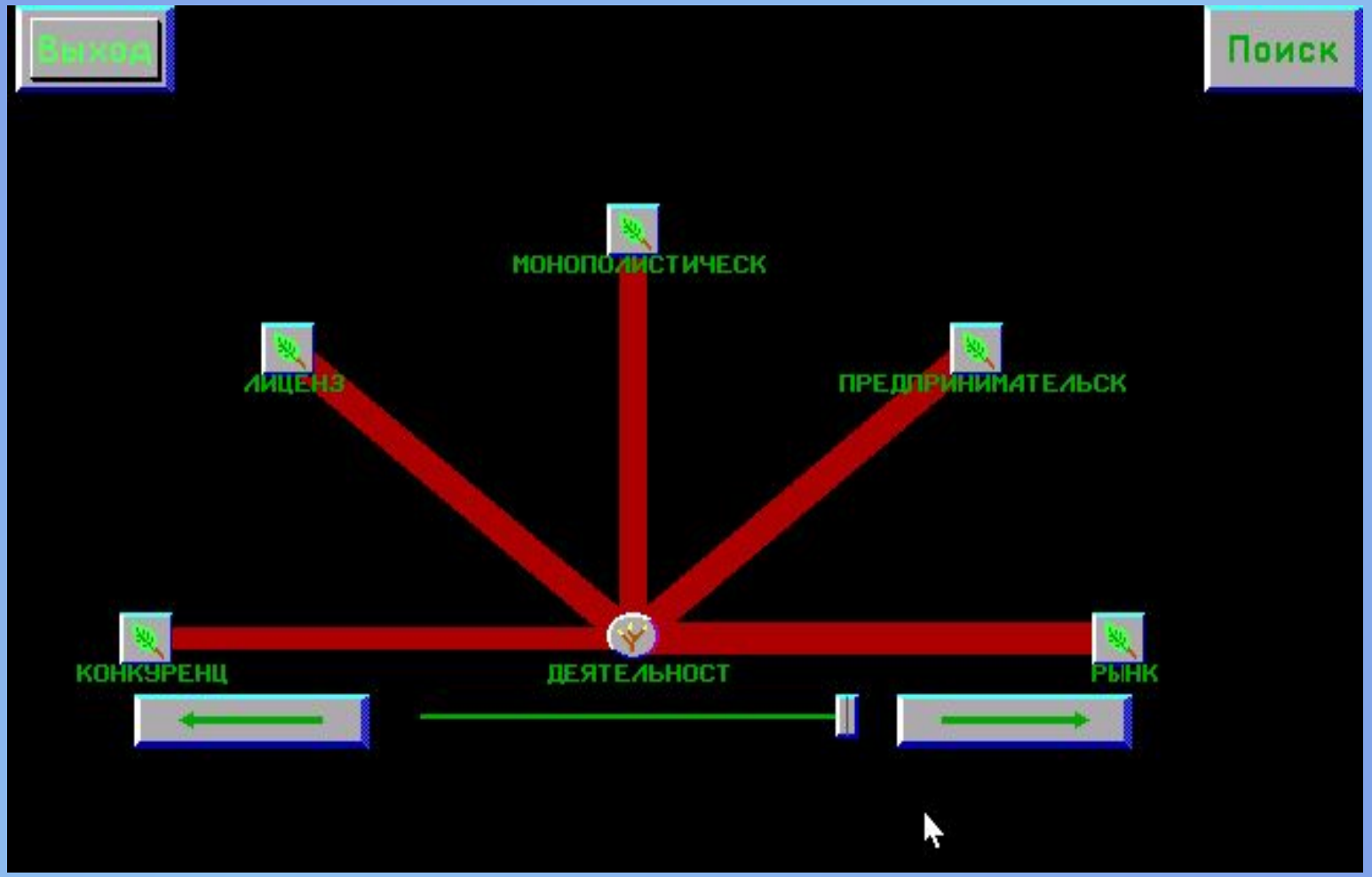
- Классификация изделий в RosettaNet производится с помощью Технического Словаря RosettaNet (RTD)
- RTD определяет классы изделий и их свойства в XML DDT
- Таким образом, для каждого типа изделия существует набор определенных признаков XML
- Каждый класс изделия также имеет соответствие в Универсальных Стандартных Изделиях и Классификацию Услуг (UNSPSC) - кодекс, в основном используемый, чтобы дифференцировать в

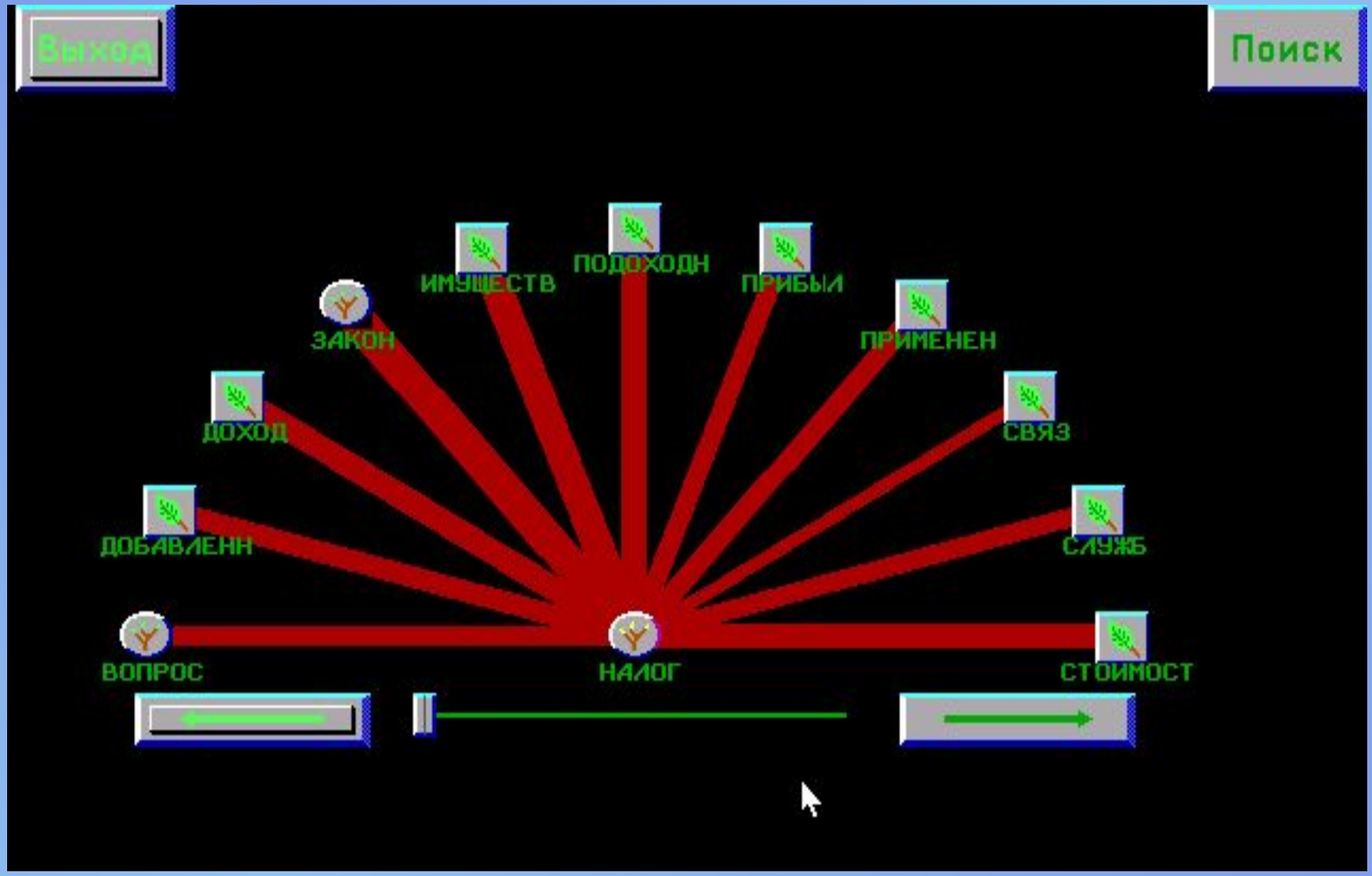
Глобальная нумерация (GTIN)

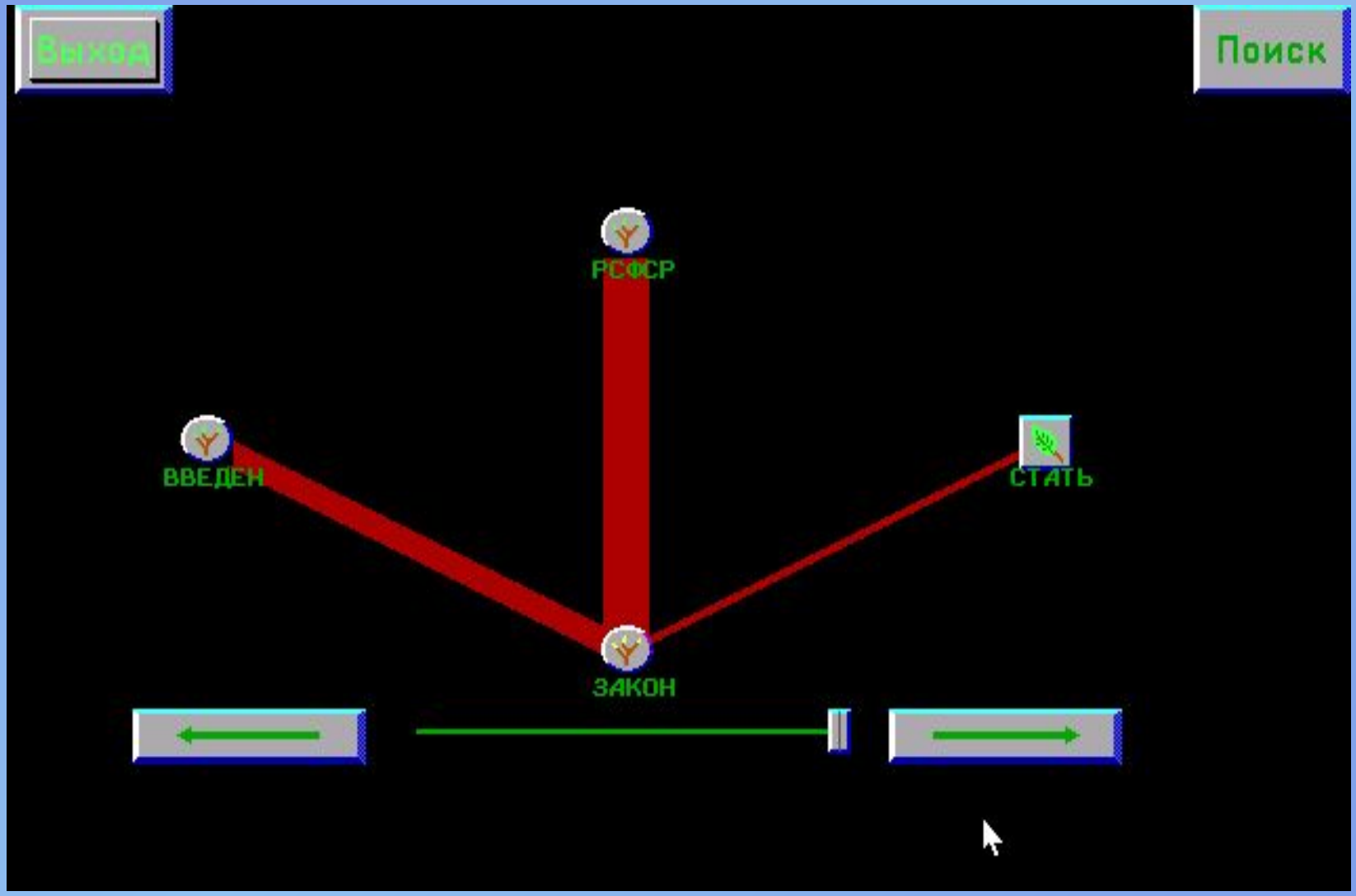
- *В RosettaNet детали информации об изделии могут быть получены с помощью просмотра каталога партнера по цепи поставки при использовании стандартных средств через "PIP2A5/EC Проверка Технической Информации. чтобы известить сразу несколько партнеров о продукции через GTIN*
- *Следовательно RTD используется в соединении данных изделия с GTINs*
- *Чтобы внедрить Технический словарь, организация должна категоризировать все продаваемые продукты на классы*

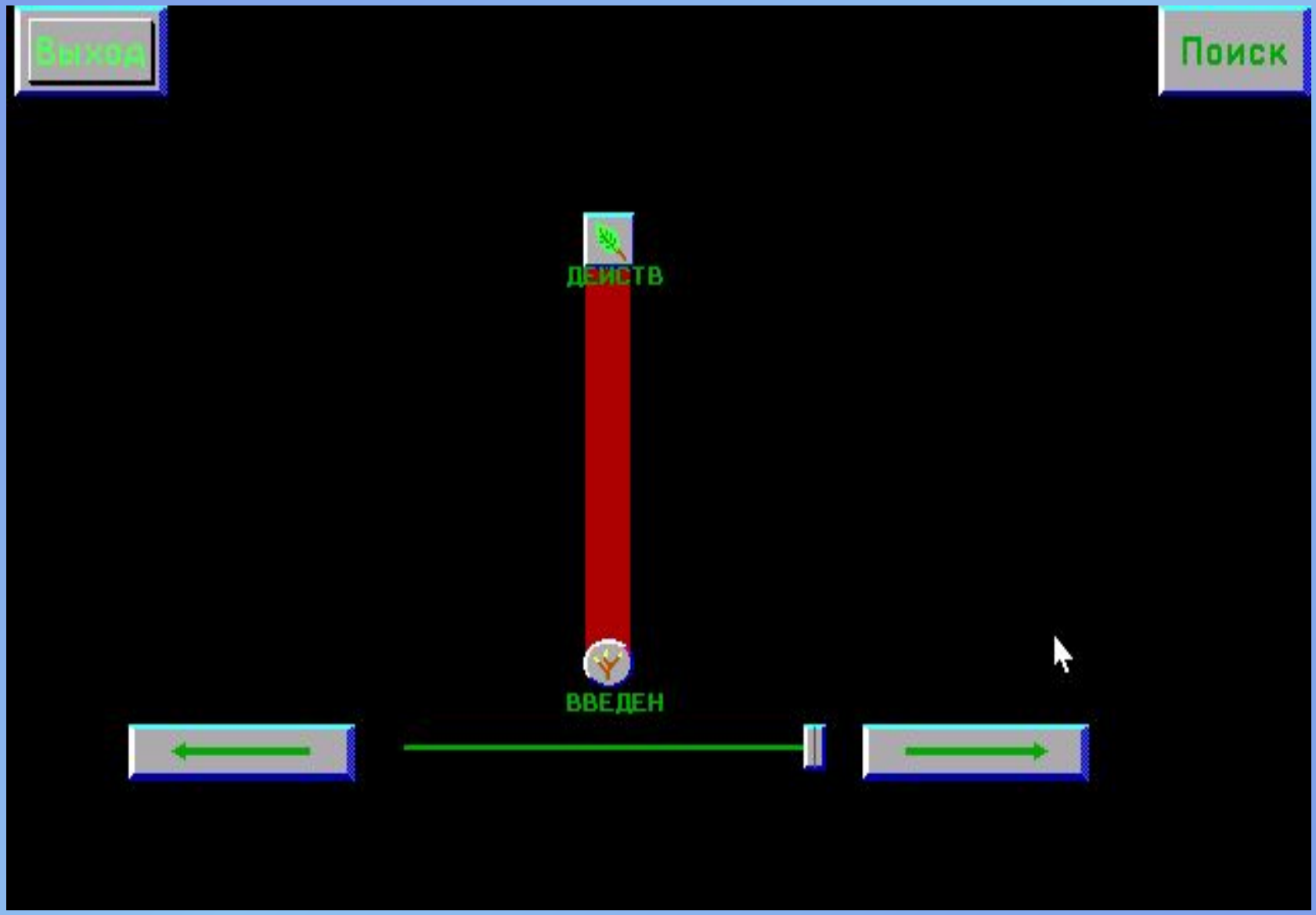

Выход
Поиск
ОТКРЫТ
ПОДДЕРЖК
ПОШЛИН
ОСОБЕННОСТ
ПРЕДПРИЯТ
ОБЩЕСТВ
ПРИНЯТ
ОБЛИГАЦ
ПРОВЕДЕН
ОБЕСПЕЧЕН
ГОСУДАРСТВ
РЕАЛИЗАЦ

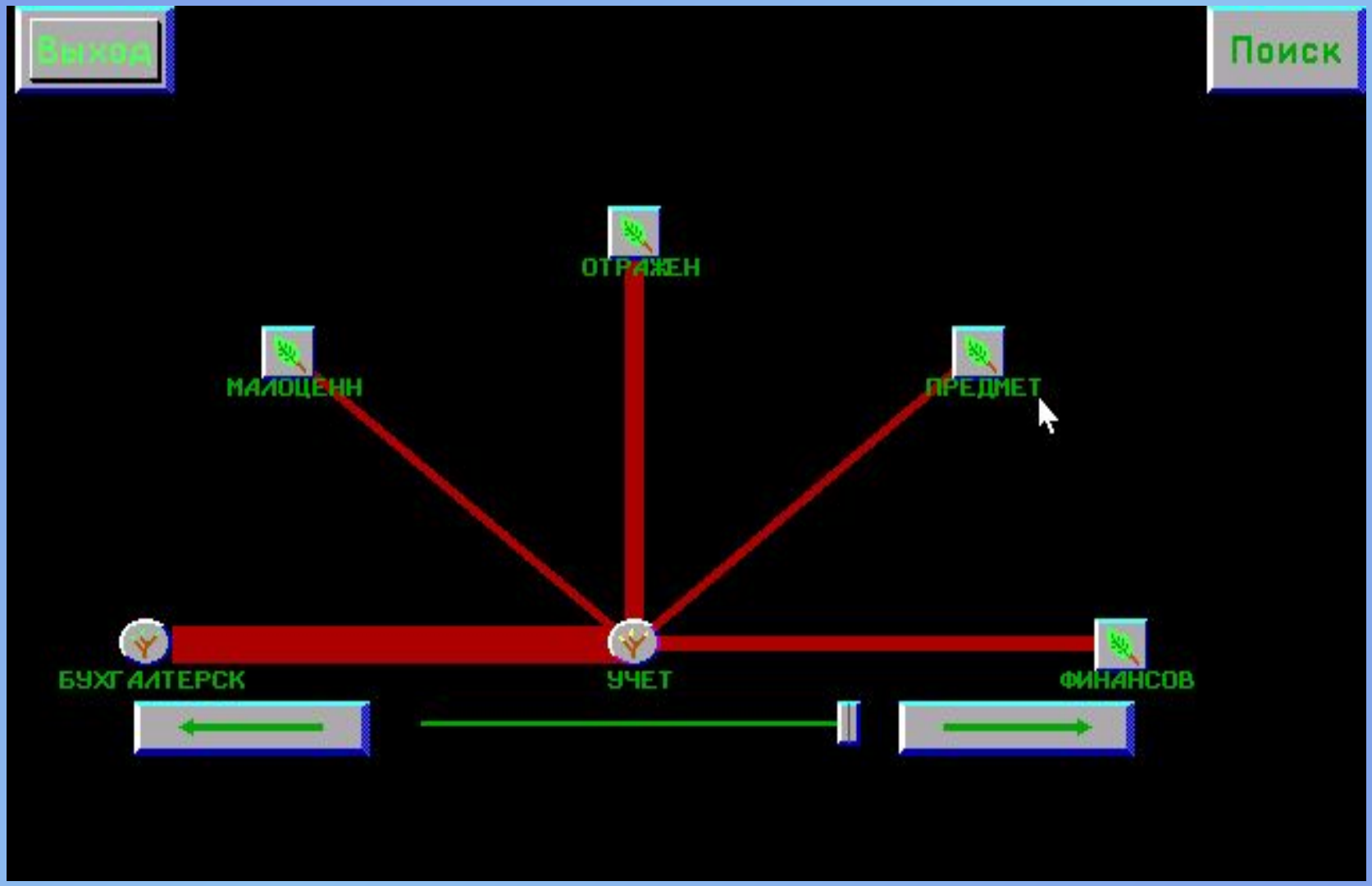

Выход
Поиск
ИСЧИСЛЕН
КВАРТАЛ
КОНТРОЛ
ИНСТРУКЦ
НАЛОГ
ДОВЕРЕННОСТ
НАСЛЕДОВАН
ДЕЯТЕЛЬНОСТ
НОВ
ДАРЕН
ПОРЯДК
ОБЯЗАТЕЛЬН

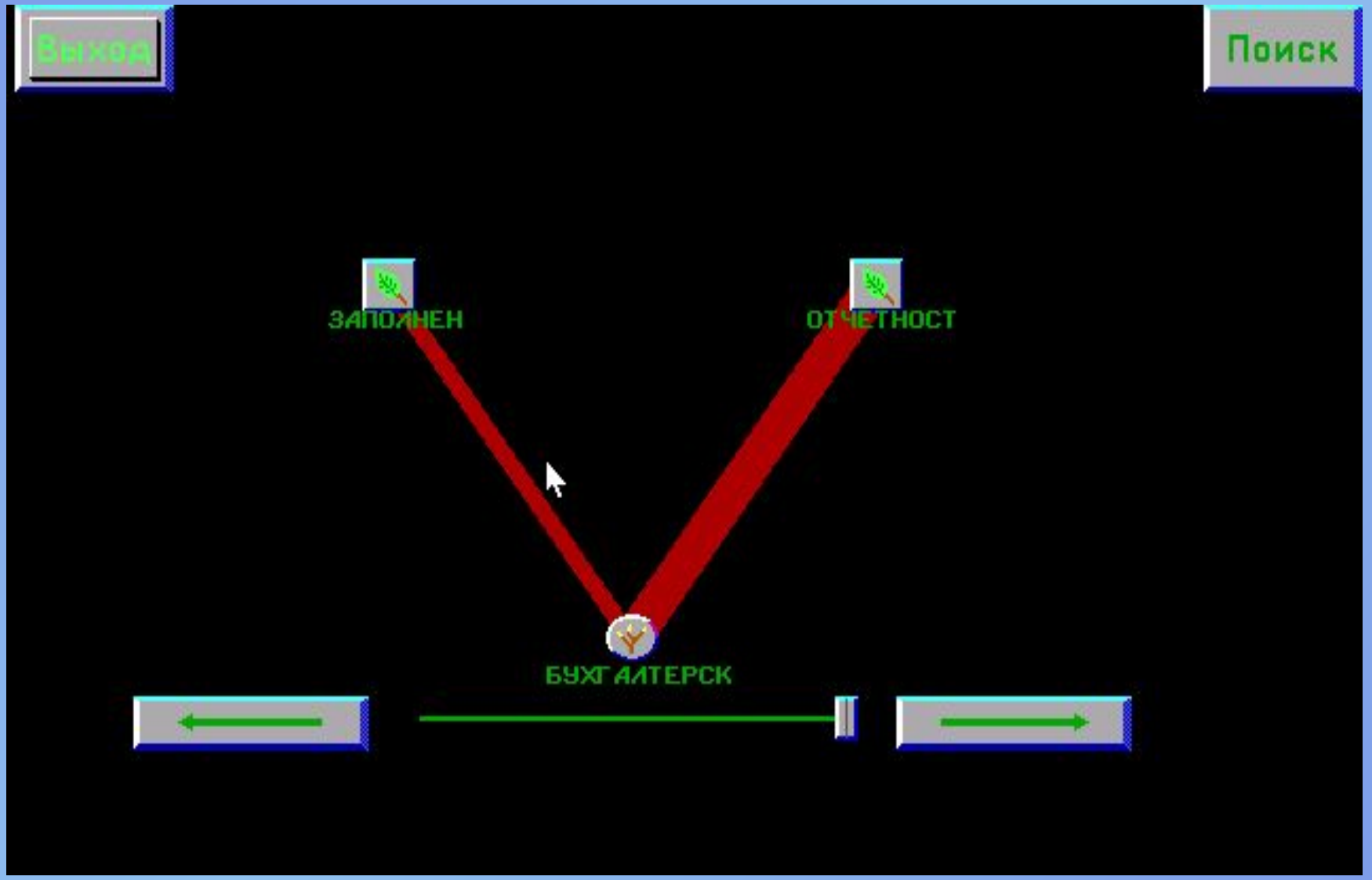

Выход
Поиск
МОНОПОЛИСТИЧЕСК
ЛИЦЕНЗ
ПРЕДПРИНИМАТЕЛЬСК
КОНКУРЕНЦ
ДЕЯТЕЛЬНОСТ
РЫНК


Выход
Поиск
ЗАКОН
ИМУЩЕСТВ
ПОДОХОДН
ПРИБЫЛ
ПРИМЕНЕН
ДОХОД
СВЯЗ
ДОБАВЛЕНН
СЛУЖБ
ВОПРОС
НАЛОГ
СТОИМОСТ


Выход
Поиск
РСФСР
ВВЕДЕН
СТАТЬ
ЗАКОН


Выход
Поиск
ДЕЙСТВ
ВВЕДЕН


Выход
Поиск
ОТРАЖЕН
МАЛОЦЕННН
ПРЕДМЕТ
БУХГАЛТЕРСК
УЧЕТ
ФИНАНСОВ


Выход
Поиск
ЗАПОЛНЕН
ОТЧЕТНОСТ
БУХГАЛТЕРСК