

Semantic web: interneti tulevik?

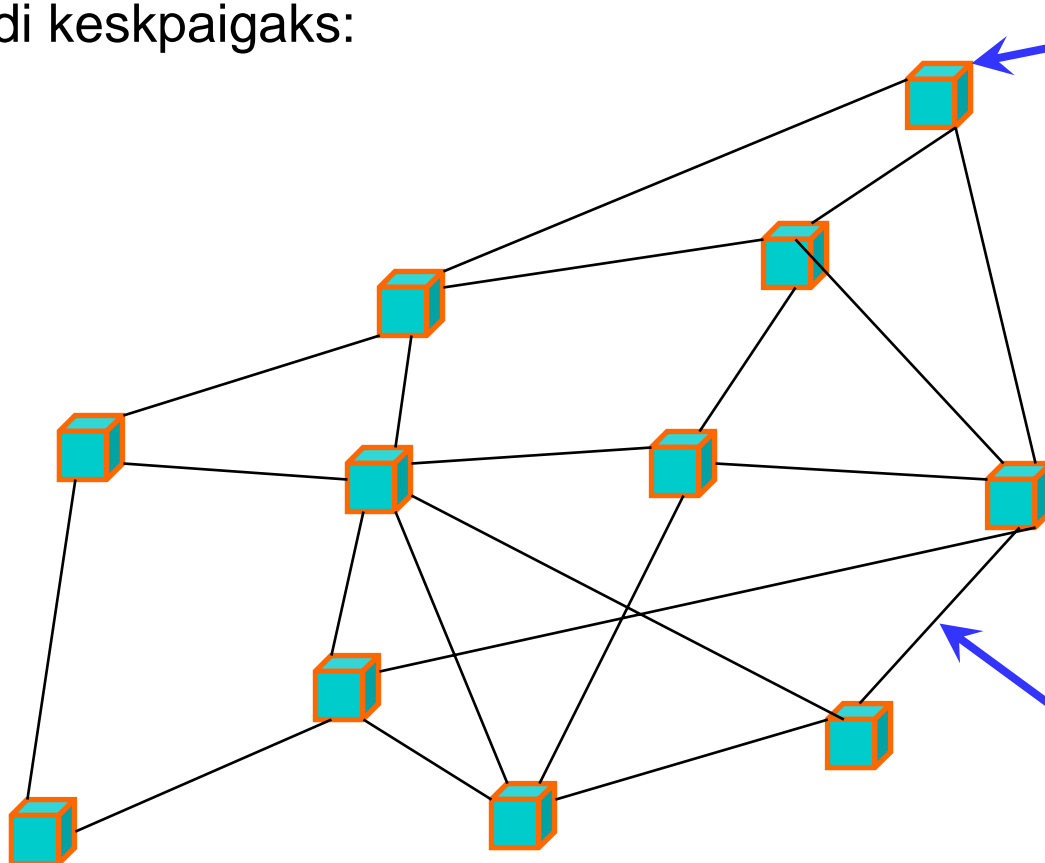
Tanel Tammet

Sisukord

- Internet: minevik, olevik, tulevik
- HTML ja XML
- Võrguteenused ja .NET
- Mis puudub?
- Vajaduste näited
- Semantic web: kihid
- Namespaces
- RDF
- Ontoloogiad
- Loogika
- Lingid

Intelligentne võrk!?

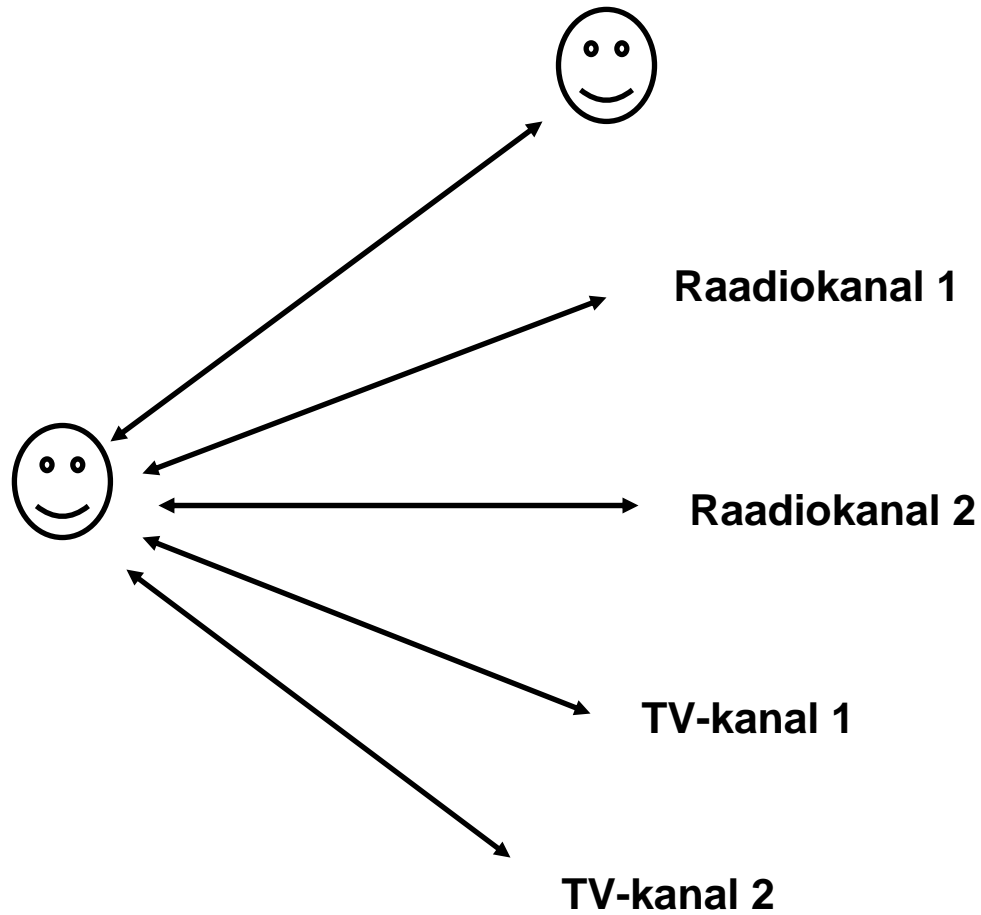
Sajandi keskpaigaks:



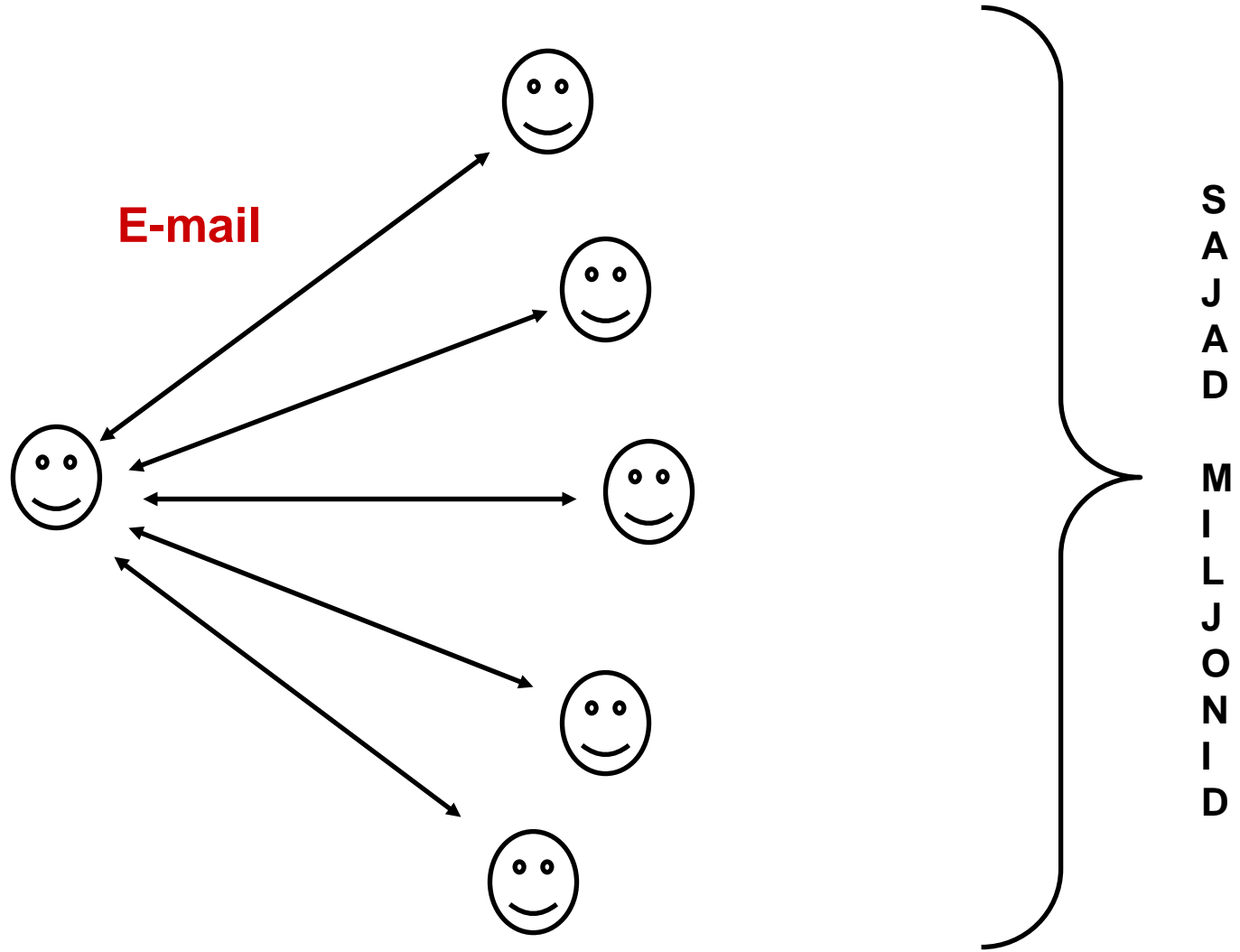
Iga arvuti
on praegustest
tuhandeid
kordi
võimsam

Võrgus on
praegusest
tuhandeid
kordi rohkem
infot ja
teenuseid

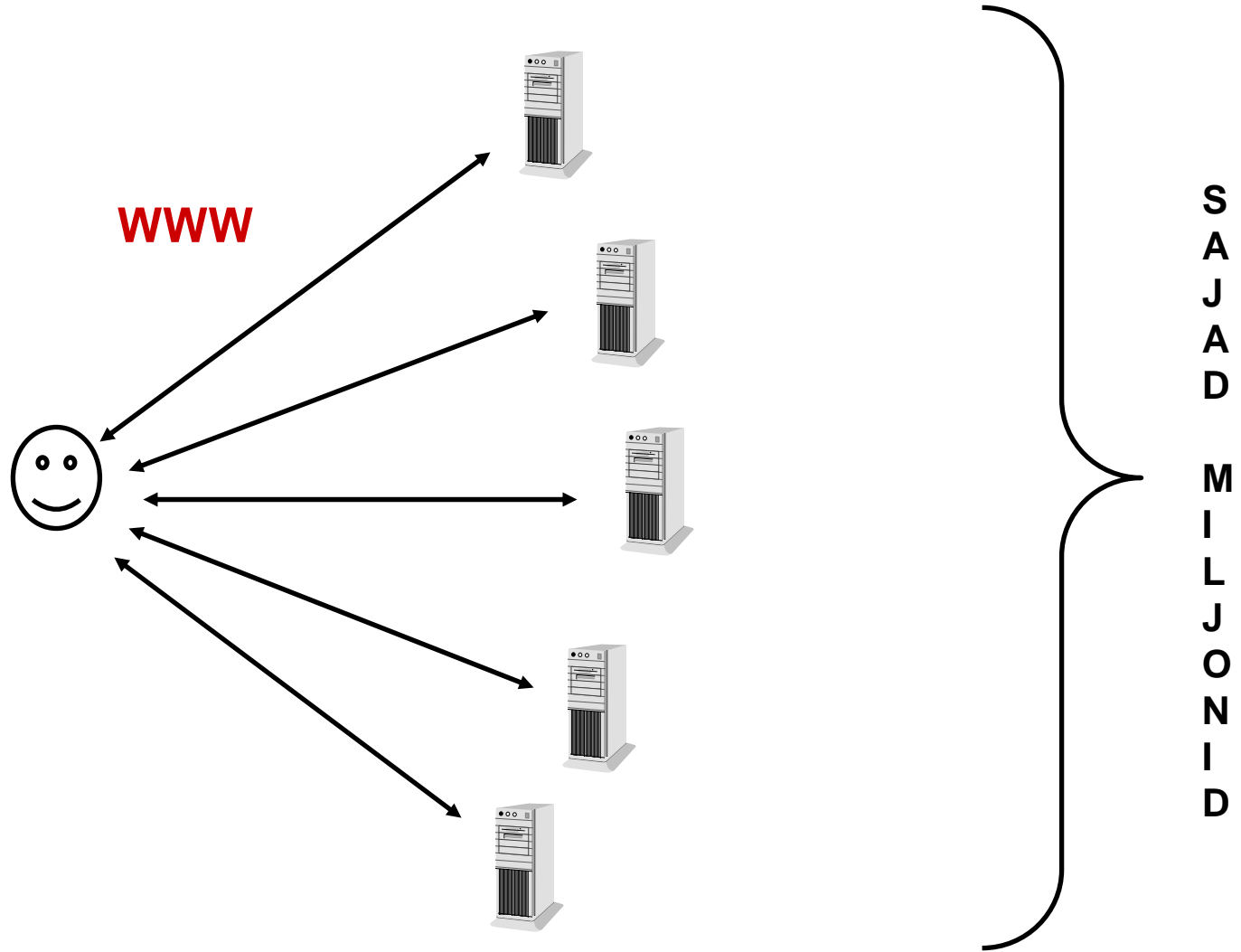
Internet 0: inimene <--> veidi inimesi, raadio, TV



Internet 1: inimene <--> hulga inimesi

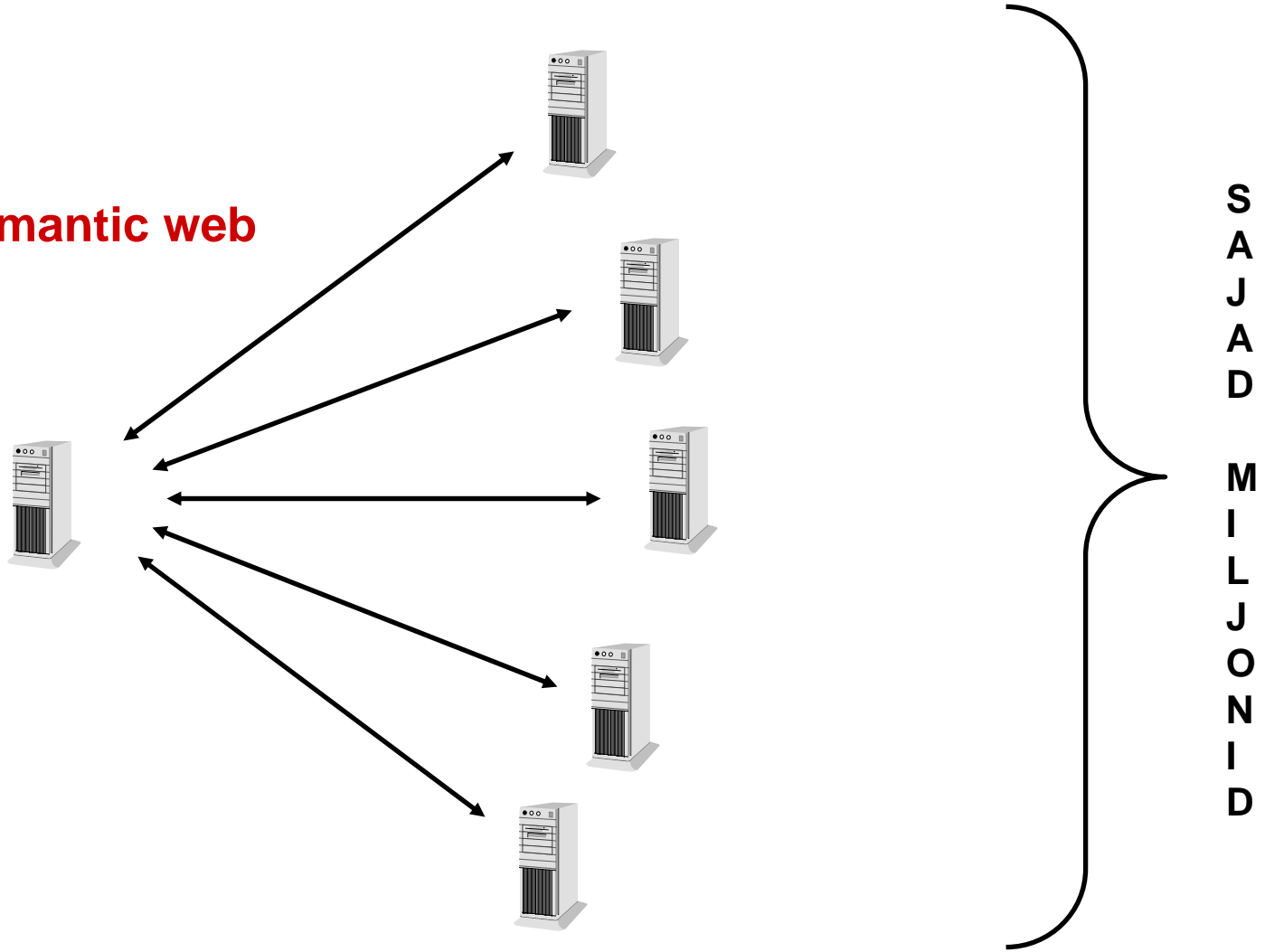


Internet 2: inimene <--> hulga masinaid

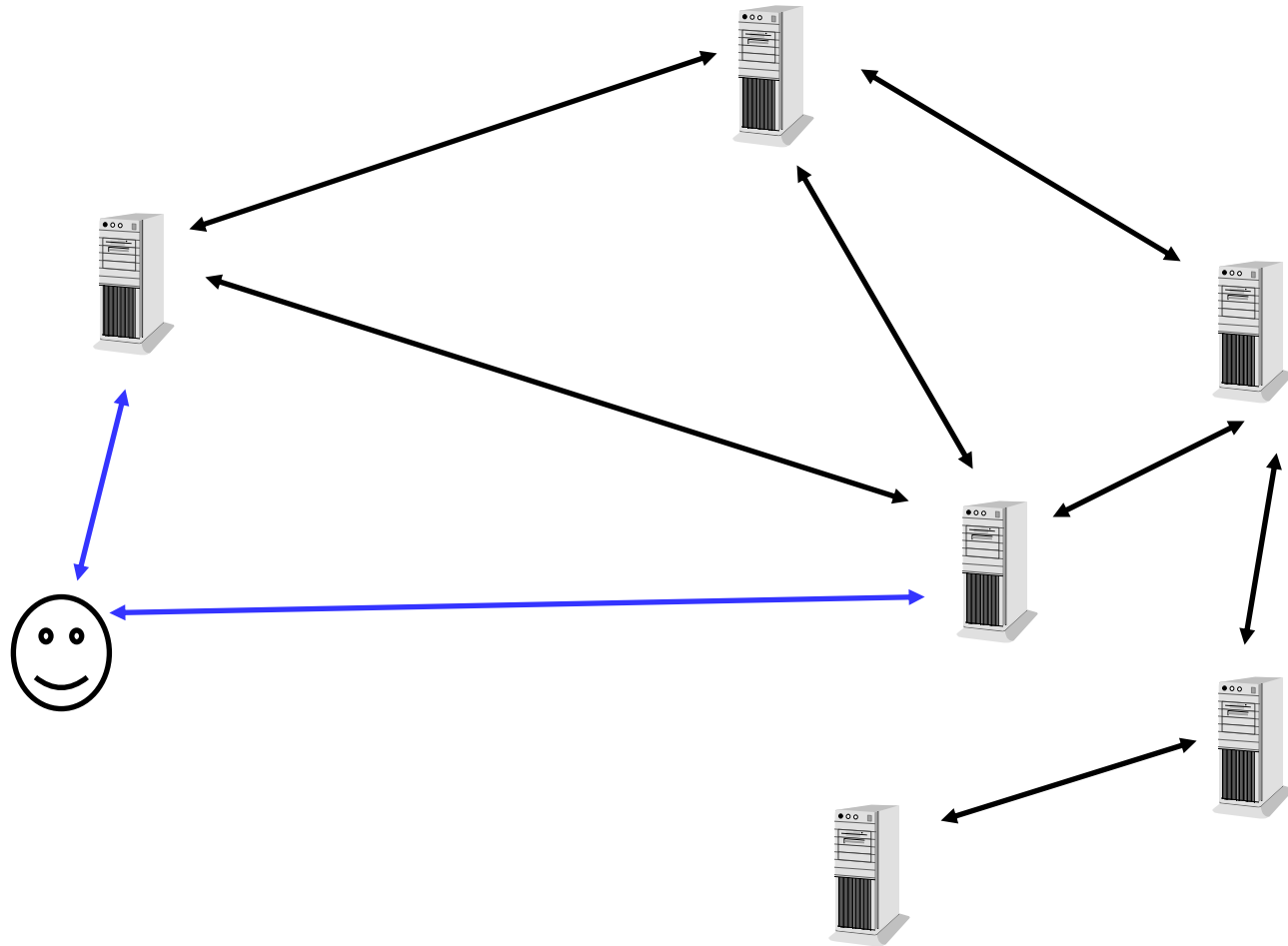


Internet 3: masinad <--> hulga masinaid

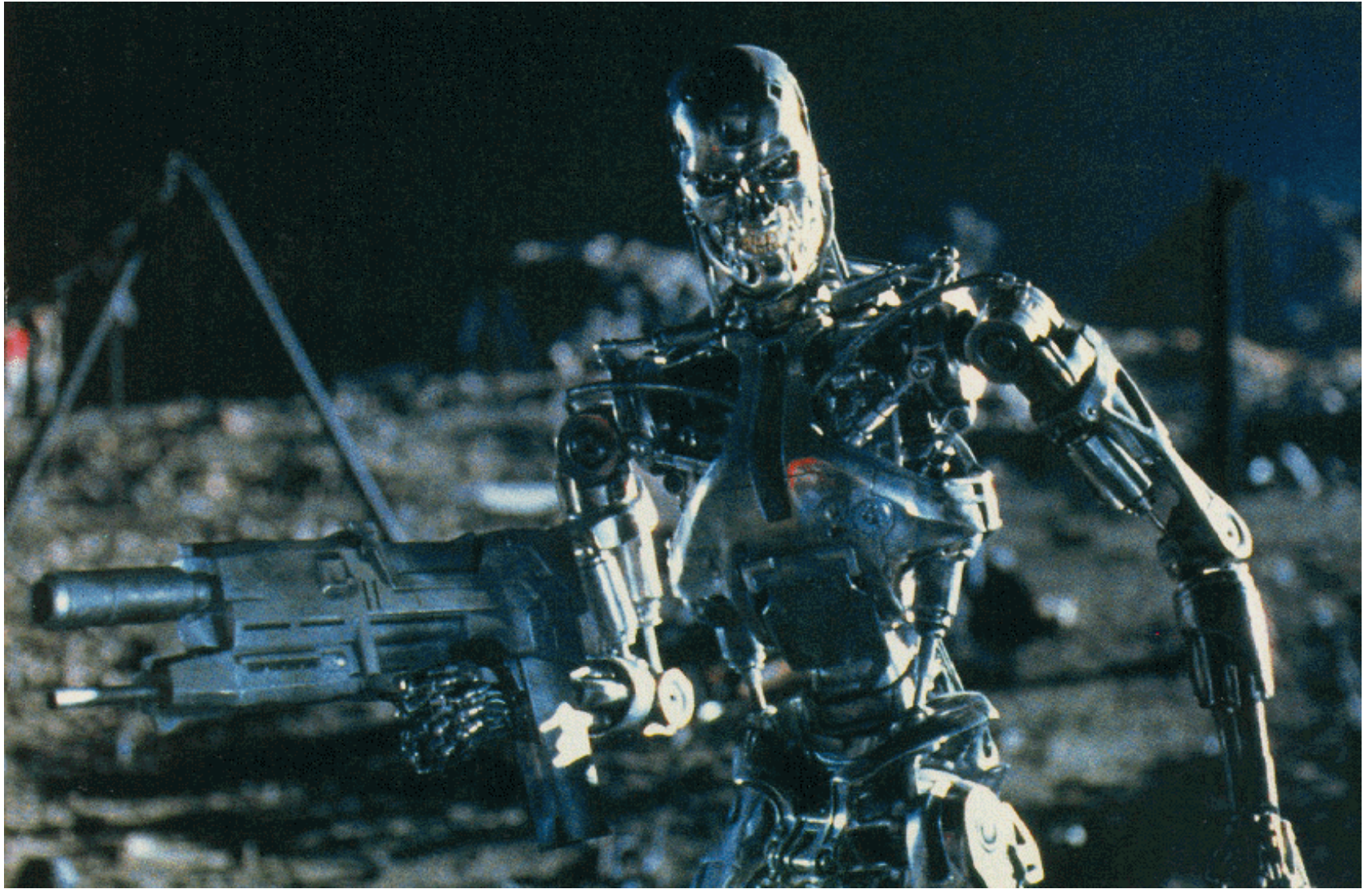
XML + Semantic web



Internet 3: inimene ??



Internet 4: Skynet?



HTML ja XML

- Info pannakse “tag”-ide vahele: infol on sildid
- HTML: “tag”-del visuaalne semantika

Siin on paks tekst

<i>Siin on kaldkirjas tekst</i>

- XHTML: “tag”-del semantika puudub

<autor>Tanel Tammet</autor>

<minuaadress>Kuiv 9</minuaadress>

HTML ja XML

- Info pannakse “tag”-ide vahele: infol on sildid
- HTML: “tag”-del visuaalne semantika

****Siin on paks tekst****

<i>Siin on kaldkirjas tekst**</i>**

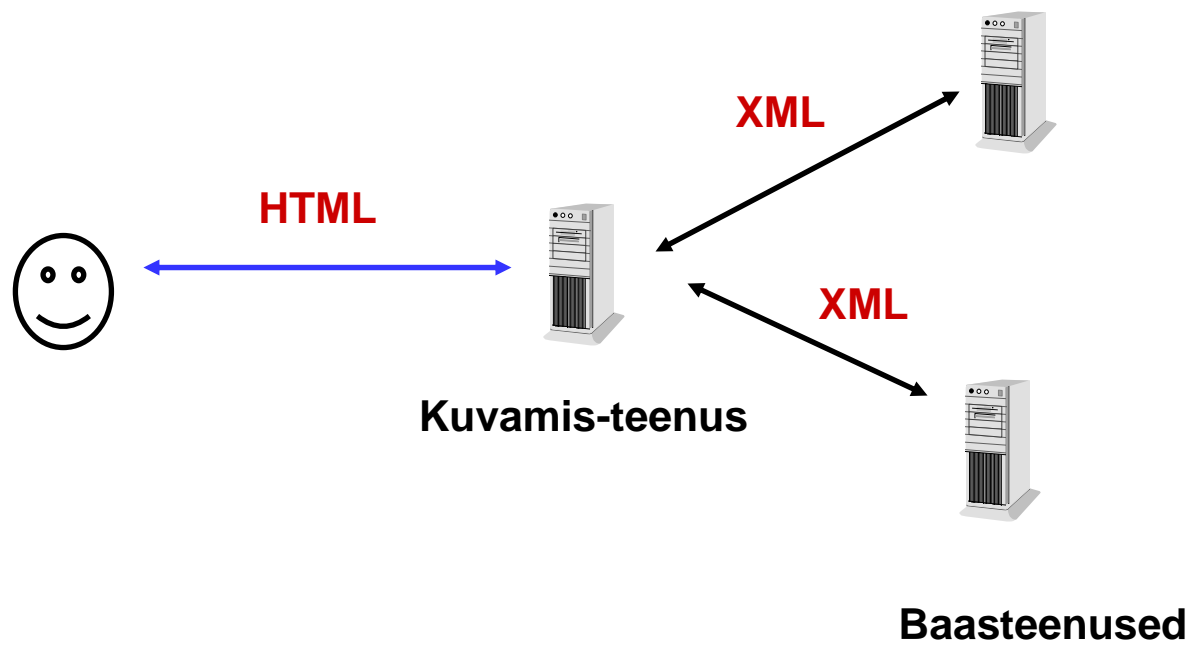
- XHTML: “tag”-del semantika puudub

<autor>Tanel Tammet**</autor>**

<minuaadress>Kuiv 9**</minuaadress>**

Võrguteenused

- Probleem: väga raske, pea võimatu on teha programmi, mis loeks teisest serverist HTML-lehekülgi.
- Idee: teeme võrgulehekülgi XML-is, nii et programm teises arvutis suudaks neid lugeda



Võrguteenused ja .NET

- Ballmer:

“.NET is at the core of what we're trying to do as a company.“

- .NET: võrguteenuste loomise platvorm

- Pogrammeerimisplatvorm (C#, CLR, ...)
- XML-rakendused XML-keeltele SOAP, WSDL, UDDI
- Serveri tugi rakendustele

- SOAP:

- XML-RPC edasiarendus: XML-suhtluse keel üle http
- TOETUS: MSFT, IBM, SUN, ...

Rakendusi ...

- Intelligentne otsing võrgust
- Business-to-business lahendused:
 - Hulgimüüjad panevad võrku oma hinnakirja, jaemüüjad otsivad sobiva hinna ja tingimustega kaupa
 - Erinevate infosüsteemide ühendamise (kaks panka ühinevad, data vaja kokku viia, pangas palju väikesi süsteeme)
- Koosolekute planeerimine: leida ühised ajad
- Ligipääsuõiguste süsteem: kes kuhu kataloogi/infole ligi pääseb
- Intelligentne assistent: otsib ise mind huvitavaid uudiseid, hindu jne, teatab, kui midagi leiab

Protseduraalne vs deklaratiivne

- Protseduraalne info esitus:

info on programmi sisse kodeeritud.

- Deklaratiivne info esitus:

info on eraldi reeglitena (failis), mida algoritm reegleid kasutav algoritm rakendab

Mis puudub?

- Üks programm peab teise programmi antud XML-ist SISULISELT aru saama
- Õllepartiide otsimise server:

Hulgimüüja A annab infot:

```
<beer>
  <name>Guinness</name>
  <price>100</price>
</beer>
```

Hulgimüüja b annab infot:

```
<olu>
  <mark>Guinness</mark>
  <hind>100</hind>
</olu>
```

Hulgimüüja C annab infot:

```
<porter>
  <name>Guinness</name>
  <price>100</price>
</porter>
```


Erinevad keeled

- Peame teadma, et:

BEER = ÕLU
PORTER on ka BEER

- Ei ole lootust teha “ette valmis” kõiki keeli
- On olemas alamkeeled (üldine toodete keel, õllede keel Eestis, Tshehhis, Hiinas, õllede keel Tartu Õllevabrikus ja Sakus, ...)
- Keeli on vaja omavahel tõlkida
- Tõlkimiseks on vaja reegleid ja elementaarset arusaamist

Sama probleem XML andmebaasis ÜHES MASINAS

- SQL keeles on meil tabelid:

ID	Nimi	PID	Amet
1	Jaan Tamm	1	Degustaator
2	Ants Raud	2	Laomees
3	Jaan Tamm	1	Müügimees
		3	Juhataja

- Reeglid:
 - ID esimeses tabelis ei tohi korduda
 - PID teises tabelis peab olema esimeses tabelis
- Vaja: nimede seosed, reeglid struktuurile

Probleem on laial skaalal üks

üks andmebaas ↔ mitu masinat ↔ terve internet

Semantic web: projekt

- Eesmärk: pakkuda XML-põhiseid keeli ja luua standardeid, mille abil kirjeldada asju (internetis) ja (XML) keeltevahelisi seoseid.
- Soovitav tulemus: saab teha tarkvara, mille abil XML-datat saab ühest programmist teise saata, nii et info kaduma ei lähe.
- Esmajoones vaja:
 - objektide kirjeldamise keel
 - mõistete omavaheliste seoste ja reeglite keel
- Projekti juht: Tim Berners-Lee ([http, www](http://www) ja esimese brauseri autor, w3c juht).

Kihiline süsteem

- Ülevalt alla:

Tõestuste keel

Täisloogika

Ontoloogiad

Ontoloogikeel (DAML+OIL)

Asjade kirjelduskeel (RDF ja RDFS)

Nimede unikaliseerimine (XML namespaces)

Süntaks (XML)

Transport (http, https, SOAP, tcp/ip ...)

Nimede unikaliseerimine: XML namespaces

■ Idee:

- iga tagi ette saab kirjutada, mis keeles ta on <keel:tag>
- keelte nimedeks kasutatakse URI-sid (URL-e)

■ Näide:

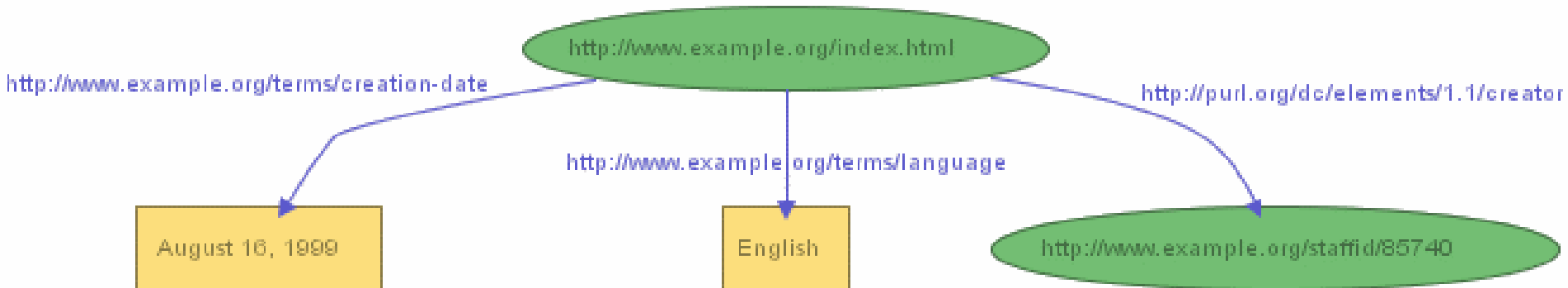
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:ex="http://www.example.org/terms/">
  <rdf:Description rdf:about="http://www.example.org/index.html">
    <ex:creation-date>August 16, 1999</ex:creation-date>
    <ex:language>English</ex:language>
    <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
  </rdf:Description>
</rdf:RDF>
```

Asjade kirjeldamine: RDF

- Idee:
 - Kõik kirjeldused on kolmikud: [omadus, ressurss, objekt]
 - Predikaadina: omadus(ressurss, objekt)
 - Kõik need kolm võivad olla URI-d

Kirjelduskeel RDF

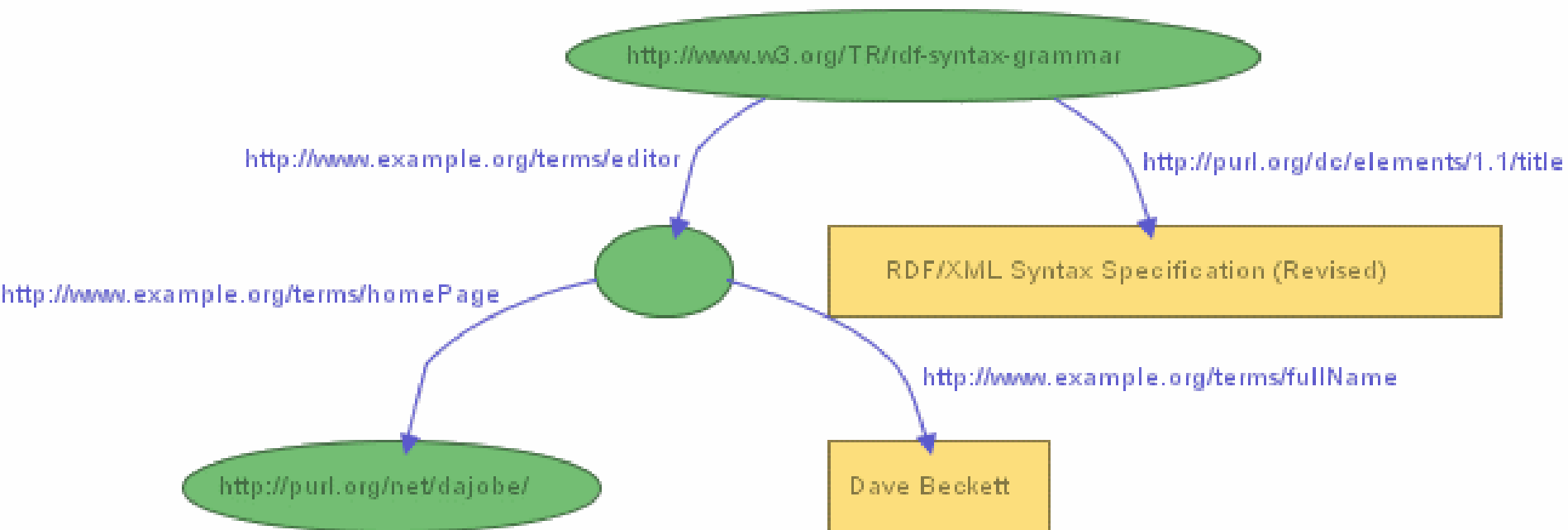
- <http://www.example.org/index.html> has a creation-date whose value is August 16, 1999
- <http://www.example.org/index.html> has a language whose value is English
- <http://www.example.org/index.html> has a creator indicated by <http://www.example.org/staffid/85740>



kirjelduskeel RDF

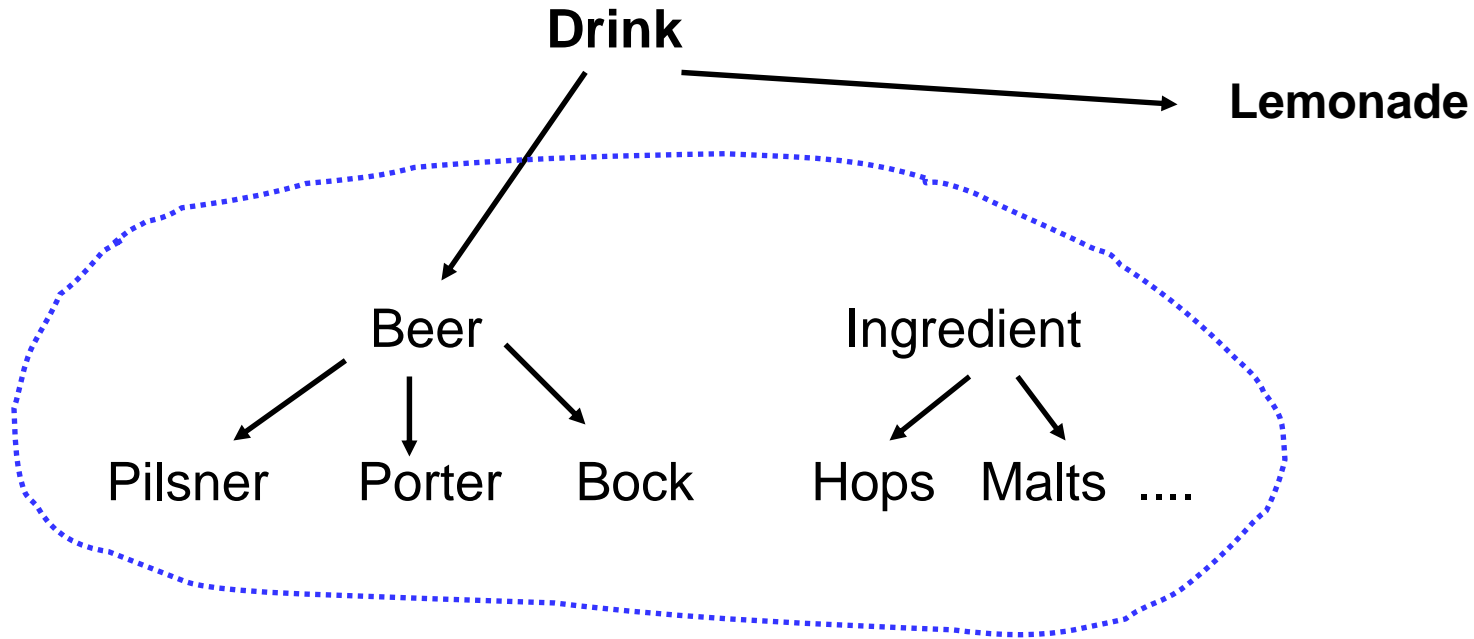
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:ex="http://www.example.org/terms/">
  <rdf:Description rdf:about="http://www.example.org/index.html">
    <ex:creation-date>August 16, 1999</ex:creation-date>
    <ex:language>English</ex:language>
    <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
  </rdf:Description>
</rdf:RDF>
```

kirjelduskeel RDF: sügavam puu



Ontoloogiad

- Mis on ontoloogiad: “maailma mudelid” ehk mõistete hierarhiad



- Eri valdkondades on oma ontoloogiad
- Ilmselt ei õnnestu panna kõiki programme kasutama samu “standardontoloogiaid”

Ontoogiakeeled DAML ja OIL

```
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns = "http://www.daml.org/2001/03/daml+oil#"
  xmlns:daml = "http://www.daml.org/2001/03/daml+oil#"
  xmlns:gen =
    "http://www.cs.umd.edu/projects/plus/DAML/onts/general1.0.daml#" >

<Ontology about="">
  <versionInfo>beer-ont, v.1.0</versionInfo>
  <comment>An ontology that models brewers and types of
    beer.</comment>
  <imports
    resource=
      "http://www.cs.umd.edu/projects/plus/DAML/onts/general1.0.daml" />
</Ontology>
```

■ ...

Ontoloogiakeeled DAML ja OIL

```
<Class ID="ScotchAle">  
  <subClassOf resource="#Ale" />  
</Class>
```

```
<Class ID="Brewery">  
  <subClassOf  
resource=  
"http://www.cs.umd.edu/projects/plus/DAML/onts/general1.0.daml#Organiza  
tion" />  
</Class>
```

```
<Class ID="Perle">  
<subClassOf resource="#Hops" />  
</Class>
```

```
<Class ID="Chinook">  
<subClassOf resource="#Hops" />  
</Class>
```

Ontoogiakeeled DAML ja OIL

```
<Class ID="Porter">  
  <subClassOf resource="#Beer" />  
</Class>
```

```
<Class ID="Caramel">  
  <subClassOf resource="#Malt" />  
</Class>
```

....

Loogika: järelduste tegemise teadus

Fundamentaalseid mõtlemismeetodeid:

- **Deduktsioon** - järelduste tegemine teadaolevatest faktidest ja reeglitest.
- **Induktsioon** - faktide ja reeglite õppimine ehk üldistuste tegemine.

Loogika
uurimisvaldkond



Mis on loogika?

Loogika on keel koos järeldamise algoritmiga

Formaalsed keeled, mille jaoks on võimalik konstrueerida algoritmi (s.o. selged, ühemõttelised, mehhaaniliselt järgitavad juhised) õigete lausete konstrueerimiseks.

Iga niisuguse loogikakeele K jaoks konstrueeritakse algoritm M , mille abil saab kontrollida, kas suvaline antud keeles K kirjutatud väide on õige või ei.

Taoline algoritm esitatakse enamasti loogikareeglite koguna.

Väite tõestus võib olla väga pikk ja keeruline.

Näide: sugulaste andmebaas loogika keeles

Faktid:

mees(Mihkel)

naine(Leena)

...

ema(Leena,Jaan)

isa(Mihkel,Jaan)

isa(Jaan,Mart)

ema(Maia,Ants)

isa(Mihkel,Ants)

...

Näide: sugulaste andmebaas loogika keeles

Faktid:

mees(Mihkel)

naine(Leena)

...

ema(Leena,Jaan)

isa(Mihkel,Jaan)

isa(Jaan,Mart)

ema(Maia,Ants)

isa(Mihkel,Ants)

...

omadus

objekt

Näide: sugulaste andmebaas loogika keeles

Faktid:

mees(Mihkel)

naine(Leena)

...

ema(Leena, Jaan)

isa(Mihkel, Jaan)

isa(Jaan, Mart)

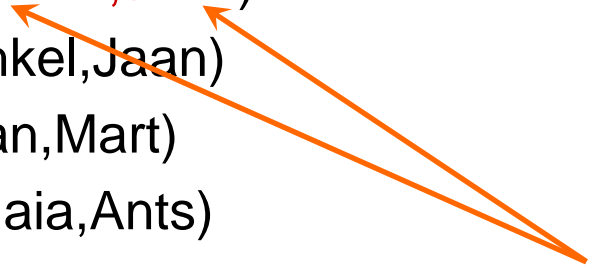
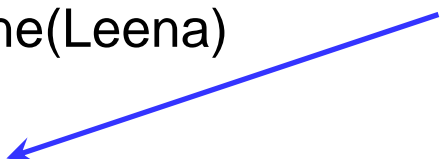
ema(Maia, Ants)

isa(Mihkel, Ants)

...

Seos või
suhe

objektid



Näide: sugulaste andmebaas loogika keeles

Reeglid:

Vanaisaks olemise saab defineerida järgmise väite abil:

``x on y-i **vanaisa** siis ja ainult siis, kui on olemas selline z, et x on z-i isa ja z on y-i isa või z on y-i ema".

Predikaatarvutuse formaalses keeles:

$$\forall x \forall y (\text{vanaisa}(x,y) \Leftrightarrow \exists z \text{ isa}(x,z) \ \& \ (\text{isa}(z,y) \vee \text{ema}(z,y)))$$

Näide: sugulaste andmebaas loogika keeles

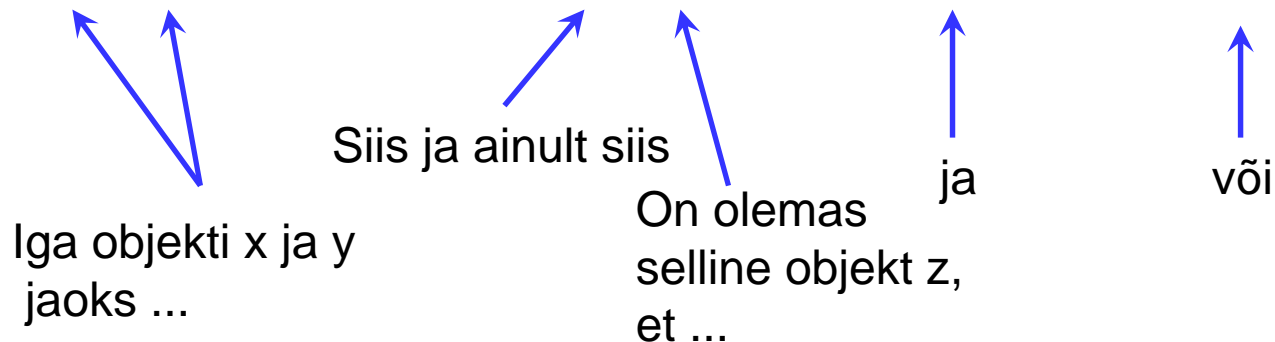
Reeglid:

Vanaisaks olemise saab defineerida järgmise väite abil:

"x on y-i vanaisa siis ja ainult siis, kui on olemas selline z, et x on z-i isa ja z on y-i isa või z on y-i ema".

Predikaatarvutuse **formaalses keeles**:

$$\forall x \forall y (\text{vanaisa}(x,y) \Leftrightarrow \exists z \text{ isa}(x,z) \& (\text{isa}(z,y) \vee \text{ema}(z,y)))$$



Näide: sugulaste andmebaas loogika keeles

Ülesanne: kas antud faktidest ja reeglitest järeldub meid huvitava lause tõesus?

``kas Mihkel on Mardi vanaisa?``

Peame formaalselt tõestama lause:

Faktid & Reeglid \Rightarrow vanaisa(Mihkel,Mart)

Näide: sugulaste andmebaas loogika keeles

Tõestus:

$$\forall x \forall y (\text{vanaisa}(x,y) \Leftrightarrow \exists z \text{ isa}(x,z) \ \& \ (\text{isa}(z,y) \ \forall \text{ ema}(z,y)))$$

Näide: sugulaste andmebaas loogika keeles

Tõestus:

$$\forall x \forall y (\text{vanaisa}(x,y) \Leftrightarrow \exists z \text{ isa}(x,z) \ \& \ (\text{isa}(z,y) \ \vee \ \text{ema}(z,y)))$$

$$\text{vanaisa}(\underline{\text{Mihkel}},\underline{\text{Mart}}) \Leftrightarrow \exists z \text{ isa}(\underline{\text{Mihkel}},z) \ \& \ (\text{isa}(z,\underline{\text{Mart}}) \ \vee \ \text{ema}(z,y))$$

Näide: sugulaste andmebaas loogika keeles

Tõestus:

$$\forall x \forall y (\text{vanaisa}(x,y) \Leftrightarrow \exists z \text{ isa}(x,z) \ \& \ (\text{isa}(z,y) \ \vee \ \text{ema}(z,y)))$$

$$\text{vanaisa}(\text{Mihkel},\text{Mart}) \Leftrightarrow \exists z \text{ isa}(\text{Mihkel},z) \ \& \ (\text{isa}(z,\text{Mart}) \ \vee \ \text{ema}(z,y))$$

$$\text{vanaisa}(\text{Mihkel},\text{Mart}) \Leftarrow \exists z \text{ isa}(\text{Mihkel},z) \ \& \ (\text{isa}(z,\text{Mart}) \ \vee \ \text{ema}(z,y))$$

Näide: sugulaste andmebaas loogika keeles

Tõestus:

$$\forall x \forall y (\text{vanaisa}(x,y) \Leftrightarrow \exists z \text{ isa}(x,z) \ \& \ (\text{isa}(z,y) \ \vee \ \text{ema}(z,y)))$$

$$\text{vanaisa}(\text{Mihkel},\text{Mart}) \Leftrightarrow \exists z \text{ isa}(\text{Mihkel},z) \ \& \ (\text{isa}(z,\text{Mart}) \ \vee \ \text{ema}(z,y))$$

$$\text{vanaisa}(\text{Mihkel},\text{Mart}) \Leftarrow \exists z \text{ isa}(\text{Mihkel},z) \ \& \ (\text{isa}(z,\text{Mart}) \ \vee \ \text{ema}(z,y))$$

$$\text{vanaisa}(\text{Mihkel},\text{Mart}) \Leftarrow \text{isa}(\text{Mihkel},\text{Jaan}) \ \& \ \text{isa}(\text{Jaan},\text{Mart})$$

Näide: sugulaste andmebaas loogika keeles

Tõestus:

$$\forall x \forall y (\text{vanaisa}(x,y) \Leftrightarrow \exists z \text{ isa}(x,z) \ \& \ (\text{isa}(z,y) \ \vee \ \text{ema}(z,y)))$$

$$\text{vanaisa}(\text{Mihkel},\text{Mart}) \Leftrightarrow \exists z \text{ isa}(\text{Mihkel},z) \ \& \ (\text{isa}(z,\text{Mart}) \ \vee \ \text{ema}(z,y))$$

$$\text{vanaisa}(\text{Mihkel},\text{Mart}) \Leftarrow \exists z \text{ isa}(\text{Mihkel},z) \ \& \ (\text{isa}(z,\text{Mart}) \ \vee \ \text{ema}(z,y))$$

$$\text{vanaisa}(\text{Mihkel},\text{Mart}) \Leftarrow \text{isa}(\text{Mihkel},\text{Jaan}) \ \& \ \text{isa}(\text{Jaan},\text{Mart})$$

vanaisa(Mihkel,Mart)

Lahenduvus

- Küsimuse $F \Rightarrow K$ vastuseks võib sisuliselt olla:
 - K järeldeb F-ist (tõsi)
 - K eitub järeldeb F-ist (vale)
 - ei K ega K eitub järeldeb F-ist (ebapiisav info)
- Täisloogika on poollahenduv: kui K on tõsi või vale, siis algoritm suudab seda näidata.
- Täisloogika ei ole lahenduv: ebapiisava info olukorras otsib algoritm tõestust üha edasi, saamata aru, et tõestust ei ole.

Ontoloogiad ja lahenduvad loogikaklassid

- Ontoloogiad on teisendatavad väga lihtsat tüüpi reegliteks loogikas.
- Ontoloogiast saadud reeglitesüsteemid on lahenduvad: tõestuse otsimise algoritm suudab alati tuvastada, kas valem tõestub või ei.
- Lahenduvates klassides tõestamine on üldiselt ka “suhteliselt” kiire.

Linke

- Üldist (XML, RDF jne):
 - <http://www.w3c.org/>
- Semantic web:
 - <http://www.w3c.org/2001/sw/>
 - www.semanticweb.org
- Daml:
 - www.daml.org