Applications of Description Logics

Application Areas I

Application Areas I

☞ Terminological KR and Ontologies

Application Areas I

☞ Terminological KR and Ontologies

- DLs initially designed for terminological KR (and reasoning)

Application Areas I

- ☞ Terminological KR and Ontologies
  - DLs initially designed for terminological KR (and reasoning)
  - Natural to use DLs to build and maintain ontologies

Application Areas I

- Terminological KR and Ontologies
  - DLs initially designed for terminological KR (and reasoning)
  - Natural to use DLs to build and maintain ontologies
- Semantic Web

Application Areas I

- Terminological KR and Ontologies
  - DLs initially designed for terminological KR (and reasoning)
  - Natural to use DLs to build and maintain ontologies
- Semantic Web
  - **Semantic** markup will be added to web resources

Application Areas I

- ☞ Terminological KR and Ontologies
  - DLs initially designed for terminological KR (and reasoning)
  - Natural to use DLs to build and maintain ontologies
- ☞ Semantic Web
  - **Semantic** markup will be added to web resources

    → Aim is “machine understandability”

Application Areas I

- ☞ Terminological KR and Ontologies
  - DLs initially designed for terminological KR (and reasoning)
  - Natural to use DLs to build and maintain ontologies
- ☞ Semantic Web
  - **Semantic** markup will be added to web resources
    - ➜ Aim is “machine understandability”
  - Markup will use **Ontologies** to provide common terms of reference with clear semantics

Application Areas I

- Terminological KR and Ontologies
  - DLs initially designed for terminological KR (and reasoning)
  - Natural to use DLs to build and maintain ontologies
- Semantic Web
  - **Semantic** markup will be added to web resources
    - ➜ Aim is “machine understandability”
  - Markup will use **Ontologies** to provide common terms of reference with clear semantics
  - Requirement for web based ontology language

Application Areas I

☞ Terminological KR and Ontologies

- DLs initially designed for terminological KR (and reasoning)
- Natural to use DLs to build and maintain ontologies

☞ Semantic Web

- **Semantic** markup will be added to web resources
  - → Aim is “machine understandability”
- Markup will use **Ontologies** to provide common terms of reference with clear semantics
- Requirement for web based ontology language
  - → Well defined semantics

Application Areas I

- Terminological KR and Ontologies
  - DLs initially designed for terminological KR (and reasoning)
  - Natural to use DLs to build and maintain ontologies
- Semantic Web
  - **Semantic** markup will be added to web resources
    - Aim is “machine understandability”
  - Markup will use **Ontologies** to provide common terms of reference with clear semantics
  - Requirement for web based ontology language
    - Well defined semantics
    - Builds on existing Web standards (XML, RDF, RDFS)

Application Areas I

☞ Terminological KR and Ontologies

- DLs initially designed for terminological KR (and reasoning)
- Natural to use DLs to build and maintain ontologies

☞ Semantic Web

- **Semantic** markup will be added to web resources
  - ➔ Aim is “machine understandability”
- Markup will use **Ontologies** to provide common terms of reference with clear semantics
- Requirement for web based ontology language
  - ➔ Well defined semantics
  - ➔ Builds on existing Web standards (XML, RDF, RDFS)
- Resulting language (DAML+OIL) is **based on a DL** ($\mathcal{SHIQ}$)

Application Areas I

- ☞ Terminological KR and Ontologies
  - DLs initially designed for terminological KR (and reasoning)
  - Natural to use DLs to build and maintain ontologies
- ☞ Semantic Web
  - **Semantic** markup will be added to web resources
    - ➔ Aim is “machine understandability”
  - Markup will use **Ontologies** to provide common terms of reference with clear semantics
  - Requirement for web based ontology language
    - ➔ Well defined semantics
    - ➔ Builds on existing Web standards (XML, RDF, RDFS)
  - Resulting language (DAML+OIL) is **based on a DL** ($\mathcal{SHIQ}$)
  - DL **reasoning** can be used to, e.g.,

Application Areas I

- ☞ Terminological KR and Ontologies
  - DLs initially designed for terminological KR (and reasoning)
  - Natural to use DLs to build and maintain ontologies
- ☞ Semantic Web
  - **Semantic** markup will be added to web resources
    - → Aim is “machine understandability”
  - Markup will use **Ontologies** to provide common terms of reference with clear semantics
  - Requirement for web based ontology language
    - → Well defined semantics
    - → Builds on existing Web standards (XML, RDF, RDFS)
  - Resulting language (DAML+OIL) is **based on a DL** ($\mathcal{SHIQ}$)
  - DL **reasoning** can be used to, e.g.,
    - → Support ontology design and maintenance

Application Areas I

- ☞ Terminological KR and Ontologies
  - DLs initially designed for terminological KR (and reasoning)
  - Natural to use DLs to build and maintain ontologies
- ☞ Semantic Web
  - **Semantic** markup will be added to web resources
    - ➔ Aim is “machine understandability”
  - Markup will use **Ontologies** to provide common terms of reference with clear semantics
  - Requirement for web based ontology language
    - ➔ Well defined semantics
    - ➔ Builds on existing Web standards (XML, RDF, RDFS)
  - Resulting language (DAML+OIL) is **based on a DL** ($\mathcal{SHIQ}$)
  - DL **reasoning** can be used to, e.g.,
    - ➔ Support ontology design and maintenance
    - ➔ Classify resources w.r.t. ontologies

Application Areas II

Application Areas II

☞ Configuration

Application Areas II

☞ Configuration

- **Classic** system used to configure telecoms equipment

Application Areas II

☞ Configuration

- **Classic** system used to configure telecoms equipment
- Characteristics of components described in DL KB

Application Areas II

☞ Configuration

- **Classic** system used to configure telecoms equipment
- Characteristics of components described in DL KB
- Reasoner checks validity (and price) of configurations

Application Areas II

☞ Configuration

- **Classic** system used to configure telecoms equipment
- Characteristics of components described in DL KB
- Reasoner checks validity (and price) of configurations

☞ Software information systems

Application Areas II

☞ Configuration

- **Classic** system used to configure telecoms equipment
- Characteristics of components described in DL KB
- Reasoner checks validity (and price) of configurations

☞ Software information systems

- LaSSIE system used DL KB for flexible software documentation and query answering

Application Areas II

- ☞ Configuration
  - **Classic** system used to configure telecoms equipment
  - Characteristics of components described in DL KB
  - Reasoner checks validity (and price) of configurations
- ☞ Software information systems
  - LaSSIE system used DL KB for flexible software documentation and query answering
- ☞ Database applications

Application Areas II

- ☞ Configuration
  - **Classic** system used to configure telecoms equipment
  - Characteristics of components described in DL KB
  - Reasoner checks validity (and price) of configurations
- ☞ Software information systems
  - LaSSIE system used DL KB for flexible software documentation and query answering
- ☞ Database applications
- ☞ . . .

Database Schema and Query Reasoning

Database Schema and Query Reasoning

- ☞ $\mathcal{DLR}$ (n-ary DL) can capture semantics of many conceptual modelling methodologies (e.g., EER)

Database Schema and Query Reasoning

- $\mathcal{DLR}$ (n-ary DL) can capture semantics of many conceptual modelling methodologies (e.g., EER)
- Satisfiability preserving mapping to $\mathcal{SHIQ}$ allows use of DL reasoners (e.g., FaCT, RACER)

Database Schema and Query Reasoning

- ☞ $\mathcal{DLR}$ (n-ary DL) can capture semantics of many conceptual modelling methodologies (e.g., EER)
- ☞ Satisfiability preserving mapping to $\mathcal{SHIQ}$ allows use of DL reasoners (e.g., FaCT, RACER)
- ☞ DL Abox can also capture semantics of conjunctive queries

Database Schema and Query Reasoning

- $\mathcal{DLR}$ (n-ary DL) can capture semantics of many conceptual modelling methodologies (e.g., EER)
- Satisfiability preserving mapping to $\mathcal{SHIQ}$ allows use of DL reasoners (e.g., FaCT, RACER)
- DL Abox can also capture semantics of conjunctive queries
  - Can reason about query containment w.r.t. schema

Database Schema and Query Reasoning

- ☞ $\mathcal{DLR}$ (n-ary DL) can capture semantics of many conceptual modelling methodologies (e.g., EER)
- ☞ Satisfiability preserving mapping to $\mathcal{SHIQ}$ allows use of DL reasoners (e.g., FaCT, RACER)
- ☞ DL Abox can also capture semantics of conjunctive queries
  - Can reason about query containment w.r.t. schema
- ☞ DL reasoning can be used to support

Database Schema and Query Reasoning

- $\mathcal{DLR}$ (n-ary DL) can capture semantics of many conceptual modelling methodologies (e.g., EER)
- Satisfiability preserving mapping to $\mathcal{SHIQ}$ allows use of DL reasoners (e.g., FaCT, RACER)
- DL Abox can also capture semantics of conjunctive queries
  - Can reason about query containment w.r.t. schema
- DL reasoning can be used to support
  - Schema design, evolution and query optimisation

Database Schema and Query Reasoning

- ☞ $\mathcal{DLR}$ (n-ary DL) can capture semantics of many conceptual modelling methodologies (e.g., EER)
- ☞ Satisfiability preserving mapping to $\mathcal{SHIQ}$ allows use of DL reasoners (e.g., FaCT, RACER)
- ☞ DL Abox can also capture semantics of conjunctive queries
  - Can reason about query containment w.r.t. schema
- ☞ DL reasoning can be used to support
  - Schema design, evolution and query optimisation
  - Source integration in heterogeneous databases/data warehouses

Database Schema and Query Reasoning

- ☞ $\mathcal{DLR}$ (n-ary DL) can capture semantics of many conceptual modelling methodologies (e.g., EER)
- ☞ Satisfiability preserving mapping to $\mathcal{SHIQ}$ allows use of DL reasoners (e.g., FaCT, RACER)
- ☞ DL Abox can also capture semantics of conjunctive queries
  - Can reason about query containment w.r.t. schema
- ☞ DL reasoning can be used to support
  - Schema design, evolution and query optimisation
  - Source integration in heterogeneous databases/data warehouses
  - Conceptual modelling of multidimensional aggregation

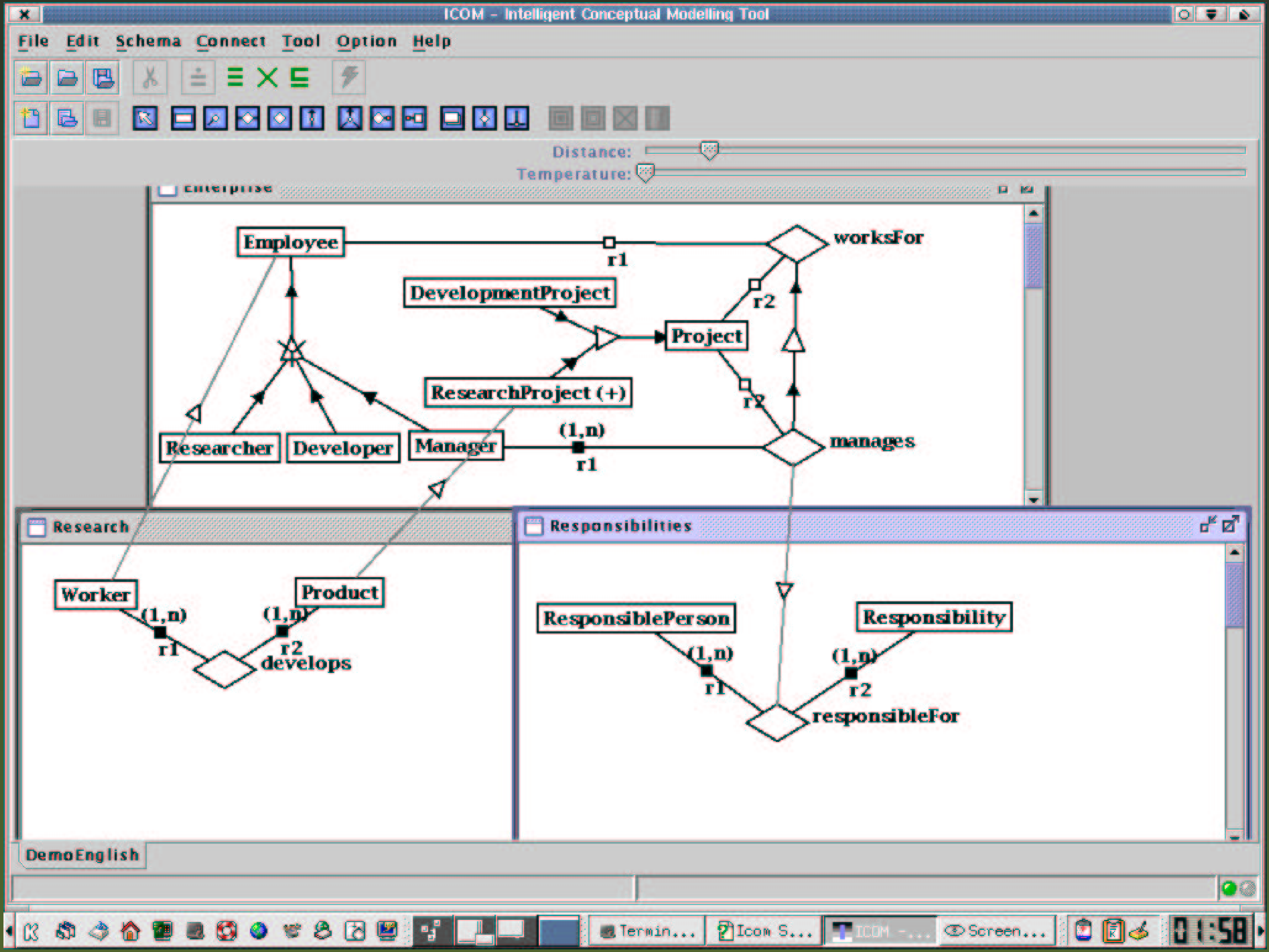
Database Schema and Query Reasoning

- ☞ $\mathcal{DLR}$ (n-ary DL) can capture semantics of many conceptual modelling methodologies (e.g., EER)
- ☞ Satisfiability preserving mapping to $\mathcal{SHIQ}$ allows use of DL reasoners (e.g., FaCT, RACER)
- ☞ DL Abox can also capture semantics of conjunctive queries
  - Can reason about query containment w.r.t. schema
- ☞ DL reasoning can be used to support
  - Schema design, evolution and query optimisation
  - Source integration in heterogeneous databases/data warehouses
  - Conceptual modelling of multidimensional aggregation
- ☞ E.g., **I.COM** Intelligent Conceptual Modelling tool (Enrico Franconi)

Database Schema and Query Reasoning

- $\mathcal{DLR}$ (n-ary DL) can capture semantics of many conceptual modelling methodologies (e.g., EER)
- Satisfiability preserving mapping to $\mathcal{SHIQ}$ allows use of DL reasoners (e.g., FaCT, RACER)
- DL Abox can also capture semantics of conjunctive queries
  - Can reason about query containment w.r.t. schema
- DL reasoning can be used to support
  - Schema design, evolution and query optimisation
  - Source integration in heterogeneous databases/data warehouses
  - Conceptual modelling of multidimensional aggregation
- E.g., **I.COM** Intelligent Conceptual Modelling tool (Enrico Franconi)
  - Uses FaCT system to provide reasoning support for EER

I.COM Demo


ICOM - Intelligent Conceptual Modelling Tool
File Edit Schema Connect Tool Option Help
Distance:
Temperature:
Enterprise
Employee
r1
worksFor
r2
DevelopmentProject
Project
ResearchProject (+)
r2
Researcher
Developer
Manager
(1,n)
r1
manages
Research
Worker
Product
(1,n)
(1,n)
r1
r2
develops
Responsibilities
ResponsiblePerson
Responsibility
(1,n)
(1,n)
r1
r2
responsibleFor
DemoEnglish
Termin...
Icon S...
ICOM -...
Screen...
01:58

Terminological KR and Ontologies

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

- ☞ General requirement for medical terminologies

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

☞ General requirement for medical terminologies

☞ Static lists/taxonomies difficult to build and maintain

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

- ☞ General requirement for medical terminologies
- ☞ Static lists/taxonomies difficult to build and maintain
  - Need to be very **large** and highly interconnected

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

- General requirement for medical terminologies
- Static lists/taxonomies difficult to build and maintain
  - Need to be very **large** and highly interconnected
  - Inevitably contain many **errors** and **omissions**

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

- General requirement for medical terminologies
- Static lists/taxonomies difficult to build and maintain
  - Need to be very **large** and highly interconnected
  - Inevitably contain many **errors** and **omissions**
- Galen project aims to replace static hierarchy with DL

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

- ☞ General requirement for medical terminologies
- ☞ Static lists/taxonomies difficult to build and maintain
  - Need to be very **large** and highly interconnected
  - Inevitably contain many **errors** and **omissions**
- ☞ Galen project aims to replace static hierarchy with DL
  - **Describe** concepts (e.g., spiral fracture of left femur)

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

- ☞ General requirement for medical terminologies
- ☞ Static lists/taxonomies difficult to build and maintain
  - Need to be very **large** and highly interconnected
  - Inevitably contain many **errors** and **omissions**
- ☞ Galen project aims to replace static hierarchy with DL
  - **Describe** concepts (e.g., spiral fracture of left femur)
  - Use DL classifier to **build taxonomy**

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

- General requirement for medical terminologies
- Static lists/taxonomies difficult to build and maintain
  - Need to be very **large** and highly interconnected
  - Inevitably contain many **errors** and **omissions**
- Galen project aims to replace static hierarchy with DL
  - **Describe** concepts (e.g., spiral fracture of left femur)
  - Use DL classifier to **build taxonomy**
- Needed expressive DL **and** efficient reasoning

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

- ☞ General requirement for medical terminologies
- ☞ Static lists/taxonomies difficult to build and maintain
  - Need to be very **large** and highly interconnected
  - Inevitably contain many **errors** and **omissions**
- ☞ Galen project aims to replace static hierarchy with DL
  - **Describe** concepts (e.g., spiral fracture of left femur)
  - Use DL classifier to **build taxonomy**
- ☞ Needed expressive DL **and** efficient reasoning
  - Descriptions use transitive/inverse roles, GCIs etc.

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

- General requirement for medical terminologies
- Static lists/taxonomies difficult to build and maintain
  - Need to be very **large** and highly interconnected
  - Inevitably contain many **errors** and **omissions**
- Galen project aims to replace static hierarchy with DL
  - **Describe** concepts (e.g., spiral fracture of left femur)
  - Use DL classifier to **build taxonomy**
- Needed expressive DL **and** efficient reasoning
  - Descriptions use transitive/inverse roles, GCIs etc.
  - Very large KBs (tens of thousands of concepts)

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

- ☞ General requirement for medical terminologies
- ☞ Static lists/taxonomies difficult to build and maintain
  - Need to be very **large** and highly interconnected
  - Inevitably contain many **errors** and **omissions**
- ☞ Galen project aims to replace static hierarchy with DL
  - **Describe** concepts (e.g., spiral fracture of left femur)
  - Use DL classifier to **build taxonomy**
- ☞ Needed expressive DL **and** efficient reasoning
  - Descriptions use transitive/inverse roles, GCIs etc.
  - Very large KBs (tens of thousands of concepts)
    - → Even prototype KB is very large (≈3,000 concepts)

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

- ☞ General requirement for medical terminologies
- ☞ Static lists/taxonomies difficult to build and maintain
  - Need to be very **large** and highly interconnected
  - Inevitably contain many **errors** and **omissions**
- ☞ Galen project aims to replace static hierarchy with DL
  - **Describe** concepts (e.g., spiral fracture of left femur)
  - Use DL classifier to **build taxonomy**
- ☞ Needed expressive DL **and** efficient reasoning
  - Descriptions use transitive/inverse roles, GCIs etc.
  - Very large KBs (tens of thousands of concepts)
    - → Even prototype KB is very large (≈3,000 concepts)
    - → Existing (incomplete) classifier took ≈**24 hours** to classify KB

Terminological KR and Ontologies

Initial motivation for work on **FaCT** system was **Galen** project

- ☞ General requirement for medical terminologies
- ☞ Static lists/taxonomies difficult to build and maintain
  - Need to be very **large** and highly interconnected
  - Inevitably contain many **errors** and **omissions**
- ☞ Galen project aims to replace static hierarchy with DL
  - **Describe** concepts (e.g., spiral fracture of left femur)
  - Use DL classifier to **build taxonomy**
- ☞ Needed expressive DL **and** efficient reasoning
  - Descriptions use transitive/inverse roles, GCIs etc.
  - Very large KBs (tens of thousands of concepts)
    - ➜ Even prototype KB is very large (≈3,000 concepts)
    - ➜ Existing (incomplete) classifier took **≈24 hours** to classify KB
    - ➜ FaCT system (sound and complete) takes **≈60 seconds**

Reasoning Support for Ontology Design

Reasoning Support for Ontology Design

☞ DL reasoner can be used to support design and maintenance

Reasoning Support for Ontology Design

- ☞ DL reasoner can be used to support design and maintenance
- ☞ Example is OilEd ontology editor (for DAML+OIL)

Reasoning Support for Ontology Design

- DL reasoner can be used to support design and maintenance
- Example is OilEd ontology editor (for DAML+OIL)
  - Frame based interface (like Protegé, OntoEdit, etc.)

Reasoning Support for Ontology Design

- ☞ DL reasoner can be used to support design and maintenance
- ☞ Example is OilEd ontology editor (for DAML+OIL)
  - Frame based interface (like Protegé, OntoEdit, etc.)
  - Extended to clarify semantics and capture whole DAML+OIL language

Reasoning Support for Ontology Design

- DL reasoner can be used to support design and maintenance
- Example is OilEd ontology editor (for DAML+OIL)
    - Frame based interface (like Protegé, OntoEdit, etc.)
    - Extended to clarify semantics and capture whole DAML+OIL language
        - Slots explicitly existential or value restrictions

Reasoning Support for Ontology Design

- ☞ DL reasoner can be used to support design and maintenance
- ☞ Example is OilEd ontology editor (for DAML+OIL)
  - Frame based interface (like Protegé, OntoEdit, etc.)
  - Extended to clarify semantics and capture whole DAML+OIL language
    - ➔ Slots explicitly existential or value restrictions
    - ➔ Boolean connectives and nesting

Reasoning Support for Ontology Design

- ☞ DL reasoner can be used to support design and maintenance
- ☞ Example is OilEd ontology editor (for DAML+OIL)
  - Frame based interface (like Protegé, OntoEdit, etc.)
  - Extended to clarify semantics and capture whole DAML+OIL language
    - ➔ Slots explicitly existential or value restrictions
    - ➔ Boolean connectives and nesting
    - ➔ Properties for slot relations (transitive, functional etc.)

Reasoning Support for Ontology Design

- DL reasoner can be used to support design and maintenance
- Example is OilEd ontology editor (for DAML+OIL)
  - Frame based interface (like Protegé, OntoEdit, etc.)
  - Extended to clarify semantics and capture whole DAML+OIL language
    - Slots explicitly existential or value restrictions
    - Boolean connectives and nesting
    - Properties for slot relations (transitive, functional etc.)
    - General axioms

Reasoning Support for Ontology Design

- ☞ DL reasoner can be used to support design and maintenance
- ☞ Example is OilEd ontology editor (for DAML+OIL)
  - Frame based interface (like Protegé, OntoEdit, etc.)
  - Extended to clarify semantics and capture whole DAML+OIL language
    - → Slots explicitly existential or value restrictions
    - → Boolean connectives and nesting
    - → Properties for slot relations (transitive, functional etc.)
    - → General axioms
- ☞ Reasoning support for OilEd provided by FaCT system

Reasoning Support for Ontology Design

- ☞ DL reasoner can be used to support design and maintenance
- ☞ Example is OilEd ontology editor (for DAML+OIL)
  - Frame based interface (like Protegé, OntoEdit, etc.)
  - Extended to clarify semantics and capture whole DAML+OIL language
    - ➔ Slots explicitly existential or value restrictions
    - ➔ Boolean connectives and nesting
    - ➔ Properties for slot relations (transitive, functional etc.)
    - ➔ General axioms
- ☞ Reasoning support for OilEd provided by FaCT system
  - Frame representation translated into $\mathcal{SHIQ}$

Reasoning Support for Ontology Design

- ☞ DL reasoner can be used to support design and maintenance
- ☞ Example is OilEd ontology editor (for DAML+OIL)
  - Frame based interface (like Protegé, OntoEdit, etc.)
  - Extended to clarify semantics and capture whole DAML+OIL language
    - ➔ Slots explicitly existential or value restrictions
    - ➔ Boolean connectives and nesting
    - ➔ Properties for slot relations (transitive, functional etc.)
    - ➔ General axioms
- ☞ Reasoning support for OilEd provided by FaCT system
  - Frame representation translated into $\mathcal{SHIQ}$
  - Communicates with FaCT via CORBA interface

Reasoning Support for Ontology Design

- DL reasoner can be used to support design and maintenance
- Example is OilEd ontology editor (for DAML+OIL)
  - Frame based interface (like Protegé, OntoEdit, etc.)
  - Extended to clarify semantics and capture whole DAML+OIL language
    - Slots explicitly existential or value restrictions
    - Boolean connectives and nesting
    - Properties for slot relations (transitive, functional etc.)
    - General axioms
- Reasoning support for OilEd provided by FaCT system
  - Frame representation translated into $\mathcal{SHIQ}$
  - Communicates with FaCT via CORBA interface
  - Indicates inconsistencies and implicit subsumptions

Reasoning Support for Ontology Design

- ☞ DL reasoner can be used to support design and maintenance
- ☞ Example is OilEd ontology editor (for DAML+OIL)
  - Frame based interface (like Protegé, OntoEdit, etc.)
  - Extended to clarify semantics and capture whole DAML+OIL language
    - ➔ Slots explicitly existential or value restrictions
    - ➔ Boolean connectives and nesting
    - ➔ Properties for slot relations (transitive, functional etc.)
    - ➔ General axioms
- ☞ Reasoning support for OilEd provided by FaCT system
  - Frame representation translated into $\mathcal{SHIQ}$
  - Communicates with FaCT via CORBA interface
  - Indicates inconsistencies and implicit subsumptions
  - Can make implicit subsumptions explicit in KB

DAML+OIL Medical Terminology Examples

E.g., DAML+OIL medical terminology ontology

DAML+OIL Medical Terminology Examples

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture transitive partonomy, causality, etc.

DAML+OIL Medical Terminology Examples

E.g., DAML+OIL medical terminology ontology

☞ Transitive roles capture transitive partonomy, causality, etc.

Smoking $\sqsubseteq$ $\exists$causes.Cancer plus Cancer $\sqsubseteq$ $\exists$causes.Death

$\Rightarrow$ Cancer $\sqsubseteq$ FatalThing

DAML+OIL Medical Terminology Examples

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture transitive partonomy, causality, etc.

  Smoking $\sqsubseteq$ $\exists$causes.Cancer plus Cancer $\sqsubseteq$ $\exists$causes.Death

  $\Rightarrow$ Cancer $\sqsubseteq$ FatalThing

- ☞ GCIs represent additional non-definitional knowledge

DAML+OIL Medical Terminology Examples

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture transitive partonomy, causality, etc.

  $\mathsf{Smoking} \sqsubseteq \exists \mathsf{causes}.\mathsf{Cancer}$ plus $\mathsf{Cancer} \sqsubseteq \exists \mathsf{causes}.\mathsf{Death}$

  $\Rightarrow \mathsf{Cancer} \sqsubseteq \mathsf{FatalThing}$

- ☞ GCIs represent additional non-definitional knowledge

  $\mathsf{Stomach\text{-}Ulcer} \doteq \mathsf{Ulcer} \sqcap \exists \mathsf{hasLocation}.\mathsf{Stomach}$ plus

  $\mathsf{Stomach\text{-}Ulcer} \sqsubseteq \exists \mathsf{hasLocation}.\mathsf{Lining\text{-}Of\text{-}Stomach}$

  $\Rightarrow \mathsf{Ulcer} \sqcap \exists \mathsf{hasLocation}.\mathsf{Stomach} \sqsubseteq \mathsf{OrganLiningLesion}$

DAML+OIL Medical Terminology Examples

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture transitive partonomy, causality, etc.

  $\mathsf{Smoking} \sqsubseteq \exists \mathsf{causes.Cancer}$ plus $\mathsf{Cancer} \sqsubseteq \exists \mathsf{causes.Death}$

  $\Rightarrow \mathsf{Cancer} \sqsubseteq \mathsf{FatalThing}$

- ☞ GCIs represent additional non-definitional knowledge

  $\mathsf{Stomach\text{-}Ulcer} \doteq \mathsf{Ulcer} \sqcap \exists \mathsf{hasLocation.Stomach}$ plus

  $\mathsf{Stomach\text{-}Ulcer} \sqsubseteq \exists \mathsf{hasLocation.Lining\text{-}Of\text{-}Stomach}$

  $\Rightarrow \mathsf{Ulcer} \sqcap \exists \mathsf{hasLocation.Stomach} \sqsubseteq \mathsf{OrganLiningLesion}$

- ☞ Inverse roles capture e.g. causes/causedBy relationship

DAML+OIL Medical Terminology Examples

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture transitive partonomy, causality, etc.

  Smoking $\sqsubseteq$ $\exists$causes.Cancer plus Cancer $\sqsubseteq$ $\exists$causes.Death

  $\Rightarrow$ Cancer $\sqsubseteq$ FatalThing

- ☞ GCIs represent additional non-definitional knowledge

  Stomach-Ulcer $\doteq$ Ulcer $\sqcap$ $\exists$hasLocation.Stomach plus

  Stomach-Ulcer $\sqsubseteq$ $\exists$hasLocation.Lining-Of-Stomach

  $\Rightarrow$ Ulcer $\sqcap$ $\exists$hasLocation.Stomach $\sqsubseteq$ OrganLiningLesion

- ☞ Inverse roles capture e.g. causes/causedBy relationship

  Death $\sqcap$ $\exists$causedBy.Smoking $\sqsubseteq$ PrematureDeath

  $\Rightarrow$ Smoking $\sqsubseteq$ CauseOfPrematureDeath

DAML+OIL Medical Terminology Examples

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture transitive partonomy, causality, etc.

  $\text{Smoking} \sqsubseteq \exists\text{causes.Cancer}$ plus $\text{Cancer} \sqsubseteq \exists\text{causes.Death}$

  $\Rightarrow \text{Cancer} \sqsubseteq \text{FatalThing}$

- ☞ GCIs represent additional non-definitional knowledge

  $\text{Stomach-Ulcer} \doteq \text{Ulcer} \sqcap \exists\text{hasLocation.Stomach}$ plus

  $\text{Stomach-Ulcer} \sqsubseteq \exists\text{hasLocation.Lining-Of-Stomach}$

  $\Rightarrow \text{Ulcer} \sqcap \exists\text{hasLocation.Stomach} \sqsubseteq \text{OrganLiningLesion}$

- ☞ Inverse roles capture e.g. causes/causedBy relationship

  $\text{Death} \sqcap \exists\text{causedBy.Smoking} \sqsubseteq \text{PrematureDeath}$

  $\Rightarrow \text{Smoking} \sqsubseteq \text{CauseOfPrematureDeath}$

- ☞ Cardinality restrictions add consistency constraints

DAML+OIL Medical Terminology Examples

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture transitive partonomy, causality, etc.

  $\mathsf{Smoking} \sqsubseteq \exists\mathsf{causes.Cancer}$ plus $\mathsf{Cancer} \sqsubseteq \exists\mathsf{causes.Death}$

  $\Rightarrow \mathsf{Cancer} \sqsubseteq \mathsf{FatalThing}$

- ☞ GCIs represent additional non-definitional knowledge

  $\mathsf{Stomach\text{-}Ulcer} \doteq \mathsf{Ulcer} \sqcap \exists\mathsf{hasLocation.Stomach}$ plus

  $\mathsf{Stomach\text{-}Ulcer} \sqsubseteq \exists\mathsf{hasLocation.Lining\text{-}Of\text{-}Stomach}$

  $\Rightarrow \mathsf{Ulcer} \sqcap \exists\mathsf{hasLocation.Stomach} \sqsubseteq \mathsf{OrganLiningLesion}$

- ☞ Inverse roles capture e.g. causes/causedBy relationship

  $\mathsf{Death} \sqcap \exists\mathsf{causedBy.Smoking} \sqsubseteq \mathsf{PrematureDeath}$

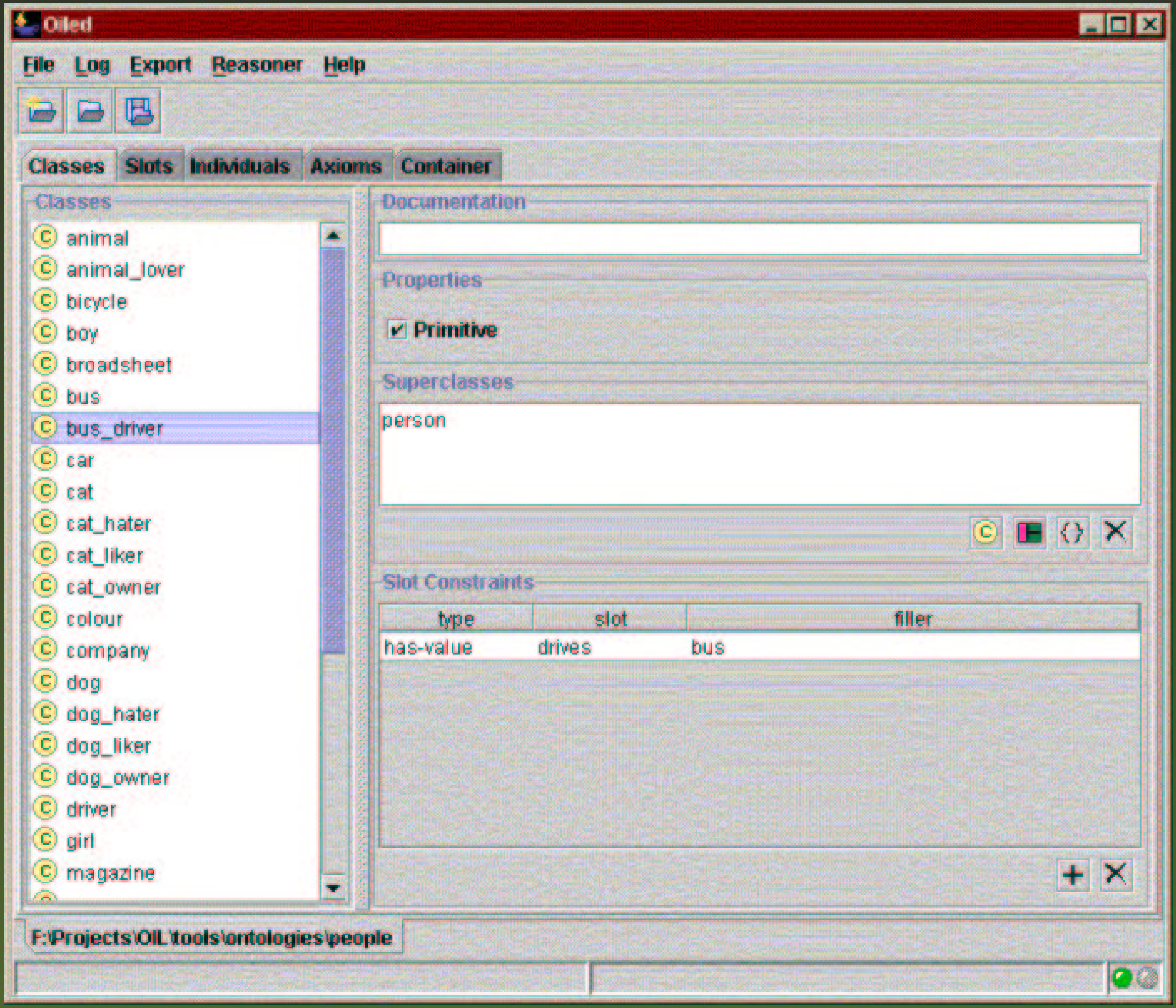
  $\Rightarrow \mathsf{Smoking} \sqsubseteq \mathsf{CauseOfPrematureDeath}$

- ☞ Cardinality restrictions add consistency constraints

  $\mathsf{BloodPressure} \sqsubseteq \exists\mathsf{hasValue}.(\mathsf{High} \sqcup \mathsf{Low}) \sqcap \leqslant 1 \mathsf{hasValue}$ plus

  $\mathsf{High} \sqsubseteq \neg\mathsf{Low} \Rightarrow \mathsf{HighLowBloodPressure} \sqsubseteq \bot$

OilEd Demo



☑ Primitive

Superclasses: person

type	slot	filler
has-value	drives	bus

F:\Projects\OIL\tools\ontologies\people