Representing and Validating Digital Business Processes

(toward the Digital Business Ecosystem)

Ernesto Damiani
DTI – Università di Milano
Italy

Summary

- Introduction
- Business Modeling
  - Business Rules
    - Business Rules Team (BRT) approach
  - Business Process Models
- Logical Models
  - Relationship with Semantic Web style knowledge representation
- Conclusion and Open Issues

Introduction

Our research lab

- Software Architectures and Knowledge-based systems lab
- Part of the Department of Information Technology, University of Milan, Italy

Located on the new campus in Crema, 40 km south-east of Milan

Research on Process Engineering at DTI

- KIWI – Knowledge Innovation for the Web Infrastructure
  - Extracting knowledge from process data
- MAPS – Agile Software Methodologies
  - Quantitative assessment of agile processes
- TEKNE – Technologies for Extracting Process Knowledge
  - Measuring process performance, both a priori and in itinere
  - Online process re-engineering if something goes wrong

Business Process Representation

Extended Digital Enterprise

- Why “Digital”: it is the digitalization of an enterprise’s traditional business processes, adding new processes enabled by e-business technology (e.g. CRM)
- Why “Extended”: digital business processes may include suppliers, consultants and even customers
- EDEs’ Business processes are *specified* using declarative languages and *enacted* via message interchanges (e.g., on a SOA)

Digital Business Ecosystem

- Distributed environment for online creation and operation of e-enterprises
  - Supports (semi)-automatic digital business process setup, matching and integration
  - Encourages co-evolution of business processes toward innovation
- An emerging field: FP6 European projects (DBE, OPAALS) – IEEE conference just started (IEEE-DEST), first edition in Cairns

Process Transparency

- EDEs must support a higher degree of *process transparency* than conventional enterprises
- Inter-organizational processes need to be monitored and tuned
  - Examples:
    - Need to know how an outsourced process is carried out, e.g. for quality assurance purposes
    - Need to know the impact of an advertising campaign while it is going on

Dynamic Integration

- A key success factor for EDEs is achieving seamless integration of business processes
  - Timeliness a major issue, e.g. when multiple companies join a EDE to provide a complex packaged product to a third party
- Organizations wishing to enter a EDE need tools for automatically testing their business models and services compatibility.

Business Modeling

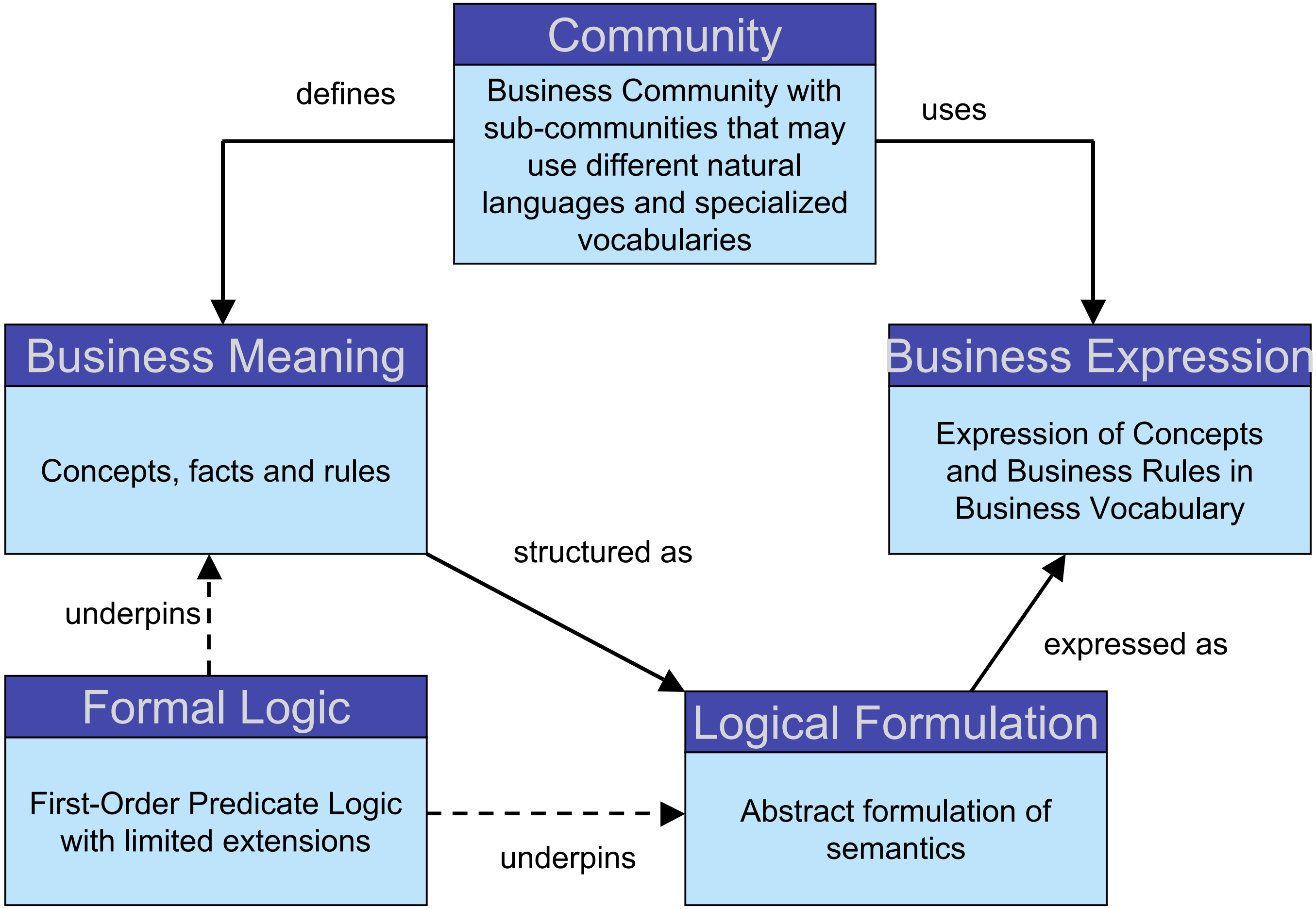
- Business Modeling is about Machine-Understandable description of businesses
- Two main layers:
  - Business Rules (BR): declarative statements describing business domains
    - Easy enough to be understood by humans
  - Business Process Models (BPM): stepwise specifications of individual business processes
    - Machine-understandable and detailed enough to be translated into computer-supported workflows
- Many approaches exist – more later

OMG – SBVR

- OMG has defined a Specification on Semantics of Business Vocabulary & Business Rules (SBVR), inserted in the wider Model Driven Architecture (MDA) vision
- The purpose of SBVR is to describe formally and without ambiguities the semantics of a business model.

Business Rules: the BRT Proposal

- Business Rules Team (BRT): 17 Organizations from 6 countries
- 3 of the proposers are also proposers for Business Process Definition Metamodel (BPDM)



Rules' modalities

- **Definition (something true by definition)**

  Example: "The sales tax rate for a rental is the sales tax rate at the pick-up branch of the rental on the drop-off date of the rental".

- **Business Rule (obligation or prohibition)**

  Example: "It is required that the drop-off date of a rental precedes the expiration date on the driver's license of the customer reserving the rental".

- **Authorization (permission)**

  Example: "A customer reserving a rental may provide what credit card guarantees the rental if the name on the credit card is the name of the customer or the credit card is for a corporate account authorized for use by the customer".

Vocabulary — Symbol types

- Terms
  - account, assigned car, branch, credit card, driver's license, drop-off branch, drop-off date, pick-up branch
- Names
  - EU-Rent, Milan Branch
- Verbs (Forms of Expression)
  - credit card guarantees rental
  - customer has driver's license
  - customer reserves rental
  - sales tax rate is charged at branch on date

Concepts

- **One concept can be represented by many terms**

    car, automobile

    voiture

- **One fact type can be represented in many forms**
    - **Sentential Forms**

        driver's license expires on date

        expiration date is on driver's license

        driver's license has expiration date
    - **Noun Forms**

        expiration date of driver's license

        driver's license expiring on date

        driver's license having expiration date

Why logics?

- Fully modeling business behavior using SBVR vocabularies can be difficult for complex systems.
- SBVR needs strong formal logic underpinning in order to enable automated reasoning on rules
  - spotting inconsistencies that may lead to contradictory situations.
  - discovering implicit knowledge implied by business rules.

From Basic Language to FOL

“**each**”, “**some**”, “**no**”	– Quantifiers
“**if**”, “**and**”, “**or**”	– Logical Connectives
“**required**”, “**prohibited**”	– Modalities
“**a**”, “**an**”, “**the**”	– Quantification and variable binding

Example:

Basic Language: For class C rentals, if pick-up branch does not coincide with drop-off, then rental duration cannot be less than two days

Semi-formal (parsed): `Foreach` rental X, `if` X.class = `'C' and` X.pickup<> X.dropoff, `then days(`X.duration`) > 1`

FOL: *√x∈Rental, IsEqual(Class(x),”C”)∧¬IsEqual(PickUp(x),DropOff(x))→IsGreater(Duration(x),1)*

Business Statements in Color

- **Requirement**

  It is required that the drop-off date of a rental precedes the expiration date on the driver's license of the customer reserving the rental.

- **Definition**

  The sales tax rate for a rental is the sales tax rate at the pick-up branch of the rental on the drop-off date of the rental.

- **Authorization**

  A customer reserving a rental may provide what credit card guarantees the rental if the name on the credit card is the name of the customer or the credit card is for a corporate account authorized for use by the customer.

Why modalities

- Modalities distinguish *alethic* from *deontic* rules.
- The former modality express reality as it is (e.g. by a law of nature), while the latter specifies how things should be (e.g., according to company regulations)
  - Alethic vision is expressed at design time, e.g. by UML diagrams
  - Deontic vision = company policy
- Can be compared for compliance checking
- Will deal with modalities later

Business Rule – Parsed

It is required that the drop-off date of a rental precedes the expiration date on the driver's license of the customer reserving the rental.

It is required that

precedes

the expiration date on

the driver's license of

the customer reserving

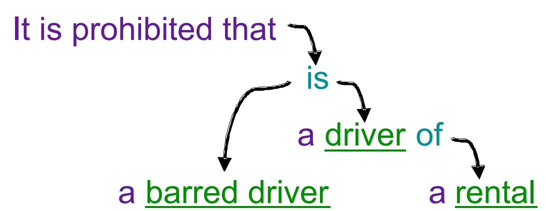
the drop-off date of a rental

From Business Rule to XML

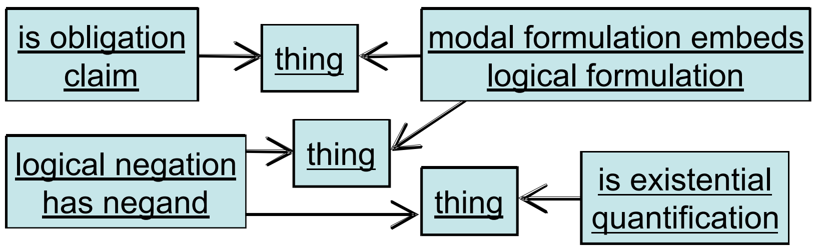
1. Start with a business rule statement.
2. Identify symbols in vocabulary.
3. Parse according to language rules.
4. Restate as facts of logical formulation.
5. Represent facts of logical formulation as objects.
6. Write objects as XML.

It is prohibited that a barred driver is a driver of a rental.

It is prohibited that a barred driver is a driver of a rental.



An obligation claim embeds a logical negation. [See next slide]



<is-obligation-claim .../>
[See following slides]

Logical Formulation

It is prohibited that a barred driver is a driver of a rental.

<u>obligation claim</u>
. embeds a <u>logical formulation</u> that is a <u>logical negation</u>
. . has a <u>negand</u> that is an <u>existential quantification</u>
. . . introduces a <u>variable</u>
. . . . has the <u>type</u> <u>*barred driver*</u>
. . . scopes over an <u>existential quantification</u>
. . . . introduces a <u>variable</u>
. . . . . has the <u>type</u> <u>*rental*</u>
. . . . scopes over an <u>atomic formulation</u>
. . . . . is based on the <u>fact type</u>: '<u>rental</u> has <u>driver</u>'
. . . . . has a <u>role binding</u>
. . . . . . is of the <u>fact type role</u> that is '<u>rental</u>' of '<u>rental</u> has <u>driver</u>'
. . . . . . binds to the <u>variable</u> that has the <u>type</u> <u>*rental*</u>
. . . . . has a <u>role binding</u>
. . . . . . is of a <u>fact type role</u> that is '<u>driver</u>' of '<u>rental</u> has <u>driver</u>'
. . . . . . binds to the <u>variable</u> that has the <u>type</u> <u>*barred driver*</u>

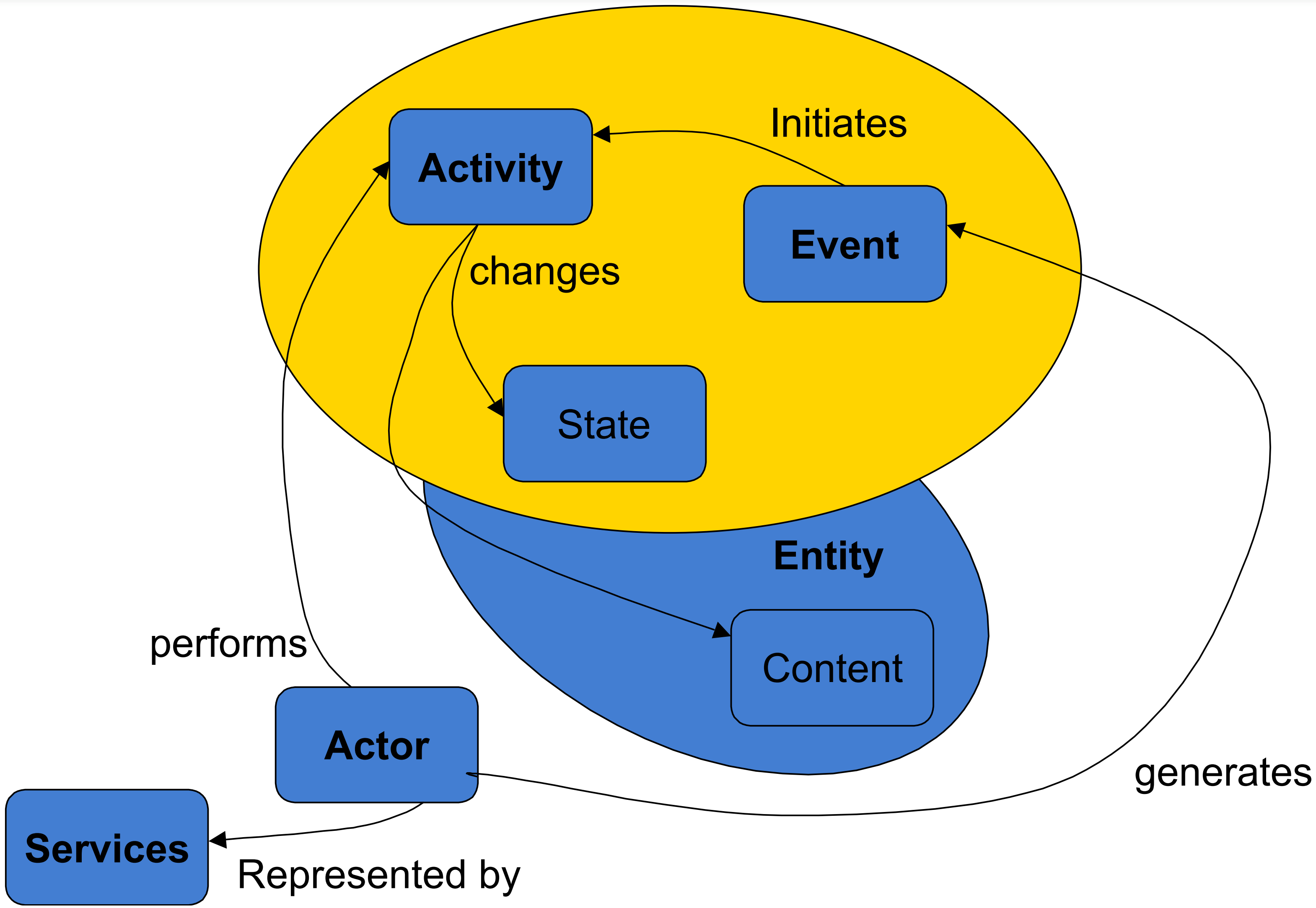
XML encoding for FOL

```
<is-obligation-claim obligation-claim="oc"/>
<modal-formulation-embeds-logical-formulation modal-formulation="oc" logical-formulation="n"/>
<logical-negation-has-negand logical-negation="n" negand="eq1"/>
<is-existential-quantification existential-quantification="eq1"/>
<quantification-introduces-variable quantification="eq1" variable="v2"/>
<variable-has-type variable="v1" type="bdt"/>
<quantification-scopes-over-logical-formulation quantification="eq1" logical-formulation="eq2"/>
<is-existential-quantification existential-quantification="eq2"/>
<quantification-introduces-variable quantification="eq2" variable="v2"/>
<variable-has-type variable="v2" type="rt"/>
<quantification-scopes-over-logical-formulation quantification="eq2" logical-formulation="af"/>
<is-atomic-formulation atomic-formulation="af"/>
<atomic-formulation-is-based-on-fact-type atomic-formulation="af" fact-type="ft"/>
<atomic-formulation-has-role-binding atomic-formulation="af" role-binding="rb1"/>
<role-binding-is-of-fact-type-role role-binding="rb1" fact-type-role="ftr1"/>
<atomic-formulation-has-role-binding atomic-formulation="af" role-binding="rb2"/>
<role-binding-is-of-fact-type-role role-binding="rb2" fact-type-role="ftr2"/>
<esbr:thing xmi:id="oc"/> <esbr:thing xmi:id="n"/> <esbr:thing xmi:id="eq1"/>
<esbr:thing xmi:id="v1"/> <esbr:thing xmi:id="bdt"/> <esbr:thing xmi:id="eq2"/>
<esbr:thing xmi:id="v2"/> <esbr:thing xmi:id="rt"/> <esbr:thing xmi:id="af"/>
<esbr:thing xmi:id="ft"/> <esbr:thing xmi:id="rb1"/> <esbr:thing xmi:id="rb2"/>
<esbr:thing xmi:id="ftr1"/> <esbr:thing xmi:id="ftr2"/>
```

BR vs. BPM

- Business rules define what a business domain is all about
- Can be used as a starting point to develop business processes or as a way to validate them
- Business process descriptions are based on suitable BPMs

Elements of a BPM



BPM languages

- Two approaches
  - **Event Oriented**
    - BPML, BPEL
    - Event Calculus
  - **Activity Oriented**
    - WfMC
    - Petri nets
- These approaches will eventually converge, incorporating state as a part of the model

BPEL

- An XML language for defining the decomposition of (business) services into (web) services
- A simple Orchestration language, not a Business Process Language
- BPEL assumes that business processes can be fully captured in a single definition, including all possible exception paths
  - Not sure this is the right assumption

Business Rules Semantics

BR semantics issues

- Semantics of BRs has been traditionally described by mapping the business rules syntax to some well-known logic formalism.
- We identified three (potentially conflicting) basic requirements for logical models underlying BRs
  - 1. High expressive power. A basic overall require- ment for the underlying logics is a high expressive power in specifying rules, in order to provide most of the description capabilities of controlled English. Business modelers accustomed to using English syntax would not accept any too severe limitation to the modeling language's expressive power.
  - 2. Tractability. The need for performing rule checking and for executability, however, requires the underlying logics expressive power to be carefully balanced against tractability.
  - 3. Non-falsifiability. BR logical semantics should rely on monotonic reasoning, i.e. on inference whose conclusions cannot be contradicted by simply adding new knowledge.
- These requirements have been often emphasized by business analysts and researchers as guidelines toward finding a correct logical interpretation of business models, but are NOT satisfied by current SBVR logical modeling.

Dealing with modalities

- Much research work is being done to provide a logics-based model for BR including modalities.
- Indeed, supporting modalities does not mean that it is mandatory to map the BRs to a modal logic.
  - Work by the BR OMG team and Terry Halpin addresses logical formalization for SBVR by mapping BR's deontic modalities Required, Prohibited into modal operators obligatory (O), permitted (P) (used when no modality is specified in the rule) and forbidden (F).

Dealing with modalities (2)

- Halpin's NORMA approach (see package aty http://sourceforge.net/projects/orm) represents BRs as rules where the only modal operator is the main rule operator, thus avoiding the need for a modal logics model.
  - Some SBVR formulations that violate this restriction may be transformed into an equivalent NORMA expression, e.g. by applying modal negation rules and the Barcan formulae
- Does not always work.. But let's assume it does

An hybrid approach

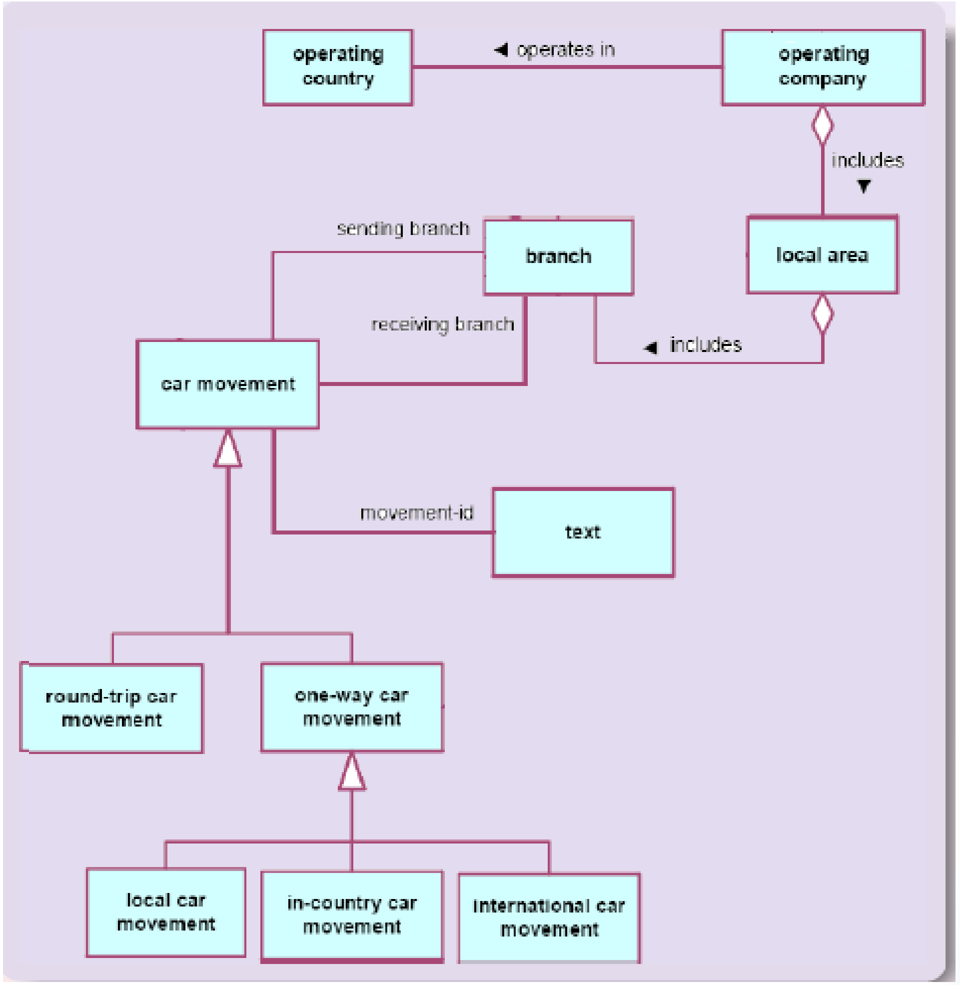
- Factor out modalities and deal with each model separately
- Express as much of each model as you can using a monotonic logic
- Express the rest as a rule system (e.g., using Horn rules)
- Carry out reasoning using a monotonic reasoner and a Prolog engine in an integrated fashion

DLs as a logic model for BR

- OWL DL is based on a subset of FOL providing
  - i) a declarative formalism for representation and expression of knowledge
  - ii) sound, tractable reasoning methods.
- BUT:
  - In all its flavors (Lite, DL, and Full) OWL is NOT expressive enough to model all the possible semantics of business vocabularies.
- When needed, business entities can be modeled using Horn Rules expressed in SWRL, an extension to OWL made by the combination of RuleML and OWL
  - grounded in FOL, with more expressive power than DL.

An Example

- We provide the translation between BRs encoding a small fragment of the EU-Rent example and OWL DL enhanced with SWRL.
- We are interested in identifying:
  - which aspects of business vocabularies and rules can be effectively modelled with OWL DL;
  - which require closed-world theorem provers to enforce more general FOL constraints.



Example (2)

- Vocabularies portray static information and requirements that maps fairly well to DL concept and role definitions.
- Fragment of EU-Rent description of business entity car movement:
  - Plain english: a car movement has a sending branch
  - BR: Necessity: each car movement has exactly one sending branch

From fact types to DL axioms

- Our example introduces:
  - concept definitions (CarMovement and Branch);
  - role associations (hasSendingBranch).
- Each keyword defines a necessary condition that ought to be shared by all instances of concept CarMovement.
- Property hasSendingBranch is made functional to express the “exactly one” constraint.

$$\mathrm{CarMovement} \sqsubseteq \exists\, \mathrm{hasSendingBranch.Branch}$$
$$\top \sqsubseteq\ \leq 1\ \mathrm{hasSendingBranch}$$

From fact types to DL axioms

- Another set of facts specializes the concept “car movement”.
- E.g., “one-way car movement” is further specialized according to the geographic distance between sending branch and receiving branch
  - round-trip car movement is a car movement that is round-trip
  - one-way car movement car movement that is not round-trip
  - local car movement one-way car movement that is local
  - international car movement one-way car movement that is international
  - in-country car movement one-way car movement that is not local and is not international
- Descriptions may refer to each other for their definition.

From fact types to DL axioms

- The fact types categorizing the specific types of car movement can be expressed with the following inclusion axioms:

$$
\begin{aligned}
\mathrm{OneWayCarMovement} &\sqsubseteq \mathrm{CarMovement}\\
\mathrm{RoundTripCarMovement} &\sqsubseteq \mathrm{CarMovement}\\
\mathrm{RoundTripCarMovement} &\sqsubseteq \neg\, \mathrm{OneWayCarMovement}\\
\mathrm{LocalCarMovement} &\sqsubseteq \mathrm{OneWayCarMovement}\\
\mathrm{InternationalCarMovement} &\sqsubseteq \mathrm{OneWayCarMovement}\\
\mathrm{InCountryCarMovement} &\sqsubseteq \mathrm{OneWayCarMovement}\\
\mathrm{InCountryCarMovement} &\sqsubseteq \neg \mathrm{LocalCarMovement} \sqcap\\
&\quad \neg \mathrm{InternationalCarMovement}
\end{aligned}
$$

- Complementarity between concepts OneWayCarMovement and RoundTripCarMovement is rendered by defining the latter as the complement of the former.
- In the OWL representation, being round-trip does not contribute to the definition of subsumption relationships between these concepts.

Applying DL reasoning

- These axioms can be loaded onto a reasoner to:
  - check the vocabulary consistency,
  - classify concept definitions,
  - populate the ontology with individuals representing the current state of the business domain
- This is done according to the open-world assumption (OWA) distinguishing both OWL semantics and SBVR default interpretation of a business domain.
- OWA: If a fact does not hold, we cannot assume the contrary to be true

DLs are not enough

- To model characteristics, the notions of branch, local area, operating company, and operating country have to be defined.
- We need to model part-of relationships between entities that cannot be properly modeled with OWL built-in constructs.

branch *is included in* local area

Necessity: **each** branch *is included in* **exactly one** local area

local area *is included in* operating company

Necessity: **each** local area *is included in* **exactly one** operating company

operating company *operates in* operating country

Necessity: **each** operating company *operates in* **exactly one** operating country

DLs are not enough

- The decidable fragment of FOL that can be expressed with OWL-DL is too limited to support all possible semantics of business vocabularies.

branch *has* country

Necessity: the country of a branch *is* the operating country of the operating company that *includes* the local area that *includes* the branch

$$\text{hasCountry}(?x,?c) \leftarrow \text{Branch}(?x) \wedge \text{OperatingCountry}(?c) \wedge \text{componentOf}(?x,?y) \wedge \text{LocalArea}(?y) \wedge \text{componentOf}(?y,?z) \wedge \text{OperatingCompany}(?z) \wedge \text{operatesIn}(?z,?c)$$

- It is not possible to render the rule above in OWL DL
  - Can be done un OWL 1.1 – but this does not solve all problems.

DLs are not enough

- Also, the simple characteristic of being round-trip, cannot be expressed with OWL DL:

car movement *being* round-trip

Definition: car movement *having* sending branch that *is* the receiving branch of the car movement

$$\text{RoundTripCarMovement}(?x) \leftarrow \text{CarMovement}(?x) \wedge \text{hasReceivingBranch}(?x,?y) \wedge \text{hasSendingBranch}(?x,?y)$$

Comparing individuals (fillers of variable ?y) is generally NOT achievable with DL formalisms

Using Horn Rules

- Horn Rules (e.g., expressed in the SWRL formalism) can compensate for this limitation.

car movement *being* local

Definition: car movement *having* receiving branch that *is included in* the local area of the sending branch of the car movement

$$\text{LocalCarMovement}(?x) \longleftarrow \text{CarMovement}(?x) \wedge \text{hasReceivingBranch}(?x,?y) \wedge \text{hasSendingBranch}(?x,?z) \wedge \text{LocalArea}(?a) \wedge \text{componentOf}(?y,?a) \wedge \text{componentOf}(?z,?a)$$

- The definition of property hasCountry becomes straightforward:

car movement *being* international

Definition: car movement *having* country of sending branch that *is* not the country of the receiving branch of the car movement

$$\text{InternationalCarMovement}(?x) \longleftarrow \text{OneWayCarMovement}(?x) \wedge \text{hasReceivingBranch}(?x,?y) \wedge \text{hasSendingBranch}(?x,?z) \wedge \text{hasCountry}(?y,?a) \wedge \text{hasCountry}(?z,?b) \wedge ?a \neq ?b$$

Consequences

- In the Horn part of the model, reasoning in the open-world is no longer possible: if a fact does not hold ot is assumed the contrary to be true
- Need to integrate the open-world approach provided by DL reasoners and the closed-world assumption (CWA) of Prolog-like execution models.
  - This integration cannot be accomplished by merely interleaving open- and closed-world evaluations of the business model.
- Need sound and complete decision procedures that can be applied to hybrid knowledge bases of this kind.

Conclusions

- When modeling business domains, SBVR does not deal with the actual enforcement of rules, e.g. the interchange of alethic and deontic interpretations of facts that may coexist in the business domain.
- Translating vocabularies and rules into logic structures and defining a reasoning strategy is a first step toward simultaneously enforcing these different categories of constraints.

Additional readings

- P. Ceravolo, E. Damiani, C. Fugazza, K. Reed, Representing and Validating Digital Business Processes, Proc. of WEBIST 2007, Barcelona, Spain, March 2007
- C. V. Damasio: Open and Closed Reasoning in the Semantic Web http://centria.di.fct.unl.pt/cd/publicacoes/openclosed.pdf
- Business Rules Team (BRT): Semantics of Business Vocabulary and Business Rules http://www.businessrulesgroup.org/sbvr.shtml.
- Semantic Web Rule Language (SWRL) http://www.w3.org/Submission/SWRL/