Making Data Meaningful: The Business Intelligence Model and Its Formal Semantics in Description Logics

Jennifer Horkoff[1], Alex Borgida[2], John Mylopoulos[1], Daniele Barone[1], Lei Jiang[1], Eric Yu[3], and Daniel Amyot[4]

[1] Dept. of Computer Science, University of Toronto, Canada
[2] Dept. of Computer Science, Rutgers University, USA
[3] Faculty of Information, University of Toronto, Canada
[4] EECS, University of Ottawa, Canada
{jenhork,jm,barone,leijiang}@cs.toronto.edu,
borgida@cs.rutgers.edu, eric.yu@utoronto.ca,
damyot@eecs.uottawa.ca

Abstract. Business Intelligence (BI) offers great opportunities for strategic analysis of current and future business operations; however, existing BI tools typically provide data-oriented responses to queries, difficult to understand in terms of business objectives and strategies. To make BI data meaningful, we need a conceptual modeling language whose primitive concepts represent business objectives, processes, opportunities and threats. We have previously introduced such a language, the Business Intelligence Model (BIM). In this paper we consolidate and rationalize earlier work on BIM, giving a precise syntax, reducing the number of fundamental concepts by using *meta-attributes*, and introducing the novel notion of "pursuit". Significantly, we also provide a formal semantics of BIM using a subset of the OWL Description Logic (DL). Using this semantics as a translation, DL reasoners can be exploited to (1) propagate evidence and goal pursuit in support of "what if?" reasoning, (2) allow extensions to the BIM language, (3) detect inconsistencies in specific BIM models, and (4) automatically classify defined concepts relative to existing concepts, organizing the model.

Keywords: Business Intelligence, Business Model, Goal Modeling, Situation Analysis, Model Reasoning, Goal Reasoning, Formal Semantics, Description Logics.

1 Introduction

Business Intelligence (BI) offers considerable potential for gaining insights into business operations, including identification of opportunities and threats. By now, most competitive organizations have a significant investment in BI; however, much of the information provided by BI tools is data- and technology-oriented, often consisting of masses of low-level data that is difficult to use for business analysis purposes. Instead, business people are interested in having their data interpreted and analyzed in terms of strategic objectives, business models, processes, markets, trends and risks.

R. Meersman et al. (Eds.): OTM 2012, Part II, LNCS 7566, pp. 700–717, 2012.

Several existing modeling techniques (e.g., the Business Motivation Model (BMM) [1], Strategy Maps (SM) [2], Balanced Scorecards (BSC) [3], SWOT Analysis [4], and Goal Modeling (GM) [5], [6]) offer concepts that are potentially useful in bridging the business landscape to BI data (e.g., goals, performance measures, initiatives, threats, opportunities). The above languages offer many concepts useful for BI, but, except for GM, are not state-of-the art with respect to other conceptual modeling languages (e.g., [6], [7]), are only described informally, and do not support formal reasoning. GM, on the other hand, is missing many useful business concepts.

To bridge the gap between the data and business realms for BI purposes, our previous work [8–11] proposed the Business Intelligence Model (BIM), a modeling language for the business world, making use of familiar concepts from existing languages (goals, processes, situations, influences, and indicators) to enable decision making during strategic business analysis. The previous proposals were mainly informal, describing how BIM models can be mapped to goal models or influence diagrams to do reasoning. In this paper we consolidate and rationalize BIM, offering both an abstract syntax and a formal semantics via translation to Description Logic (DL), and then utilize the reasoning capabilities inherent in the target DL.

Our consolidated BIM distills essential concepts from prior proposals, such as situations, goals, indicators, and tasks, and relationships (e.g., *refines* and *influences*); it allows the introduction of more specialized concepts (e.g., vision, mission, initiative), as needed, via orthogonal meta-properties. The new syntax and semantics of BIM are more uniform, thanks to the introduction of an attribute that represents *evidence* for or against the satisfaction (occurrence, performance, etc.) of *all* things, thus allowing for "what if?" scenario analysis on all aspects. We introduce the novel concept of goal *pursuit*, used in BIM analysis.

The semantics of BIM are provided by translation to a subset of the OWL2 DL [7], including a schema for translating specific BIM models into DL axioms. This precisely and formally captures the constraints on valid BIM models, the interaction of *evidence/pursuit* with *refines/influences*, and the use of a fixed set of meta-attributes to specify specialized concepts. DL reasoners can be used to detect inconsistencies in BIM schemas, and perform "what if" analysis. DLs provide additional benefits such as allowing more detailed conceptual modeling of entities, tasks, etc.; representing incomplete information; defining concepts and automatically organizing them in subclass hierarchies. In addition, translation to OWL allows publishing of generic BIM models as ontologies on the semantic web. We briefly compare BIM with other existing business languages with respect to concept coverage.

The translation of BIM to DL raised issues of interest to the DL community. A companion workshop paper provides a summary of the BIM concepts and DL translations provided in Sections 2 to 4; describes how parameterized concepts and rules can be used to simplify the encoding, reducing replication; and describes how to deal with meta-attributes in a general way [12].

The rest of the paper is structured as follows. Section 2 describes the core of BIM and its translation to OWL, using a running example. Section 3 describes reasoning in BIM, including reasoning over evidence and pursuit. Use of meta-properties in BIM

to capture many other BI language notions is described in Section 4. Section 5 considers related work while Section 6 provides conclusions and outlines future work.

2 The Core of the Business Intelligence Model (BIM)

In this section, we introduce the core concepts and relationships of BIM, by first describing each informally, illustrating them via a realistic running example, and then providing their formal semantics by giving axioms in the OWL2 DL.[1]

2.1 Running Example: Credit Card Industry Analysis

We have used realistic sources to create an illustrative BIM schema focusing on strategic industry analysis for a generic company providing credit and charge cards. To create our BIM model, we have incorporated elements and relationships from a DataMonitor profile of the Global Credit and Charge Cards Industry [13]. We show an excerpt from the developed model in Fig. 1. Although the example may seem complex at this point, it will be gradually explained and used for illustration in the rest of the paper.

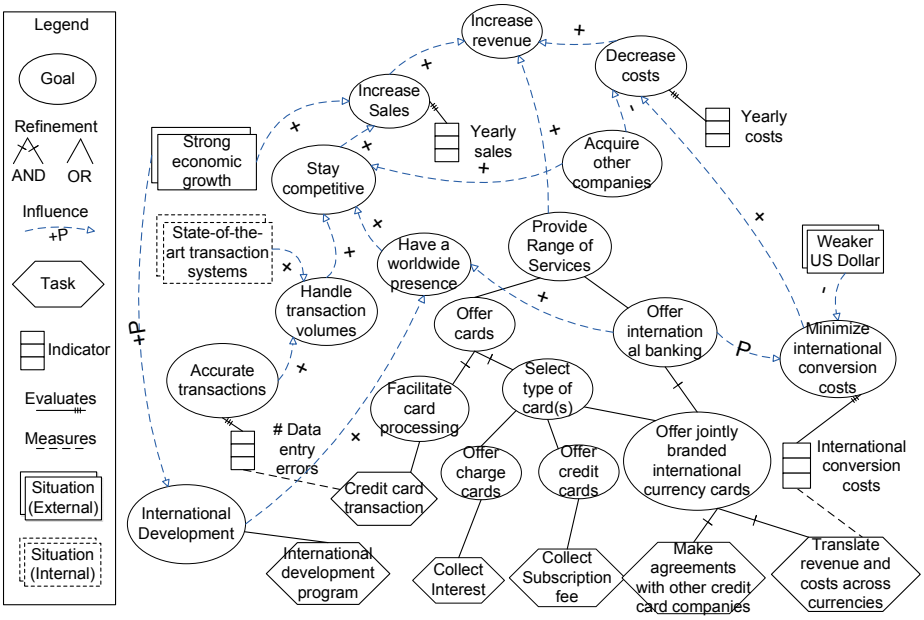

Fig. 1. BIM for a Generic Credit Card Company

[1] Since OWL2 is known to correspond to a subset of First Order Logic (FOL), this is equivalent to providing a semantics via FOL axiomatization. The advantage is that OWL2 has decidable reasoners.

2.2 BIM Concepts and Relationships

As usual in object-centered representations, individual things are grouped into generic classes, may have attributes, and are connected by binary relationships. BIM focuses on four types of things**:** ***Entities, Tasks, Situations*** and ***Indicators***, which are subclasses of (***BIM_)Thing*****.** Situations are specialized into ***Goals*** and ***Organizational situations***. BIM focuses on four relationships between things**:** ***influences*****,** ***refines***, ***evaluates*** and ***measures***. Fig. 2 shows a model introducing the significant terms of BIM. In the remainder of this section we present these notions, in each case describing and illustrating them, and then offering DL axioms for precise semantics.

2.3 BIM Things and Their Semantics

Thing. Rather than asserting the binary existence, execution, or satisfaction of each Thing, BIM considers multiple sources and degrees of evidence, either for or against each. This is similar to the treatment of goal satisfaction in goal models [14–16]. In BIM, these values are accumulated in a set-valued attribute called ***evidence***, whose values may be set externally by the modeler (especially when doing "What if" scenario analysis), or constrained via relationships to other model elements. We use a set of evidence values, instead of a single value, to allow for the collection of possibly conflicting evidences from multiple sources. The question "Evidence for…?" is answered depending on the specific type of thing: evidence for the *satisfaction* of goals, the *occurrence* of situations, the *performance* of indicators, the *execution* of tasks, and the *existence* of entities.

We represent the evidence for a particular thing using a qualitative scale similar to the satisfaction/denial scale used in GM [14–16]. With this scale, we capture strong/weak evidence for/against a thing, using the abbreviations SF, WF, WA, and SA. The value of *evidence* is a subset of these labels, allowing BIM to capture multiple sources of evidence. Naturally, strong evidence for (against) a thing implies weak evidence for (against) it.

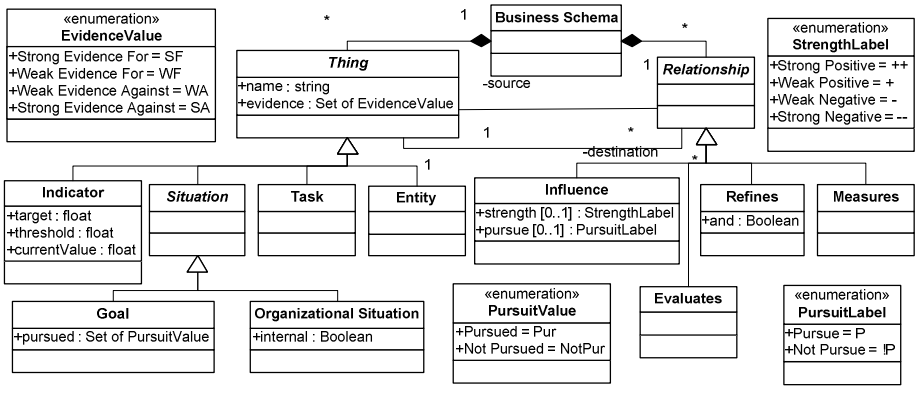

Fig. 2. BIM Concepts and Relationships

To formalize evidence in OWL, we introduce primitive class (BIM_)Thing, and represent an attribute as an ordinary OWL property, with its domain and range properly constrained, expressed in the following axiom[2]

Property: evidence **Domain**: Thing **Range**: {SF,WF,WA,SA}

We will find it convenient to create defined subclasses of Thing, such as

Class: SFThing **EquivalentTo:** Thing **and** (evidence **value** SF)

where **and** denotes concept intersection, and (R **value** V) denotes objects with V as one of the values of property R. SFThing allows us to consider class membership rather than *evidence* attribute contents. We then capture strong evidence implying weak by the axioms

(SFThing **SubClassOf**: WFThing) (SAThing **SubClassOf**: WAThing)

Following the standard translation of UML diagrams into DL (see [17]) we would get axioms describing the BIM hierarchy of things, such as

Class: Situation **SubClassOf**: Thing
Class: Goal **SubClassOf**: Situation

as well as disjointness axioms for sibling classes in the hierarchy

DisjointClasses: Situation Indicator Task Entity
DisjointClasses: Goal OrganizationalSituation

Situation. In order to effectively monitor the status of business objectives, we must take into account situations that may affect such objectives. "Situation", adopted from SWOT analysis [4], is used as an atomic proposition in BIM. However, given the ontology of entities and general relationships available in BIM, situations could be expanded to have associated partial world-state descriptions, as suggested in [18]. Since we are interested in strategic business models, we focus only on situations which have an impact on business objectives, *organizational situations*, inspired by [19]. For example, both "Heavy rainfall" and "Weaker US Dollar" are situations, describing the state of things in the world, but likely only the latter is relevant to our example organization.

It is important to note that in creating a business schema, we assume that the schema is drawn from the point of view of a particular organization. This allows us to omit arguments representing the viewpoint of particular relationships/attributes (e.g., *pursuit*); BIM has chosen to be less expressive in this regard in order to make modeling easier. As an analogy, contrast building an UML model of library operations where *lending(byLibrary, ofMaterial, toBorrower, onDate)* is 4-place, vs. the model for a single library, where the first argument is implicit and is omitted, but prevents us from talking about inter-library loans. Future work can consider ways to add multiple viewpoints or agents to the language.

The use of a single viewpoint allows organizational situations to be classified as either internal or external. In our running example, "Weaker US Dollar" is a situation

[2] We introduce notation by example, using the Manchester Syntax [7].

that is external from the viewpoint of the organization, while "State-of-the-art transaction system" is internal. This aspect is captured in BIM using the boolean attribute ***isInternal***.

For greater expressiveness, we can represent *levels of occurrence* of situations, using a specialization *occurrence* of the *evidence* attribute. For example, some of the propositions comprising a situation may occur, while others may not.

Representing in OWL the constraint that *isInternal* is a functional Boolean attribute is straightforward, and such aspects will not be repeated below. We can specify in OWL part of the notion that *occurrence* is the restriction of *evidence* to the domain of *Situation* via the axioms

Property: occurence **Domain**: Situation **SubpropertyOf:** evidence

Unfortunately, this does not capture the fact that every case of *evidence* applied to a *Situation* is an *occurrence*. (i.e., the logical implication $evidence(x,y) \wedge Situation(x) \rightarrow occurrence(x,y)$). Our translation could be expanded using an OWL 2 encoding provided in [20].

Goal. Goals have a long history as part of requirements modeling (e.g., [5], [6]), and are also included as part of several business analysis techniques such as BMM [1], SM [2] and BSC [3]. We include this concept in BIM, thinking of it as an intentional situation that is desired by the (viewpoint) organization. In our example, the credit card company wants to "Increase revenue" and "Offer international banking".

In addition to the *evidence* attribute, called *satisfied* for goals, goals also have a ***pursuit*** attribute, used to indicate whether or not a goal instance is actively being pursued by the viewpoint organization. This set-valued attribute is assigned values from {*Pur* (is pursued), *NonPur* (is not pursued)}. The pursuit of goals can be set by modelers, or, as we shall see later, can be influenced by the pursuit or satisfaction of other situations (including goals). For this reason, *pursuit*, like *evidence*, is set-valued. If a goal is both satisfied and pursued (and not simultaneously denied or not pursued), it is said to be ***Fulfilled***.

The OWL axiomatization of goal semantics therefore starts with

Property: satisfied **Domain**: Goal **SubpropertyOf:** evidence

Property: pursuit **Domain**: Goal **Range**: {Pur, NotPur}

Class: Fulfilled **EquivalentTo**: Goal **and** (evidence **value** SF) **and not** (evidence **value** WA) **and** (pursuit **only** {Pur})

Indicator. In BIM, indicators constitute a conceptual bridge connecting a business schema to data found in a variety of data sources. Conceptually, an indicator is linked to a situation, which we say it evaluates (expressing *why* we are interested in the indicator), and a task, whose operation it measures (expressing what the indicator does). For example, in Fig. 1 the indicator "# (of) data entry errors" evaluates the goal "To have accurate transactions" while measuring an aspect of the "Credit card transaction" task.

We collect strong/weak evidence for or against the *performance* of indicators. In this case, by associating so-called *target* and *threshold* values to an indicator, we can convert current numerical values to performance levels. Indicators can be composite,

having values computed via the values of other indicators, or can be atomic, having values computed directly from data sources. More details on indicators in BIM, including unit conversion, normalization, and reasoning can be found in [9], [11].

The formalization of the structural relationships between *Indicator*, the *Situation* it *evaluates*, and the *Task* it *measures* is similar to that of previously defined relationships and is omitted. The much more interesting aspects of indicator computations involve arithmetic, and are therefore outside the competence of OWL[3].

Entity. Entities are the "bread and butter" of all Information System models, and we have nothing new to add except that BIM can represent *evidence for/against the existence* of individual entities using the evidence attribute. BIM allows users to represent entities relevant to the business schema, for example, a Credit_Card, User_Account, etc., by creating, as necessary, appropriate domain-specific sub-classes of *Entity* and *Relationship*.

Task. A task in BIM represents a process or a set of actions. Tasks may be atomic, typically representing a simple action, or non-atomic, composed of sub-tasks (see Section 2.4 on *refines*). In BIM, we can collect *evidence* for/against *the execution* of tasks.

The ontology and modeling of entities and processes/events has been well-studied, and we point to Unified Enterprise Modelling Ontology (UEMO) [21] for (among others) a rich OWL axiomatization of commonly useful constructs.

2.4 BIM Relationships and Their Semantics

In this section, we provide the basic semantics of BIM relationships. Further semantics are provided in Section 3, describing reasoning concerning evidence and pursuit using relationships between BIM things.

Influences. The *influences* relationship is used to represent the transmission of (un)favorable effects on situations, including goals and organizational situations. For example, the goal "Acquire other companies" has a partial negative effect on "Decrease Costs". Borrowing from goal model notation [14–16], there are four kinds of *influences* links: a ++/+ (make/help) influences link represents strong/partial positive effect on *evidence* from the source to the destination situation, and a --/- (break/hurt) link represents strong/partial negative effect. We defer to Section 3 the detailed specification of how values of *evidence* are propagated along various *influences* links.

In addition to affecting *evidence* of things, *influences* links can be used to affect the *pursuit* of goals. For this purpose, additional optional *influences* label annotations are used: P and !P, for pursuit and the denial of pursuit, respectively. For example, "Strong economic growth" gives partial positive *evidence* to "International Development" but also transmits the *pursuit* of the source goal to the destination. *Influences* interacts in subtle ways with *evidence* and *pursuit*, as described in Section 3.

Through the use of influence relationships, internal or external situations may be declared favorable (strength/opportunity) or unfavorable (weakness/threat) to

[3] Ianone & Rector [28] provide a suggestion on describing properties defined by mathematical function as *annotations*, but since annotations are just comments, they are not appropriate for defining semantics.

objectives within an organization, or to other situations. For example, "Weaker US Dollar" is a threat with respect to "Minimize international conversion costs", but could be an opportunity with respect to a different goal, for example, "Decrease Costs". Such analyses are essential to business analysis and BI.

Basic Semantics of Influences. As with all relationships, we introduce axioms for the domain and range of the property representing it in the translation to DL, and the name of the inverse property (allowing us to "traverse edges backward"):

Property: influences **Domain**: Situation **Range**: Situation **InverseOf**: infBy

The different kinds of labels on *influences* are represented as sub-properties of influences and infBy, with the labels as prefixes separated by an underscore (e.g., +_InfBy, −−_Influences). Because the semantics of the pursuit label interacts with that of the strength label, we introduce a taxonomy of OWL properties, starting with leafs like −−P_InfBy, +!P_InfBy, etc. These are then made subproperties of more general infBy properties, like InfByP, infBy!P, infBy−− and infBy+; in turn, infBy++ and infBy+ are subproperties of infByPositively. This allows us to state some axioms once, for a higher property, rather than repeat it for each subproperty. The following is a sample of the OWL axioms needed for this

Property: influences **Domain**: Situation **Range**: Situation **InverseOf**: infBy
Property: −−P_infBy **InverseOf**: --P_influences **SubpropertyOf**: infByP, infBy--
Property: infByP **InverseOf**: influencesP **SubpropertyOf**: infBy
Property: infBy−− **InverseOf**: influences−− **SubpropertyOf:** infByNegatively

DLs are particularly well suited to describing BI notions such *Threat:*

Class: Threat **EquivalentTo:** (OrganizationalSituation **and** (isInternal **value** False) **and** (influencesNegatively **some** Situation)

(where generally (R **some** C) denotes objects that have at least one R-value in class C) because reasoners automatically recognize individuals meeting the conditions on the right hand side, and classify them as instances of *Threat.*

Refines. Refinement of a thing provides direct *evidence* for/against it through *evidence* for/against the refining things. For example, "Collect Interest" is a refinement of "Offer charge cards", so strong/weak evidence for/against the former implies strong/weak evidence for/against of the latter. Refinement is the only way things other than situations (Indicators, Tasks, Entities) can accumulate evidence.

In goal modeling, refinement allows AND/OR goal decomposition into sub-goals. In BIM, we expand the concept of refinement: any concept can always be refined into other, usually more specific, concepts of the same type. For example, entities can be refined into sub-entities (usually parts), tasks into sub-tasks, etc. Goals are one exception, since they can also be refined into tasks, representing the so-called "means-ends" relationships as in [5]. For example, the goal "Facilitate card processing" is refined into the task "Credit card transaction".

Refinements are by default interpreted disjunctively (ORed), e.g., the goals "Offer cards" and "Offer international banking" refine "Provide range of financial products and

services" independently. Refinement relationships can be indicated as explicitly conjoined (ANDed) by adding a perpendicular hash mark: e.g., both "Facilitate card processing" and "Select type of card(s)" are required to satisfy the "Offer cards" goal. On any particular concepts, we want all refinements to be only ANDed or ORed.

Since the refiner is supposed to be more specific than the refined, it does not make sense to have circularities in *refines* relationships. Precise semantics for evidence propagation via *refines* will be given in Section 3.

Basic Semantics of Refines. Refines is defined as a property, with the inverse refinedBy. Axioms restrict the types of the refiner and refined to be the same type of Thing, with the exception of Goals, which can be refined into Tasks. These axioms use the OWL constructor **only**, where (R **only** C) denotes objects that are only related by property R to individuals in C:

Property: refines **Inverse**: refinedBy
Class: Situation **SubClassOf:** (refines **only** Situation)
Class: (refines **some** Situation) **SubClassOf:** Situation.
(similar axioms for all Thing sub-classes except Task and Goal)

Class: Goal **SubClassOf:** (refines **only** Goal)
Class: (refines **some** Goal) **SubClassOf:** (Goal **or** Task)

Since on any particular node, we want all refinements to be ANDed or ORed, we add a subclass of Thing, called AND_Thing, and leave refinement relations themselves unmarked. The appropriate DL specifications for this are

Class: AND_Thing **SubClassOf**: Thing
Class: OR_Thing **EquivalentTo:** Thing **and not** AND_Thing

The idea is that the axioms in Section 3 will treat the *refines* links as ANDed or ORed depending on whether their destination is an AND-node or an OR-node.

To capture the constraint on the non-circularity of *refines*, we introduce a transitive property refinesClosure, which is asserted to be anti-symmetric and to contain the refines property:

Property: refinesClosure **Domain**: Thing **Range**: Thing
Characteristics: Asymmetric, Transitive
Property: refines **subPropertyOf:** refinesClosure

Evaluates. Indicators *evaluate* situations (and hence goals). The current value of an indicator can be mapped to an indicator performance level (*evidence*), which then evaluates evidence for/against the occurrence of a situation. For example, "International conversion costs" evaluates the goal "Minimize international conversion costs".

Measures. Indicators can be associated with a particular task via the *measures* relationship. The measure link is intended to be an abstract relationship that associates a particular indicator with the task that it measures. For example, "International conversion costs" is associated with data produced by the "Translate revenue and costs across currencies" task.

2.5 Representing Specific BIMs in DL

A BIM graphical model (e.g., Fig.1) can be translated into DL axioms in a manner that respects the following intuition of GM users: given a set of top-level goals, for every instance of this set there are instances of all of the rest of the nodes in the graph, including their connections. This results in an isomorphic copy of the (concept level) graph. In our example, for every instance of "Increase Sales" (e.g., for June, 2012) there is a corresponding instance of the sub-graph. If we had more than one top-level goal instantiated for the same period (e.g., "Increase Sales" and "Decrease Costs" for June 2012) these instantiated goals would share a connected instance of the sub-graph. Two, separate instances of "Increase Sales" (for June 2012 and July 2012) would produce two, separate, instantiated copies of the graph. Essentially, this means that we need to provide axioms to describe the graph for each desired instantiation, taking into account the Open World Semantics of DLs (see step 4). The following steps create an instance of a specific BIM in DL:

1. For every node create a concept that is a subclass of the concept representing the node's type, and describing whether its refinement, if any, is of AND or OR kind. For example, node "Offer Cards", which is an AND goal, would generate:

Class: OfferCards **SubClassOf:** Goal **and** AND_Thing

2. Represent that the nodes are distinct, by adding disjointness axioms between all the concepts introduced in Step 1.

3. Represent all the edges/relationships and their inverses using axioms illustrated for the *+_influences* edge from StayCompetitive to IncreaseSales:

Class: StayCompetitive **SubClassOf**: (+_influences **some** IncreaseSales)
Class: IncreaseSales **SubClassOf**: (+_infBy **some** StayCompetitive)

4. Finally, add cardinality constraints for every edge type to express that there are no other edges applicable to that node (this support **only** inferences in Section 3). For example,

Class: OfferCards **SubClassOf:** (refinedBy **exactly** 2) **and** (refines **exactly** 1) **and** (influences **exactly** 0) **and** (infBy **exactly** 0)

3 Evidence, Pursuit and Reasoning in BIM

This section describes reasoning with BIM, including rules for evidence and pursuit, "What if?" and "Is this possible?" analysis, consistency checks on schemas, and other benefits of using DLs in our axiomatization.

3.1 Effect of Relationships on Evidence and Pursuit

In this section we provide the precise rules for relating both *evidence* and *pursuit* values in the presence of various kinds of relationships, especially *refines* and *influences*. (Recall that each BIM thing has an *Evidence* property, whose values are a subset of {SF, WF, WA, SA} and *Pursuit* property, whose values are a subset of {*Pur*, *NonPur*}.)

Measures. Does not affect *evidence* and *pursuit*.

Evaluates. The *performance* value of the source indicator is the *evidence* value for the target. In DL, this requires separate axioms for each possible value of *performance,* so we need four axioms like

(evaluatedBy **some** SFThing) **SubClassOf** SFThing

Since such repetitious groups of axioms will appear frequently we turn to simple axiom schemas:

(evaluatedBy **some** Thing<V>) **SubClassOf** Thing<V> for V=SF,WF,WA,SA

Pursuit is not affected.

Refines. We use the rules for combining evidence on AND and OR refinements from [14], [15]. Positive *evidence* values from the sources are propagated to the target according to its node kind: on an OR node, it is enough to have one refiner with V=SF,WF to get V; on an AND node, all refiners must have V:

OR_Thing **and** (refinedBy **some** Thing<V>) **SubClassOf**: Thing<V>
AND_Thing **and** (refinedBy **only** Thing<V>) **SubClassOf**: Thing<V>

For negative evidence, the converse holds: for V=SA,WA

OR_Thing **and** (refinedBy **only** Thing<V>) **SubClassOf**: Thing<V>
AND_Thing **and** (refinedBy **some** Thing<V>) **SubClassOf**: Thing<V>

Note that we have essentially replaced propositional sub-goal conjunction/disjunction in previous goal model approaches ([14], [15]) by DL universal/existential quantification over *refinedBy* parts, better allowing for expansions to any additional type of refinement (e.g., XOR, at least-K).

Influences on evidence. The *evidence* values of the target are related to that of the source depending on the influence label. The rules from [14], [15] are shown in Table 1. One can then simply write 16 axioms, including

(infBy++ **some** SFThing) **SubClassOf** SFThing
(infBy−− **some** SFThing) **SubClassOf** SAThing

Influences with pursuit. In case both the source and destination node are goals (i.e., have *Pursuit* attributes), the effect of *influence* links with P or !P is summarized in Table 2. If an influence edge does not contain a label concerning pursuit or evidence, the pursuit or evidence value of the destination, respectively, remain unchanged. After creating two abbreviations PurGoal and NotPurGoal

Class: PurGoal **EquivalentTo:** Goal **and** (pursuit **value** Pur)
Class: NotPurGoal **EquivalentTo:** Goal **and** (pursuit **value** NonPur)

we provide 4 axioms for the entries in Table 2, including the following for the first row

(infByP **some** PurGoal) **SubClassOf** PurGoal
(infBy!P **some** PurGoal) **SubClassOf** NotPurGoal

Influences without pursuit. Recall that only goals have a *Pursuit* attribute, so we may have cases where one or the other end of the edge is missing it. If the source does not have a *Pursuit* attribute, e.g. a situation influencing a goal, the satisfaction polarity of the source determines the pursuit of the target in P and !P influence types. For example, if a source situation strongly/weakly occurs and the influence link is P, then the destination goal is partially satisfied and pursued. If the label is –!P, the goal is partially denied and not pursued. These are 2 of the 4 axioms:

(InfByP **some** (**not** Goal **and** (SFThing **or** WFThing)) **SubClassOf** PurGoal
(InfBy!P **some** (**not** Goal **and** (SFThing **or** WFThing)) **SubClassOf** NotPurGoal

If the source has a *Pursuit* attribute but the destination does not, e.g. a goal influencing a situation, then the influence of the source *Evidence* on the destination *Evidence* only occurs when the goal is pursued, in case P is on the label, or not pursued in the case of !P. For example, if the goal is satisfied and pursued, and the label is +P, the situation partially occurs. If the goal is satisfied and not pursued, the situation has no incoming evidence from that goal.

Table 1. Evidence propagation depending on influence label (destination Evidence value in grey)

	Link Label Contains			
Source Evidence Set Contains	**++**	**+**	**-**	**--**
SF	SF	WF	WA	SA
WF	WF	WF	WA	WA
WA	WA	WA	WF	WF
SA	SA	WA	WF	SF

Table 2. Pursuit value propagation depending on influence label (destination Pursuit value in grey)

	Link Label Contains	
Source Pursuit Set Contains	**P**	**!P**
Pur	Pur	NonPur
NonPur	NonPur	Pur

3.2 Reasoning with BIM Models

"What if?" Scenarios. The above axioms allows us to explore "What if" scenarios, such as *"How is the evidence for/against any particular model element affected if our organization "offers cards" but does not "offer international banking"*. To explore this scenario, one approach would add the following axioms

Class: OfferCards **SubClassOf:** SF_Thing
Class: OfferInternationBanking **SubClassOf:** SA_Thing

and then check whether ProvideRangeOfServices is classified as a subclass of SF_, SA_, WF_ and WA_Goal, respectively. One can similarly check the classification of IncreaseRevenue to see the effect of these assumptions on it [4].

[4] An alternative approach, more conducive to those familiar with DLs, is to create an ABox with individual instances to represent a particular situation and then reason only with the classification of individuals.

While the above features can be obtained by existing goal-model reasoners ([14–16]), the DL formulation allows exploration of less precise scenarios ("What if we offer cards ***or*** international banking") by simply adding instead

(OfferCards **or** OfferInternationBanking) **SubClassOf:** SF_Thing

"What if?" scenarios can also involve *pursuit* values: e.g., we could investigate what happens if "Offer International Banking" is not pursued

OfferInternationBanking **SubClassOf:** NotPurGoal

Furthermore, at this point it is easy to extend BIM to allow intermediate forms of refinement between AND and OR, such as "if at least two of the refiners for class G are SF, then make G be SF"; these are translated into qualified number restrictions in OWL2 DL. Another example is an XOR node, which would replace (**some** refines SFThing) of OR_Thing by (**exactly** 1 refines SFThing). These allow the implementation of propagation semantics alternative to [14], [15], such as those described in [16].

"Is it possible?" Scenarios. One might also want to answer a different question: *"Is it possible to fully satisfy ProvideRangeOfServices?"* At the simplest level, this is just adding the axiom

ProvideRangeOfServices **SubClassOf:** SF_Thing

and waiting for the reasoner to signal non-emptiness of this concept. We observe that in fact the problem is one of abduction: what minimum sets of leaf nodes must be assumed in order to achieve a particular degree of evidence for some other node. While the general problem of DL abduction is unsolved, the recent work of Du et al. [22] may be applicable, as it handles OWL2 and works by abducing atomic concepts (BIM nodes). It should be noted that this work does not handle "nominals", such as SF, in our axioms. However, we use nominals in our axioms only for convenience. Since these objects do not enter into relationships of their own, they could be replaced by data values (string `"SF"`) or disjoint atomic concepts[5]. Although we leave the integration of Du et al.'s approach into our reasoning tools to future work, we note that unlike existing approaches to backward reasoning ([15]), the integration would not require us to translate *out* from the DL representation.

Satisfaction and Classification. OWL2 and many other DLs have decidable satisfiability testing algorithms, which are at the heart of most DL implementations today. These systems allow one to verify that no class declared is "useless", in the sense that it must always be empty/inconsistent. We have tested our axiomatization of BIM for consistency. BIM users can also test the encoding of specific BIM models, such as Fig. 1, for errors in using the language constructs (e.g. influencing a Task, an Entity being pursued). Using standard DL debugging tools (pinpointing), one can find which axioms are conflicting.

Moreover, in the presence of *definitions* (necessary and sufficient conditions), DL reasoners provide automatic classification of named classes in the IsA hierarchy, organizing our model — a very important feature for large models. For example, we

[5] This is a well-known DL trick that works unless one wants to have proper cardinality constraints.

might want to extend BIM itself with the notion of AmbivalentThing — one that supports some goals but opposes others. Formally, this would be defined as

Class: AmbivalentThing **EquivalentTo:** (influencesPositively **some** Goal)
and (influencesNegatively **some** Goal)

Then, AcquireOtherCompanies from Fig. 1 would be automatically made a subclass of AmbivalentThing.

4 BIM Meta-properties

Rather than simply make BIM the union of many concepts found in other business analysis languages (e.g., Vision, Mission, Strategy (BMM), Softgoal, Hardgoal (GM), Initiative (BSC)), we performed an ontological analysis of their underlying meaning. The result is a set of six meta-properties: *duration* (long-term/short-term), *likelihood of fulfillment* (high/low), *nature of definition* (formal/informal), *scope* (broad/narrow), *number of instances* (many/few), and *perspective* from BSC [3] (financial/ customer/ internal/ learning and growth). Note that values of meta-properties at the class level do not constrain class instances, but only say something about the nature of instances, that they are likely or generally conform to the informal notions expressed by the meta-properties and their values. For example, instances of the "Stay competitive" goal are likely to have long duration, broad scope, etc., but are not restricted from behaving differently.

New, more specialized, BIM subconcepts can now be obtained using values for these meta-properties. For example, the BMM concept of a **Vision** is a "goal with a long duration, broad scope, low chance of fulfillment, informal definition, and few instances". Examples of Visions from our credit card organization could include "Stay competitive" or "Have a worldwide presence". Similarly, we could use meta-properties to specify other things, such as "International development program", "Make agreements with other credit card companies", "Credit card transaction", as a **Mission** (BMM), **Initiative** (BSC), and **Business_Process** (BMM), respectively. Not all meta-properties must take on specific values in order to express a more specialized BIM concept. For example, **Soft/Hard Goals** from GM can just be goals with an informal/formal *definition*.

The representation of meta-properties in ordinary DLs is known to be problematic, especially in our case, where we want the meta-properties to behave so that restricting their possible values results in subclasses. However, this is exactly what one would get if the meta-properties were treated as ordinary properties. So we propose axioms like

Property: duration **Domain**: Thing **Range**: {long_term, short_term}
Characteristics: Functional

and then define classes such as

Class: Vision **EquivalentTo:** Goal **and** (duration **value** long-term) **and … and** (nature_of_definition **value** informal).
Class: SoftGoal **EquivalentTo:** Goal **and** (nature_of_definition **value** informal).

DL reasoning would then automatically classify Vision as a subclass of SoftGoal.

One slight difficulty with the above approach is that this conceptual model associates with individual goal *g* belonging to Vision, (which we might be pursuing tomorrow) the property *number_of_instances*, with value *few,* which does not make sense intuitively.

Similar to the approach in [23], the DL translation could be expanded with constraints over meta-properties, checking the well-formedness of the BIM Schema. For example, concepts classified as SoftGoals should not refine concepts classified as HardGoals. Future work could be directed towards the design of such constraints.

5 Related Work

The appropriateness of BIM can be assessed by considering its ability to express many of the concepts used in existing business modeling languages (e.g., BMM, SWOT, BSC, SM, GM). We have already described the coverage of several concepts in existing languages via the introduction of BIM core concepts and through the use of meta-properties. We summarize further coverage of concepts in Table 3.

Table 3. Coverage of Concepts in Existing Business Languages using BIM Concepts

BIM Concept/ Relationship	Covers Concept (Language), possibly using metaproperties
Goal	End, Vision, Objective, Goal (BMM); Soft/Hardgoal (GM), Objective (SWOT); Mission, Vision, Goal/Objective (BSC/SM);
Task	Means, Course of action, Mission, Strategy, Tactic, Business process (BMM); Task (GM); Strategy, Initiative (BSC/SM);
Situation	Internal/External Influencer (BMM), Issue (SWOT)
Situation + influence	Strength, Weakness, Opportunity, Threat (SWOT)
Indicator	Metric (BMM), Measure (BSC/SM)
Indicator target	Target (SWOT), Target (BSC/SM)
AND/OR Refinement	AND/OR Decomposition (GM); aggregation (UML)
Influence	Contribution (GM)

Previous work has introduced early versions of BIM concepts, relationships, and syntax [8–11]. However, these earlier works provided language descriptions that were often inconsistent, described concepts and relationships without providing formal semantics, relied on mappings to several different existing languages/systems to support various forms of reasoning, and generally focused on the use of BIM in strategic analysis. This paper provides a consistent, more concise description of BIM, which is made up orthogonal notions such as *evidence* and *refines* that are uniformly applicable to all things. The translation to DL provides a formal semantics, and other benefits. Reasoning capabilities, similar to the goal models reasoning in [14–16], are now inherent in the language design, and derive from its semantics. The capability to express and reason about the *pursuit* of goals, including concurrent consideration of *evidence*, has been added to the language. On the other hand, previous papers on BIM have given more attention to indicators, numeric/probabilistic evaluation of evidence, and have presented extensive case studies [24].

Prior work formalizing reasoning about UML class diagrams by translation to DL was an initial foundation to this work [17]. We go beyond this approach by including business-level concepts not included in UML, such as goal and situation.

The UEML (Unified Enterprise Modelling Language) project has led to, among others, the creation of an ontology UEMO (Unified Enterprise Modelling Ontology) [21], which is philosophically founded in Bunge's work [25] and has precise semantics. The purpose of UEML/UEMO is to help align modeling languages (for which a special methodology is provided). UEMO semantics are also presented as a set of OWL axioms (plus Semantic Web Rule Language rules), which can be used to describe and compare formally modeling constructs in languages, and possibly facilitate interoperation [26]. Interesting future work would be to try to map the constructs of BIM into the UEMO ontology, thereby acquiring deeper philosophical foundations and seeing to what extent the axiomatization presented here falls out of the process.

The notion of meta-properties such as rigidity, identity, unity, and dependence was introduced as part of the OntoClean approach [23]. Future work could add constraints to BIM metaproperties similar to the constraints applied in OntoClean.

When designing the semantics for reasoning with evidence, we encoded the rules provided for goal model satisfaction in [14], [15]. This approach formalizes goal models using propositional logic, relying on a SAT solver for both "forward" and "backward" reasoning. Our formalization of BIM provides many similar benefits, by virtue of being based on formal logic. However, as our formalization uses DL and thus first-order logic, we provide benefits beyond the approach in [14], [15]. Specifically, in addition to covering additional concepts and relationships (situation, evaluates), our approach allows for reasoning by classification via the easy addition of new concepts (e.g., Supported_Goal **EquivalentTo**: SFGoal **or** WFGoal), allows for relationship and attribute domain/range checking using satisfiability, and considers the notion of goal pursuit along with evidence. The introduction of the DL formalism also better facilitates future reasoning extensions, including reasoning at the instance level via the easy addition of class instances, support for alternative approaches of combining multiple sources of evidence (e.g., [16]) via use of cardinality constraints (e.g., at least two influences are SF), and reasoning over meta-property classification via the introduction of property constraints.

6 Conclusions and Future Work

This work has consolidated existing proposals for BIM, providing consistent syntax and formal semantics using a translation to DL. Specifically, this paper offers the following contributions:

1) We provide and illustrate a consolidated version of BIM, **a)** showing that BIM covers many previous BI language features, without being a "kitchen sink" of ideas; **b)** instead, there are aspects that make the language uniform (e.g., evidence, influence and refinement are applicable to all things) and easier to learn/use (e.g., use of meta-attributes and DL defined concepts to create new BI notions on demand, rather than burdening the language design with lots of terminology); **c)** cover new ground with respect to previous proposals by considering *pursuit* of goals.

2) We provide a formal semantics for BIM by translation into OWL axioms for the basic primitive notions in Fig.2, forming an upper ontology that precisely describes the semantics of the language constructs. **b)** Specific BIM models can also be translated according to a schema we offer into extensions of the upper ontology. Generic BIM domain models (such as Fig. 1) can then be published as OWL ontologies

on the Semantic Web, where specific organizations (e.g., a bank) may then use and augment it for their own purposes. Conversely, OWL ontologies can be imported for modeling entities, etc., for the bank. **c)** The translation into OWL provides opportunities for extending the language with useful new constructs (e.g., XOR) and terms by directly defining them as OWL concepts, and makes use of DL's abilities to describe and reason with partial/incomplete information [27]. **d)** OWL offers reasoning support for inconsistency detection, "what if" scenario evaluation, ability to define new model concepts and classify them, all using existing OWL reasoners.

We have partially evaluated our proposal by encoding the BIM to OWL2 translation, including our Fig. 1 example, in the Protégé tool. The resulting ontology is available online (www.cs.utoronto.ca/~jenhork/BIM/BIMLanguage.owl). As a complementary activity for defining the semantics of BIM primitives, we have categorized them with respect to the DOLCE+ upper ontology [19][6].

Our approach is not without limitations. Previous work on BIM introduced quantitative as well as probabilistic reasoning using values for indicators, evidence [9]. Such reasoning cannot be captured in OWL2, as arithmetic reasoning is undecidable.

Future work should introduce appropriate user interfaces and APIs, hiding the formal details. Although we have provided an example visual syntax for the language, future work could consider the design of an effective concrete syntax. Language extensions could consider the representation of multiple viewpoints, including external goals and agents or actors as in goal modeling (e.g., [5]).

References

[1] Object Management Group, Business Process Modeling Notation (BPMN) Version 2.0 (2009), http://www.omg.org/spec/BPMN/2.0

[2] Kaplan, R.S., Norton, D.P.: Strategy Maps: Converting Intangible Assets into Tangible Outcomes, pp. 1–4. Harvard Business School Press (September 2004)

[3] Kaplan, R.S., Norton, D.P.: Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press (1996)

[4] Dealtry, T.R.: Dynamic SWOT Analysis. Dynamic SWOT Associates (1994)

[5] Yu, E.: Towards modelling and reasoning support for early-phase requirements engineering. In: Proceedings of 3rd IEEE International Symposium on Requirements Engineering (RE 1997), vol. 97, pp. 226–235 (1997)

[6] Dardenne, A., Lamsweerde, A.V., Fickas, S.: Goal-directed requirements acquisition. Science of Computer Programming 20(1-2), 3–50 (1993)

[7] W3C, OWL 2 Web Ontology Language Manchester Syntax (2009), http://www.w3.org/TR/owl2-manchester-syntax/

[8] Jiang, L., Barone, D., Amyot, D., Mylopoulos, J.: Strategic Models for Business Intelligence. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 429–439. Springer, Heidelberg (2011)

[9] Barone, D., Jiang, L., Amyot, D., Mylopoulos, J.: Composite Indicators for Business Intelligence. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 448–458. Springer, Heidelberg (2011)

[6] A preliminary report on the DOLCE+ categorization is available online www.cs.utoronto.ca/~jenhork/BIM/StrategicModelsforBusinessIntelligence.pdf

[10] Barone, D., Yu, E., Won, J., Jiang, L., Mylopoulos, J.: Enterprise Modeling for Business Intelligence. In: van Bommel, P., Hoppenbrouwers, S., Overbeek, S., Proper, E., Barjis, J. (eds.) PoEM 2010. LNBIP, vol. 68, pp. 31–45. Springer, Heidelberg (2010)
[11] Barone, D., Jiang, L., Amyot, D., Mylopoulos, J.: Reasoning with Key Performance Indicators. In: Johannesson, P., Krogstie, J., Opdahl, A.L. (eds.) PoEM 2011. LNBIP, vol. 92, pp. 82–96. Springer, Heidelberg (2011)
[12] Borgida, A., Horkoff, J., Mylopoulos, J., Rosati, R.: Experiences in Mapping the Business Intelligence Model to Description Logics, and the Case for Parametric Concepts. In: Proceedings of the 2012 International Workshop on Description Logics, DL 2012 (2012)
[13] Datamonitor, Global Credit & Charge Cards Industry Profile (2004)
[14] Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R.: Formal Reasoning Techniques for Goal Models. In: Spaccapietra, S., March, S., Aberer, K. (eds.) Journal on Data Semantics I. LNCS, vol. 2800, pp. 1–20. Springer, Heidelberg (2003)
[15] Sebastiani, R., Giorgini, P., Mylopoulos, J.: Simple and Minimum-Cost Satisfiability for Goal Models. In: Persson, A., Stirna, J. (eds.) CAiSE 2004. LNCS, vol. 3084, pp. 20–35. Springer, Heidelberg (2004)
[16] Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., Yu, E.: Evaluating goal models within the goal-oriented requirement language. International Journal of Intelligent Systems 25(8), 841–877 (2010)
[17] Berardi, D., Calvanese, D., Degiacomo, G.: Reasoning on UML class diagrams. Artificial Intelligence 168(1-2), 70–118 (2005)
[18] Wetzel, T.: States of Affairs. The Stanford Encyclopedia of Philosophy, Fall 2008 Edition (2008), http://plato.stanford.edu/archives/fall2008/entries/states-of-affairs
[19] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: WonderWeb Deliverable D18. The WonderWeb Library of Foundational Ontologies and the DOLCE ontology (December 2003)
[20] Krötzsch, M., Rudolph, S., Hitzler, P.: Description Logic Rules. In: Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008), pp. 80–84 (2008)
[21] Opdahl, A.L., Berio, G., Harzallah, M., Matulevicius, R.: An ontology for enterprise and information systems modelling. Applied Ontology 7(1), 49–92 (2012)
[22] Du, J., Pan, J.Z.: Towards Practical ABox Abduction in Large OWL DL Ontologies. In: Artificial Intelligence, pp. 1160–1165 (2009)
[23] Guarino, N., Welty, C.A.: An Overview of OntoClean. Handbook on Ontologies 48(1), 1–20 (2004)
[24] Barone, D., Topaloglou, T., Mylopoulos, J.: Business Intelligence Modeling in Action: A Hospital Case Study. In: Ralyté, J., Franch, X., Brinkkemper, S., Wrycza, S. (eds.) CAiSE 2012. LNCS, vol. 7328, pp. 502–517. Springer, Heidelberg (2012)
[25] Bunge, M.: Treatise on Basic Philosophy: Vol 1-8. Ontology I The Furniture of the World Boston Reidel (1977)
[26] Opdahl, A.L.: Anatomy of the Unified Enterprise Modelling Ontology. In: van Sinderen, M., Johnson, P. (eds.) IWEI 2011. LNBIP, vol. 76, pp. 163–176. Springer, Heidelberg (2011)
[27] Borgida, A.: Description Logics in Data Management. Data Engineering 7(5), 671–682 (1995)
[28] Iannone, L., Rector, A.: Calculations in OWL. In: Proceedings of the 5th Workshop OWL: Experiences and Directions, OWLED (2008)