Towards a first Ontology for Customer Relationship Management

Diego Magro
Dipartimento di Informatica, Università di Torino
Corso Svizzera 185
10149 Torino, Italy
magro@di.unito.it

Anna Goy
Dipartimento di Informatica, Università di Torino
Corso Svizzera 185
10149 Torino, Italy
goy@di.unito.it

ABSTRACT

This paper presents some results of an ongoing project aimed at modeling the main concepts related to Customer Relationship Management (CRM). More precisely, the paper presents O-CREAM, a CRM ontology based on DOLCE and on two DOLCE-based modules, DnS (exploited for modeling *roles* and for handling reification) and OIO (exploited for modeling business knowledge by means of *information objects*). The project relies on the belief that all the actors involved in CRM could benefit from an ontological investigation of this field, aimed at providing a core set of formally described concepts and relations, useful both for describing CRM processes and for specifying the functionality of CRM applications. In particular, a well-formed CRM ontology would support communication and interoperability both in intra-organization and in inter-organization CRM processes. The paper discusses in details the axiomatization for the *sale* and *customer relationship* concepts, as well as for the corresponding business knowledge items (i.e., *sale* and *customer records*). It concludes by sketching a possible concrete exploitation of O-CREAM.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms

Management

Keywords

Enterprise Ontology, CRM Ontology, Customer Relationship Management

1. INTRODUCTION

Nowadays, the ability to maintain profitable relationships with customers is of paramount importance for almost all enterprises. In fact, the competition in the current markets forces many enterprises to continuously reduce the time to market; to understand and to anticipate the needs of both actual and potential customers; to enhance the capability of retaining them (and of acquiring new ones); to increase their fidelity and their profitability and to envisage the possibility of any customer defection (churn risk assessment). These goals require a huge amount of knowledge on the customers and the ability to manage personalized relationships with them. For these reasons, in the last two decades, the Customer Relationship Management (CRM) [7, 6] has gained more and more importance in the business strategies of many companies worldwide and it is now considered an autonomous discipline within the field of business organization[1]. CRM denotes all those business activities and processes supporting the above-stated goals. By giving the customer a central role in the business strategy, it aims at enabling the companies to personalize not only their offers (products and services) but also their pricing and payment policies, the delivery modalities, the packaging, the after-sale services, etc. Besides Marketing, several other business units cooperate in supporting CRM, among them Sales and Customer Service, but also Production, Logistics and Finance. CRM activities process, manage, use and produce a great bulk of data and knowledge and they usually exploit an heterogeneous set of enabling technologies and services, spanning from more classical communication media (phone, fax and also ordinary mail and personal communications) to e-mail and Web applications; from Databases Management Systems to Data Warehouses, ERP systems and Business Intelligence applications.

The interactions among the business units involved in CRM, as well as those between the companies that use the CRM solutions and those ones that provide them, require a common language and a shared set of concepts.

In the last twenty years ontologies have gained popularity in computer science as a means for improving the interaction between computational systems, between humans and computational systems, and also between humans themselves [11].

There are a lot of efforts to model concepts related to enterprise activities and business, reported in the literature (see, for instance, [14]). However, as far as CRM is concerned, to our knowledge, a fully developed semantic model of this field is still missing.

Moreover, we believe that all the actors involved in CRM can benefit from an ontological investigation of this field, aimed at providing a core set of formally described concepts and relations, useful both for describing CRM processes and tasks and for specifying the requirements and the functionality of the CRM applications.

This paper presents some results of an ongoing ontological analysis of the field: Its principal goal is to argue the rationale behind the proposed approach and to discuss its main features. In particu-

[1]...although sometimes it is only used as a fashionable keyword to name old practices, but this is another story.

CSTST 2008 October 27-31, 2008, Cergy-Pontoise, France

lar, sharing the point of view maintained in [2], we specify a CRM ontology (O-CREAM: Ontology for Customer RElAtionship Management) within the framework of DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) [5], a well-known foundational ontology developed at the Laboratory of Applied Ontology (ISTC-CNR, http://www.loa-cnr.it). In so doing, the formal descriptions of the concepts and relations specific to CRM benefit from the rich characterization of the basic concepts and relations provided by DOLCE: The main aim is to build a well-founded domain ontology, suitable for supporting interoperability and communications both within a same company and in more open environments, such as those where users and providers of the CRM technological solutions meet. Besides DOLCE, our approach relies on two other DOLCE-based ontology modules [8]: The Ontology of Descriptions and Situations (DnS) and the Ontology of Information Objects (OIO). The former is a foundational ontology, which offers, among other things, a framemork for representing roles [13] and a principled way of handling the reification. The latter provides a formal characterization for information objects and is the basis for the O-CREAM fragment that accounts for the business knowledge[2].

A full formalization of O-CREAM is out of the scope of this paper, however we prove here the suitability of our approach by providing an axiomatization, within the DOLCE framework, for the concepts (and the related properties) of *sale* and *customer relationship*, as well as for that fragment of the business knowledge that contains data about them.

The paper is organized as follows: Section 2 present the main features and objectives of O-CREAM; Section 3 briefly discuss the aspects of DOLCE, DnS, and OIO exploited by O-CREAM; Section 4 presents the axiomatization for the *sale* and *customer relationship* concepts, as well as for the corresponding business knowledge items (i.e., *sale* and *customer records*), within O-CREAM. Finally, Section 5 provides some discussion and future work directions.

2. MAIN GOALS AND FEATURES OF O-CREAM

CRM influences all the business activities that somehow involve customers [6], therefore the processes supporting it are distributed on several business units. For instance, let's consider a possible way of managing a customer complaint about the quality of both a product and the after-sales service, as informally sketched as follows:

"In response to the complaint, check whether the customer is a most valuable or a most growable one; if it is so, on the basis of her profile, make her a personalized offer in terms of a new product, a new kind of after-sales service, a special price and devise a suitable delivery strategy, otherwise [...]; store the offer in the information system [...]".

Carrying out the tasks above involves not only the Marketing unit, but it may also involve the Production, the Sales and Customer Service, the Logistics and the Finance units. The CRM is therefore an interesting benchmark for the ontologies to prove their usefulness in supporting the knowledge sharing and the communications among different business units and in improving their cooperation.

Moreover, ICT-based products and services represent enabling technologies for CRM, and a shared CRM ontology may also be exploited in supporting the match between the requirements of companies looking for ICT solutions to support their CRM activities and the offers of the ICT solution providers. This inter-organizations setting is the perspective in which the ARNEIS framework [10] has been designed (see Section 5 for a brief sketch of this framework).

O-CREAM aims at providing the enterprises with a CRM reference ontology that formally describes a set of concepts and relations that capture the main features of this domain. A complete description of O-CREAM is out of the scope of this paper, nevertheless it is worth discussing here some of its core features. Let's consider again the example above. Some CRM activities may be driven by events (such as a communication from a customer that complains about something), they may produce events (such as an offer), they may use some business knowledge (such as the sale data for defining a customer profile), they may store new data (e.g., for keeping trace of the offers or of the communications with the customers), they may produce new knowledge (such as the segmentation of customers into some categories, such as *most valuable*, *most growable* and *below zero* customers), and they may exploit and/or enforce the relationships with actual or potential customers.

From our analysis of the CRM field, carried on in conjunction with CRM experts from Italian small-to-medium sized enterprises and CRM software producers, we found that there are three main concept types that need to be modeled in order to provide a semantic representation of the CRM field, namely: (i) business events (e.g., sales, offers, communications, appointments, etc.); (ii) (business) relationships which the enterprises are involved in (e.g., those between the company and its actual or potential customers); (iii) concepts and relations relevant to the knowledge that an enterprise has on (or derives from) business events and relationships.

On the basis of this rough grouping, we designed O-CREAM as composed by two big "pieces": the first one including the modeling of business events and relations, and the second one representing the enterprise knowledge about such events and relations.

Both concepts representing events and concepts representing relationships are basically characterized by the roles that are involved in the event or in the relation. For instance, a sale event is characterized by the presence of a buyer that pays a price to buy a good from a vendor. Moreover, a relationship is often linked to some event; e.g., if there has been a sale, then there will be a relationship between the actor that played the role of vendor in the event (maybe a company) and the actor that played the role of buyer (maybe a customer). Finally, if an enterprise is involved in some business event (e.g. a sale), then it will probably have some knowledge about such en event (e.g. a sale record in its database) and about the relations stemmed from it (e.g. a customer record).

All the concepts and the relations in O-CREAM are formalized according to a general framework provided by the DOLCE foundational ontology. By exploiting the reach axiomatization of DOLCE, we aim at strengthening the well-foundedness of O-CREAM and at improving its suitability to support the interoperability and the communication among the stakeholders also in open environments.

Obviously, DOLCE was not the only possible option, other upper level ontologies exist: for example, [12] lists, discusses, and compares several upper level ontologies. Following the considerations that emerge from this discussion, we selected DOLCE, basically for two reasons (besides the fact that it is freely available): it is inspired and oriented towards commonsense and linguistic knowledge, which seems particulary suitable for modeling fields like CRM; it is exploited for web-based applications, which are an important technology for CRM support.

As we stated above, in O-CREAM, we need to characterize business events and relationships, as well as those parts of the business knowledge that "speak of" them. To this purpose, the reification

[2]In this paper, we use the expression business "knowledge" in a broad sense to refer both to data and information, and to the knowledge in the strict sense.

of the predicates and the tuples that represent the events and the relationships appears to be a natural solution. Basically, such predicates and tuples are placed into the domain of discourse of the ontology by giving them a first-order nature, according to the reification framework provided by DnS (Sections 4.1 and 4.2 illustrate the approach by presenting the formalization of the notions of *sale* and of *customer relationship*, respectively). The formal characterization of those aspects of the business knowledge that are relevant to CRM is centered on the notion of *information object*, as it is provided by OIO, and it exploits the possibility of predicating on the above-mentioned reified predicates and tuples (Section 4.3 illustrate the approach by describing the formalizations of the concepts of *sale record* and *customer record*).

Finally, the notion of *role*, which is recognized by many researchers as a basic one in ontologies [1], is used within O-CREAM (in its reified formal representation contained in DnS) to specify the part taken by each particular in a business event or relationship.

3. BRIEF DESCRIPTION OF THE REUSED ONTOLOGIES

As already stated, O-CREAM relies on the foundational ontology DOLCE [5] and on two other ontologies [8], the ontology of Descriptions and Situations (DnS) and the Ontology of Information Objects (OIO), which extend DOLCE[3].

In this section, we provide a brief description of those aspects of DOLCE, DnS and OIO that are relevant to understand O-CREAM.

The predicates of DOLCE, DnS and OIO are prefixed by dol, dns and oio, respectively.

3.1 DOLCE

DOLCE is a foundational ontology of particulars. It distinguishes between *endurants* and *perdurants*, i.e. between those "particulars that are wholly present [...] at any time they are present" [5] (such as cars, people, laws, etc.), and those entities that "happen in time", i.e. that have temporal parts (such as processes, activities, etc). All the DOLCE categories (i.e. unary predicates) used in this paper are subconcepts of the endurant category. In particular, a $dol : agentive\_physical\_object$ is a physical object (therefore a physical endurant) to which we ascribe some kind of agentivity (e.g. a human person). A $dol : non\_agentive\_physical\_object$ is a physical object to which we do not ascribe any agentivity (e.g. a hammer, a car, etc.). A $dol : agentive\_social\_object$ is a non physical object (therefore a non physical endurant) to which we ascribe some kind of agentivity and that depends on a community of agents (e.g. a legal person). On the opposite, to a $dol : non\_agentive\_social\_object$ we cannot ascribe any agentivity (such as a law, a description, etc.).

Finally, any $dol : amount\_of\_matter$ is a physical endurant representing, for instance, some iron, some meat, etc. O-CREAM makes use also of the DOLCE primitive relation $dol : part(x, y, t)$, which expresses the parthood relationship between an endurant whole x and the part y at time t.

DOLCE reference formal characterization is specified in modal logic S5 plus the Barcan Formula, but *lighter* versions of DOLCE, also expressed in OWL (Web Ontology Language) [16], do exist as well[4].

3.2 DnS

DnS [8] extends DOLCE with a set of concepts, relations, definitions and axioms whose main purpose is to support the reification of the tuples and of both the intension and the extension of the classes and relations. DnS is organized around the two main concepts of $dns : description$ and $dns : situation$ and the satisfaction relation ($dns : satisfies$) between situations and descriptions [9]. A $dns : situation$ is a $dol : non\_agentive\_social\_object$ which represents "a state of affairs or relationship, a tuple, or fact" [8]. Each situation involves a set of particulars ($dns : setting(x, y)$ expresses the fact that the particular x *is in the setting of* the situation y) that is given a structure (a conceptualization) by means of a $dns : description$ (which is a $dol : non\_agentive\_social\_object$). In other words, a $dns : description$ represents a conceptualization that makes possible to view a set of particulars as a $dns : situation$, i.e., as a "referent of a cognitive disposition towards a world". For instance, the description of the concept of *sale* allows one to interpret a state of affairs involving a person, a company, a car and an amount of money as a situation in which a car is sold by a company to a person who pays to get it. The relationship between a situation and a description is formally captured by the $dns : satisfies$ relation and each situation must $dns : satisfies$ at least one $dns : description$. At a formal level, a predicate may be reified as a description, while an instance of its extension may be reified as a situation that satisfies the description. Each description must use ($dns : uses$) at least one $dns : concept$ (which is a $dol : non\_agentive\_social\_object$). In DnS, the roles ($dns : role$) are reified as individuals of $dns : concept$ [13] and they are the most important kind of $dns : concept$ for O-CREAM. Each role may be used by different descriptions, but it must be defined by exactly one ($dns : defines$ is a specialization of $dns : uses$; $dns : defined\_by$ is the inverse of $dns : defines$). At any time, each role may be played by ($dns : played\_by$) any endurant. Each endurant can play more than one role at the same time and, conversely, each role can be played by more than one endurant simultaneously, but it can also be empty. A $dns : agent\_driven\_role$ is a $dns : role$ that can be played only by agents. The relation $dns : played\_by$ is a specialization, of $dns : classifies$: the latter is more general, since it involves elements of $dns : concept$ and DOLCE's particulars, while the former restricts to $dns : role$ (a subconcept of $dns : concept$) and endurants (i.e. specific types of DOLCE's particulars).

As regards the general notion of satisfaction, DnS specifies that if a situation x $dns : satisfies$ a description y, then there must be at least one $dns : concept$ z, used by y, which classifies at least one particular in the setting of the situation x. This is a rather general notion of satisfaction, which can be specified to adapt it to various domains. DnS also provides some more specific notions of satisfaction. Among them, we mention the *proactive satisfaction* ($dns : p\_sat$), which requires that the proactively satisfied description exists prior to at least one particular in the setting of the satisfying situation.

Finally, to understand O-CREAM, we need also to know that a $dns : rational\_physical\_object$ is a $dol : agentive\_physical\_object$ to which we can ascribe rationality, besides agentivity (e.g., a human person).

3.3 OIO

OIO [8] extends DOLCE with a set of concepts, relations, definitions and axioms that account for information objects. In OIO, an information object ($oio : information\_object$) is a $dol : non\_agentive\_social\_object$ and it represents an information content. OIO distinguishes both between an information and its physical realization, and between an information and its encoding sys-

[3]Some parts of O-CREAM are based on the Plan Ontology [8], but in this paper we do not discuss them.

[4]See http://www.loa-cnr.it/Ontologies.html.

tem. A physical realization is any entity that acts as a physical support for an information (e.g. a paper sheet, a sound, a content of a sequence of cells in the main memory of a computer, etc). Moreover, information contents are ordered according to some encoding systems, that is according to a set of rules (i.e. a language, a code, etc.), represented in the ontology by a $dns : description$. Finally, at any time t, an information content x can express a meaning y, formalized as a $dns : description$ ($oio : expresses(x, y, t)$), and can be about any particular, i.e. any element in the domain of discourse of the ontology, z ($oio : about(x, z, t)$).

4. FORMAL CHARACTERIZATION OF (A FRAGMENT OF) O-CREAM

Both the business events (e.g. sales, communications, publications, appointments, offers, etc.) and the business relationships (e.g. relationships with actual or potential customers) relevant to CRM are represented within the DnS framework as situations satisfying specific descriptions. Such descriptions provide the conceptualizations for the situations and they specify the roles[5] that the particulars involved in the situations play.

Placing the particulars relevant to a business event or relationship in the setting of a situation means, in some sense, representing and handling them as a unity, since the situation represents the event or the relationship as a unique reified element (which can be predicated on within axioms that characterize the business knowledge).

In this section, we illustrate our approach by providing a formal characterization for a business event type (the *sale*), for a business relationship type (the *customer relationship*) and for the two corresponding types of knowledge elements (the *sale* and *customer records*). The free variables in the formulas are intended to be universally quantified[6].

4.1 Sale

A *sale* is a business event in which a buyer pays a price to buy a good from a vendor.

In O-CREAM, a description type is introduced for each kind of business event. The following axiom characterizes a sale description as a $dns : description(x)$ that uses the four roles $vendor$, $buyer$, $exchanged\_good$, $paid\_price$.

$sale\_description(x) \rightarrow$
$dns : description(x) \wedge$
$(\exists v, b, eg, pp)(vendor(v) \wedge buyer(b) \wedge exchanged\_good(eg) \wedge$
$paid\_price(pp) \wedge dns : uses(x, v) \wedge dns : uses(x, b) \wedge$
$dns : uses(x, eg) \wedge dns : uses(x, pp))$

In the axiom we have used the relation $dns : uses(x, y)$, which is more general than $dns : defines(x, y)$, since a $sale\_description$ may use roles defined elsewhere. Moreover, no closure is specified on the relation $dns : uses(x, y)$ within the axiom above that limits the possible role types to the four specified, since we do not want to prevent the possibility of specifying any sale description using also other types of DnS concepts.

The roles used by any sale description are specified by the following four axioms. All these roles are always defined by some $sale\_description$. Moreover, both vendor and buyer are $dns : agent\_driven\_role$, since only agents can play them.

[5]We restrict our attention to the roles, even if some kinds of business events requires also other $dns : concept$s in their corresponding descriptions.

[6]Even if this formal characterization is given in classic First Order Logic, it is worth bearing in mind that O-CREAM relies on DOLCE, whose reference formalization is expressed in Modal Logic.

Since we take the point of view of enterprises, we are not interested in modeling the business transactions between private individuals: this is why only the organizations, at any time t, can play the vendor role. On the opposite, the buyer role can be played either by organizations (e.g., in the Business-to-Business transactions) or by people (e.g., in the Business-to-Consumer transactions).

$vendor(x) \rightarrow$
$dns : agent\_driven\_role(x) \wedge$
$(\forall y)(dns : defined\_by(x, y) \rightarrow sale\_description(y)) \wedge$
$(\forall y, t)(dns : played\_by(x, y, t) \rightarrow organization(y))$

$buyer(x) \rightarrow$
$dns : agent\_driven\_role(x) \wedge$
$(\forall y)(dns : defined\_by(x, y) \rightarrow sale\_description(y)) \wedge$
$(\forall y, t)(dns : played\_by(x, y, t) \rightarrow (organization(y) \vee$
$human\_person(y)))$

In [3], an ontological analysis of the concept of product is carried out, which results in a complex notion of product, involving that one of a social artefact purposely created with the intention of exchanging it. In their analysis, the authors state that such a social artefact can be either a $dol : amount\_of\_matter$ or a $dol : non\_agentive\_physical\_object$. Of course, a product in the sense stated above can be an exchanged good. Since we want the role of exchanged good to be as much general as possible, we admit among its possible players also any kind of service provided by an enterprise, an amount of money (e.g. in financial transactions) or a piece of information (e.g. any data):

$exchanged\_good(x) \rightarrow$
$dns : role(x) \wedge$
$(\forall y)(dns : defined\_by(x, y) \rightarrow sale\_description(y)) \wedge$
$(\forall y, t)(dns : played\_by(x, y, t) \rightarrow (dol : amount\_of\_matter(y)$
$\vee\ dol : non\_agentive\_physical\_object(y) \vee service(y) \vee$
$amount\_of\_money(y) \vee oio : information\_object(y)))$

The following characterization of the $paid\_price$ role could seem a little counter-intuitive, since all the kinds of particulars that can play the $exchanged\_good$ role can also play the $paid\_price$ one: this is motivated by the sake of generality. As in [15], in O-CREAM the notion of sale comprises the usual sales in which a good is paid by money, as well as the currency exchanges and also the barters (in which any good can be exchanged with any other good).

$paid\_price(x) \rightarrow$
$dns : role(x) \wedge$
$(\forall y)(dns : defined\_by(x, y) \rightarrow sale\_description(y)) \wedge$
$(\forall y, t)(dns : played\_by(x, y, t) \rightarrow (dol : amount\_of\_matter(y)$
$\vee\ dol : non\_agentive\_physical\_object(y) \vee service(y) \vee$
$amount\_of\_money(y) \vee oio : information\_object(y)))$

The roles introduced above are pairwise disjoint:

For each $P, Q \in \{vendor, buyer, exchanged\_good, paid\_price\}$ $(P \neq Q)$: $\neg(P(x) \wedge Q(x))$

A sale is defined as a situation that satisfies a $sale\_description$:

$sale(x) \leftrightarrow$
$dns : situation(x) \wedge (\exists y)(sale\_description(y) \wedge$
$dnd : satisfies(x, y))$

As stated in Section 3.2 , DnS contains a general notion of satisfaction, which can be futher specified to tailor it to specific domains. Here we require that any sale contains players for all the roles introduced above and used by the $sale\_description$ that it satisfies:

$sale(x) \wedge sale\_description(y) \wedge dns : satisfies(x, y) \rightarrow$
$(\forall r)(dns : uses(y, r) \wedge (vendor(r) \vee buyer(r) \vee$
$exchanged\_good(r) \vee paid\_price(r)) \rightarrow$
$(\exists p, t)(dns : played\_by(r, p, t) \wedge dns : setting(p, x)))$

Conversely, any situation that contains players for all the above-

stated roles used by a $sale\_description$ satisfies that description (and, therefore, it represents a sale):

$dns : situation(x) \wedge sale\_description(y) \wedge$
$(\exists t)(\forall r)(dns : uses(y, r) \wedge (vendor(r) \vee buyer(r) \vee$
$exchanged\_good(r) \vee paid\_price(r)) \rightarrow$
$(\exists p)(dns : played\_by(r, p, t) \wedge dns : setting(p, x))) \rightarrow$
$dns : satisfies(x, y)$

The notions formalized so far, make use of four concepts whose characterization is out of the scope of O-CREAM, namely those of human person, organization, amount of money and service. We provide only a minimal set of axioms for these entities. The first three axioms are intuitive, while the last two state that, in O-CREAM, a service is supposed to be a situation that proactively satisfies a service description (i.e. the description of the service exists prior to at least one particular in the setting of the service itself):

$human\_person(x) \rightarrow dns : rational\_physical\_object(x)$
$organization(x) \rightarrow dol : agentive\_social\_object(x)$
$amount\_of\_money(x) \rightarrow dol : endurant(x)$
$service\_description(x) \rightarrow dns : description(x)$
$service(x) \leftrightarrow$
$dns : situation(x) \wedge$
$(\exists y)(service\_description(y)) \wedge dns : p\_sat(x, y)$

Clearly, service description and sale description are two disjoint concepts:

$\neg(service\_description(x) \wedge sale\_description(x))$

4.2 Customer Relationship

A company is in a *customer relationship* with those people or enterprises to which it has sold some good. Sometimes, a person is explicitly specified as a customer contact.

First of all, following the same formalization pattern as for the sale, a customer relationship description is a description that uses the three roles *supplier*, *customer* and *customer_contact*:

$customer\_relationship\_description(x) \rightarrow$
$dns : description(x) \wedge$
$(\exists s, c, cc)(supplier(s) \wedge customer(c) \wedge customer\_contact(cc) \wedge$
$dns : uses(x, s) \wedge dns : uses(x, c) \wedge dns : uses(x, cc))$

Moreover, customer relationship description is disjoint with respect to sale and service description:

for each $P \in \{service\_description, sale\_description\}$:
$\neg(customer\_relationship\_description(x) \wedge P(x))$

All the three roles mentioned above are $dns : agent\_driven\_role$ and they can be defined only by some $customer\_relationship\_description$. The supplier role can be played only by organizations, as it is for the vendor role and for the same reasons:

$supplier(x) \rightarrow$
$dns : agent\_driven\_role(x) \wedge$
$(\forall y)(dns : defined\_by(x, y) \rightarrow$
$customer\_relationship\_description(y))$
$\wedge(\forall y, t)(dns : played\_by(x, y, t) \rightarrow organization(y))$

Like the buyer role, the customer one can be played either by organizations or by people:

$customer(x) \rightarrow$
$dns : agent\_driven\_role(x) \wedge$
$(\forall y)(dns : defined\_by(x, y) \rightarrow$
$customer\_relationship\_description(y)) \wedge$
$(\forall y, t)(dns : played\_by(x, y, t) \rightarrow$
$(organization(y) \vee human\_person(y)))$

Only people can be contacts in customer relationships:

$customer\_contact(x) \rightarrow$
$dns : agent\_driven\_role(x) \wedge$
$(\forall y)(dns : defined\_by(x, y) \rightarrow$
$customer\_relationship\_description(y)) \wedge$
$(\forall y, t)(dns : played\_by(x, y, t) \rightarrow human\_person(y))$

Furthermore, the following axioms state the pairwise disjointness between the roles:

$P, Q \in \{customer, supplier, customer\_contact\}(P \neq Q)$
and $R \in \{vendor, buyer, exchanged\_good, paid\_price\}$: $\neg(P(x) \wedge Q(x))$ and $\neg(P(x) \wedge R(x))$

A customer relationship is simply a situation that satisfies a customer relationship description:

$customer\_relationship(x) \leftrightarrow$
$dns : situation(x) \wedge$
$(\exists y)(customer\_relationship\_description(y) \wedge$
$dns : satisfies(x, y))$

The following axiom specifies the $dns : satisfies$ relation for the customer relationship. Any customer relationship contains players both for customer and supplier roles, while a player for the customer contact role is not mandatory:

$customer\_relationship(x) \wedge$
$customer\_relationship\_description(y) \wedge$
$dns : satisfies(x, y) \rightarrow$
$(\forall r)(dns : uses(y, r) \wedge (customer(r) \vee supplier(r)) \rightarrow$
$(\exists p, t)(dns : played\_by(r, p, t) \wedge dns : setting(p, x)))$

The notions of customer relationship and that of sale are obviously related. The following two axioms capture such relation.

First of all, if there is a customer relationship that involves a particular p_1 as customer and a particular p_2 as supplier, then p_1 must have bought something from p_2, i.e. there must be at least a sale in which p_1 plays the role of buyer and p_2 plays that one of vendor[7]:

$customer\_relationship(cr) \wedge$
$customer\_relationship\_description(crd) \wedge$
$customer(c) \wedge supplier(s) \wedge dns : satisfies(cr, crd) \wedge$
$dns : setting(p_1, cr) \wedge dns : setting(p_2, cr) \wedge$
$dns : uses(crd, c) \wedge dns : uses(crd, s) \wedge$
$dns : played\_by(c, p_1, t_1) \wedge dns : played\_by(s, p_2, t_1) \rightarrow$
$(\exists sa, sd, v, b, t_2)(sale(sa) \wedge sale\_description(sd) \wedge vendor(v) \wedge$
$buyer(b) \wedge dns : satisfies(sa, sd) \wedge$
$dns : uses(sd, v) \wedge dns : uses(sd, b) \wedge$
$dns : setting(p_1, sa) \wedge dns : setting(p_2, sa) \wedge$
$dns : played\_by(b, p_1, t_2) \wedge dns : played\_by(v, p_2, t_2))$

The converse must also be true: if a buyer p_1 buys something from a vendor p_2, a customer relationship holds in which p_1 is the customer and p_2 the supplier:

$sale(sa) \wedge sale\_description(sd) \wedge vendor(v) \wedge buyer(b) \wedge$
$dns : satisfies(sa, sd) \wedge dns : uses(sd, v) \wedge dns : uses(sd, b) \wedge$
$dns : setting(p_1, sa) \wedge dns : setting(p_2, sa) \wedge$
$dns : played\_by(b, p_1, t_1) \wedge dns : played\_by(v, p_2, t_1) \rightarrow$
$(\exists cr, crd, c, s, t_2)(customer\_relationship(cr) \wedge$
$customer\_relationship\_description(crd) \wedge$
$customer(c) \wedge supplier(s) \wedge dns : satisfies(cr, crd) \wedge$
$dns : setting(p_1, cr) \wedge dns : setting(p_2, cr) \wedge$
$dns : uses(crd, c) \wedge dns : uses(crd, s) \wedge$
$dns : played\_by(c, p_1, t_2) \wedge dns : played\_by(s, p_2, t_2))$

4.3 Sale and Customer Records

Each element of business knowledge (i.e. the whole bulk of the business knowledge owned by an enterprise, as well as any part of this knowledge) is an $oio : information\_object$ (see Section 3.3). Therefore each business knowledge element *is about* some particular and it *expresses* some meaning.

In this formalization, we assume neither any particular encoding

[7]Note that the relations with potential customers are modeled by a different concept, i.e. Potential Customer Relationship.

system for the business knowledge, nor any specific physical realization. Rather, in general, a same part of business knowledge can have different physical realizations (e.g. both paper sheets and sequence of bits in a hard disk) as well as several encodings (e.g. a document can be written in more languages).

In the following, we do not provide a full formalization for all the notions related to the business knowledge, but we formally characterize only those fragments of the business knowledge bulk that refer to the concepts of sale and customer relationship introduced above.

As the following axiom states, each business knowledge element is an information object whose parts are still business knowledge elements:

$business\_knowledge\_element(x) \rightarrow$
$oio : information\_object(x) \wedge (\forall y, t)(dol : part(x, y, t) \rightarrow$
$business\_knowledge\_element(y))$

Usually, the operational business knowledge contains data about sales, ranged in a set of records (possibly stored in a database table on some hard disk, but this is not mandatory in our formalization).

Having reified both the sales and their descriptions, it is easy to characterize a sale record as a business knowledge element, which is about a sale and whose meaning is specified by a corresponding sale description:

$sale\_record(x) \rightarrow$
$business\_knowledge\_element(x) \wedge$
$(\exists y, z, t)(sale\_description(y) \wedge sale(z) \wedge dns : satisfies(z, y) \wedge$
$oio : expresses(x, y, t) \wedge oio : about(x, z, t))$

For the sake of simplicity, the relation expressing the ownership between an organization and a business knowledge element is not described here. However, it is worth stating, at least informally, that each sale record is owned by exactly one organization, which plays the role of vendor in the sale that the sale record is about.

In an enterprise, a customer record contains data on an actual customer of its. Therefore, actually, a customer record is about a customer relationship and it expresses a customer relationship description:

$customer\_record(x) \rightarrow$
$business\_knowledge\_element(x) \wedge$
$(\exists y, z, t)(customer\_relationship\_description(y) \wedge$
$customer\_relationship(z) \wedge dns : satisfies(z, y) \wedge$
$oio : expresses(x, y, t) \wedge oio : about(x, z, t))$

As for the sale record, we can informally state that each customer record is owned by exactly one organization, which plays the role of supplier in the customer relationship that the customer record is about.

5. DISCUSSION AND FUTURE WORK

In this paper we presented O-CREAM, a reference heavyweight core ontology for CRM. It is worth noting that here we presented only a small subset of the set of notions formalized in O-CREAM, i.e. the most suited to present the core ideas of the approach. We would like now to conclude the paper by discussing some open issues, especially concerning the possible exploitation of O-CREAM.

First of all, O-CREAM would surely benefit from the integration with an ontology of organizations [4], formalizing concepts such as roles and norms underlying organizational settings. For instance, such an ontology would clarify the relation between the notion of *customer_contact* and the one of *customer*, by allowing one to specify that whenever the customer role is played by an organization, the contact should play, in her turn, some role in that organization (e.g., she can be an employee of the organization).

Moreover, in order to exploit O-CREAM in real applications, and in particular in Web-based systems, a lighter version (expressed, for instance, in a language such as OWL) is required. This implies the need for a mechanism to deal with relations with a cardinality greater than two, and reification, adopted in a principled way in our approach (see Section 3), represents a Best Practice for facing this issue.

As a concrete example of exploitation of O-CREAM in a Web-based application, we plan to integrate it within the ARNEIS system [10]. ARNEIS (Advanced Repository for Needs of Enterprises and Innovative Software) is a framework for building "intelligent" web-based repositories of software solutions for small and medium enterprises (*Piccole e Medie Imprese*, PMI). PMI that aim at finding ICT products and services to support their business activities can contact the ARNEIS repository and express their requirements. ARNEIS, in turn, provides them with the most suitable solutions fulfilling their requirements, by relying on its advanced reasoning capabilities, that support the matching between PMI requirements and stored ICT solutions. The ARNEIS framework and a prototype implementation, are described in [10].

In particular, ARNEIS Knowledge Base is represented by a set of ontologies, encoding the system knowledge about various areas of business activities (e.g. Customer Relationship Management, Office Automation, Financial Analysis, and so on). On the basis of such ontologies, ARNEIS dynamically generates a User Interface that supports the elicitation of information from the user (a software house aiming at publishing a description of its products/services, or a PMI looking for ICT support for its business). On the basis of such information, ARNEIS builds a semantic description of the software solution/PMI requirements that represent the input for the matching engine. The ARNEIS ontologies, as well as the description of ICT products/services and of PMI requirements based on such ontologies, are represented in OWL. Thus, the exploitation of O-CREAM within the ARNEIS framework is a further motivation to develop an OWL version, that is currently under construction.

6. ACKNOWLEDGMENTS

This work has been partially supported by TOCAI.IT FIRB national project (http://www.dis.uniroma1.it/~tocai/).

7. REFERENCES

[1] G. Boella, L. van der Torre, and H. Verhagen, editors. *Applied Ontology, Special Issue: Roles, an interdisciplinary perspective*, volume 2. IOS Press, 2007.

[2] S. Borgo and P. Leitão. The role of foundational ontologies in manufacturing domain applications. *LNCS*, 3290:670–688, 2004.

[3] S. Borgo and L. Vieu. From physical artefacts to products. In *Proc. Second Workshop FOMI*, pages 85–89, 2006.

[4] E. Bottazzi and R. Ferrario. Preliminaries to a dolce ontology of organizations. *International Journal of Business Process Integration and Management*, 2008. To appear.

[5] C.Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. *WonderWeb deliverable D18*. ISTC-CNR, 2003. Technical Report.

[6] A. Devalle. Customer relationship management. In *V. Cantino et al. Management Information Systems*, pages 343–368. McGraw-Hill, 2005.

[7] J. Freeland. *The Ultimate CRM Handbook*. McGraw-Hill, 2005.

[8] A. Gangemi, S. Borgo, C. Catenacci, and J. Lehmann. *Task Taxonomies for Knowledge Content*. 2005. Metokis Deliverable D07.

[9] A. Gangemi and P. Mika. Understanding the semantic web through descriptions and situations. *LNCS*, 2888:689–706, 2003.

[10] A. Goy, D. Magro, and F. Prato. ARNEIS: A framework for building intelligent web-based repositories of ICT products and services. Technical report, Computer Science Department, University of Torino (Italy), 2008.

[11] M. Hepp, P. D. Leenheer, A. de Moor, and Y. Sure, editors. *Ontology Management*. Springer, 2008.

[12] V. Mascardi, V. Cordì, and P. Rosso. A comparison of upper ontologies. Technical report, Computer Science Department, University of Genova (Italy), 2006.

[13] C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, and N. Guarino. Social roles and their descriptions. In *Proc. KR 2004*, pages 267–277, 2004.

[14] P. Rittgen, editor. *Handbook of Ontologies for Business Interaction*. Idea Group, 2007.

[15] M. Uschold, M. King, S. Moralee, and Y. Zorgios. The enterprise ontology. *The Knowledge Engineering Review*, 13(1):31–89, 1998.

[16] W3C. Web Ontology Language. http://www.w3.org/2004/OWL, 2007.