# Semantic Programming for Semantic Web

**Andrei Mantsivoda**
Irkutsk State University, Russia

**Аннотация**

We propose an approach to knowledge processing, which is based on the semantic programming paradigm. The term superstructure over the data type model, together with naming constraints is the basic construction, on which this approach rests. The talk is mostly focused on the basic concepts of the approach.

The formal descriptions of knowledge domains play a key role in many fields including the Semantic Web [2], in which knowledge domains are used for description of information resources and metadata development. We develop here an approach to knowledge domain representation, which is based on the ideas of semantic programming [1]. Domains are considered as integrated collections of objects. Objects of a domain are stratified into classes and have their own properties which distinguish them from other objects. Objects have relations with other objects. Also we assume that some objects can have *names* (identifiers). Naming is important. In particular, based on naming, the notion of *a resource* is defined [4]. A resource is considered as anything that has identity (a unique name). Thanks to its unique name a resource can be distinguished from other resources and associated with a description (metadata). Sometimes a unique name determines a method for access to the resource itself. A name also can play the role of a magnet, which attracts diverse portions of information, and makes them a uniform resource.

## 1 Semantic Programming

Semantic programming determines the following order of describing $D$. First, a data type model $\Re = \langle M_1, \dots, M_s; \Omega \rangle$ is established, which supplies atomic elements for descriptions. Then $D$ is stratified and the main classes and attributes of objects are formed. Third, a hereditary finite superstructure $T_\Re^D$ over the model $\Re$ is constructed. This superstructure consists of syntactic objects (terms), which are interpreted as object descriptions. Fourth, reflecting the hierarchical nature of $D$, a hierarchical system of classes as special subsets of $T_\Re^D$ is defined. Fifth, an ontology in the form of a constraint system [3] based on naming constraints is established.

In general, terms contain incomplete information about objects, and some terms contain more information than others. We say that $t_1$ approximates $t_2$ ($t_1 \sqsubseteq t_2$), if $t_2$ contains all information contained in $t_1$. Approximation determines a partial order over terms. And the less informative a term, the greater number of objects it describes, that is, if $t_1 \sqsubseteq t_2$, then $\{d \mid t_2 \triangleq d, d \in D\} \subseteq \{d \mid t_1 \triangleq d, d \in D\}$. Here $t \triangleq d$ means that $t$ describes the object $d$.

The concept of the name is introduced in our formalism via the naming relation (constraint) which has the form $id :: t$, and is intuitively interpreted as "an object with the name $id$ is described by the term $t$".

## 2 The Term Superstructure

In this section we construct the term superstructure $T_{\Re}^{D}$ over the basic model $\Re$. Let $\mathbf{CN} = \{C_1, \ldots, C_p\}$ be the collection of all classes established in $D$, $C_i \subseteq D$. We also assume that the partial order relation $\prec_D$ of inheritance is determined, such that if $C_i \prec_D C_j$, then $C_j \subseteq C_i$.

$T_{\Re}^{D}$ is built of syntactic material and has the form of a set of terms. Its elements will serve as descriptions of objects of $D$. For this, a special language of terms $L$ is introduced. We assume that all elements of $\Re = \langle M_1, \ldots, M_s; \Omega \rangle$ are distinguished. The set of all constants corresponding to elements of $\Re$ is denoted by $\bar{M} = \bar{M}_1 \cup \ldots \cup \bar{M}_s$. The *signature* of $L$ includes: (1) the set of constants $\bar{M}$; (2) the 'top' constant $\top$; (3) the finite set of 'name' constants $ID = \{id_1, id_2, \ldots, id_q\}$; (4) the finite set of unary functional symbols $Attr = \{p_1, p_2, \ldots, p_k\}$, which denote attributes relevant to classes of $D$; (5) the set of binary functional symbols $CN = \{cn_1, cn_2, \ldots, cn_p\}$ containing separate $cn_i$ for each class $C_i$ of $D$; (6) the term set constructor $\{\ldots\}$. The constant $\bar{\emptyset}$ denotes the empty set.

The language $L$ consists of terms of two types: description terms, and attribute terms. They have the following mutually recursive definitions:

**Definition 2.1 (description terms)** *A description term is a term of the form $cn(c, a)$, where $cn \in CN$, $c$ is a description set and $a$ is an attribute set. A description set is either $\bar{\emptyset}$ or an expression of the form $\{t_1, \ldots, t_q\}$, where $t_i$ are description terms. An attribute set is either $\bar{\emptyset}$ or an expression of the form $\{a_1, \ldots, a_q\}$, where $a_i$ are attribute terms.*

**Definition 2.2 (attribute terms)** *An attribute term is a term of the form $p(t)$, where $p \in Attr$ and $t$ is either a description term or $t \in ID$, or $t \in \bar{M}$.*

Intuitively, a term $cn_i(c, a)$ describes an object $d$ as a member of the class $C_i$. Its first argument $c$ contains the set of description terms, any of which characterizes $d$ as an element of some superclass of $C_i$. We call such terms *description* subterms. In general description subterms are those, which are not subterms of attribute terms. The fact that $t_1$ is a description subterm of $t_2$ we denote by $t_1 \prec t_2$ (it is a syntactical equivalent of $\prec_D$). Multiple inheritance is brought about by including in $c$ more than one description term. The second argument contains attributes of $d$ associated with $C_i$.

Not any term can describe objects of $D$. The term $cn_i(\{cn_j(c_j, a_j)\}, a_i)$ means that $C_j \prec_D C_i$. In particular, the term $person(\{professor(\bar{\emptyset}, \bar{\emptyset})\}, \bar{\emptyset})$ describes an object in a domain, in which the class *person* is a subclass of the class *professor*, and this is invalid for the standard domain of people. Attributes also must be bound to classes they specify. For instance, the attribute *surname* is correct for the class of persons but irrelevant to the class of cars. In order to bind attributes to classes we introduce the relation $p \lhd_D \langle C, R\rangle$, where $p \in Attr, C \in \mathbf{CN}, R \in \mathbf{CN} \cup \{\bar{M}_1, \dots, \bar{M}_s\}$. This relation determines $p$ as the attribute of the class $C$, and the values of this attribute belong to $R$.

This brings about the notion of *a correct term* with respect to the relations $\prec_D$ and $\lhd_D$. For instance, the term $professor(\{person(\bar{\emptyset}, \{surname(\text{"}Smith\text{"}), spouse(id_5)\})\}, \{university(id_7)\})$ is correct in the domain of people. Here $surname \lhd_D \langle C_{person}, \bar{M}_{String}\rangle$, $spouse \lhd_D \langle C_{person}, C_{person}\rangle$, $university \lhd_D \langle C_{teach\_staff}, C_{university}\rangle$; $C_{person} \prec_D C_{teach\_staff} \prec_D C_{professor}$; $id_5$ and $id_7$ are names of objects of the classes $C_{person}$ and $C_{university}$, respectively. The definition of correctness can be formulated syntactically and is omitted due to the lack of space.

**Definition 2.3 (A term superstructure)** *Let $D$ be a domain. The term superstructure over $\Re$ is the set $T^D_{\Re} = \{t \mid t$ is a correct description term of $L$ w.r.t $\prec_D$ and $\lhd_D\}$*

Let us define the semantics of terms of $T^D_{\Re}$.

**Definition 2.4** *Let $D$ be a domain. An interpretation $I_D : L \to D$ is any mapping such that: (1) For any $id \in ID$, $I_D(id) \in D$. All $I_D(id_i)$, $i = 1..q$ are pairwise different. (2) For any $p_i \in Attr$ if $p_i \lhd_D \langle C, R\rangle$ then $I_D(p_i) \subseteq C \times R$ , and for any $d \in D$ the set $\{m \mid \langle d, m\rangle \in I_D(p_i)\}$ is finite (the restriction of the finite range).*

**Definition 2.5** *A term $t \in T^D_{\Re}$ is the description of an object $d \in D$ ($t \triangleq d$) if for any $cn_i(c, a) \prec t$ the object $d \in C_i$, and if $p(e) \in a$ then (1) if $e \in ID$ and $p \lhd_D \langle C_i, C'\rangle$ for some $C'$ then $I_D(e) \in C'$ and $\langle d, I_D(e)\rangle \in I_D(p)$; (2) if $e \in \bar{M}_i$ then $\langle d, e\rangle \in I_D(p)$; (3) if $e \in T^D_{\Re}$ and $p \lhd_D \langle C_i, C'\rangle$ then there exists $d' \in C'$ such that $e \triangleq d'$ and $\langle d, d'\rangle \in I_D(p)$.*

## 3 Approximations and Amalgams

Approximation is a relation, which is determined on terms of $T^D_{\Re}$ and denoted $t_0 \sqsubseteq t_1$. It allows us to compare the amounts of information stored in terms. Intuitively, $t_0 \sqsubseteq t_1$ holds if $t_1$ contains all information coded in $t_0$.

**Definition 3.1 (Approximation)** *(1) For any $id_1, id_2 \in ID$ , $id_1 \sqsubseteq id_2$ iff $id_1 = id_2$. (2) For any $m_1, m_2 \in \bar{M}$, $m_1 \sqsubseteq m_2$ iff $m_1 = m_2$. (3) $e \sqsubseteq \top$ for any $e \in T_{\Re}^{D} \cup \bar{M} \cup ID$ . (4) $cn(c,a) \sqsubseteq t$ iff there exists $cn(c_1, a_1) \prec t$ , such that (a) for any $t' \in c$ there exists $t'' \in c_1$ such that $t' \sqsubseteq t''$; (b) for any $p(t') \in a$ there exists $p(t'') \in a_1$ such that $t' \sqsubseteq t''$.*

Two elements $t_1, t_2 \in T_{\Re}^{D}$ are equivalent ($t_1 \equiv t_2$) iff $t_1 \sqsubseteq t_2$ and $t_2 \sqsubseteq t_1$. $t_2$ is strictly greater than $t_1$ ($t_1 \sqsubset t_2$) iff $t_1 \sqsubseteq t_2$ and $t_2 \not\sqsubseteq t_1$.

**Proposition 3.1** *The relation of approximation is reflexive and transitive.*

Thus, the pair $\langle T_{\Re}^{D}, \sqsubseteq \rangle$ is a partial order with the greatest element $\top$. Let us introduce the operation of taking the least upper bound of two terms $t_1 \sqcup t_2$ which amalgamates information of the two descriptions.

**Definition 3.2** *Let $t, t_1, t_2 \in T_{\Re}^{D}$. $t$ is the least upper bound (amalgam) of $t_1, t_2$ ($t = t_1 \sqcup t_2$), if $t_1 \sqsubseteq t$, $t_2 \sqsubseteq t$ , and for any $t_0 \in T_{\Re}^{D}$: $t_1 \sqsubseteq t_0 \wedge t_2 \sqsubseteq t_0$ implies $t \sqsubseteq t_0$.*

**Theorem 3.1** *Let $D$ be a domain and $T_{\Re}^{D}$ the corresponding superstructure over $\Re$. Then for any $t_1, t_2 \in T_{\Re}^{D}$ there exists $t_1 \sqcup t_2 \in T_{\Re}^{D}$.*

Intuitively this theorem means that two descriptions $t_1$ and $t_2$, describing the same object $d \in D$, can merge within $T_{\Re}^{D}$ into a combined description – the exact 'sum' of $t_1$ and $t_2$. If $t_1 \sqcup t_2 = \top$ then amalgamation of the two terms is senseless (e.g. when these terms are inconsistent and can not describe the same object).

Let us denote $[D]_t = \{d \mid t \triangleq d, d \in D\}$ . $[D]_t$ is the set of domain objects described by the term $t$.

**Proposition 3.2** *If $t_1 \sqsubseteq t_2$, then $[D]_{t_2} \subseteq [D]_{t_1}$. If $t_1 \equiv t_2$, then $[D]_{t_1} = [D]_{t_2}$. $[D]_{t_1 \sqcup t_2} = [D]_{t_1} \cap [D]_{t_2}$. In particular, if $t_1 \triangleq d$ and $t_2 \triangleq d$ then $t_1 \sqcup t_2 \triangleq d$.*

**Definition 3.3 (Semantics of naming constraints)** *$I_D \models id :: t$ if $I_D(id) \in [D]_t$.*

# 4 Classes

In this section we introduce the concept of a class description in $T_{\Re}^{D}$.

**Definition 4.1** *A term $t \in T_{\Re}^{D}$ is complete w.r.t. the domain $D$, if $|[D]_t| > 0$ and for any $t_0 \in T_{\Re}^{D}$, such that $t \sqsubset t_0$, $|[D]_t| = 0$.*

Thus, complete elements contain all possible information, which is expressible within the language $L$, and can not be augmented. We denote by $\bar{T}_{\Re}^{D}$ the set of all complete terms of $T_{\Re}^{D}$, and by $\bar{T}_{\Re i}^{D}$ the set of all complete terms $t \in \bar{T}_{\Re}^{D}$ such that $cn_i(\bar{\emptyset}, \bar{\emptyset}) \sqsubseteq t$.

**Proposition 4.1** *Let $I_D$ be an interpretation. For any $t \in T^D_{\Re}$, such that $t \triangleq d$ for some $d \in D$, there exists a complete term $\bar{t} \in T^D_{\Re}$ such that $t \sqsubseteq \bar{t}$, and any strictly ascending chain $t \sqsubset t_1 \sqsubset \ldots \sqsubset t_k \sqsubset \bar{t}$ is finite.*

The language $L$ and $T^D_{\Re}$ describe $D$ with some precision. We say that a description $\triangleq$ of the domain $D$ *distinguishes* an object $d \in D$ if there exists a complete term $\bar{t} \in \bar{T}^D_{\Re}$ such that $[D]_{\bar{t}} = \{d\}$. A description $\triangleq$ can have indistinguishable elements for two reasons. First, several objects can have the same complete description. Second, some $d \in D$ can not have it at all.

**Definition 4.2** *A description $\triangleq$ is complete if for any object $d \in D$ there exists a complete element $\bar{t} \in \bar{T}^D_{\Re}$ such that $\bar{t} \triangleq d$. $\triangleq$ is precise if any object of the domain is distinguishable.*

**Proposition 4.2** *(1) Let $\bar{t}$ be a complete term and $\bar{t} \triangleq d$. Then for any $t$ such that $t \triangleq d$, $t \sqsubseteq \bar{t}$ holds. (2) A precise description is complete.*

**Definition 4.3 (Class approximation and description)** *Let $C_i$ be a class of the domain $D$. The set $T^D_{\Re i}$ is the approximation of $C_i$. The set $\bar{T}^D_{\Re i} = T^D_{\Re i} \cap \bar{T}^D_{\Re}$ is the description of $C_i$.*

## 5 Conclusion and Future Work

Based on the constructions introduced above we are developing a knowledge representation technique in the form of a constraint system $\langle C, T^D_{key}, \vdash \rangle$, where $C$ is a set of primitive naming constraints (assertions), $T^D_{key}$ is a key set (containing terms $t$ which uniquely define objects, such that $|[D]_t| = 1$), and $\vdash$ is a finite entailment relation, which describes naming constraints behavior. Now we try to apply this technique to various problems of resource manipulation in the Web and other tasks within logic programming, data mining, operation systems, etc. A number of theoretical issues (such as constraint space behavior and propagation strategies) are considered as well.

## Список литературы


[1] S.S. Goncharov, Yu.L.Ershov, D.I.Sviridenko. "Semantic programming", Information processing, Proc. IFIP 10-th World Comput. Congress, Dublin, v.10, 1986, 1093-1100.

[2] T. Berners-Lee, J. Hendler, O. Lassila. The Semantic Web. Scientific American, May, 2001.

[3] V. A. Saraswat. Concurrent Constraint Programming Languages. The MIT Press, Cambridge, MA, 1993.

[4] RFC 2396 - Uniform Resource Identifiers (URI). http://rfc.net/rfc2396.html.