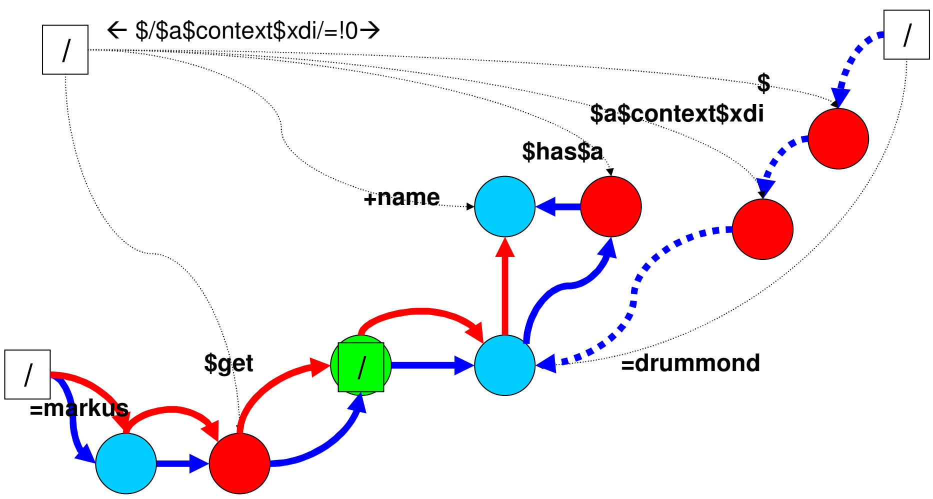
University of Rome Tor Vergata

*G. Bartolomeo*

# *XDI-RDF Global Graph Model Proposal*

*Version: 1.0*

*Date: Jan, 21 2009*




This document describes a proposal for the XDI-RDF Global Graph under definition at OASIS XDI TC.

Comments and feedback are welcome at the author's email address giovanni.bartolomeo@uniroma2.it

| ***Change log:*** |
| --- |
| *v.1.0 Jan 21, 2009 Initial Version* |

# 1. Key Principle #1 (syntactic correctness)

Graph and metagraph MUST use the same graphical conventions. Semantics of arcs, nodes and labels MUST be the same. Moreover, as the XDI RDF model uses the very same RDF statements to express relations both at a instance level and at a meta-level, there MUST be no distinction between graph and metagraph: they are both subgraphs of the same, unique, global XDI graph.

Any distinction in the syntax used for graphs and the one used for metagraphs may result in the inability to express specific concepts in XDI, thus limiting its expressiveness.

# 2. Key Principle #2 (semantic consistency)

Any graph must be syntactically correct and properly reflect the relations described in the XRI3.0 statements describing the context (syntactic correctness). This property is similar to the well-formed property of an XML document.

However this is not sufficient to claim that such a graph properly depicts an XDI document as XDI documents add semantics to the XRI statements they are made of.

Semantics may impose some constraints on the set of XRI statements which could be contained in the XDI document, thus on the set of arcs and nodes that MUST or MAY appear in the corresponding graph.

In this paper we use the convention to describe in *italics* those constraints.

# 3. Nodes and arcs

A statement like +x/$has/+y involves **instances** of subjects (+x,+y,$has). Therefore, nodes in the graph represent specific instances of those subjects which, in turn, MAY be used in statements as subjects, predicates or objects.

Those nodes are connected through arcs originating from the context in which the corresponding subjects have been defined.

# 4. Foundation and contexts ($)

The XDI global graph MUST have a foundation root. This is a XDI context in which the very basic XDI subjects are defined. To this context is assigned the XRI3.0 address =!0.

```
$
   $a$context$xdi
      =!0
=
   $has
      !0
=!0
```

The corresponding graph for these statements is reported below.

NOTE: for simplicity's sake we omit to report the definition of $a$context$xdi.

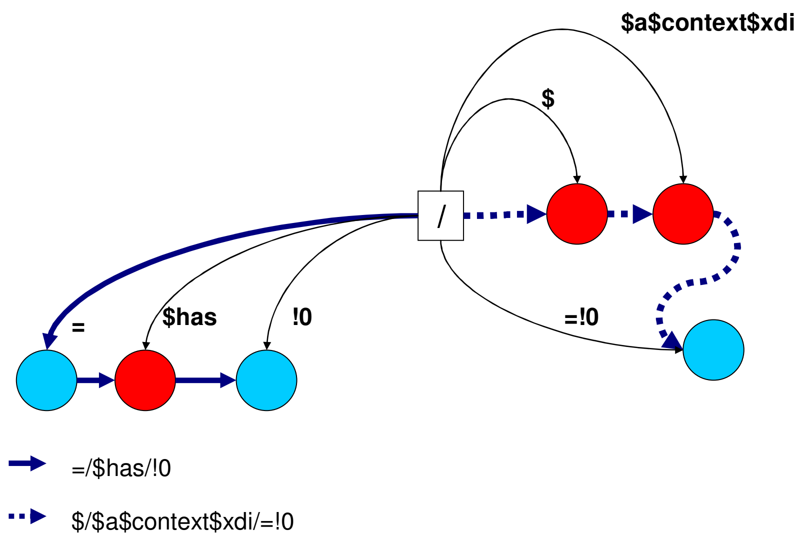



**Figure 1**

All the subjects in this context are represented by arcs originating from the root. XRI statements are represented through paths. XRI statements always start with a subject, therefore all paths have their origin in the context root.

*In this graph the predicate $has has been used to build the XRI address =!0. In general, if a node +x+y is encountered in a given context, the expression +x/$has/+y must be present in that context (Figure 3), otherwise the graph, though syntactically correct, is not semantically consistent.*

NOTE: this context uses the $has relationship which will be illustrated in the following. Because $has has an inverse predicate, $a$has, for semantic consistency's sake, the statement !0/$a$has/= should be present in the document, and properly reflected in the graph. However, for simplicity's sake we omit this detail. An example of inverse predicate is reported in Figure 7.

# 5. $is

Now we open a second context and experiment how the four fundamental predicates are implemented. We start from the identity predicate $is. This second context is defined as:

```
$
   $a$context$xdi
      =!f83.6b1.44f
+x
   $is
      +y
```

Therefore it defines the two XDI subjects +x +y and their identity relationship.

NOTE: for simplicity's sake we omit to report the definition of +x and +y, i.e. (+/$has$/x, +/$has/y), which must be defined in this context (otherwise the graph, though semantically correct, is inconsistent).

The corresponding graph is hereafter reported. The root at the top right corner corresponds to the one illustrated in Figure 1, and is henceforth named "main root". The root at the left is instead the root of a new context identified by the XRI =!f83.6b1.44f. The new introduced root is henceforth named "local root".

Focus now on the definition of this context. There are two different sets of arcs involved in this definition. The first set are the arcs originates from the main root (context identified with the XRI =!0). These arcs are used to create local instances of the concepts they represents. This is reflected in the three nodes at the right side of the local root. These nodes are linked together by a second set of arcs which forms a path describing the XRI statement $/$a$context$xdi/=!f83.6b1.44f.

NOTE: for simplicity's sake we omit to report the definition of =!f83.6b1.44f, which is similar to the one for =!0 discussed in the previous section.

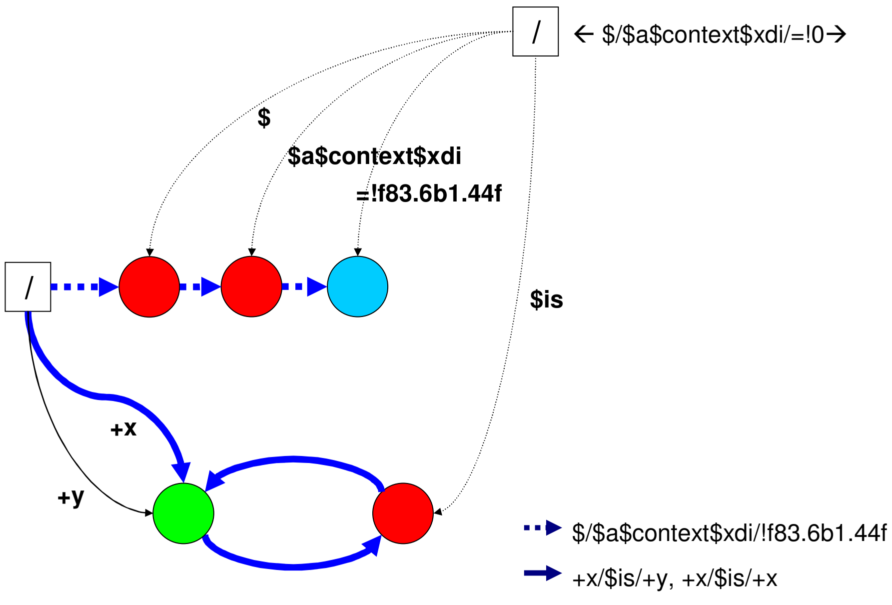



Figure 2

The identity relationship +x/$is/+y is described in the context by a path involving two nodes: one is referenced by the local root (by two arcs, +x and +y) and the other is referenced from the main root. The second node is in fact an instance of the predicate identity $is, defined in the main root.

The first node instead is an instance of the subject +x which is locally defined in this context. It is also an instance of the subject +y, defined in this context too.

*By pointing the two arcs labelled with +x and +y to the same node, we enforce the described identity relationship at instance level, making the graph semantically consistent.*

# 6. $has

The $has predicate is modelled in the graph similarly.

```
$
   $a$context$xdi
      =!f83.6b1.44f
+x
   $has
      +y
```

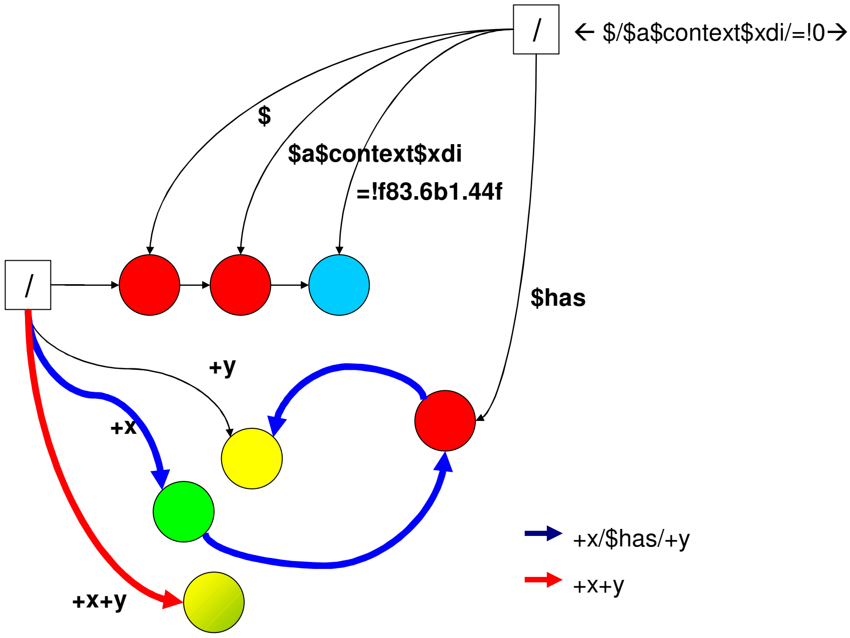



**Figure 3**

+x+y is a subject in the local context and may be used as any other subject in the same context. The graph also illustrates the definition of the subject +x+y, through the $has predicate. No additional nodes or arcs are needed in the graph to describe these statements.

*If the node +x+y is encountered in a given context, the expression +x/$has/+y must be present in that context (Figure 3), otherwise the graph, though syntactically correct, is not semantically consistent.*

NOTE: In this example we omitted to report the inverse relationship, e.g. +y/$a$has/+x, however for semantic consistency's sake, this statement should be present in every context defining +x/$has/+y, and properly reflected in the graph. An example of inverse predicate is reported in Figure 7.

# 7. $has$a

The $has$a predicate is modelled in the graph similarly.

```
$
   $a$context$xdi
      =!f83.6b1.44f
+x
   $has$a
      +y
```

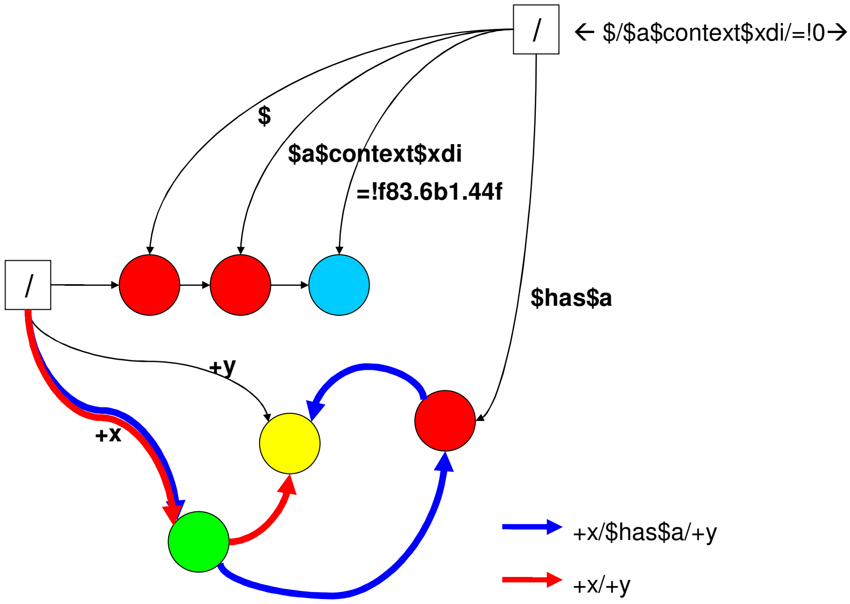



**Figure 4**

The statement +x/+y in the local context is reported in a graph by an arc linking two nodes pointed from the local root by the arcs +x and +y. The graph also illustrates the definition of the statement +x/$has$a/+y, through the $has$a predicate. No additional nodes or arcs are needed in the graph to describe these statements.

*If the statement +x/+y is encountered in a given context, the expression +x/$has$a/+y must be present in that context (Figure 4), otherwise the graph, though syntactically correct, is not semantically consistent.*

NOTE: In this example we omitted to report the inverse relationship, i.e. +y/$a$has$a/+x, however, for semantic consistency's sake, this statement should be present in every context defining +x/$has$a/+y, and properly reflected in the graph. An example of inverse predicate is reported in Figure 7.

# 8. $is$a

The $is$a predicate is modelled in the graph similarly.

```
$
   $a$context$xdi
      =!f83.6b1.44f
+x
   $is$a
      +y
```

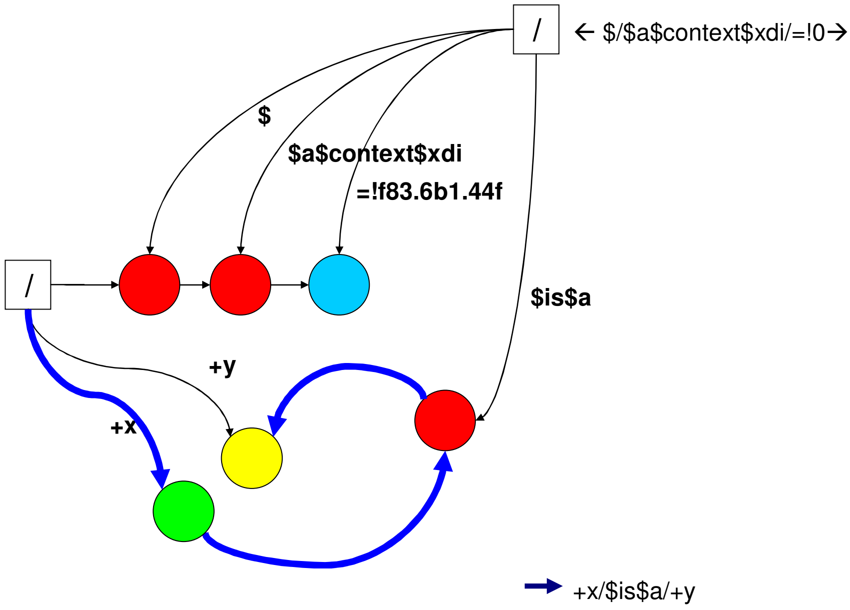



**Figure 5**

*If the statement +x/$is$a/+y is present in the context, any composition ($has$a) relation, inheritance ($is$a) relation or aggregation ($has) relation defined for subject +x should be also valid for ("inherited by") subject +y.*

Figure 6 illustrates an example of a context in which the following statements are defined:

```
+y
   $has$a
      +z
+x
   $is$a
      +y
   $has$a
      +z
+x/+z
+y/+z
```

NOTE: in this example, the vertical arrows pointing +y and +z might originate from a root different than the local root, i.e. +y and +z may be defined in another context (not shown in the picture). See also Figure 8 and Figure 9 for further examples about distribution.

NOTE: inverse predicates are not described here for simplicity's sake. An example of inverse predicate is reported in Figure 7.

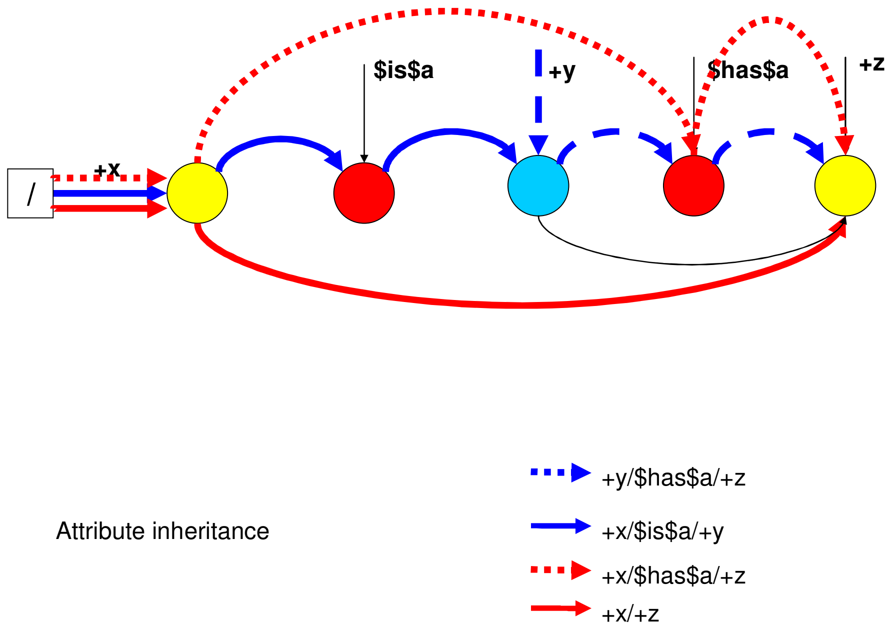



Figure 6

# 9. Inverse predicates

Inverse predicates are modelled similarly. We limit to illustrate in the following graph how the inverse predicate $a is modelled.

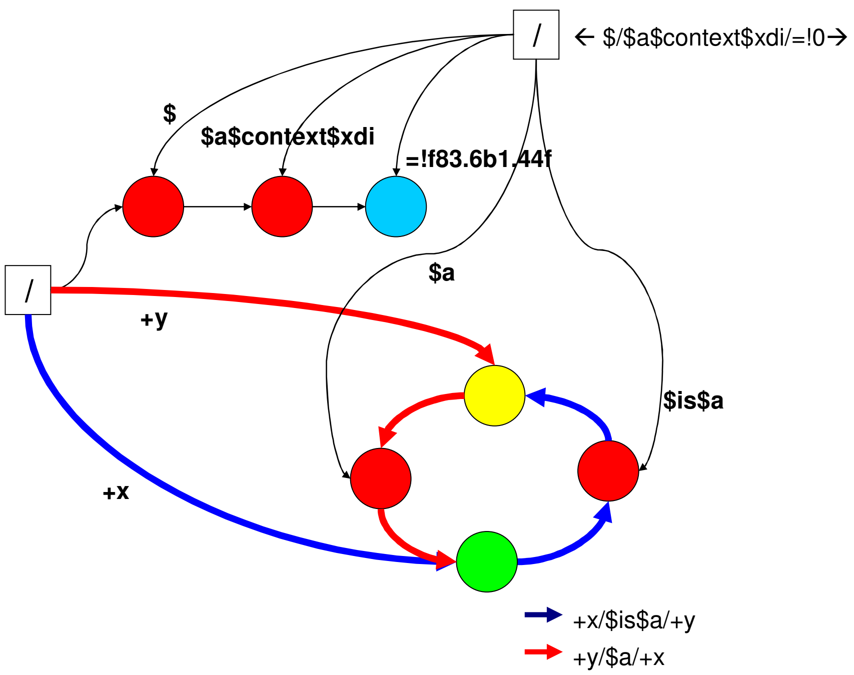



Figure 7

*For semantic consistency, if +x/$is$a/+y is encountered in the graph, then +y/$a/+x should be presenta as well and viceversa. This applies also to any other predicate which can be associated to an inverse predicate (e.g., $has, $has/$a).*

# 10. XDI messages. Query. Distribution.

XDI messages are themselves XDI documents, thus they are reflected with nodes and arcs in the graph.

A message contains:

- the sender of the message as a subject
- one or more XDI graph operations as predicates
- for each operation, a subcontext the operation will act upon, introduced by the context transition symbol put on behalf of the object.

In the following two messages

```
=markus
   $get
      /
         =drummond
            $has$a
               +name
```

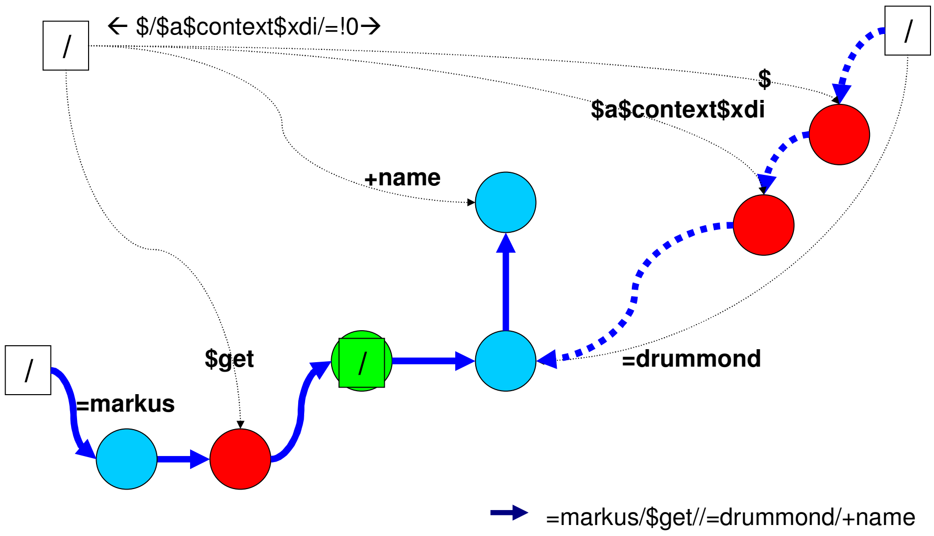
```
=markus
   $get
      /
         =drummond
            +name
```

=markus performs a query to know whether =drummond has the attribute +name and the value of this attribute. The corresponding graphs are depicted in Figure 8. The context transition is depicted by a pointer to a new local root node which represents the root of the new context.

NOTE: XDI client can request different representations of a resource by specifying different types in the request message by using references to types. The first message could be modified as follow:

```
=markus
   $get$a$xsd$boolean
      /
         =drummond
            $has$a
               +name
```

This would return a reference to the Boolean values true or false. However, for simplicity's sake we omit to report this use case.



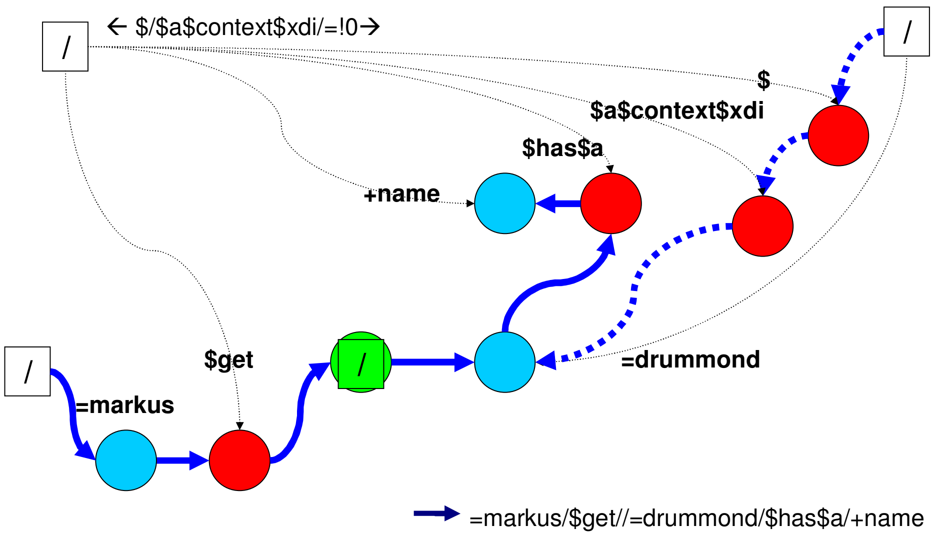




Figure 8

NOTE: the aforementioned messages could be grouped into one single message, which uses the same local root to :

```
=markus
   $get
      /
         =drummond
            $has$a
               +name
            +name
```

The corresponding graph is depicted in Figure 9.

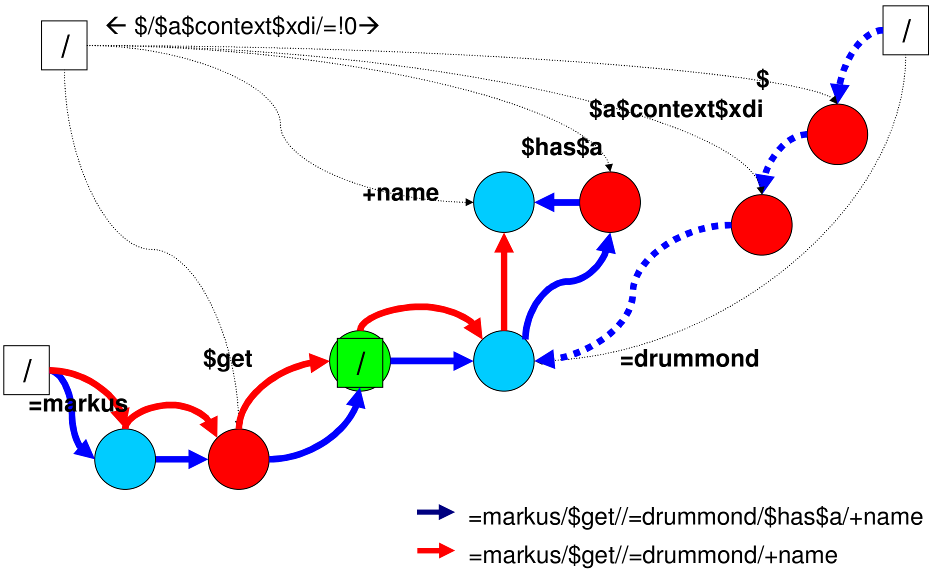



Figure 9

# 11. References

[1] OASIS XDI TC, The XDI RDF Model, version 11, **http://www.oasis-open.org/committees/download.php/29748/xdi-rdf-model-v11.pdf**