Theory Comput Syst (2015) 56:291–308
DOI 10.1007/s00224-014-9536-x

On the Turing Completeness of the Semantic Web

Amir Pourabdollah · Tim Brailsford

Published online: 8 March 2014
© Springer Science+Business Media New York 2014

Abstract The evidenced fact that "Linking is as powerful as computing" in a dynamic web context has lead to evaluating Turing completeness for hypertext systems based on their linking model. The same evaluation can be applied to the Semantic Web domain too. RDF is the default data model of the Semantic Web links, so the evaluation comes back to whether or not RDF can support the required computational power at the linking level. RDF represents semantic relationships with explicitly naming the participating triples, however the enumeration is only one method amongst many for representing relations, and not always the most efficient or viable. In this paper we firstly consider that Turing completeness of binary-linked hypertext is realized if and only if the links are dynamic (functional). Ashman's Binary Relation Model (BRM) showed that binary relations can most usefully be represented with Mili's pE (predicate-expression) representation, and Moreau and Hall concluded that hypertext systems which use the pE representation as the basis for their linking (relation) activities are Turing-complete. Secondly we consider that RDF –as it is- is a static version of a general ternary relations model, called TRM. We then conclude that the current computing power of the Semantic Web depends on the dynamicity supported by its underlying TRM. The value of this is firstly that RDF's triples can be considered within a framework and compared to alternatives, such as the TRM version of pE, designated pfE (predicate-function-expression). Secondly, that a system whose relations are represented with pfE is likewise going to be Turing-complete. Thus moving from RDF to a pfE representation of relations would give far greater power and flexibility within the Semantic Web applications.

A. Pourabdollah (✉) · T. Brailsford
School of Computer Science, The University of Nottingham, Jubilee Campus,
Wollaton Road, Nottingham NG8 1BB, UK
e-mail: amir.pourabdollah@nottingham.ac.uk

T. Brailsford
e-mail: tim.brailsford@nottingham.ac.uk

Keywords Relations · Model · RDF · Hypertext · Semantic Web

1 Introduction

1.1 The Power of Links

Today we extensively use the Web for our day-to-day jobs. The Web's initial design was not calculated to have this capacity and may not be optimal, however it has been successfully grafted onto the existing Web and has much the appearance of a carefully-designed and flexible open hypertext system [1]. Many Web scripting and programming technologies within and of Web pages exist today that allows the realization of a dynamic, programmable and adoptable Web, a very different from the initial static Web. On the other hand, since the Web is nothing more than some interconnected contents, the Web programmability mainly lies in its links.

The Semanic Web [2] is a web of data designed to be primarlily used by computers in response to the semantic discovery of the data originated by humans. Links in the context of the Semantic Web may not be necessarily traversed by human but this does not matter when it comes to their computational power. Even if not directly, one can expect a similar level of programmability at the links level in the Semantic Web as it is expected in the Web.

As a hypertext system, the Web may be viewed in terms of relations and relationships between objects, which may take many forms. These can be better comprehended and compared with a unifying model. The Binary Relation Model (BRM) [3] defines a means of exhaustively listing and subsequently comparing different implementations of binary relations, irrespective of their purpose or use. The BRM indicates which of the possible representations were useable, but also which had advantages in some operational contexts that others lacked. Hypertext systems can obviously benefit from analysis within the BRM as it can clearly delineate the limitations and advantages of some implementations and show how those limitations and advantages arise.

1.2 Static vs. Dynamic Links

Moreau and Hall [4] discussed the power of various hypertext systems in terms of the Chomsky hierarchy, and concluded that by implementing hypertext linking as binary relations expressed in one of the representations from the BRM meant that "linking is as powerful as computing", i.e. linking is Turing-complete, as long as it is implemented with the pE (predicate-expression) representation of relations. Interestingly, this most powerful representation of a binary relation itself had roots in the modelling of software and verification. The pE, or predicate-Expression, form of a function was how Mili represented binary relations of pre- and post-process states [5]. Namely, the binary relation was not necessarily expressing a semantic relationship but could just as easily express a process. In this context, the Turing-completeness of the pE representation is perfectly natural; as it expresses programs which themselves by definition express anything that is computable.

So if the links in a hypertext system are implementations of pE representation of binary relations, the system inherits this Turing-completeness, and can do anything that is computable essentially because it is programmable. This is where much of the real power of the Web arises, despite its very basic linking model, because it has a means by which the pE implementation of binary relations can be achieved, combining both process and hypertext linking.

1.3 Binary vs. Ternary (or Semantic) Links

In the context of the Semantic Web, RDF is the basic data model for describing Web resources, as recommended by W3C [6]. Among the layers of the Semantic Web, RDF is the core layer that makes basic statements about resources. It is used to write descriptive "statements" about each resource, using other resources, in the form of subject-predicate-object (or resource-property-value) triples. Each resource is denoted by a URI, thus a RDF relation is an ordered enumeration of three URIs. Although a single RDF triple is about relating two resources by a property, there are many indirect solutions about how to use RDF to express n-ary relations (like in [7]). It is noticeable that RDF itself does not provide any implementation of a language, and that XML is only one option for describing data in RDF. There are alternative syntaxes, like n-Triples [8], Turtle [9] and Notation-3 (N3) [10].

One feature missing from the BRM representations but present in RDF is the ability to explicitly represent the names of relation incidences from one or more named relations – the BRM representations are unable to incorporate this naming of the appropriate relation or incidence, except implicitly. On the other hand, RDF's semantic advantage is compromised by the same constraints as those present in the BRM's enumeration of incidences representation, this latter being RDF's relation representation, setting aside the semantics. These constraints include its inability to represent infinite relations, its unsuitability for dynamic use, and its proneness to error over a changeable set of relation elements (in the hypertext case, the endpoints of links).

1.4 Can RDF Support Turing Completeness in the Sematic Web?

The work in this paper unifies the computational power of the pE representation with the explicit relation naming/designation capability of RDF. Extending the Binary Relation Model to a Ternary Relation Model (TRM) exactly addresses this by providing explicit naming/typing/semantic meaning of each binary relation and hence to all binary relation incidences. RDF is equivalent to the enumeration of incidences representation of relations in the TRM, with the explicit naming/typing/meaning of the relation present in every relation incidence. Yet RDF is not "powerful" as such, being a data representation, not a programming mechanism. However the step from data representation to programming mechanism is conceptually very simple and consists merely of implementing a different relation representation. The work of this paper shows how simple it is to incorporate Turing-completeness, perhaps not into RDF as it is at present, but into a marginally modified RDF. If nothing else, the use of an extended pE relation representation can simplify the creation and management of RDF triples within the current infrastructure.

This paper covers this work by starting with a background in the Binary Relation Model, with a focus on the pE representation, and in RDF. It then describes how the Ternary Relation Model extends the Binary Relation Model, and why this is generally useful. We subsequently consider the impact of changing the RDF relation representation to include a pE form of representation, gaining the advantages of computability and control, and then conclude.

1.5 Notations

Some of the notation and terms used in the paper are clarified here. It is important to keep in mind that "relation" as used throughout this paper refers to a collection of relation incidences, and should not be confused with individual incidences.

Relation: this refers to a collection of entities and the connections between them. Different "types" of connections are usually represented with different relations.
Relation incidence: one or more entities from a relation along with the connection that joins them.

For example, a single RDF triple is a relation incidence, while the set of all triples whose *property* (see Section 2.3) is the same, forms a relation in full.

2 Related Works

This section overviews the features of the BRM and how it relates to RDF. BRM has been reviewed in relatively more details since its principle and concepts are essential in the rest of the paper. However, full treatments of the BRM [11], the pE representation of relations [5] and its power [4] and of RDF [6] can be found elsewhere.

2.1 The Binary Relation Model

The BRM [11, 12] is a way of enumerating all the possible ways of implementing link types in a hypertext system. It begins by identifying the salient features of binary relations from a hypertext point of view. This hypertext sensibility influenced the necessity of considering different representations, since the pure mathematical models of binary relations were not subject to real-world distresses such as the volatility of the underlying set of elements in a relation, which in a hypertext and Web context, are manifested in implementation difficulties such as broken or disoriented links, and link completeness. The BRM abstracted out of real-world hypertext systems basic differences in the underlying link creation and maintenance processes, which are described in terms of the different representations within BRM.

The BRM firstly considers how relations are comprised, determining that there are three features:

- The source set – elements which occur on the left of the relation, the "from" elements;

- The destination set – elements on the right of the relation, the "to" elements; and
- The incidences – marking which of the sources is connected/ related to which of the destinations.

It also considers how relations are utilised, primarily from a hypertext viewpoint, but with more general applicability. It does this through asking a series of "navigational" questions which between them describe the source, destination and incidences for whole or partial sets of relation incidences from a binary relation.

The BRM formulates all the possible ways of implementing link types in a hypertext system, by considering purely the navigation model, and focuses on general representation of binary relations regardless of their application or visualization. Some of the definitions of BRM are briefly as follows:

a. An *endpoint* is any addressable thing.
b. A *link* is a connection from an endpoint to another endpoint.
c. A (binary) *relation* R is a subset of S^2 (the Cartesian product of S and S) while the model *space* S is the set of all endpoints.

For any potential endpoint (or node or element), four main navigational questions can be answered to characterize its navigational position:

a. *Source Existence:* Is this node the source of any link?
b. *Destination Identification:* Where can I go from this node?
c. *Destination Existence:* Is this node the destination of any link?
d. *Source Identification:* What nodes are linked to this node?

Questions a and b represent linking in the usual, "forward" direction, while c and d represent linking in the "backward" direction, so that bidirectional linking is modellable.

It then goes on to consider what are the ways to represent these individual sets, with three ways being:

1. *Enumeration* – the explicit naming of all participating elements;
2. *Predicate* – the "filtering" of a set from a larger set by applying a set-membership selection test; and
3. *Expression* – a calculation (parameterised or not) that returns a set of elements.

The predicate and expression states are also called 'computed', and the computation itself can be implemented in two different modes: *pre-computed* or *dynamically computed* [12, 13]. For pre-computed links, the link anchor in the source document is clearly specified after all the necessary computations, but in dynamically computed links, the eligibility of each node to be source or destination, is computed in run-time on user request (like non-advertised links that are being advertised by hovering the mouse over them).

This is reflected in the remaining questions that can be asked of a representation in the BRM, namely

a. *Link Existence:* Is there a link between these two elements?
b. *Source enumeration:* What are all the possible sources of this set of links?

c. *Destination enumeration:* What are all the possible destinations of this set of links?
d. *Link enumeration:* What are all the links in this set?

The first of these is not a true navigation question because the identification of both source and destination endpoints is already known, the only question being asked is whether there is a corresponding pair of entry and exit points between them, i.e. ”can one go from here to there?”. The last three are not navigation questions involving decisions about if and where one can go from a given endpoint, but rather are queries about the whole set of links, whose results are independent of the reader’s current position in the data collection.

Pre-computation of all relation incidences (links) is the application of either a predicate or expression to calculate all the participants in any of the three constituent sets of a relation.

Having established what sets form the relation, and how these may be represented, then a comprehensive enumeration of the representations for relations are counted off simply by considering all the possible combinations of possibilities for the sets making up a relation, paying particular attention to how these representations occur in real-world hypertext systems. These real-world observations support many of the theoretical observations, many being motivated by the challenges of maintaining valid hypertext links (equivalent to relation incidences) in a highly changeable information collection, such as the Web. This is a key limitation of those representations that use enumeration for any or all of their constituent sets, and the various representations of the BRM are discussed in terms of their ability to answer the navigational questions in a volatile and potentially infinite information collection.

2.2 pE as a Relation Representation

The pE representation (named more fully, the “predicate- expression representation” in [3]) had fewer constraints than those representations incorporating enumeration. However it is not able to guarantee “backwards” linking, i.e. bidirectional linking, but otherwise is able to answer the navigational questions within a greater range of scenarios and contexts than other representations.

Moreau and Hall considered real-world examples of some of the representations from BRM in terms of their relative computing power [4] and found similar, complementary contrasts. For example, one of the linking mechanisms in Microcosm [14] which was in part enumerated was found to be equivalent to a pushdown automaton. More importantly, they found that one early implementation of pE relation representation, functional linking [15], was essentially Turing-complete and as such, that “linking was as powerful as computing” since linking was, in this implementation at least, actually enabled by computing.

Moreau and Hall’s work thus indicates a significant difference between different relation representations. Enumeration of relations is essentially a data structure. The pE representation renders relations as processes. These processes can always be reduced to data structures by fully evaluating all possible relation incidences, which may be desirable for pragmatic purposes such as performance [13]. However the converse is not necessarily true, as it is not always possible to fix on the "true" processes giving rise to a set of relation incidences.

There are clearly significant benefits arising from dynamically generating the relation elements – as opposed to statically enumerating them. Including participants in a relation by name ensures that they remain in the relation, regardless of whether subsequent activity renders them irrelevant, and at the same time it means that that other elements that might become eligible for inclusion due to meeting requisite conditions are however not included by default. There is no inherent inclusion or exclusion, based on meeting relevant criteria, so that for those relations which were created with criteria-based selection processes, the actual management of selection occurs outside the relation representation.

By contrast, describing participants in a relation by explicitly prescribing the selection process in the representation greatly simplifies the management of the relation. Eligible participants come and go from the relation as appropriate, and it is possible to extract small sub-relations on criteria-based selection as well, so that the entire relation never need be fully manifested. This is discussed further in Section 5.

2.3 Using RDF as Linkbase

The Semantic Web (and consequently, RDF-described data) is deliberately designed to be used by computers [2, 16–19]. One approach to make human-readable media while having all the computer-readability benefits of the Semantic Web is using RDF as a linkbase. By this view, data stored in RDF can be used by computers for automatic link generation in any hypertext systems, including the Web. This enriches the structure by adding links using external sources of knowledge. This approach has been used in COHSE (Conceptual Open Hypermedia ServicE) [20, 21]. In this service, conceptual metadata about hypermedia documents are used to add pre-computed links to the pre-existing navigational links. It can convert a set of conceptually-unlinked documents like normal Web pages to another set of linked documents. In order to recognize potential link anchor points, the link generator uses several software modules and external linkbases or ontology services, including RDF repositories. A significant functionality appears when a single document can be enriched by several types of knowledge, each for special groups of readers.

There are some other similar works that create human-readable media using the stored facts in the Semantic Web, like in [20]. Conversely, other works focus on providing intelligent processors for hypermedia systems (like building automatic semantic links in [22–24] or hyperstructure-based search methods in [25]). In a different approach, RDFa [26] is a W3C recommendation on how to enriche the HTML elements with RDF metadata. This can be an alternative method to embed some RDF triples inside the HTML links.

2.4 Ternary Links in Hypermedia

While classic hypermedia models (like the Dexter Model [27]) define a link as a two-element object consisting source and destination, 'association' has sometimes been considered as the third element, varying from implicit to explicit involvement. We call such three-element links ‘Ternary Links’ hereafter, though others have also used terms like ‘link typing’ in the Trellis Model of hypertext [28] and FOHM (Fundamental Open Hypertext Model) [29], ‘rich links’ in [30] or ‘meta-level links’ in [31]. Even today’s usual HTML links, where no formal or explicit association exists within a link, the association is embedded somehow in the contents of the document which the link is. This will create a conceptual understanding of the relationship between the source and the destination in the user’s mind [31] that can be either guided by the HTML anchor text, by the mental model associated to a graphical icon, or even by a previous experience of Web navigation. However, HTML also supports generic type linking like ‘class’, ‘rel’ and ‘rev’ attributes of <a> and <link> tag to handle link types, but these are infrequently used and many Web browsers ignore them [32, 33].

3 The Ternary Relation Model

In this section the concept of Ternary Relations Model (TRM) is introduced. TRM is an extension of the BRM described in the previous section. Since RDF will be shown to be a special case of TRM, it is important to formally define TRM here.

3.1 Limitations of the BRM

The Binary Relation Model focused on the four main navigational questions without which relations cannot be fully expressed. However, it represents explicitly nothing else, such as the semantics or types or meaning of links. This information may be implicit in the implementation, for example inspection of a set of links or of a process generating links may indicate its purpose; however, this is not expressed explicitly in the model.

What is semantically lacking from the BRM representations is a discriminator which encapsulates information about the semantics of the relation, such as the relation’s name, type, purpose or selection criteria. While this information is not essential for representing relations fully, it is however useful to varying degrees in different real-world applications. An obvious example is in hypertext itself, where a chooser function can be most helpful when numerous possible links are available from a given source. In another application, it might be useful to perform calculations which take into account the semantics of the relations and relation incidences, for example, finding all elements which satisfy the relation “is-a-kind-of tree” but not “lives-in tree”.

The BRM represents no information that would assist a user in selecting the most appropriate link destination for their purposes. In the BRM, the navigational questions allow one to determine where one can go and from where, but not *why* one would want to go there. The third link element physically presents in the Semantic Web links (as provided by RDF) since computers explicitly need them to traverse the links. In the human-readabile Web with binary links however, the same semantic concept implicitly exists in the page contents and/or in the users mind [31].

3.2 Description of the TRM

The Ternary Relation Model (TRM) extends the BRM with exactly this semantic limitation in mind. A hypermedia system in TRM consists of nodes and relations - where a node is an individual unit of data/information (e.g. a web page, a URI, a data record in a database, etc. same as BRM's endpoints), and a relation (in the TRM context, not to be confused with databases) is an ordered triple of any three nodes acting as source (***s***), association (***a***) and destination (***d***). Formally, the TRM consists of (*N*, *R*) couples where *N* is the set of all nodes, *R* is the set of all relations; and

$$R \subset N^3 \text{i.e.} R \subset \{(s, a, d)|s, a, d \in N\}\,.$$

Relations are maintained in the same framework as the nodes are maintained. So it is natural to define a relation as a node. This not only makes the information structure simpler but also adds the nested relations functionality (i.e. a relation can be any of source, association or destination in another relation). Thus there will be only an atomic structure called node. A node may have some attributes (e.g. id, description or any other metadata nodes e.g. provenance) as well as the identifiers of the source, association and destination of some other three nodes if this node acts as a relation. The identifiers may be replaced by the functions (e.g. URIs of the Web services responsible for generating the end points) in the full functional links mode.

The third feature of a relation is introduced whose purpose is to explicitly represent the semantics of the relation that binds elements. So for example, the BRM's pE representation, specified by a predicate p and expression E is now supplemented with a designator f which can be a relation name or any other semantically-meaningful description, and now becomes a pfE (predicate-function-expression) relation. The BRM enumeration of incidences representation which represents a binary relation as a set of pairs $\{(s_1, d_1), (s_2, d_2), ..., (s_n, d_n)\}$ now becomes $\{(s_1, a_1, d_1), (s_2, a_2, d_2), ..., (s_n, a_n, d_n)\}$ for *R* the relation to which these pairs all belong (Fig. 1).

Then "semantic functional linking" i.e. pfE linking, is the TRM equivalent of pE linking in BRM. By analogy, the most general case is when the relation is represented

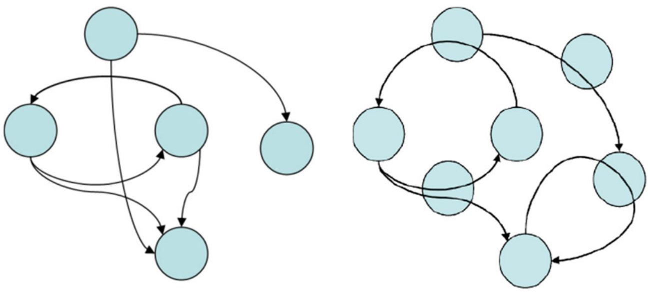

Fig. 1 Binary link structure (*left*) vs. Ternary link structure (*right*)

as a (predicate, expression, expression), or when $R=\{(s, a, d) \mid p(s)=true\ ,\ a=f(s); d=g(s, a)\}$.

This is a general case when an unadvertised source of a link can be filtered by the *p()* function, then the available associations are computed by knowing the source, and finally the available destinations are computed by knowing the source and the nominated associations. By analogy to the BRM, this state can be called semantic functional linking and using the same arguments as in [4] can support theoretical satisfaction of Turing-completeness and interaction. The proof is extensible because the BRM is a special case of the TRM where the association is fixed. Thus a hypertext system that implements pfE linking is Turing-complete, which it inherits from pE linking.

The TRM is a general relation framework which can incorporate many other relation frameworks including RDF. Like RDF, it also does not provide any specific language for implementation. Since RDF is the triple arrangement of URIs in the form of subject-predicate-object, the instant conclusion from the TRM definition is that RDF can be described in the TRM and thus RDF is a sub-framework of the TRM. Where functional links are not necessary, TRM inherits any RDF specification including complex RDF forms e.g. RDFS or OWL.

The main principles of the TRM are:

1. TRM describes the relations (or links) between nodes.
2. A relation is an ordered triple of three elements: "source", "association" and "destination".
3. A relation can itself be a node, i.e. nested relations are possible.
4. Relations in TRM are bidirectional, which means that a single relation can be read from both directions.
5. Each of the three elements of a relation can be a (dynamic) function of another.

Having added information to the representations means that additional questions can now be asked about the relation. Considering the "forwards" direction of linking for example, we can now ask:

a. Can I go anywhere from here?;
b. If so, where can I go from here?; and
c. From my options, which should I choose?

Similarly the "backwards" direction is supported, so that of available sources from a given destination, a selection of the most appropriate is possible.

Having introduced this semantic component to relations has benefits for many hypertext systems. For example, the functional linking in [15] which was found to be "as powerful as computing" in [4] maps into the TRM as a pfE representation, so that we now have "semantic functional linking" with transparent semantics to assist users in link choice. It still however retains its Turing-completeness, its ability to operate over infinite data, its ability to select out subsets of the relation incidences without full computation, and still retains the ability to translate into any of the enumerated representations.

3.3 Examples of TRM/RDF Links

The following examples illustrate pfE relation management in TRM and show how hypertext relation incidences (i.e. links) can be managed.

As captured in Fig. 2, the body text of an email has been searched by “Yahoo! Shortcuts” to find out the potential link anchors, with the found anchors being distinguished by dashed underlines. Hovering the mouse pointer over these special links results in a menu of choices which are dependent on the nature of the anchor appearing. These choices are not showing the associated destination, but rather are showing the link “types”, i.e. relation names, that are relevant to that source. The user then will be directed to other Web pages by selecting each menu choice.

The whole process is enabled with pfE links. A predicate *p()* determines whether each word is an anchor (in this case it is pre-computed). Then a function *f()* will produce menu choices (equivalent to association names or types) relevant to the anchor text (either pre-computed or dynamically computed) and finally another function *E()* will determine the destination Web page after selecting a relation type, having the source and the association (dynamically computed).

In this example, the TRM pfE functions are:

1. Is the word “Malaysia” found in a linked database?
   p(“Malaysia”)=True.
2. Which associations are originating from “Malaysia”?
   f(“Malaysia”)= Map, Web search.
3. Where can I go by selecting “Malaysia” and “View Map”?
   E(“Malaysia”,“the SemanticWebView Map”)= An updated Webpage about Malaysia map (A parametric URL)
   E(“Malaysia”,“Search”)= Yahoo search Webpage for “Malaysia” (Another parametric URL)

However this is but one simple example—there are many possibilities for defining functionally-similar relations. The choice of actual functions is up to the relation creator who may prioritise relation accuracy over response time, for example.

As another example, we consider an online real estate photo browsing system. The information has been arranged as semantic triples (in RDF). For example, statements such as “property x is owned by y” or “property x is in area z” are been stored in the system as the triples (x, owned by, y), (x, in area, z). A web page has been

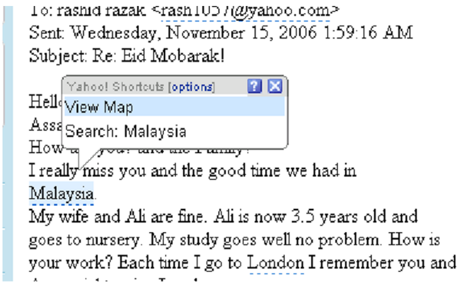

Fig. 2 TRM links in “Yahoo! Shortcuts”

designed that contains the property photos, possibly with some conventional HTML links as well as some added Ternary Links. The source of each of the ternary links is computed to be the photo of a property, and the association, together with the available link destination, will be displayed whenever the mouse pointer is over a specific image. In this way, the structure of available links originating from the current object is made accessible to the user. The semantic relation between the source and the destination will be made apparent before the link is traversed (Fig. 3). It is noticeable that although in this example the links are primarily stored in a RDF repository, they are processed and dynamically changed according to the users' context.

Where accuracy was critical, it is possible that the predicate *p()* would actually be a Boolean conjunction of the results of the function *f()*, so that if at least one relation type is found for a given potential source, then the predicate returns true, i.e. that the potential source is indeed a source, because an association (link type) was found for it.

So clearly, hypertext relations and relation incidences (links) can be managed with pfE. We now turn to the management of relations in the Semantic Web.

3.4 What This Means for RDF

As discussed in Section 1, Moreau and Hall [4] propose that a hypertext system (called a "hypertext machine") can emulate any finite computation of a Turing machine only if the system is able to implement the functional links. In other words, linking is Turing-complete in that situation. The similar approach in the automaton look at the hypertext systems has also been taken in [34, 35] where the terms "Web Machine" and "Linked Data Machine" have been used. As also discussed in Section 1, RDF is a trivial link-base choice when a semantically-enriched hypertext system (e.g. the Semantic Web) is developed. Having these two assumptions, if RDF cannot act as a functional link-base, the hypertext machine is not Turing-complete. Since by definition, a hypertext machine has not other computational power source than its link-base, the link-base (here RDF) should have provided the necessary computational power.

Fig. 3 Samples of the functional TRM link (pfE) in a personalized real estate website

The above theoretical conclusion practically means that we can now analyse the relative strengths and weaknesses of RDF and consider possible alternative relation representations which could achieve or exceed the same aims.

Hypertext links are directly analogous to RDF triples in terms of their creation and maintenance, and in fact often hypertext links are used to represent a semantic relation. We find that all the challenges of creating and maintaining relation participants, whether links or triples, are the same, and that the analysis of BRM linking in [11] is directly applicable to TRM. The following benefits counted in [11] are similarly achieved in TRM:

a. *Creation*: making new links/triples easily without too much human input especially on large data collections;
b. *Soundness* [36]: all links/triples are semantically consistent, no false positives;
c. *Completeness*: all links/triples that ought to exist are in fact made, i.e. no false negatives;
d. *Correctness*: maintaining technical accuracy of links/triples in the context of volatility within the underlying data collection. (i.e. error 404 occurrences);
e. *Infinite relations*: being able to define relations without creating them in full allows the calculation of a relevant subset of an infinite relation. For example, any measuring system that uses the real numbers cannot be calculated in full, so that relations such as "distance-to" must be discretised in an enumerated representation, but could be more accurately represented with a computation; and
f. *Targeted subsets*: selecting a small but a relevant subset of relation incidences without computing the entire relation.

The next section examines how this can be achieved for RDF.

4 Towards a Turing-Complete Semantic Web

A central premise of this work is that RDF may be viewed as one of the representations within TRM. A consequence of this is that TRM's pfE representation of semantic relations could potentially be used as either an alternative or a supplement to RDF's triples to be used in the Semantic Web. There clearly is scope to use TRM's pfE representation of semantic relations as an alternative or as a supplement to RDF's triples.

There are two immediately obvious amendments that could use pfE to enhance RDF, both backwards-compatible with current RDF:

1. *Automation*: the minimalist solution is to build software to automatically create and maintain triples in much the same way as links, with all the same considerations for pre-computation of links. This is not actually doing anything to change the Semantic Web software or the way triples are received and interpreted, although it will change the way triples are created and perhaps included in documents.

2. *Dynamic RDF*: to enable RDF to represent semantic relations in any of the TRM forms, but most importantly to expand from having just simply enumerated triples to having triples plus relations represented with pfE. This has the benefits of enabling dynamic relation management so that triples can be created "on the fly" and in response to the context.

The former option is essentially what occurs in those applications that use RDF as a link-base (see Section 2.4).

The latter option requires a more radical alteration to the RDF management software which now must be able to invoke calculations and receive results. However, it is not so far-fetched when we consider that the Web does exactly that at present, with servers calling on external applications to perform calculations even in simple situations such as CGI and PHP scripts. In a service-oriented architecture approach, a practical solution to this is to have web resources for on-the-fly link creation so the hypertext systems can use them as dynamic pfE link generators instead of the fixed RDF datasets.

Dynamic RDF (D-RDF) introduced in [37] is a solution towards the later option. D-RDF is an extension of the RDF XML vocabulary that supports basic programming structures and user-defined functions. As a result it works as a programmable shell on top of the static RDF which makes RDF triples on-the-fly by request. Although the programming still operates on the existing static data, D-RDF is a step forward a fully computational RDF extension.

In a closer look, it can be argued that the necessary computability shall be desired at the level of querying language not at the data level. A supporting analogy to this argument is the role of SQL regarding the relational databases. The evolution of SQL from the command-line style to the embedded procedural languages like PL/SQL has consequently added programming functionalities to the databases. Examples are producing dynamic views and pivot tables. SPARQL is the current W3C standard language for querying RDF [38]. SPARQL is not currently procedural so it can be argued that if the evolution of SPARQL can lead to a procedural language, then RDF can be wrapped in a programmable shell. For example SPARQLog, as introduced in [39] can be modelled by a Turing Machine.

In response, we notice that at present, the triples are generated and maintained in RDF outside of their usage context, which comes back to the enumerated nature of the RDF's URIs. A query language shell which works on an enumerated set of URIs, will deliver enumerated URIs too. In other words, the Turing-completeness of the query language exists in a different context than the context of the hypertext machine. For realising the pfE type of functional links, both pre-computed and on-the-fly generated links are required. This functionality cannot be delivered by querying a finite set of enumerated links.

5 Discussion

It is important that the Semantic Web be made Turing-complete. RDF has become much more than the metadata language that it was originally conceived as, and is now the backbone of a significant amount of Web development. Limitations in the power of RDF as it stands now will limit future Semantic Web development.

RDF is essentially at the same stage as hypertext systems, including the Web, were in during the early 1990s, where everything had to be explicitly named in order to be related. Since then, hypertext systems and the Web have progressed beyond this with dynamic computation of scripts which were in effect pE links. Relations, in hypertext systems at least, have become both process and data, depending on which of the relation representations are used.

Now we propose that it is time to do the same with the Semantic Web relations. There is a previously-existing model of binary relations, the BRM, which analyses link implementations as binary relations and diagnoses their strengths and weaknesses. If we extend that model to incorporate the notion of explicit naming/meaning of relations into the TRM, then it can be equally well applied to semantic relation management, such as in the Semantic Web.

The costs and benefits have been studied within hypertext and are known [12], and all that remains is to map the procedures onto RDF relations to gain the same benefits.

We find that RDF maps exactly into one of the representations in the TRM. This immediately suggests that the many benefits of alternative relation representations that were found in hypertext management could beneficially be applied to semantic relation management in the Semantic Web.

Considering then the management of semantic relations, there is a clear benefit from augmenting the forms of relation representations available to the Semantic Web applications. Conceptually, this augmentation is very simple indeed, and consists of adding another form of relation representation, the pfE representation, to the Semantic Web applications. In fact, it needs not even be directly added to existing applications, accessed by many, but rather could follow the open hypertext systems model of publishing sets of ready-made relations which anyone can use but are managed by the owner [40].

At its most promising, the many benefits of the pfE representation in managing the semantic relations include the ultimate in flexibility—the ability to define relations as processes, not just as data. By enabling dynamic relation management to supplement RDF's enumerated relation management, we have now gained criteria-based relation management, with many useful features, as follows.

Users frequently want to classify data according to their "nature" and thus allocate them to relations on that basis. Algorithms that specify the selection of elements for participation in a relation are obvious improvements over the manual creation of relations in such cases. Such improvements include the automation of relation incidence creation, with the attendant benefits of relation *soundness* (no false positives) and relation *completeness* (no false negatives), as well as reducing human intervention.

As a result of this, *personalisation* of relations to a user's requirements becomes simple. Every time new data are exposed to a relation's selection process, it can be assessed for its eligibility automatically. Thus users can define their own relations, based on selection criteria relevant to their own purposes, and apply them to whatever data they encounter. The example is Section 3.3 could have different pfE relations for each user, so that one user could have links from “Malaysia” but another could have links emanating from family member names instead.

Also, users often want to subsequently *search* for relations or incidences from relations based on their “nature”. When the selection is already a part of the process to make relation incidences, the process is simplified.

More importantly, it is possible to perform this selection as a dynamic process. This is important not only because of avoiding unnecessary computation, but also because it gives relation *correctness*. A relation or incidences calculated upon request will be correct at the time of the request, more likely so than something that was correct at some earlier calculation date [12].

Another benefit of the dynamic computation is that it bypasses the need to create or manifest all of the incidences in the relation, yet the required instances are still available. Dynamic computation using pfE enables this, with the predicate part of the computation selecting out the relevant subset of relation incidence sources according to the appropriate criteria.

This creation of targeted subsets of relations can be useful whenever there is no need to create all incidences in a relation just to access a subset of them. In particular this is useful in a volatile data environment, where relation correctness potentially becomes an issue as relation elements themselves change in terms of the selection criteria.

It also means that it becomes possible to represent relations that are infinite. Relations that involve infinite sets such as the real numbers either must be discretised at the cost of accuracy, or never manifested in full. The latter option becomes feasible with selection-based relation management.

6 Conclusion

The management of RDF triples is reducible to the management of ternary relations and manifests many of the same issues as are found with the management of hypertext links. The Semantic Web is currently based on massive RDF repositories that are just enumerating the knowledge facts, while they are better to be based on a dynamic and personalized knowledge web.

The Semantic Web is in some respects in the same position as hypertext systems were when only point-to-point linking was available. The problem at hand is essentially the same – how to manage the creation and maintenance of large numbers of relations and relation incidences over a changing, many-owner collection of information. There are differences in the purpose and usage of those relations, but this does not affect the technical challenges of managing them.

Casting relations into a model helps us to forget the superficial differences in the use of relations, and to apply lessons learned from one relation-management situation into another.

What we find from the management of hypertext links is that a conceptually simple change in the way we represent relations can significantly alter the capability of hypertext systems, in some cases translating what is in some representations nothing more than a data structure into what becomes, in another representation, a process, and thereby gaining all the advantages of programmability without sacrificing the ability to render those processes back into data structures as needed. This can also be done with semantic relations. So in relation management, it really is possible to have one's cake and eat it.

Acknowledgments We would like to thank Dr. Helen Ashman for her great contributions to this work. The very useful comments from anonymous reviewers are also highly appreciated.

References

1. Nürnberg, P.J., Ashman, H.: What was the question? Reconciling open hypermedia and World Wide Web research. In: Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia: Returning to Our Diverse Roots. ACM Press, Darmstadt (1999)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Sci. Am. **284**(5), 34–44 (2001)
3. Ashman, H.: Electronic document addressing: dealing with change. ACM Comput. Surv. **32**(3), 201–212 (2000)
4. Moreau, L., Hall, W.: On the expressiveness of links in hypertext systems. Comput. J. **41**, 459–473 (1998)
5. Mili, A.: A relational approach to the design of deterministic programs. Acta Informatica **20**(4), 315–328 (1983)
6. W3C: Resource Description Framework (RDF): Concepts and Abstract Syntax. Feb 2004]; Available from: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/ (2006)
7. W3C-Working-Group: Defining N-ary Relations on the Semantic Web. June 2006]; Available from: http://www.w3.org/TR/swbp-n-aryRelations/ (2006)
8. Recommendation, W.C.: RDF Test Cases. June 2006]; Available from: http://www.w3.org/TR/rdf-testcases/ (2004)
9. W3C: Turtle, Terse RDF Triple Language. July 2013]; Available from: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/ (2013)
10. Berners-Lee, T.: Notation 3 (N3). June 2006]; Available from: http://www.w3.org/DesignIssues/Notation3 (1998)
11. Ashman, H.: Relations modelling sets of hypermedia links and navigation. Comput. J. **43**, 345–363 (2000)
12. Ashman, H.: Theory and Practice of Large-Scale Hypermedia Systems. PhD Thesis, The Royal Melbourne Institute of Technology, Australia (1997)
13. Ashman, H., Garrido, A., Kukkonen, H.O.: Hand-made and computed links, precomputed and dynamic links. In: Proceedings of Hypermedia - Information Retrieval - Multimedia '97 (HIM '97). Dortmund (1997)
14. Hall, W., Hill, G., Davis, H.: The microcosm link service. In: Proceedings of the Fifth ACM Conference on Hypertext. ACM Press, Seattle (1993)
15. Verbyla, J., Ashman, H.: A user-configurable hypermedia-based interface via the functional model of the link. Hypermedia **6**(3), 193–208 (1994)
16. Berners-Lee, T., Fischetti, M., Foreword By-Dertouzos, M.L.: Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor. HarperInformation (2000)
17. Berners-Lee, T.: Semantic Web Road Map. W3C Electronic Journal (1998)
18. Antoniou, G., van-Harmelen, F.: A Semantic Web Primer. MIT Press, Cambridge (2004)
19. Palmer, S.B.: The Semantic Web: An Introduction. 15/08/2005]; Available from: http://infomesh.net/2001/swintro/ (2001)
20. Goble, C., et al.: Conceptual open hypermedia = the semantic web?. In: The Second International Workshop on the Semantic Web. Hong Kong (2001)
21. Carr, L., et al.: Conceptual linking: ontology-based open hypermedia. In: Proceedings of the 10th International Conference on World Wide Web. ACM Press, Hong Kong (2001)
22. Bhaumik, A., et al.: Towards hypermedia support for database systems. In: Proceedings of the 34th Hawaiian International Conference on System Sciences. IEEE, Hawaii (2001)

23. Bieber, M.: Dynamic Hypermedia Engine (D.H.E). CIS Department, New Jersey Institute of Technology 15/08/2005]; Available from: http://www.cis.njit.edu/~bieber/dhe-overview.html (2000)
24. Chen, X., et al.: Integrating Web Systems through Linking. www2003.org 15/08/2005]; Available from: http://www2003.org/cdrom/papers/poster/p145/p145-chen.htm (2004)
25. Qiu, Z.: Hyperstructure-Based Search Methods for the World Wide Web. PhD Thesis, Darmstadt Technical University, Denmark (2004)
26. W3C: RDFa 1.1 Primer, Rich Structured Data Markup for Web Document. July 2013]; Available from: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/ (2012)
27. Halasz, F., Schwartz, M.: The Dexter hypertext reference model. Commun. ACM **37**(2), 30–39 (1994)
28. Stotts, P.D., Furuta, R.: Adding browsing semantics to the hypertext model. In: Proceedings of the ACM Conference on Document Processing Systems. ACM (1988)
29. Millard, D.E., et al.: FOHM: a fundamental open hypertext model for investigating interoperability between hypertext domains. In: Proceedings of the Eleventh ACM Conference on Hypertext and Hypermedia. ACM Press, San Antonio (2000)
30. Oinas-Kukkonen, H.: What is inside a link? Commun. ACM **41**(7), 98 (1998)
31. Takahashi, K.: Metalevel links: more power to your links. Commun. ACM **41**(7), 103–105 (1998)
32. Bieber, M., et al.: Fourth generation hypermedia: some missing links for the World Wide Web. Int. J. Hum. Comput. Stud. **47**(1), 31–65 (1997)
33. Shum, S.B.: The missing link: hypermedia usability research and the web. ACM SIGCHI Bulletin **28**(4), 68–75 (1996)
34. Abiteboul, S., Vianu, V.: Queries and computation on the Web. In: Database Theory—ICDT'97, p. 262–275. Springer (1997)
35. Hartig, O.: SPARQL for a web of linked data: semantics and computability. In: The Semantic Web: Research and Applications, pp. 8–23. Springer (2012)
36. Thistlewaite, P.: Automatic construction and management of large open webs. Inf. Process. Manag. **33**(2), 161–173 (1997)
37. Jain, K.: D-RDF: Dynamic Resource Description Framework. ProQuest (2007)
38. W3C: XML Linking Language (XLink) Version 1.0. 17/07/2008]; Available from: http://www.w3.org/TR/xlink/ (2001)
39. Bry, F., et al.: SPARQLog: SPARQL with rules and quantification. In: Semantic Web Information Management, pp. 341–370. Springer (2010)
40. Carr, L., et al.: The distributed link service: a tool for publishers, authors and readers. In: Proceedings of the 4th International WWW Conference. Boston (1995)

Copyright of Theory of Computing Systems is the property of Springer Science & Business Media B.V. and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.