**University of Aberdeen**

**A Policy-Based Approach to Context Dependent Natural Language Generation**

Bouttaz, Thomas; Pignotti, Edoardo; Mellish, Christopher Stuart; Edwards, Pete

# A Policy-Based Approach to Context Dependent Natural Language Generation

**Thomas Bouttaz, Edoardo Pignotti, Chris Mellish, and Peter Edwards**
Computing Science, University of Aberdeen,
Aberdeen AB24 5UA, UK
{t.bouttaz, e.pignotti, c.mellish, p.edwards}@abdn.ac.uk

## Abstract

This paper presents a method for tailoring Natural Language Generation according to context in a web-based Virtual Research Environment. We discuss a policy-driven framework for capturing user, project and organisation preferences and describe how it can be used to control the generation of textual descriptions of RDF resources.

## 1 Introduction

Adaptive interfaces change the style and content of interaction according to the context of use. In particular, adaptive hypertext (O'Donnell et al., 2001) adapts the content and form of natural language text. Systems like this introduce the need for a good model of the context and how it influences language. This context can, in general, include aspects of the user themselves, general aspects of the situation and also the task the user is currently performing. Many interactive systems use sets of attribute-value pairs to implement the user and general context models. They then encode the method of decision making in each task context, this taking into account the information in the two models (e.g. Savidis et al., 2005). We are investigating a different approach where the representation of user (coming possibly from several sources), general context and task context are combined in a declarative way through the construction of *policies*. In this approach, preferences are expressed in terms of obligations, prohibitions and permissions, possibly arising from different sources, using semantic web ontologies. Combining information from multiple sources has been used in user

modelling (Kobsa, 1993) and there has also been some use of ontologies in user modelling (Hatala and Wakkary, 2005), but ours is the first system that uses semantic web ontologies for the encoding of all user actions, task contexts, permissions and user preferences.

Although policies can be used to control a number of aspects of adaptation, here we concentrate on their use within Natural Language Generation (NLG), mainly for content determination. In general NLG is often conceived as being responsive to multiple *goals* or *constraints* (e.g. Hovy, 1990). In addition, the content and form of a generated text often needs to be tailored to at least certain aspects of the user (Paris, 1988; Bateman and Paris, 1989). However, not many general mechanisms have been presented for dynamically combining different aspects of the context for guiding NLG. Plan-based tailoring (Paris, 1988; Paris et al., 2004) might provide part of such a mechanism, but it assumes a top-down approach to text planning, which is not natural for applications that just have to express some of what happens to be there in the input data (Marcu, 1997). Requirements on style, syntax, content, etc. can all be expressed and combined in constraint-based NLG (Piwek and van Deemter, 2007), but existing implementations only use general constraint-satisfaction mechanisms for particular parts of the generation problem. Generation based on Systemic Grammar (Bateman, 1997) provides a clear mechanism for decision-making and tailoring (Bateman and Paris, 1989) but is less clear on the representation of context. In generation by classification (Reiter and Mellish, 1992), contexts are complex ob-

jects classified into an ontology. Aspects relevant to particular generation decisions are then inherited according to where the context has been classified. Although this is elegant in theory, in practice, such ideas are now used more as part of object-oriented programming approaches to NLG (White and Caldwell, 1998). It thus remains to be seen both to what extent declarative representation of contexts and NLG decision making is possible, and also to what extent control of NLG can use similar mechanisms to other types of adaptation. The current work can be seen as further exploration of this territory.

In this paper, we report on policy-driven control of NLG as we have integrated it in a Virtual Research Environment (VRE) called ourSpaces[1]. This system has been developed to facilitate collaboration and interaction between researchers by enabling users to track the provenance of their digital artifacts and processes, and to capture the provenance around a user's social network, e.g. activities within the environment, relationships between members, and membership of projects and groups. Provenance (also referred to as lineage or heritage) aims to provide additional documentation about the processes that led to the creation of an artifact. Within this environment, a short textual description of an artifact, person or project can be valuable to a user. We have developed an NLG service to generate text descriptions of those resources based on the RDF metadata held by the system. This service has to perform "ontology verbalisation" (i.e. translate ontology fragments into natural language), a topic on which there has been much previous research (e.g. Sun and Mellish, 2007; Power and Third, 2010). Our own approach builds on the system of Hielkema (2010). However, work on ontology verbalisation has not yet presented general mechanisms for content determination from semantic web data. This paper discusses how policies can be used to tailor the content selected for an NLG service like ours, so that it adapts according to the context of use.

## 2 Capturing Context

Underpinning the VRE is a rich and pervasive RDF (Klyne and Carroll, 2004) metadata infrastructure built upon a series of OWL ontologies (McGuin-

ness and van Harmelen, 2004) describing aspects of the provenance of digital artifacts, projects, organisations, people and social networking activities. Through our experience with a number of case-study groups we have identified three dimensions that together characterise the context used to generate text descriptions:

**The provenance of the resource being described.** At the core of the VRE is a representation based on the Open Provenance Model (OPM) (Moreau et al., 2011). OPM provides a specification to express data provenance, process documentation and data derivation. It is based on three primary entities namely *Artifact*, *Process* and *Agent* and associated causal relationships namely *used*, *wasGeneratedBy*, *wasTriggeredBy*, *wasDerivedFrom* and *wasControlledBy*. The context behind the description of a digital resource is provided by a provenance ontology developed in OWL, which defines the primary entities of OPM and additional provenance ontologies which extend the concepts defined in the OPM ontology with domain-specific classes (see Figure 1 top).

**The user's social context.** In the VRE, the link between the social network and digital artifacts is established formally, by the integration of the FOAF social networking vocabulary (Brickley and Miller, 2010) with our provenance ontologies. FOAF characterises an individual and their social network by defining a vocabulary describing people, the links between them and the things they create and do. Moreover, we have extended our framework to allow links between people and projects, groups and organisations (see Figure 1 bottom-right).

**Specific user, project, organisation and system policies.** Within our system, users and their behaviours are managed by enforcing certain policies. Policies can be created by the user, by an administrator of a project, group or organisation, or by a system developer. For example, a user may impose certain access constraints on digital artifacts that they own, e.g. certain information about the artifact may only be accessible to users who are members of a particular project and who contributed towards the artifact itself. A project might also be required to archive artifacts to the UK Social Science

Data Archive (UKDA) [2] and follow certain documentation requirements. More specifically, a policy may be created by the Principal Investigator of the project and addressed to its members specifying that certain information about an artifact has to be provided during the upload.

In the VRE we define such policies as a combination of *Obligation*, *Prohibition* or *Permission* instances described by the properties *hasObligation\**, *hasProhibition\** and *hasPermission\** in the ontology illustrated in Figure 1 bottom-left). Each *Obligation*, *Prohibition* or *Permission* has an associated set of *Condition* instances. A condition in our ontology is a combination of a subject (an *opm:Artifact* or an *opm:Process*) and a rule describing the condition (see Figure 3 and 4).
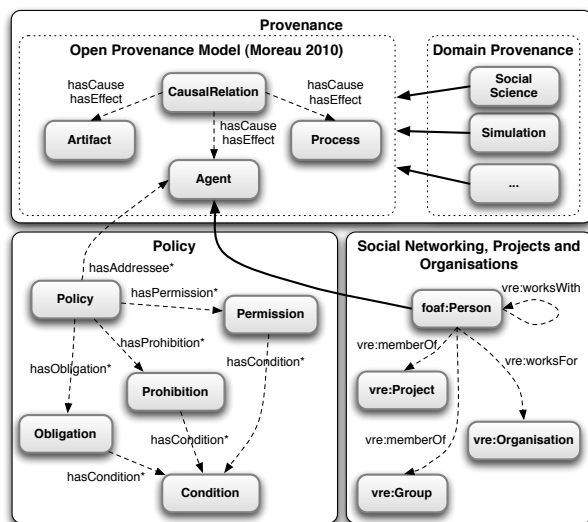


Figure 1: Capturing context in the VRE.

## 3 Generating Context-Dependent Text Descriptions

In order to enable collaboration between researchers, the VRE makes use of a number of repositories and services to store research resources, and offers a number of tools to manage and visualise such resources (see Figure 2). One of the most important components of the VRE is a *Text Generator* service which is able to generate short textual descriptions from the RDF metadata associated with

resources stored in the *Metadata Repository* (e.g. title, author, date of publication). In order to generate the text, we have implemented a RESTful service that invokes a *Text Generator* service based on the RDF ID of the resource being described, passed as a parameter by the Web interface. This service generates text containing a description of the resource using a deep model of the syntactic structure of sentences and their combinations, inspired by the work of Hielkema (2010).
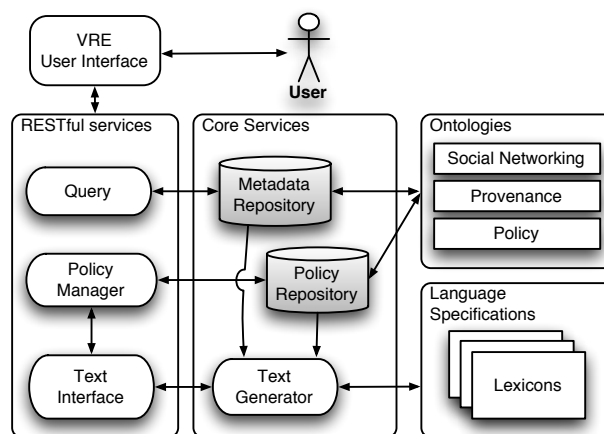


Figure 2: Architecture enabling context-dependent NLG.

The *Text Generator* builds an internal RDF model of the resource being described by querying the *Metadata Repository*. The text is then produced by converting axioms inside the model to plain text using the appropriate language specifications. A language specification is composed of a set of lexicons encoded in XML which describe how to render the text corresponding to a RDF property (e.g. syntactic category, source node, target node, verb tense). For example, if the property *transcribedBy* of a resource of type *Transcript* has a value of *"Thomas Bouttaz"*, the XML file corresponding to that property will specify that this information must be rendered as: "It was transcribed by Thomas Bouttaz" (see Figure 5 left). By following the hyperlinks available in the resource description, the user is then able to expand the text to access more information about related resources. For instance, in this example the user can click on the hyperlink *Thomas Bouttaz* to get more information about that person. This is done by invoking the *Text Generator* service with the ID of the

RDF representation of that person. The description returned by the service is then appended to the original text by the *Text Interface*.

Due to the complexity of metadata associated with a resource, context plays a vital role in supporting the selection of information to be displayed to the user. Using policies, it is possible to enforce context-dependent preferences while the text is being generated by the *Text Generator*. This is achieved in our framework by invoking the *Policy Manager* which implements a policy reasoning service based on the ontology described in Figure 1 bottom-left. Our framework is composed of a repository storing RDF triples representing policies, and a provenance policy reasoner based on the TopBraid SPIN API (Knublauch et al., 2011). In our framework, before realising the descriptive text of a resource, policies are checked against the model containing the RDF graph. The *Policy Manager* checks if any of the policies stored in the *Policy Repository* can be activated by the current RDF model by running the SPIN reasoner against the rules associated with the policies.

To illustrate the use of policies within the VRE, consider an example where the Principal Investigator of a project needs to make sure that confidential information in his project is protected. This can be achieved by constructing a policy with a set of rules similar to the one shown in Figure 3.

```
CONSTRUCT {
   _:b0 a spin:ConstraintViolation .
   _:b0 spin:violationRoot ?process .
   _:b0 spin:violationPath pggen:location .
   _:b0 spin:violationPath pggen:hasStartDate .
   _:b0 spin:violationPath pggen:hasEndDate .
}
WHERE {
   ?artifact pggen:wasGeneratedByInfer ?process .
   NOT EXISTS {
      ?artifact pggen:producedInProject ?project .
      ?project project:hasMemberRole ?role .
      ?role project:roleOf [USER_ID] .
   }.
}
```

Figure 3: Rule protecting confidential information of process artifacts.

The rule presented in Figure 3 specifies that it is not possible to view location, start date and end date of the process that generated a resource, unless the user is a member of the project which produced that artifact. Similarly, another rule could protect the identity of the person that transcribed an artifact. On the other hand, an individual user might want to express his preferences regarding what information is rendered in the textual description of a resource. For instance a user could declare that he is not interested in knowing who deposited a resource if that person is already part of his social network.

When the user requests a textual description of a resource, the VRE detects if certain policies are activated depending on the context surrounding the user and the resource being described. If policies are active, the *Text Generator* service takes into account the constraints associated with such policies. If a violation is detected, the service will remove the information described by the *spin:violationPath* property from the internal RDF model describing the resource. Therefore when the realiser generates the text from the model, those details will be omitted.

While this example demonstrates how the system can remove axioms associated with private information, this framework also allows users to add information to the description. For instance in the previous example, the Principal Investigator might want to express that if a user non-member of the project tries to generate a description of a protected resource (in this case an interview process), the description should include information about who to contact to obtain access to that artifact (e.g. the email address of the PI of that project). Again, this preference would be represented by a policy associated with a rule indicating where to retrieve those information in the RDF repository, and where to add them in the internal model.

```
CONSTRUCT {
   ?artifact nlg:forObtainingAccess ?mbox .
}
WHERE {
   ?artifact pggen:producedInProject ?project .
   ?project project:hasMemberRole ?role .
   NOT EXISTS {
      ?role project:roleOf [USER_ID] .
   } .
   ?role a project:PrincipalInvestigator .
   ?role project:roleOf ?pi .
   ?pi foaf:mbox ?mbox .
}
```

Figure 4: Rule adding information about who to contact for obtaining access to an artifact, for project non-members.

The rule shown in Figure 4 adds a *nlg:forObtainingAccess* property to the local model representing the artifact being described, if the user asking for that description is not a member of the project which produced that artifact. This property is defined in a utility ontology only used by the NLG service. In this manner that service is able to retrieve information from the repository, and to locally generate a different model, more adapted to the user's context.

The example in Figure 5 shows two text descriptions of the same interview transcript. On the left-hand side, the description is generated for a user member of the project in which the transcript was produced. On the right-hand side, the description is generated for a non-member who has expressed that he is not interested in information about users in his social network.
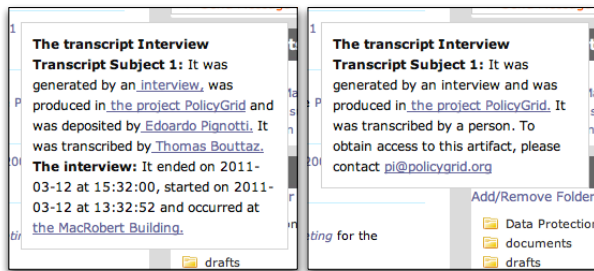


Figure 5: Two examples of text descriptions about the same transcript.

Using this framework it is possible to declare policies that apply to different contexts involving users, projects, organisations. Context may also include which VRE page the user is currently browsing. By taking into account all of these factors, this architecture allows tailored content determination for the generation of resource descriptions.

## 4 Conclusions & Future Work

In this paper we have presented a software architecture able to deliver context-dependent textual descriptions of resources described by RDF metadata. This architecture has been developed to work in a VRE to provide a tool for researchers to explore the provenance of research artifacts. Due to the volume of metadata associated with a resource in the VRE, we argued that context plays a vital role in support-ing the selection of the information to be displayed to the user. We have identified three factors to determine context: a) the provenance of the resource being described; b) the user's social context; c) specific user, project, organisation and system policies.

We discussed how policy reasoning could be used to provide a flexible mechanism to define and enforce context-dependent preferences. We presented an example where the textual description of an interview transcript was tailored to the user context to assure that confidential information about the interview was only disclosed to members of a specific project. In our future work we plan to investigate other ways in which context could be used to influence the generation of text. For example, how descriptions of resources could be generated depending on different user's domain vocabularies. Moreover, we plan to investigate other ways in which the context representation described here can influence the system in general.

Regarding scalability, conflicts may arise between policies. Therefore we need to use conflict resolution strategies, such as using ranks highlighting the level of importance of different policies. In this manner, the *Policy Manager* would be able to determine how to order several conflicting policies applying to a particular resource. To determine that two policies may conflict, we plan on using techniques similar to the ones proposed by Şensoy et al. (2010). Moreover regarding usability, we need to implement a system that would allow users to easily create SPIN rules representing their policies, possibly using NLG.

Finally, we need to evaluate the extent to which the techniques presented in this paper actually enhance the user's ability to perform tasks using the VRE. We plan to do this by comparing the use of the main system with the use of versions that have specific features (NLG service, policy-driven NLG service) disabled, following a similar methodology to that used by Hielkema (2010).

## Acknowledgments

## References

John A. Bateman. Enabling technology for multilingual natural language generation: the kpml development environment. *Nat. Lang. Eng.*, 3:15–55, March 1997.

John A. Bateman and Cecile Paris. Phrasing a text in terms the user can understand. In *Proceedings of the 11th international joint conference on Artificial intelligence - Volume 2*, pages 1511–1517, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

Dan Brickley and Libby Miller. FOAF vocabulary specification. Technical report, W3C, 2010.

Murat Şensoy, Timothy J. Norman, Wamberto W. Vasconcelos, and Katia Sycara. Owl-polar: semantic policies for agent reasoning. In *Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I*, ISWC'10, pages 679–695, Berlin, Heidelberg, 2010. Springer-Verlag.

Marek Hatala and Ron Wakkary. Ontology-based user modeling in an augmented audio reality system for museums. *User Modeling and User-Adapted Interaction*, 15:339–380, August 2005.

Feikje Hielkema. *Using Natural Language Generation to Provide Access to Semantic Metadata*. PhD thesis, University of Aberdeen, 2010.

Eduard H. Hovy. Pragmatics and natural language generation. *Artif. Intell.*, 43:153–197, May 1990.

Graham Klyne and Jeremy J. Carroll. Resource description framework (RDF): Concepts and abstract syntax. World Wide Web Consortium, Recommendation REC-rdf-concepts-20040210, February 2004.

Holger Knublauch, James A. Hendler, and Kingsley Idehen. SPIN - Overview and Motivation. Technical report, W3C, 2011.

Alfred Kobsa. User modeling: Recent work, prospects and hazards. In M. Schneider-Hufschmidt, T. Kühme, and U. Malinowski, editors, *Adaptive User Interfaces: Principles and Practice*, pages 111–128. North-Holland, Amsterdam, 1993.

Daniel Marcu. From local to global coherence: A bottom-up approach to text planning. In *Proceedings of the 14th National Conference on Artificial Intelligence*, pages 629–635, 1997.

Deborah L. McGuinness and Frank van Harmelen. Owl web ontology language overview. Technical Report REC-owl-features-20040210, W3C, 2004.

Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, Beth Plale, Yogesh Simmhan, Eric Stephan, and Jan Van den Bussche. The open provenance model core specification (v1.1). *Future Gener. Comput. Syst.*, 27:743–756, June 2011.

M. O'Donnell, C. Mellish, J. Oberlander, and A. Knott. Ilex: an architecture for a dynamic hypertext generation system. *Nat. Lang. Eng.*, 7:225–250, September 2001.

Cécile Paris. Tailoring object descriptions to a user's level of expertise. *Comput. Linguist.*, 14:64–78, September 1988.

Cécile Paris, Mingfang Wu, Keith Vander Linden, Matthew Post, and Shijian Lu. Myriad: An architecture for contextualized information retrieval and delivery. In *in AH2004: International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*, pages 205–214, 2004.

Paul Piwek and Kees van Deemter. Generating under global constraints: the case of scripted dialogue. *Research On Language and Computation (ROLC)*, page To Appear, 2007.

Richard Power and Allan Third. Expressing owl axioms by english sentences: dubious in theory, feasible in practice. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 1006–1013, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Ehud Reiter and Chris Mellish. Using classification to generate text. In *Proceedings of the 30th annual meeting on Association for Computational Linguistics*, ACL '92, pages 265–272, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

Anthony Savidis, Margherita Antona, and Constantine Stephanidis. A decision-making specification language for verifiable user-interface adapta-

tion logic. *International Journal of Software Engineering and Knowledge Engineering*, 15:1063–1094, 2005.

Xiantang Sun and Chris Mellish. An experiment on "free generation" from single rdf triples. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, ENLG '07, pages 105–108, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

Michael White and Ted Caldwell. Exemplars: A practical, extensible framework for dynamic text generation. In *In Proceedings of the Ninth International Workshop on Natural Language Generation*, pages 266–275, 1998.