# Automated Website Synthesis

Siu-wai Leung and Dave Robertson
Centre for Intelligent Systems and their Applications
School of Informatics, University of Edinburgh

**Abstract**

Automated website synthesis is more straightforward than automated program synthesis in artificial intelligence research but can yield clear benefits for large groups of people. We briefly describe our current work and approach to building websites for academic research and aviation accident reporting. Automated website synthesis is applicable to many situations which require efficiency in creating websites for emergency or experimenting website design ideas.

## 1 Introduction

Automated website synthesis facilitates the construction and maintenance of websites and their components by using knowledge engineering technologies. It will help web developers and customers cope with technical problems such as:

- Short life cycles from continual re-design
- Diversified customer preferences
- Frequent content updates
- Reliability assurance

This article will describe the basic ideas and our approach to automated website synthesis.

## 2 Generating Programs If Possible

There has been theoretical and applied research in software engineering for generating customised and optimised intermediates or end-products of the software components on demand by using automated program synthesis [Lowry and Duran, 1989, Czarnecki and Eisenecker, 2000]. Its basic idea is to incrementally refine a high-level specification until an implementation is

derived. If automated generation could be done, its advantages are numerous and so obvious, such as the increase of productivity, cost reduction, guaranteed consistency and compilance of standards. We do not need to elaborate them in this article. We know that partial automation is achievable by using automated program synthesis and generative programming, while high degree of automation was only successful in narrow domains of applications. Successful applications of program synthesis in specific domains are such as ecological model synthesis and astrophysics model synthesis [Robertson *et al.*, 1991, Lowry and Van Baalen, 1997]. The limitations of program synthesis for general use are mainly due to the fact that the existing technologies are insufficient and/or the acquired domain knowledge is inadequate. To gain an improvement, we still need to develop better techniques and acquire useful domain/task-specific knowledge.

## 3 Generating Websites If Possible

The websites which do not need much custom programming provide perhaps a chance for automated domain/task-specific synthesis to give benefit to sizeable user communities. Two key features of such websites make this possible: they often are constructed in a standard way for a given user community; and (if they do not contain complex interactive features) they are comparatively simple formal structures. We ignore the websites which require custom programming because they do not usually follow any patterns and they can be as complicated as the most sophisticated programs.

Standardisation in website design at the level of presentation is already exploited by content management systems, making intensive use of templates and stylesheets [Scharl, 2000]. There is, however, deeper standardisation between the way one models the part of an organisation relevant to a website and the information content assembled to construct that site. For example, there are thousands of websites describing academic research groups. Most of these have similar content structures, based on similar conceptual models of the organisation of these research groups. The content and websites can be formally specified and automatically synthesised [Robertson and Agusti, 1999]. These website synthesis techniques based on a computational logic approach have been used by us since 1996.

## 4 A Computational Logic Approach

We use transformational techniques of program synthesis, particularly structural synthesis in which reusable design components are parameterised, configured and refined to bridge the gap between problem description and final specification [Robertson and Agusti, 1999]. Such techniques provide a formal knowledge-based approach [Cavalcanti and Robertson, 2000]. The re-

finement steps of the reusable design components are declaratively specified by rewrite rules (similar to grammar rules in languauge processing). The declarative rewrite rules are usually more precise and concise than ordinary program codes. To interpret the simple syntax of rewrite rules, the website synthesiser can be very compact. By using computational logics, the properties of the websites can be systematically verified for higher reliability [Cavalcanti and Robertson, 2002].

We have applied this technique to synthesise websites for University institutes [Robertson and Agusti, 1999], aviation accident reporting, its causal perception experiments [Leung *et al.*, 2002], and web-based customer relationship management (CRM) [Leung *et al.*, 2003]. Website synthesis begins with an ontology (meta-description based on a Semantic Web) of a problem domain and applicable transformations which relate elements of the ontology to elements of the website content description. We therefore have a search space of rewrites from an initial domain-specific ontology leading to a detailed website content description. In simple cases, where there is exactly one route through this space, we have a fully automatic synthesis procedure. If there are multiple routes leading to different results, interactions with human or automated reasoners at choice points or additional problem-specific constraints are necessary. The final step transforms the final web content description into a website. As such, the websites can be synthesised with great ease and high reliability.

## 5 Piloting Website Synthesis

Our first use of this technique was to generate and maintain our research group website (`http://www.dai.ed.ac.uk/groups/ssp`) in 1996. We devised a simple (less than 20 terms) ontology describing the structure of the group; built, using this ontology, a knowledge base describing the content of the group; then wrote a set of transformations which convert any knowledge base written using this ontology into the web content to be presented to the website. Finally, using PiLLoW [Cabeza *et al.*, 1996], the web content is converted into web pages. This was an initial investment of 5 person-days of efforts – about 2 person-days longer than it would have taken to author the HTML web pages directly in a structure-based editor. Having made this effort, however, we can (re-)generate the entire site within a few seconds at any time to reflect the changes in the knowledge base. This has meant that for the past years the maintenance cost of our site to researchers has been close to zero. Furthermore, the transformations used to connect knowledge base to web content are sufficiently general that they have been applied, with slight modifications, to a larger institutes (e.g., the Centre for Intelligent Systems and their Applications, `http://www.cisa.informatics.ed.ac.uk`). This sort stability and broad applicability is possible because academic research

institutions often share similar organisation structure and similar ways of presenting that structure on the web. We take advantage of this to cut recurrent maintenance costs radically by intensive use of automated synthesis. In a sense, such websites possesses basic content management facilities.

## 6 Managing Changes and Variety

As seen in our cases, website synthesis serves web content management well. As long as the domain ontology (information architecture) remains the same, changes of information content would not destroy the structural integrity of a website. Even if the domain ontology is changed, the website can be rebuilt using applicable web components with little effort.

Website synthesis may also generate multiple website structures with a prescribed content either to tailor services for customers or to test perceptual hypotheses. We proposed a use of website synthesis in facilitating the interactions of web-based customer relationship (CRM) systems with customers [Leung *et al.*, 2003]. In one experiment for causal perception of aviation accidents, we synthesised two hundred websites with different arrangements of presenting aviation accident information [Leung *et al.*, 2002] and corroborated a recent theory of causal perception [Lien and Cheng, 2000]. It is promising that software agents are useful to further automate the management of synthesised websites [Cavalcanti and Vasconcelos, 2002].

## 7 Experimenting with Web Rhetorics

Same content of information may have different rhetorics in arranging the navigation and presentation patterns. The web designers/architects and target audience have their own preference but it is just too difficult (expensive in time and cost) to experiment with the possible designs. As website synthesis can produce multiple website prototypes efficiently, the same web content can be easily expressed in different web rhetorics. This would help the website developers visualise their preferred web rhetorics and communicate with their clients by using the synthesised website prototypes. We hope that the study of web rhetorics can be also facilitated. A long term goal is to have practical software tools to assist the website developers in visualising reliable (even if not really creative) possibilities of web design. This idea has been exploited in our experiment of causal perception/judgement of aviation accidents [Leung *et al.*, 2002], we generated 200 websites with slightly different arrangements (rhetorics) to study the reactions of their users. We would not be able to create so many websites without using website synthesis. We are also conducting a sophisticated experiment on visualising causality. The participants use a simple drag-and-drop interface to design and preview their websites synthesised on-the-fly so that they can easily ex-

plore/evaluate different possible designs in displaying causality information of aviation accident events.

## 8 Integrating with Semantic Web Technologies

As an alternative to the direct synthesis of web pages in HTML (Hypertext Markup Language) from computational logic specifications, we can set the synthesiser to output XML (eXtensible Markup Language) and XSLT (eXtensible Stylesheet Language Transformation) documents in compliance with the W3C standards for Semantic Web and Web Services. By using XSLT, XML documents can be transformed into HTML or other multimedia XML documents, such as SVG (Scalable Vector Graphics), for further rendering at client (browser) or server side. Such XML-based transformations, although more verbose in syntax, are straightforward in translation from our rewrite rule syntax. As many content management systems (CMS) can accept XML syntax, the synthesiser may generate the website specifications in XML-based (or whatever) for integration with the content management systems, instead of direct generation of final web pages.

## 9 Formulating Generic Website Models

Almost all established content management systems have their own website models or frameworks, such as Zope CMS Framework, Midgard Framework, Cocoon XSP, etc. Such models and frameworks enable web developers to select and/or build components. However, even if XML is being used, their site specifications/configurations in XML cannot be exchanged or easily translated. This is because different website frameworks have their own ontologies and models. To suit different preferences or requirements of technologies, it would be desirable to have a generic website model that is relatively language independent. We do not need to restrict the developers to use a specific programming languages, such as Python, Perl, PHP, or Java, and adapt to different cultures. The generic website model should be easily transformed into any established website development frameworks. To develop such a generic website model, we are studying the existing website frameworks, expressing their ontologies and models in Semantic Web languages, and finding a way to bridge their gaps. This study is also in line with website synthesis, which may transform web content information into website specifications for use with any established content management systems. It is also expected that generic website models and website synthesis may help generate the website components in different ways which are compatible with multiple content management systems.

## 10 Automating Rational Web Design

Our intention is not to synthesise websites according to all web design theories, but to focus on those websites in which design decisions are consistently justifiable by explicit knowledge i.e. rational web design. When there are conceptual gaps between information content and available web design components, we need axioms or rewrite rules to bridge those gaps before performing website synthesis. Such rules represent web design decisions or principles. We are looking into a minimal analogy approach and preference reasoning to automate web design decisions in bridging such conceptual gaps. Hopefully, such gaps will need shorter arbitrary bridges by creating/formulating better knowledge and making useful knowledge available to the website synthesis.

## Acknowledgement

This work has been supported by EPSRC grant GR/M98302 for research on communicating knowledge about accidents from synthesised websites.

## References

[Brinck *et al.*, 2002] Brinck, T., Gergle, D., and Wood, S.D. (2002) Usability for the Web: Designing Web Sites that Work. Morgan Kaufmann Publishers.

[Cabeza *et al.*, 1996] Cabeza, D., Hermenegildo, M, and Verma, S. (1996) The PiLLoW/CIAO library for Internet/WWW programming using computational logic systems. Proceedings of the 1st Workshop on Logic Programming Tools for Internet Applications (JICSLP'96), Bonn, Germany, September 1996.

[Cavalcanti and Robertson, 2000] Cavalcanti, J. and Robertson, D. (2000) Synthesis of web sites from high level descriptions, The 3rd Workshop on Web Engineering, Amsterdam, The Netherlands, May 2000.

[Cavalcanti and Robertson, 2002] Cavalcanti, J. and Robertson, D. (2002) Verifying web site properties using computational logic. In Van Bommel, P. (ed.) Information Modeling for Internet Applications, Idea Group Publishing.

[Cavalcanti and Vasconcelos, 2002] Cavalcanti, J. and Vasconcelos, W. (2002) A logic-based approach for automatic synthesis and maintenance of web sites, Conference of Software Engineering and Knowledge Engineering (SEKE-2002), Ischia, Italy, July, 2002.

[Czarnecki and Eisenecker, 2000] Czarnecki, K. and Eisenecker, U. (2000) Generative Programming: Methods, Tools, and Applications. Addison-Wesley.

[Leung *et al.*, 2002] Leung, S.W., Robertson, D., Lee, J., and Johnson, C. (2002) Using website synthesis in an experiment on the causal perception of aviation accidents. Workshop on the Investigation and Reporting of Incidents and Accidents (IRIA 2002), 17-20 July, Glasgow, U.K.

[Leung *et al.*, 2003] Leung, S.W., Leung, S.K., Kung, A., Tang, T., Wong, K.F., and Li, T. (2003) Using automated web synthesis and semantic web technologies for customer preference management in web-based CRM systems. Technovate 2003 Conference on CRM, Internet Research and New Media, Cannes, France, January 2003.

[Lien and Cheng, 2000] Distinguishing genuine from spurious causes: a coherence hypothesis. Cognitive Psychology, 40:87-137.

[Lowry and Van Baalen, 1997] Lowry, M. and Van Baalen (1997) Meta-amphion: Synthesis of efficient domain-specific program synthesis systems. Automated Software Engineering, 4:199-241.

[Lowry and Duran, 1989] Lowry, M. and Duran, R. (1989) Knowledge-based software engineering. In Barr, A., Cohen, P. and Feigenbaum, E. (ed.) The Handbook of Artificial Intelligence, Volume IV, pp 241-322.

[Robertson *et al.*, 1991] Robertson, D., Bundy, A., Muetzelfeldt, R., Haggith, M., and Uschold, M. (1991) Eco-Logic: Logic-Based Approaches to Ecological Modelling. MIT Press.

[Robertson and Agusti, 1999] Software Blueprints: Lightweight Uses of Logic in Conceptual Modelling. ACM Press and Addison-Wesley.

[Scharl, 2000] Scharl, A. (2000) Evolutionary Web Development. Springer-Verlag.