Markup UK 2021 Programme

A Conference about XML and Other Markup Technologies

Our Motivation – Markup UK began in 2018, when a group of like-minded practitioners agreed that there was an appetite for another markup conference held in Europe. Every one of the founders has contributed to and benefited from the markup technology community in many ways over a long period. As well as working in this space, we have all been enthusiastic attendees of similar events, and we continue to be involved in committees for both sister conferences and standardisation organisations. We recognise the importance of these collaborations in promoting active participation. Markup UK is our opportunity to give something back. We are committed to keeping the conference keenly priced to promote attendance and inclusion. We do not seek to make a profit: we aim to cover our costs, and in the event of a surplus it will be ploughed back in to the following year’s conference.

Contents

Vision........................................................................................................................................................... 4
Mission.......................................................................................................................................................... 4
Values .......................................................................................................................................................... 4
05-21-0930. Opening remarks...................................................................................................................... 5
05-21-0945. XForms .................................................................................................................................... 6
05-21-1300. Metamodel .................................................................................................................................... 7
05-21-1345. Schematron .............................................................................................................................. 8
05-21-1430. XSLT Forms................................................................................................................................. 9
05-21-1530. Java, C#, XML & XSLT ................................................................................................................ 10
05-21-1615. Comprehensibility ...................................................................................................................... 11
05-21-1700. Documentation ...................................................................................................................... 12
05-22-0930. Opening Remarks .................................................................................................................. 13
05-22-0945. XForms .................................................................................................................................... 14
05-22-1300. Distributed Markup ................................................................................................................ 15
05-22-1345. diff3x ..................................................................................................................................... 17
05-22-1430. Tutorials ................................................................................................................................. 18
05-22-1545. Markup UK Proceedings ........................................................................................................... 19
05-22-1630. CSS & XSLT ............................................................................................................................. 20
05-22-1715. Closing ................................................................................................................................... 21
Administrative Information ............................................................................................................................. 21
Markup UK (MUK2020)

Description:

Markup UK Conferences Limited is a limited company registered in England and Wales.

Stakeholder(s):

Markup Technology Community:

Markup UK began in 2018, when a group of like-minded practitioners agreed that there was an appetite for another markup conference held in Europe. Every one of the founders has contributed to and benefited from the markup technology community in many ways over a long period. As well as working in this space, we have all been enthusiastic attendees of similar events, and we continue to be involved in committees for both sister conferences and standardisation organisations.

Europe

Markup UK 2021 Organisation Committee

Geert Bormans:
Geert Bormans has long been an angle-bracket jack-of-all-trades. He loves the beauty of a well-architected solution or a pure and simplified process. Geert makes a living as an independent consultant providing XML or Linked Open Data solutions, mainly to the publishing industry. He does so with a broad geographical flexibility. Geert likes an interesting challenge, easily found when having two teenage daughters. However, he prefers the challenges to involve alpine ground, six strings, or markup.

Tomos Hillman:
Tom has a decade of experience with XML, XSLT, XQuery and related technologies, particularly in the field of digital publishing, quality analysis, and transformation. He has given training courses to various institutions including publishers, universities and the UN, as well as being a regular faculty member at the prestigious XML Summer School in Oxford. Originally from Wales, he now lives with his family in Yorkshire. Tom doesn’t like to conform to stereotypes, but a deep love of Star Wars, board games, fiction and animation sometimes make this hard.

Ari Nordström:
Ari is an independent markup geek based in Göteborg, Sweden. He has provided angled brackets to many organisations and companies across a number of borders over the years, some of which deliver the rule of law, help dairy farmers make a living, and assist in servicing commercial aircraft. And others are just for fun. Ari is the proud owner and head projectionist of Western Sweden’s last functioning 35/70mm cinema, situated in his garage, which should explain why he once wrote a paper on automating commercial cinemas using XML.

Andrew Sales:
Andrew is old enough not to be able to forget SGML. He began writing DTDs and Python programs for publishers in 2000 and, like a streaming processor, he hasn’t looked back. His day job involves architecting content for a London-based UK publisher. Out of hours, he helps out with conferences, maintains the Schematron international standard, and takes photographs using analog and digital devices.

Rebecca Shoob:
Rebecca provides invaluable back-office support to the conference. Originally a linguist, she is an experienced office manager by trade and enjoys gardening in her spare time. She has also recently been elected a district councillor.

Markup UK 2021 Sponsors

Evolved Binary
Saxonica
le-tex Publishing Services
Oxygen XML Editor
Mercator IT Solutions
Exeter
Mulberry Technologies
Markup UK Sister Events
XML Prague
Balisage
Declarative Amsterdam
XML Summer School
Vision
The markup technology community benefits

Mission
To share knowledge and experience with the markup technology community

Values
- **Collaboration**: We recognise the importance of these collaborations in promoting active participation. Markup UK is our opportunity to give something back.
- **Participation**
  - **Inclusion**: We are committed to keeping the conference keenly priced to promote attendance and inclusion. We do not seek to make a profit: we aim to cover our costs, and in the event of a surplus it will be ploughed back in to the following year's conference.
05-21-0930. Opening remarks

[To be described]

<table>
<thead>
<tr>
<th>Performance Indicators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>Target</td>
</tr>
<tr>
<td>Actual</td>
</tr>
</tbody>
</table>
05-21-0945. XForms

*Learn about the structure and workings of XForms.*

**Stakeholder(s)**

*Steven Pemberton*: (CWI)

XForms Hands-on (Tutorial) – In the 50s, when the first programming languages were designed, computers cost millions, and relatively, programmers were almost free. Those programming languages therefore reflected that relationship: it didn’t matter if it took a long time to program, as long as the resulting program ran as fast as possible. Now, that relationship has been reversed: compared to the cost of programmers, computers are almost free. And yet we are still programming them in direct descendants of the programming languages from the 50s: we are still telling the computers step by step how to solve the problem. Declarative programming is a new approach to applications: rather than describing exactly how to reach the solution, it describes what the solution should look like, and leaves more of the administrative parts of the program to the computer. One of the few declarative languages available is XForms, an XML-based language that despite its name is not only about forms. Large projects, at large companies such as the National Health Service, the BBC and Xerox, have shown that by using XForms, programming time and cost of applications can be reduced to a tenth! This hands-on tutorial is in two parts, each consisting of a rapid-fire sequence of around a dozen exercises, consisting of 5 minutes of presentation, followed by 5 minutes of coding. Each exercise consists of modifying an existing XForm to use the newly-learnt feature. The tutorial allows you to learn about the structure and workings of XForms, and gives you the opportunity to work on useful working programs. Part 1 introduces the main elements of the language; part 2 builds on part 1, and takes you through the more advanced parts. It is a “bring your own device” tutorial, requiring the installation of some files beforehand, and checking they are working. Attendees can work using the text editor of their choice.

**Performance Indicators**

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
05-21-1300. Metamodel

Make families of tag sets flexible and compatible.

Stakeholder(s)

Tommie Usdin : Mulberry Technologies, Inc. ~ B. Tommie Usdin is President of Mulberry Technologies, Inc., a consultancy specializing in XML for prose documents. Ms. Usdin has been working with generic markup since 1985 and has been a supporter of XML since 1996. She chairs “Balisage: The Markup Conference” and is a frequent speaker at JATS-Con, MarkupUK, and other conferences. Ms. Usdin has developed DTDs, Schemas, and XML/SGML application frameworks for applications in academia, government, and industry. Projects include reference materials in medicine, science, engineering, and law; semiconductor documentation; historical and archival materials, and text books. Distribution formats have included print books, magazines, and journals, and both web- and media-based electronic publications. She is co-chair of the NISO Z39-96, JATS: Journal Article Tag Suite Working Group, and member of the NISO STS Standing Committee and of the BITS Committee. You can read more about her at http://www.mulberrytech.com/people/usdin/index.html

Encouraging Tag Set Branching without Creating a Briar Patch ~ Customizing a tag set can be an easy way to get the vocabulary you need. It can also be a journey filled with dead ends, trap doors, and slowly-revealed and difficult to identify problems. Like many public tag sets, JATS (the Journal Article Tag Suite) was designed to be customized. Our original expectation was that individual users would customize it, and while a few have done that to good effect, we have found that the major customizations have been by groups of users. BITS (the Book Interchange Tag Suite), NISO STS (Standards Tag Suite), and Manuscript Exchange Common Approach (MECA) are widely adopted customizations of JATS. When users customize a tag set they expect to be able to use the existing infrastructure associated with that tag set, making changes to accommodate the changes they made. They often expect to intermingle their new documents with documents tagged to the original tag set and perhaps with documents tagged to other customizations of the source tag set. They expect this to work gracefully, easily, seamlessly. Sometimes it does, but sometimes it does not! The "JATS Compatibility Meta-Model Description" was developed to help people who customize JATS create tag sets to create models that will coexist peacefully with existing JATS documents and with documents tagged to other JATS customizations. It seems unlikely that the particulars of the JATS Compatibility Model will apply to other tag sets, but the principles behind the Meta-Model might be useful to other groups thinking about ways to make their families of tag sets flexible and compatible.

Performance Indicators

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
05-21-1345. Schematron

Discuss three problem areas covered by Schematron.

Stakeholder(s)

David Maus:
Staats- und Universitätsbibliothek Hamburg ~ David Maus (dmaus.name) is Head of Research & Development at Staats- und Universitätsbibliothek Hamburg. He is the lead developer and maintainer of SchXslt, an XSLT-based Schematron processor.

What's in a Schematron? ~ Schematron is a rule based validation language for structured documents. It was designed by Rick Jelliffe in 1999 and standardized as ISO/IEC 19757-3 in 2006, 15 years ago. In 2020 work on the specification concluded: The working group was disbanded and the last edition of the specification, its draft not shared publicly for discussion with the Schematron community, put an end to Schematron as an Open Standard. While losing an Open Standard is annoying, especially given the circumstances that lead to the specification becoming proprietary, it gives the opportunity to look at the specification as an object of an analytic inquiry. What's in a Schematron? Which problems does Schematron address and how does it do it? How would a Schematron-like language look like in 2021? In this paper I will discuss three problem areas covered by Schematron: 1. Rule based validation; 2. Schema composition; and 3. Reporting. For each area I will show the issues involved, how Schematron addresses the problem, what conclusions can be drawn from it, and how a new rule based validation language could build on the Schematron experience in this particular area.

Performance Indicators

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
05-21-1430. XSLTForms

[To be described]

Stakeholder(s)

Alain Couthures:
agenceXML ~ Alain Couthures is a non-conventional expert programmer finding solutions to problems many would not even imagine being resolvable. He has already implemented XForms at client-side (XSLTForms) and is now implementing XQuery in Javascript. He is a member of the W3C XForms Users Community Group.

XSLTForms is a non-conventional expert programmer finding solutions to problems many would not even imagine being resolvable. He has already implemented XSLTForms and is now implementing XQuery in Javascript. He is a member of the W3C XForms Users Community Group.

XSLTForms for the ’20s ~ Since 2009, XSLTForms has been compatible with successive browser versions starting from Internet Explorer 4. It is based on XSLT 1.0, CSS and vanilla Javascript. XSLTForms modernization is required for perennity and evolutivity. HTML5 notation for XForms and CSS: XSLTForms is now using an HTML5 notation for XForms elements replacing numerous DIV and SPAN embedded elements with only CSS classes to differentiate them. HTML5 is allowing custom elements and CSS classes can be defined for them. To avoid name conflicts, XForms elements are transformed into “xforms-*” elements (“xforms-input”, for example) and XForms attributes into “xf-*” attributes (“xf-selected”, for example). Inline XML instances and XML schemas have to be protected within SCRIPT elements with a not-supported type (“application/xml”, for example) to preserve letter cases. Because of CSS attribute selectors, CSS classes are powerful enough to manage custom elements visibility. For example, an XForms group without a binding will always be visible but, with a binding, an extra attribute (“xf-bound”) will be programmatically added. More custom sub elements are added, such as “xforms-body”, “xforms-help”, … to render XForms controls with HTML and CSS.

XQuery/XPath parser in Javascript: The XPath parser has been rewritten in Javascript to be executed just-in-time when an expression is to be evaluated for the first time. XQueryX 3.1 to Javascript transpiler: The Javascript arrays resulting from the XPath/XQuery parser can now be converted into Javascript functions. Its body contains a sequence of instructions with a Reverse Polish Notation order. Path evaluations require “if” statements to stop when a step returns an empty sequence. Predicates are performed with inline functions to be called for each sequence item.

XForms 2.0 support in XSLTForms: XSLTForms is already supporting some major XForms 2.0 new features such as variables and AVT.

Performance Indicators

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
05-21-1530. Java, C#, XML & XSLT

[To be described]

Stakeholder(s)

Michael H Kay: Saxonica

<transpile from="Java" to="C#" via="XML" with="XSLT"/>

This paper describes a project to convert a substantial piece of software (an XSLT processor, as it happens, but it could have been anything) from Java to C#, using an XML representation as the intermediate format, and using XSLT as the transformation language. Outline:

- The business requirement (why did we need to do this, and what were the constraints)
- The technical challenge (alternative approaches considered; why is the problem difficult; examples of transformations that are needed)
- The benefits of using XML and XSLT (XML as a human-readable and machine-processable representation of tree-structured information; XSLT as a transformation language focused on matching patterns in trees; examples of some of the complex patterns that need to be matched)
- Overall design of the solution (the processing pipeline)
- Selected points of interest (particular aspects of the solution showing innovative or advanced use of XSLT that might contain useful ideas for other projects)
- Achievements / conclusion / further work

Performance Indicators

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
05-21-1615. Comprehensibility

Provide directions, tips and tricks on how to make XML-related code more understandable.

Stakeholder(s)

Erik Siegel:
Xatapult ~ Erik is a self-employed content engineer and XML specialist, working from the Netherlands. Most clients are from the publishing industry or involved in standardization. Coming from a technical background, Erik is deliberately looking for content and XML-related projects on all levels: from the strategic use of standards to developing processing applications. Documenting and explaining difficult subjects, whether in prose or as a course, is something he likes to as an addition to all the technical work.

Comprehensible XML ~ Writing software, it is all too easy to forget that there is another side to it than just: it works. Most software goes through a life cycle of writing, testing, debugging and maintenance. This makes it important that what you write is comprehensible, both for somebody else and yourself in a few months’ time. It reduces the chance of mistakes and bugs and shortens development time. We have probably all seen, heard, or read something about how to write good code. We probably all try to comply more or less, but given what we see around us, we do not always succeed. This talk will try to provide directions, tips and tricks on how to make (in this case XML-related) code more understandable. It will also provide background on why this important and why we should try to comply. How can we do this with minimum effort. It will be a mixture of things from literature and personal experience after 40 years of programming computers.

Performance Indicators

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
05-21-1700. Documentation

Discuss how much documentation of tags sets is needed.

Stakeholder(s)

Deborah A Lapeyre :
Mulberry Technologies, Inc. ~ Deborah Aleyne Lapeyre (Debbie) is a Senior Consultant for Mulberry Technologies, Inc., a consulting firm specializing in helping their clients toward publishing and documentation solutions through XML, XSLT, and Schematron. She works with Tommie Usdin as Secretariat and architects for JATS (ANSI NISO Z39.96-2019 Journal Article Tag Suite), BITS (Book Interchange Tag Suite), and NISO STS (ANSI/NISO Z39.102-2017 Standards Tag Suite). She is on the organizing Committee for Balisage: The Markup Conference. Debbie has taught hands-on XML, XSLT, DTD and schema construction, and Schematron courses as well as numerous technical and business-level introductions to XML and JATS. She has been working with XML and XSLT since their inception and with SGML since 1984 (before SGML was finalized as an ISO standard). In a previous life, she wrote code for systems that put ink on paper and used, taught, and documented an early 1980’s proprietary generic markup system named “SAMANTHA”. Hobbies include pumpkin carvering parties and many too many books.

How Much Tag Set Documentation is Needed? How much is Too Much? ~ The more the better. Documentation is expensive, stick to the basics. If it isn't well documented people won't use it or, worse, won't use it consistently. JATS (The Journal Article Tag Suite) has documentation. A LOT of documentation. Documentation designed to introduce new users to the tag set. Documentation designed to support experienced users. Documentation to support people who are customizing JATS, including both advice on the mechanics and logic of making customizations. There are definitions, helpful remarks, tagged examples, extended essays. There is an International Standard that meets political needs and a site with non-normative documentation that meets practical needs. There are third party sites advising users on how to use the tag sets for best interoperability, and many organizations that ingest JATS provide (ans may insist on) local rules. It is entirely possible that this is the most heavily documented XML tag set of all time. Do other tags need this much documentation? Are there any parts of this would others find useful? After a guided tour of the JATS documentation, the audience can chime in: How much documentation does a tag set need? How much documentation does YOUR tag set need? What is useful? What is overkill? What audience most needs to be served?

Performance Indicators

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
05-22-0930. Opening Remarks

[To be described]

Performance Indicators

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
05-22-0945. XForms

[To be described]

Stakeholder(s)

Steven Pemberton:
CWI ~ Steven Pemberton is a researcher at CWI Amsterdam, the Dutch national research centre for mathematics and informatics. His research is in interaction, and how the underlying software architecture can support users. He co-designed the ABC programming language that formed the basis for Python, and was one of the first handful of people on the open internet in Europe, when the CWI set it up in 1988. Involved with the Web from the beginning, he organised two workshops at the first Web Conference in 1994. For the best part of a decade he chaired the W3C HTML working group, and has co-authored many web standards, including HTML, XHTML, CSS, XForms and RDFa. He continues to chair the XForms group at W3C, and was until recently a member of the ODF (Open Document Format) technical committee. More details at http://www.cwi.nl/~steven

XForms Hands-on (Tutorial) ~ In the 50s, when the first programming languages were designed, computers cost millions, and relatively, programmers were almost free. Those programming languages therefore reflected that relationship: it didn’t matter if it took a long time to program, as long as the resulting program ran as fast as possible. Now, that relationship has been reversed: compared to the cost of programmers, computers are almost free. And yet we are still programming them in direct descendants of the programming languages from the 50s: we are still telling the computers step by step how to solve the problem. Declarative programming is a new approach to applications: rather than describing exactly how to reach the solution, it describes what the solution should look like, and leaves more of the administrative parts of the program to the computer. One of the few declarative languages available is XForms, an XML-based language that despite its name is not only about forms. Large projects, at large companies such as the National Health Service, the BBC and Xerox, have shown that by using XForms, programming time and cost of applications can be reduced to a tenth! This hands-on tutorial is in two parts, each consisting of a rapid-fire sequence of around a dozen exercises, consisting of 5 minutes of presentation, followed by 5 minutes of coding. Each exercise consists of modifying an existing XForm to use the newly-learnt feature. The tutorial allows you to learn about the structure and workings of XForms, and gives you the opportunity to work on useful working programs. Part 1 introduces the main elements of the language; part 2 builds on part 1, and takes you through the more advanced parts. It is a “bring your own device” tutorial, requiring the installation of some files beforehand, and checking they are working. Attendees can work using the text editor of their choice.

---

Performance Indicators

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Page 14
05-22-1300. Distributed Markup

Create specifications that everyone can use to transfer data.

Stakeholder(s)

Karin Bredenberg:
Kommunalförbundet Sydarkivera ~ Karin Bredenberg is a Metadata Strategist at Kommunalförbundet Sydarkivera. She currently serves as the chair of PREMIS EC, co-chair of TS EAS, chair of the DILCIS Board, and a member of the METS Board. Currently, Bredenberg is the activity lead for specifications in the project E-ARK3 and the European Commission’s eArchiving Building block.

Jaime Kaminski:
University of Brighton ~ Jaime Kaminski BA, MA, PhD, is a training manager at Highbury R&D. Before joining Highbury, Jaime spent 14 years in academia researching in the field of Digital Humanities where he has been working on EC-funded projects including EPOCH, 3D-COFORM, V-MUST, E-RHIS, ROMOR, E-ARK and E-ARK4ALL, as well as the UKRO-funded SEAHA project. Before joining academia, he spent seven years as a technical briefings manager and senior technology analyst for a Blue Chip ICT consultancy.

The Connecting Europe Facility
European Commission
Swiss Federal Archives

2021 The Future of Distributed Markup Systems or ‘Help my package became too big!’ ~ In 2019 you were introduced to the wonders of digital archiving and the eArchiving Building Block. We explored how common specifications have been used for describing both Information Packages and the Content Information Type Specifications (CITS) that can be placed in an Information Package. This is a new world for some and an old world for others, so let’s do a short recap and introduce you to the challenges that come next. The eArchiving Building Block supported by The Connecting Europe Facility and the European Commission creates common specifications based on pre-existing standards that everyone can use to transfer data. The birth took place in 2018 and was a long one ending in 2019. The result was several specifications for the different types of information packages that are found in the OAIS Reference model [https://www.iso.org/standard/57284.html] and Content Information Type Specifications (CITS) for structuring the content to be placed in the package. Tools were also developed, but let’s focus on the foundation, the specifications. In the fall of 2019, a new two-year project was started, which is now developing more specifications and enhancing those that have already been created. But where does this winding path take us? To more specifications and more data to transfer? The path will become more and more winding, the bigger the child gets and the more we feed it. One of the challenges is the databases. They are the food that the baby needs to grow. You might think this was an easy task, but I can assure you it is not. Databases need to be transformed into a sustainable format. The eArchiving Building Block uses the SIARD standard (Software Independent Archiving of Relational Database) developed by the Swiss Federal Archives. With the help of the available tools, SIARD transforms the database into an XML-format. But, many other files can be hidden in a database in the form of BLOBs and or CLOBs, which means that the total size of a database can be huge. How do you transform that to an XML format with the files extracted and referenced in the XML-document? Well, it is referenced in two XML-documents. First, in the XML documents that are produced by the transformation, and second, in an information package XML document. This later document tells us what files that are supposed to be transferred in the package and a checksum so we can check that the file is the correct one when it arrives. Our child is getting grumpy. Can we put all the files in one folder and describe them in just the two XML-documents? Of course, we can, but that might give us an XML-document that takes 24 hours to validate if the database was filled with files. How do we make our child happy? We just need to give them one piece of candy and not the whole bag. This means that we need to set up recommendations and even rules about splitting the package into more packages, so the candy is divided into more than one bag. This is where we are today, deciding which candy goes into which bag. At the same time, we can see that it is not only databases that are problematic. One of the new Building Block specifications specifies how to transfer medical records. Here we run into the same problem with the package being too big to handle. The solution is not to get more powerful computers to handle the packages. It needs to be possible to do in different kinds of hardware and software environments and with different staff skillsets. We cannot count on having access to the funding to get what we want in the form of hardware, but we still have the task of preserving the now for the future and thus need more candy bags. Where do I find my packages?
### Performance Indicators

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## 05-22-1345. diff3x

Discuss diff3 as an XML format.

### Stakeholder(s)

**Robin La Fontaine**: DeltaXML Ltd – Robin is the founder and CEO of DeltaXML. He holds an Engineering Science degree from Oxford University and an MSc in Computer Science. His background includes computer aided design software in mechanical, electrical and electronic applications, solar energy system simulation. Since founding DeltaXML, he has been addressing the challenges and exploring the opportunities associated with information change in data and documents.

**Nigel Whitaker**: DeltaXML Ltd – Nigel is the Chief Architect at DeltaXML and explores new product development as well as the more mundane aspects of software development. He studied Physics and Computer Science at Manchester University, where after his PhD, he worked on computer aided design for electronics.

An improved diff3 format using XML: diff3x – There is no doubt that the diff and diff3 format has established itself as a well-used de facto standard. It might seem presumptuous to suggest that it could be improved; or indeed that it needs to be improved. However, the original premise of line-based text files as the subject matter is now out of date with more structured information being the norm. Often this is in the form of programming source code where the layout tends to remain fairly consistent through edit cycles, but increasing use of JSON and XML pose particular difficulties for the simple line-based structure of diff3. In our paper at MarkupUK in 2019, we discussed some of the issues and suggested some minor improvements to diff3. These changes suffered from the common complaints of a retro-fit in that they did not sit comfortably with the original and only did half a job. The prevalence of GUIs also suggest that the actual syntax of a diff file is not as important as it was in that the emphasis has changed from human readability to interchange between two applications. For these reasons it seemed better to consider a different approach using the tools and formats that are now in common use, for example XML or JSON. What might diff3 look like as an XML format? Would the advantages of a new format make it worth swapping from the tried and tested diff3? Could existing GUI software easily adapt to a new format and, perhaps, even be simpler as a result?

### Performance Indicators

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
05-22-1430. Tutorials

*Build tutorials in XForms.*

**Stakeholder(s)**

*Steven Pemberton :*

*CWI*

On the design of a self-referential tutorial – XForms is a declarative programming language, with a different paradigm compared to traditional languages. Consequently learning materials are paramount, since it requires a new way of thinking about programming. A tutorial is a challenging educational setting, both for those learning as those instructing. Time is severely limited, and there is inevitably far more material on the subject than can possibly be taught in the time available. Unavoidably, compromises have to be made: do you make it deep and narrow, or broad and shallow? Should it be a teaser to tempt attendees to later self-study, or a starter, so that attendees have at least a working knowledge of some of the material. Most XForms tutorials had to date been of the lecture style, covering most of the language. However, a request for a hands-tutorial inspired a new approach. After attending several tutorials, the author realised that to optimally use the time available for exercises, attendees shouldn't be required to start from scratch, since precious time is lost dealing with trivial issues. Rather, the exercises should all require making a change to an existing, working, example, using the new knowledge, giving the advantage of hands-on, while minimising trivial details. The tutorial was thus designed as a rapid-fire sequence of exercises, with 5 minutes of presentation, and 5 minutes of coding. The exercises themselves are not stand-alone, but cumulative, each one building on an earlier one, so that at the end the attendee has a handful of small, but in themselves useful, applications. Although the tutorial was designed to be part of a live event, it also supports the use of self-study. The resulting tutorial is interesting in that it is not only *about* XForms, but it is also built *in* XForms, which in itself gave surprising possibilities. This paper discusses the method used to build the tutorial, the decisions taken, the techniques used, and the benefits that the approach gave.

### Performance Indicators

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
05-22-1545. Markup UK Proceedings

*Improve the styles for publishing the Markup UK proceedings.*

**Stakeholder(s)**

**Tony Graham:**

*Antenna House ~* Tony Graham is a Senior Architect with Antenna House, where he works on their XSL-FO and CSS formatter, cloud-based authoring solution, and related products. He also provides XSL-FO and XSLT consulting and training services on behalf of Antenna House. Tony has been working with markup since 1991, with XML since 1996, and with XSLT/XSL-FO since 1998. He is Chair of the Print and Page Layout Community Group at the W3C and previously an invited expert on the W3C XML Print and Page Layout Working Group (XPPL) defining the XSL-FO specification, as well as an acknowledged expert in XSLT. Tony is the developer of the 'stf' Schematron testing framework and also Antenna House’s 'focheck' XSL-FO validation tool, a committer to both the XSpec and Jaxy XSLT testing frameworks, the author of “Unicode: A Primer”, and a qualified trainer. Tony's career in XML and SGML spans Japan, USA, UK, and Ireland. Before joining Antenna House, he had previously been an independent consultant, a Staff Engineer with Sun Microsystems, a Senior Consultant with Mulberry Technologies, and a Document Analyst with Uniscope. He has worked with data in English, Chinese, Japanese, and Korean, and with academic, automotive, publishing, software, and telecommunications applications. He has also spoken about XML, XSLT, XSL-FO, EPUB, and related technologies to clients and conferences in North America, Europe, Japan, and Australia.

"FYI we're not looking to go to print": Restyling the Markup UK Proceedings ~ Markup UK is a markup conference, and its conference proceedings start life as DocBook XML. DocBook has a standard set of XSLT 1.0 stylesheets for transforming DocBook XML markup into other formats. Markup UK 2018 was put together very rapidly, so it having proceedings at all is a tribute to the usefulness of the DocBook XSLT stylesheets. However, the PDF proceedings were the stock DocBook styles done with a Garamond font and with the addition of some frontmatter with sponsors logos and acknowledgements. The Markup UK 2019 proceedings were produced using the same customisation. Shortly after the conference, it was agreed that the styles should be improved. The full instructions for what to do were "FYI we're not looking to go to print, if that influences any of your decisions." It did, but it didn't make the task any easier. This presentation discusses the changes to the proceedings and how the DocBook stylesheets were customised to achieve them. The changes fall into three areas: not going to print; Markup UK look-and-feel; and accessibility.

**Performance Indicators**

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
05-22-1630. CSS & XSLT

Keeping XSLT, HTML and CSS synchronized.

Stakeholder(s)

Liam R E Quin:
Delightful Computing ~ Liam Quin (www.delightfulcomputing.com) was in charge of XML development at W3C; they are now an independent XML consulting and training business in Canada.

CSS Within: CSS From XSLT ~ A common problem when using XSLT to make HTML and CSS is keeping the XSLT, HTML and CSS synchronized. Changes to the XSLT that affect what HTML is generated necessitate corresponding changes to CSS; changes to the CSS may necessitate changes to the XSLT. Over time there will be some elements in the HTML for which there is no style information in the CSS, or for which the style information is out of date; there will be rules in the CSS that are no longer needed, or, worse, that are out of date and are only sometimes triggered. Generating CSS from within XSLT, perhaps with one large xsl:text element, helps to reduce the divergence by making it easier to find CSS rules that affect a given element. But it is still easy to forget to update the CSS. A search for div.beer might not show up anything, but the CSS might have a selector fridge>* which matches beer when it's in the fridge. So it is not always trivial to locate the appropriate CSS rule to update. With XSLT 3 also comes expand-text="yes" and curly braces being potentially special inside text node constructors, conflicting with CSS syntax. CSS Within is a new way to integrate CSS generation with HTML generation to help reduce or eliminate these problems. The method used is to embed rule and media elements, in a CSS Within namespace, inside XSLT templates. This puts the CSS styles exactly where the XSLT developer needs them: right next to the XSLT instructions producing the elements that they style. CSS Within supports multiple stylesheets (for example for Web and PDF), and can run in pure XSLT or with extension instructions currently available for Saxon, in Java). Limited usage experience suggests that CSS Within makes HTML generation easier to maintain and more robust over time.

<table>
<thead>
<tr>
<th>Performance Indicators</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
05-22-1715. Closing

Deliver closing remarks and thanks from the organisers

Performance Indicators

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Administrative Information

Start Date: 2021-05-21
End Date: 2020-05-22
Publication Date: 2021-05-16
Source: https://markupuk.org/schedule.html

Submitter:
Given Name: Owen
Surname: Ambur
Email: Owen.Ambur@verizon.net
Phone: