

XML For Digital Preservation: XML Implementation Options for E-Mails

Author: Maureen Potter. Experiment Operator, Digital Preservation Testbed, The Netherlands.

This paper is based on a presentation to the *Erpanet* workshop on *XML and Digital Preservation* in Urbino, Italy on 11 October 2002. Slides from the original presentation can be found at <http://www.digitaleduurzaamheid.nl> or <http://www.erpanet.org>

Introduction

This paper presents some of the work we have undertaken at the Digital Preservation Testbed in the Netherlands using XML as a preservation approach. I first introduce the background and scope of the Testbed project, putting our work into context. I will then discuss the advantages and disadvantages of XML for archival preservation, and identify the attributes of email as a record type that make it particularly suitable for conversion to XML. This paper will then focus on the three different implementation options we have designed for conversion or provision of emails in XML. The first two are intended for use on emails that have already been transmitted and can be used to convert existing sent and received mail into XML format. The third is an 'add-in' that is integrated with the email application (in our case Outlook), and that creates and stores an XML representation of the message at the same time that it is originally transmitted. This paper closes with a description and screenshots of the email to XML demonstrator and shows how this captures additional record keeping and archival metadata.

Background

The Digital Preservation Testbed was established in October 2000 by the Ministry of the Interior and the Ministry of Education, Culture and Sciences (of which the National Archives is a linked institution). The Testbed is a three-year research project with the overall goal of investigating options to secure sustained accessibility to authentic archival government records over the long term. The Testbed is part of a wider Dutch initiative called the Digital Longevity Programme. The other projects in the Digital Longevity Programme include the Digital Longevity Taskforce, a project on Record Keeping systems, and a Quality Survey. The different projects work together to complement each other.

The Testbed is a practical research project that carries out experiments in a controlled and secure environment. This allows us to ascertain the effects of undertaking preservation action on archival records. Our direction is dictated by the Research Questions laid down at the beginning of the project.

Research Questions

The Research Questions have three main areas of interest: General; Metadata; and Attribute-based. General research questions include:

- What are the advantages and disadvantages of implementing the different preservation approaches?

- How can the effectiveness of each approach be measured and or demonstrated?
- What are the factors that affect the effectiveness or appropriateness of each preservation approach? For example, Cost? Record type? Authenticity requirements and retention periods?
- What are the basic requirements for preservation functions? For example, what are the requirements for accessing and retrieving records from the preservation function?

Metadata research questions address such issues as:

- What factors affect the metadata required for preservation? For example, record type and preservation approach, and how?
- What are the options for associating metadata with records?

We also consider attribute research questions. The Testbed classifies electronic records according to the five attributes identified by Rothenberg¹. These are: Content; Context; Structure; Appearance and Behaviour. We consider such aspects as

- What are the options for preserving record attributes?
- What is the relationship between the preservation of specific attributes and the cost of preservation?

In all, these Research Questions cover a very broad spectrum.

Scope

Such a broad spectrum, in fact, that our scope is limited to four specific record types. We consider digital preservation from the file level upwards and are concerned only with records that are born digital. The four record types chosen for inclusion in our experiments are:

- Text documents - for example, MS Word or WordPerfect documents
- Emails – from, amongst others, Outlook, Eudora, Novell Groupwise, Hotmail, and KMail
- Spreadsheets – including MS Excel and Lotus
- Databases – for example, Access and Oracle.

Within these four record types, we examine three preservation approaches:

- Migration – the transfer of digital materials from one hardware and/or software platform to another;
- Emulation - the recreation of one hardware and/or software environment on another; and
- XML. The XML approach can be implemented in various different ways. For example, conversion from another format into XML can be considered as a particular type of migration technique. XML is also a highly promising original data format for archiving and interoperability. It therefore deserves to be considered as a preservation approach in its own right.

Not every approach is suitable for every record type. For example, we do not consider it worthwhile to attempt full-blown emulation for emails. Our preliminary results with XML for emails were very encouraging, and we quickly realised that this would be a suitable approach for their long-term preservation.

¹ Rothenberg, Jeff & Bikson, Tora: *Digital Preservation - Carrying Authentic, Understandable and Usable Records Through Time* (Digitale Duurzaamheid, The Hague, 1999).

Advantages and Disadvantages of XML for Archival Preservation

The beauty of using XML for the preservation of digital archival records is that it can be used in so many different ways. It can be used for:

- Metadata storage and exchange
- As an original file format for new records and a target conversion format for existing records
- Object linking and referencing between and within files
- Encapsulation (using XML in a wrapper approach to describe the contents of the files contained within).

These multiple use scenarios provide a huge advantage. If Archives employ XML for different archival purposes and in different scenarios, they can significantly reduce the number of different file formats they have to deal with. This limits the complexity of future preservation action on the records, and also on the archive itself.

XML can also be used to represent nearly all of the five attributes of a digital record. Record Content and Context are easily described in XML, using its most basic ability to represent content in a clear, straightforward and human readable manner. The Structure of the record can be preserved with XML, a well-structured mark-up language. Specifying an XML DTD or Schema allows us to accurately re-assemble and render the record and ensure that it is structurally authentic and integrally whole. Combining the XML file with a corresponding Style Sheet can help ensure the authentic Appearance of the record. The style sheet contains instructions for the appearance of the record, and will apply specific formatting and layout styles to any number of XML files².

In addition to these factors that are specific to digital preservation, there are also some more general attributes that make XML a good file format. As a standard, it is reliably controlled and developed by the W3C, and the specifications are freely available. It is planned for interoperability – the fact that it doesn't have to be adapted to be transmitted and understood across different platforms is a distinct advantage. Its human readable form makes it appear less-code like for those who aren't quite so technical. This means that it is also relatively easy for people to understand the processes behind it and to eventually manipulate and work with it³. Finally, XML is also good for generating indexes and searching aids, and is supported by most major software manufacturers and tools. It has a sound support base and while it has taken a few years to reach a sustainable and useful level of popularity, we are now at the stage where most people agree that the practical advantages of XML make it suitable for – well, pretty much most things!

However, not everyone is convinced⁴. One of the foremost touted 'disadvantages' is user scepticism. Users are indeed often sceptical – XML has been hailed as the silver bullet for digital preservation and information exchange for a few years, but is only now beginning to come into mainstream archival preservation work. This means that although nearly everyone has heard of it, many users have yet to experience it. This can lead to some scepticism: firstly

² There is some uncertainty about the extent to which this XSL/XML combination will produce consistent results. Some say that the end results may vary, depending on the browser and processing software used; this is something we will investigate in the Testbed environment.

³ It also increases the likelihood that the mark-up will be understood in the future if we have unexpected and vastly different software and hardware changes.

⁴ See for example to recent article by Terje Hillesund in JODI, the Journal of Digital Information: <http://jodi.ecs.soton.ac.uk/Articles/v03/i01/Hillesund/> - note that there is also a response to this article at <http://jodi.ecs.soton.ac.uk/Articles/v03/i01/Walsh>

because they think that they will have to learn a programming language; and secondly because they already have a way of doing things and they don't want to change⁵. A second alleged disadvantage is the fact that you still have to pay for software to convert and process your files to XML⁶. Although XML is perceived as cheaper than PDF (for example), and XML is an open non-proprietary standard, it still needs software. This software needs to be paid for and includes tools not only to carry out the conversion, but also to process the results back into a record-like form.

A more commonly heard objection is the stability and expected lifetime of XML. It is often suggested that XML will be superseded by another file format in five to ten years. If this is the case, then conversion from XML to another format should still be relatively straightforward compared to conversion from, for example, a proprietary format. Even so, many people do not believe that will be the case. There is no reason why XML will only last a short period of time. If it does the job sufficiently, then why replace it? Previous file formats and strategies have been replaced because they have been found wanting, not just because someone came up with something else. Various groups and institutions are experimenting with XML for Digital Preservation, and it is undergoing vigorous research. That is why it has achieved prominence, and not because a multinational company provided it for free or because we are still waiting for something better to come along.

A final, oft-heard criticism is that XML tends to be verbose and makes files bigger. Whilst this may be true, it is not necessarily significant. Storage is relatively cheap, and the cost of storage is a small price to pay for gains in ease of use and reliability.

XML For Emails

Our research revolves around four mail record types – text documents, emails, spreadsheets and databases. So far, email has proved to be a particularly suitable record type for XML treatment. There are many similarities between XML and Email formats, and conversion between the two is thus relatively straightforward.

Both are highly specified. Emails must follow the Internet Message Format to be interoperable on different platforms. This format is well laid out and defines the component parts of a basic email transmission file. (The standard currently in use with emails is RFC 2822, with the MIME extensions specified in RFC 2045 – 2049.) It is controlled by a non-profit organisation, the Internet Engineering Task Force⁷, and is well defined, well structured, and text based.

XML is a standardised format, as well as a mark-up language. Again, it is highly specified and controlled by a non-profit organisation – in this case the World Wide Web Consortium⁸. The W3C are responsible for organising and maintaining the XML Specification, Schema, Standard and XSLT Recommendation. XML, as the name denotes, is extensible. It can be

⁵ On this note, record creation practices are probably one of the biggest challenges that we are facing in digital preservation. We all talk about digital obsolescence and interoperability, but if the records aren't created properly in the first place, they will not be stable for tomorrow, never mind over the long term

⁶ This 'disadvantage' applies to all digital objects. It is therefore dubious to refer to it as a disadvantage at all; it is more of a requirement.

⁷ See: <http://www.ietf.org/>

⁸ See: <http://www.w3.org/>

adapted and extended for any purpose while still remaining true to its spirit. It can operate on any hardware and/or software platform, and can be read on any plain text editor.

The similarities between the two mean that conversion is a relatively straightforward procedure. All individual sections are plainly marked in the email transmission file and can easily be transformed into a similarly well-structured XML file.

Implementation Options

Through the course of our experiments, we have developed three processes for conversion of emails into XML. These three processes are suitable for two different scenarios: *Post-Use Conversion*, and *Pre-Use Provision*.

Processes in the Post-Use Conversion scenario are suitable for agencies with large numbers of emails that have already been used for the purpose for which they were created, and which must now be retained for a currently unspecified time period as evidence of a transaction. Processes in this scenario take the Transmission file of the email as their starting point and NOT files saved in the proprietary format of the email application.

There are two implementation options in this first scenario. The first option, which we refer to as the All-in-One Option, involves conversion of the transmission file directly to XML. Headers and values are individually tagged, and the message content and attachments remain in their original transmission format. This approach can be likened to the 'XML Wrapper' approach. The second option, which we refer to as the Split-Files Option, involves a more complex record object. The transmission file is broken down into its component parts, headers and values are paired and marked up, and the message body and attachments are saved in their native formats. Each part is saved in the format most appropriate for the record at the time.

The second scenario, which we refer to as Pre-Use Provision, relies on the integration of archival and good record keeping practices into current email usage. It is the first step towards formalising emails, using a shared central filing system in a way that is easy and reliable for long-term storage. We have developed an add-in for Microsoft Outlook 2000 that restricts the amount of freedom users currently have with formal and official emails. Additional metadata is added to the file before it is sent, the structure of the message body is clearly defined, a preview of the record is provided in HTML and when it is sent, an XML version of the message is stored on a central server for long term storage.

The All-In-One Option

This first option is suitable for messages that have already been used in the course of business and now need to be authentically retained for the long term. The first step in implementing this option is to receive the transmission file⁹. The transmission file is structurally divided into several parts. The component parts dictate the overall make-up of the email, so each will look different depending on its content and treatment.

⁹ Receiving the transmission file, instead of the message in its 'native' format, allows you to check that the email is complete and uncorrupted at the point of accessioning, and Archives do not have to retain old copies of email applications in order to open the file. It is also the most complete and definitive file format for the record as it contains all of the information that passed through time and space in the purposes of carrying out a transaction.

Generally, all transmission files start with the basic header information. There is much more to this than is displayed by Outlook, for example. The headers include the date and time the message was received by the servers it has passed through, the MIME type, whether there are any attachments, whether the email is part of a thread, as well as the usual To, From, Subject and so on. After the headers, the message is clearly delineated into parts. Each part has a header that identifies its content type. The character set is specified, as is any encoding and file name. The header is followed by the content for that part, be that the body of the message, an attachment, an inserted item or image, or any other content object you may want to include in an email. The parts are often encoded – for example an MS Word attachment will be encoded in Base64 and will not be human readable until the parts have been reassembled.

The transmission file is uploaded into the Testbed system as an Original Record Object (ORO), and the Testbed system takes this as its starting point. The conversion to XML is carried out automatically in the Testbed, using tools written with Java. We use Java for this purpose because the Testbed system is built on an Oracle Internet File System (IFS) database, and the tools available via the API (Application Programming Interface) for the Oracle IFS require Java manipulation.

The transmission file is converted directly to XML, with XML tags representing each MIME Header-Name and Value:

```
- <record>
+ <metadata>
- <email>
- <headers>
- <header>
  <headerName>Received</headerName>
  <headerValue>from ms01.dh02.ictu.nl ([10.19.5.2]) by tb1 (Build 101 8.9.3/NT-8.9.3) with ESMTMP id IAA
  <Testbed@10.18.1.241>; Tue, 12 Mar 2002 08:55:04 +0100</headerValue>
</header>
- <header>
  <headerName>Received</headerName>
  <headerValue>from gw01.dh01.ictu.nl (unverified) by ms01.dh02.ictu.nl (Content Technologies SMTPRS
  ESMTMP id <T59961a58390a1305020b3@ms01.dh02.ictu.nl> for <Testbed@10.18.1.241>; Tue, 12 Mar
  08:39:58 +0100</headerValue>
</header>
- <header>
  <headerName>X-MimeOLE</headerName>
  <headerValue>Produced By Microsoft Exchange V6.0.5762.3</headerValue>
</header>
- <header>
  <headerName>content-class</headerName>
  <headerValue>urn:content-classes:message</headerValue>
</header>
- <header>
  <headerName>MIME-Version</headerName>
```

Figure 1: Screenshot of the start of the XML file produced by the All-In-One Option

The XML file (see figure 2) consists of parts, like the transmission file. The first part is a manual set of metadata that has been entered at the point of upload and associated with the file before the conversion process has taken place. This is archival metadata and consists of such metadata items as Submitter, Date of Transfer, and Registration Number. The second part of the XML file begins with the Record Header. This is the section we are all accustomed to seeing at the top of our email messages, plus additional metadata picked up as the message is transmitted such as the Received fields we can see above. All of this ‘Top Header’ information is marked up in XML.

The third 'part' of the XML file contains the message Body, or the message content. This is not coded into XML, but is stored in a Character Data section that will not be parsed by the browser.

Sometimes messages contain more than one Body section. If a HTML email has been sent, the transmission file will usually contain a plain text version of the message, as well as the HTML version. This is so that people whose applications do not have HTML capability can still see the message content, if not the intended appearance. This extra section has another header that defines the content type definition.

The XML file in this instance is a kind of wrapper. The metadata and headers are marked-up in XML, but the body – the attachments and the message content – are not. This means that although the digital components of the file are securely stored together, the attachments have not been decoded (from, for example Base64), and the metadata cannot be manipulated without requiring change permissions. Allowing changes to be made to the file increases the chance that the record will be altered and its authenticity compromised.

```

- <record>
+ <metadata>
- <email>
+ <headers>
- <multipart-mixed>
- <part>
+ <headers>
- <body>
+ <![CDATA[ ]]>
</body>
</part>
- <part>
- <headers>
+ <header>
+ <header>
+ <header>
+ <header>
</headers>
- <body>
+ <![CDATA[ ]]>
</body>
</part>
</multipart-mixed>
</email>
</record>

```

Fig 2: Collapsed XML File, All-In-One Option

To cater for this problem, we developed our second email process: the Split Files option, again for a Post-Use scenario.

The Split Files Option

Once again we start with the flat text raw email transmission file. The Testbed again uses a Java tool, for the same reasons as last time, to perform the conversion. You can see from the screenshot below (figure 3) that the XML file produced has evolved from the previous process.

```

<?xml version="1.0" ?>
- <record>
- <email>
- <headers>
  <Received>from ms01.dh02.ictu.nl ([10.19.5.2]) by tb1 (Build 101 8.9.3/NT-8.9.3) with ESMT
  <Testbed@10.18.1.241>; Tue, 12 Mar 2002 08:55:04 +0100</Received>
  <Received>from gw01.dh01.ictu.nl (unverified) by ms01.dh02.ictu.nl (Content Technologies
  id <T59961a58390a1305020b3@ms01.dh02.ictu.nl> for <Testbed@10.18.1.241>; Tue, 12
  +0100</Received>
  <X-MimeOLE>Produced By Microsoft Exchange V6.0.5762.3</X-MimeOLE>
  <content-class>urn:content-classes:message</content-class>
  <MIME-Version>1.0</MIME-Version>
  <Content-Type>multipart/mixed; boundary="----_NextPart_001_01C1C99A.A6A40E16"<
  <Subject>FW: mr-KMail-msg-24</Subject>
  <Date>Tue, 12 Mar 2002 08:51:00 +0100</Date>
  <Message-ID><C3719FE945884C4EBEFA3C4C72EEFE98330689@gw01.dh01.ictu.nl></Messa
  <X-MS-Has-Attach>yes</X-MS-Has-Attach>
  <X-MS-TNEF-Correlator />
  <Thread-Topic>mr-KMail-msg-24</Thread-Topic>
  <Thread-Index>AChJMPDEdvxIrIXzQ7+EV6bGW3W32wAaauKw</Thread-Index>

```

Figure 3: Start of XML File - Split Files Option

This time, the headers are not simply divided into header item and header value tags; the item becomes the XML tag, and the value is recorded inside. This makes for a much cleaner and smaller XML file. However, in this process, only the headers are transformed into XML. If we examine the record object contents, we can see that this record object consists of several component parts. (See figure 4)

The image presented here is a screenshot from our Testbed system. You can see the collapsed record tree and its various component parts. The first part is metadata. This is stored within the Testbed database and is rendered simply as a list of items. The next part is the Body. In the case of HTML or RTF emails, there can be more than one body. Attachments are listed separately and are left in their native format. Finally we have the XML file. The XML file contains all of the header information that was in the original transmission file. The headers are standard type headers; the only issue we have left to resolve around this is maintaining the link between the content type definition headers and the message parts – for example, we need to know whether an image was an inserted image, whether it was part of the background, or whether it was an attachment. The content type definition helps clarify this information – if we get it wrong, the authenticity of the record is in danger.



Fig 4: Sample Collapsed Record Trees

The Split Files option is suitable for messages with complex content, such as formal emails with background images and attachments. This is because it allows the preserver to carry out separate preservation action on the different parts of the record, as and when required, without endangering the rest of it – for example, to migrate an attachment from Word to PDF and update the metadata but leave the message body untouched.

However, although we had developed two successful approaches for conversion of existing emails into XML, we realised that the transmission file, whilst an essential source of metadata, did not hold all that we wanted and did not provide enough context. We needed extra metadata – for example, it is not enough to simply have an email address for a recipient: the address must be associated with a person, and preferably also an institution. To address this and other aspects, we developed a demonstrator that allows users to incorporate this essential information into any future messages they send.

The Forward Facing Option

The Forward Facing Option is essentially an Add-In that currently works on Outlook 2000 but which can easily be adapted to work on any email application. It has two basic components: an add-in for Outlook, converting the email to XML behind the scenes; and a Web Service – validating the XML, transforming the message to HTML and sending that separately.

The demonstrator allows for two types of e-mail message: informal or formal. Informal messages are those not related to the official business of the department and are not archived. The only restriction that the e-mail tool applies to an informal e-mail is to insert a disclaimer at the top of the message to tell the recipient that this is not an official message.

Formal messages are assumed to have the same status as a letter on the department's headed paper and we have used that analogy in several aspects of our approach. In a formal e-mail, the user is prompted to fill in a number of metadata items (Dossier, Program, Urgency etc.) Other items, such as the user's name, organisation and contact details only need to be entered once and thereafter are filled in automatically. Outlook interacts with the user's Address Book or Contacts folder to extract additional information about the recipients of the message, and includes this in the message metadata.

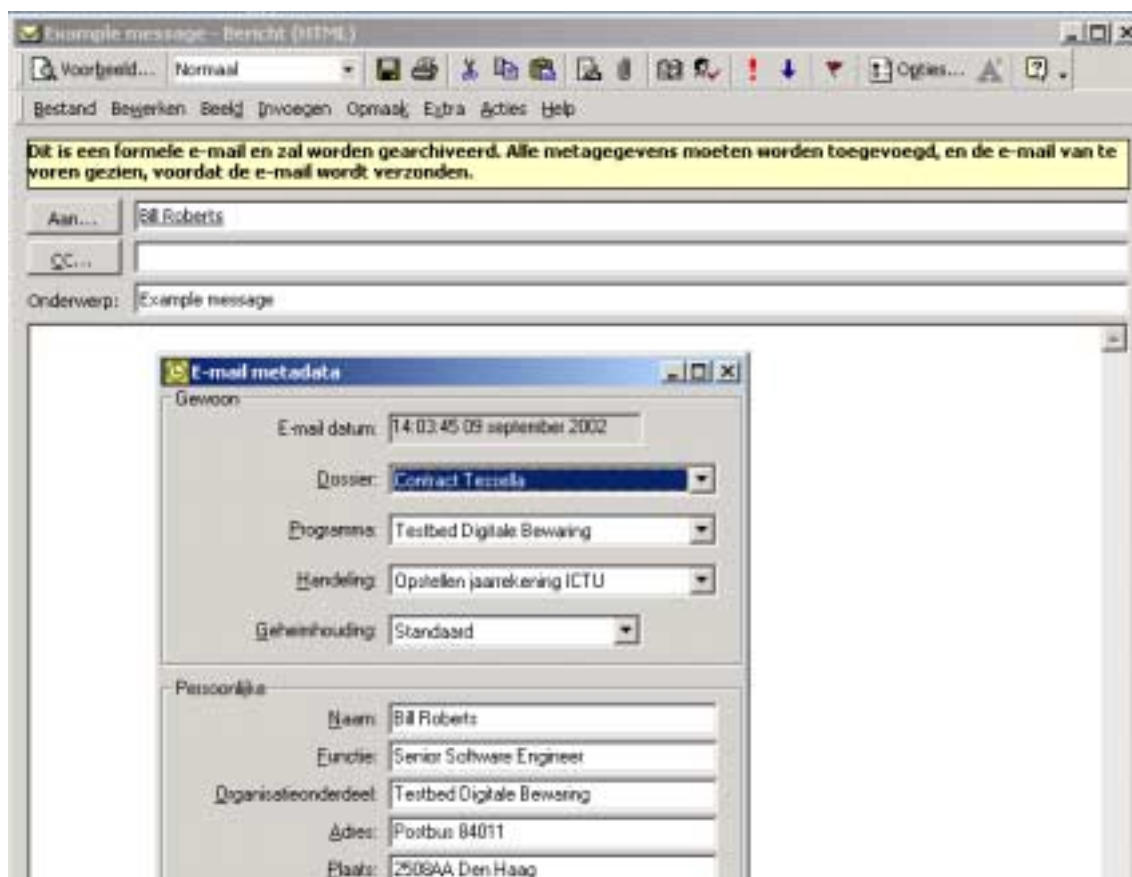


Figure 5: Additional Compulsory Metadata fields: the top half of the frame must be filled in each time a new message is sent; the lower half of the frame must only be filled in the first time the demonstrator is used.

The message content is filled in as normal, and the user must apply styles to each part of the message, in a similar manner to MS Word allowing users to specify heading styles. All messages must begin with an opening phrase, include standard content, and finish with a closing phrase. The customised version of Outlook combines the metadata and the message content into an XML file. This XML file is then transmitted to a central server, which checks the XML against a schema and stores the XML file in the archive. The server also applies an XSL (Extensible Style sheet Language) style sheet to the XML file to produce a formatted

HTML file in the house style of the organisation, applying fonts, colours and logos for example. This HTML format message is sent back to Outlook to become the actual e-mail message that is sent to the recipients. The User sees this preview; if it is satisfactory then the HTML message is sent and the XML version is stored in a central repository for later archiving.

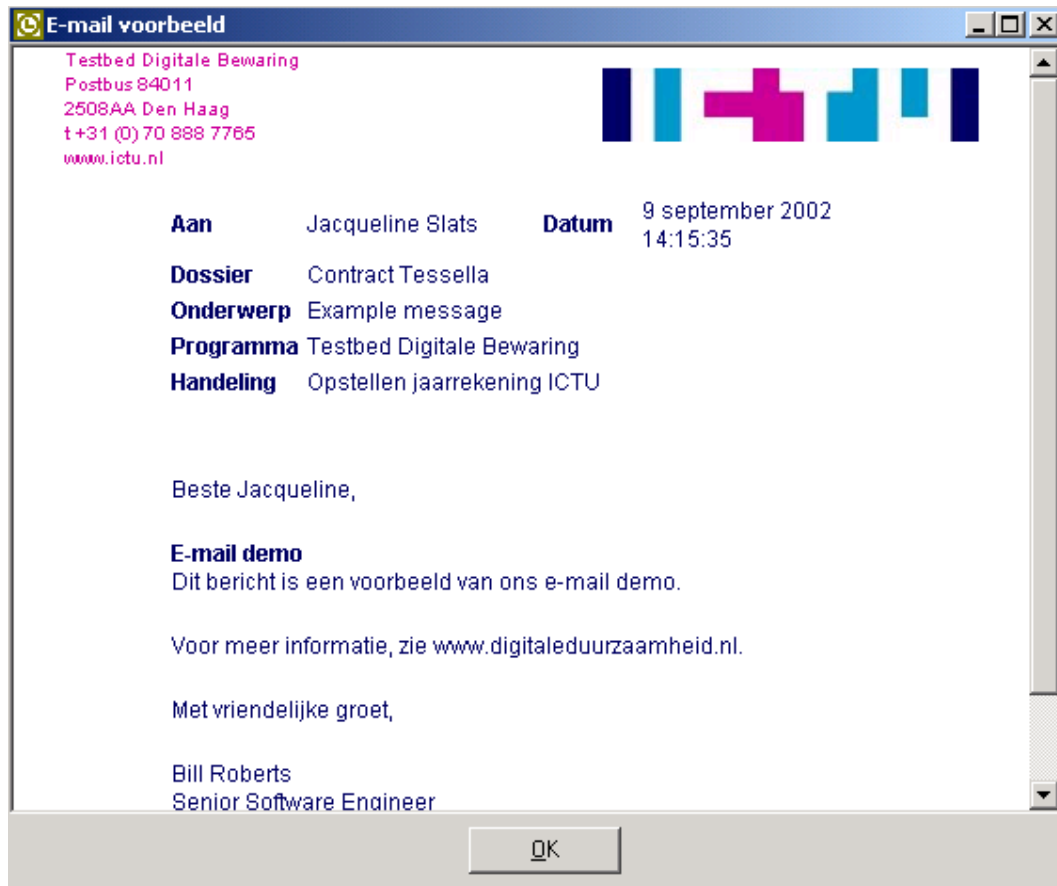


Figure 6: Preview of the formatted message in HTML, including additional metadata.

In this way, the application can ensure that the e-mail message is archived together with the required metadata and at the same time, can apply a uniform house style to the official messages produced within an organisation. The second generation of the demonstrator is now in development. We will soon begin a pilot programme with the demonstrator and will be monitoring use and receiving feedback for further development.

Conclusion

Our experiments have led us to believe that XML is a highly suitable and promising format for the long-term retention of authentic email messages. The implementation and specifics of the approach depend largely on the requirements, abilities, and legal constraints of an institution. We have developed these three approaches during the course of experiments in our Testbed, and whilst these are not the only ways in which an email/XML preservation

approach can be implemented, their basic premises can also apply to adaptations of these approaches. Agencies and organisations should pick the approach that is best suited to their needs and modify it as necessary. If possible, message creators should always store additional record keeping metadata with the message at the point of creation, and all users should be trained in the correct construction of, not only emails, but also all other applications used in the course of day-to-day business. Messages and records created in a stable manner today stand a far greater chance of lasting over the long term than badly created records, whose integrity and authenticity may not last as long as the following day! Storage in a central location allows an institution to regain control over their records, instead of leaving messages in individuals' mailboxes. Storage of the message as an XML file helps ensure that the record can be retrieved and reread for many years to come.

For further information, visit our website: <http://www.digitaleduurzaamheid>, or contact our team directly: Testbed@ictu.nl