

E-mail-XML Demonstrator: Technical description

Introduction

In the Testbed Digitale Bewaring project we have recently developed prototype software to allow us to investigate in detail the issues around the long-term preservation of e-mail messages and to illustrate possible solutions to the problems that many government organisations are faced with.

Objectives

There were three aspects of e-mail use that we considered:

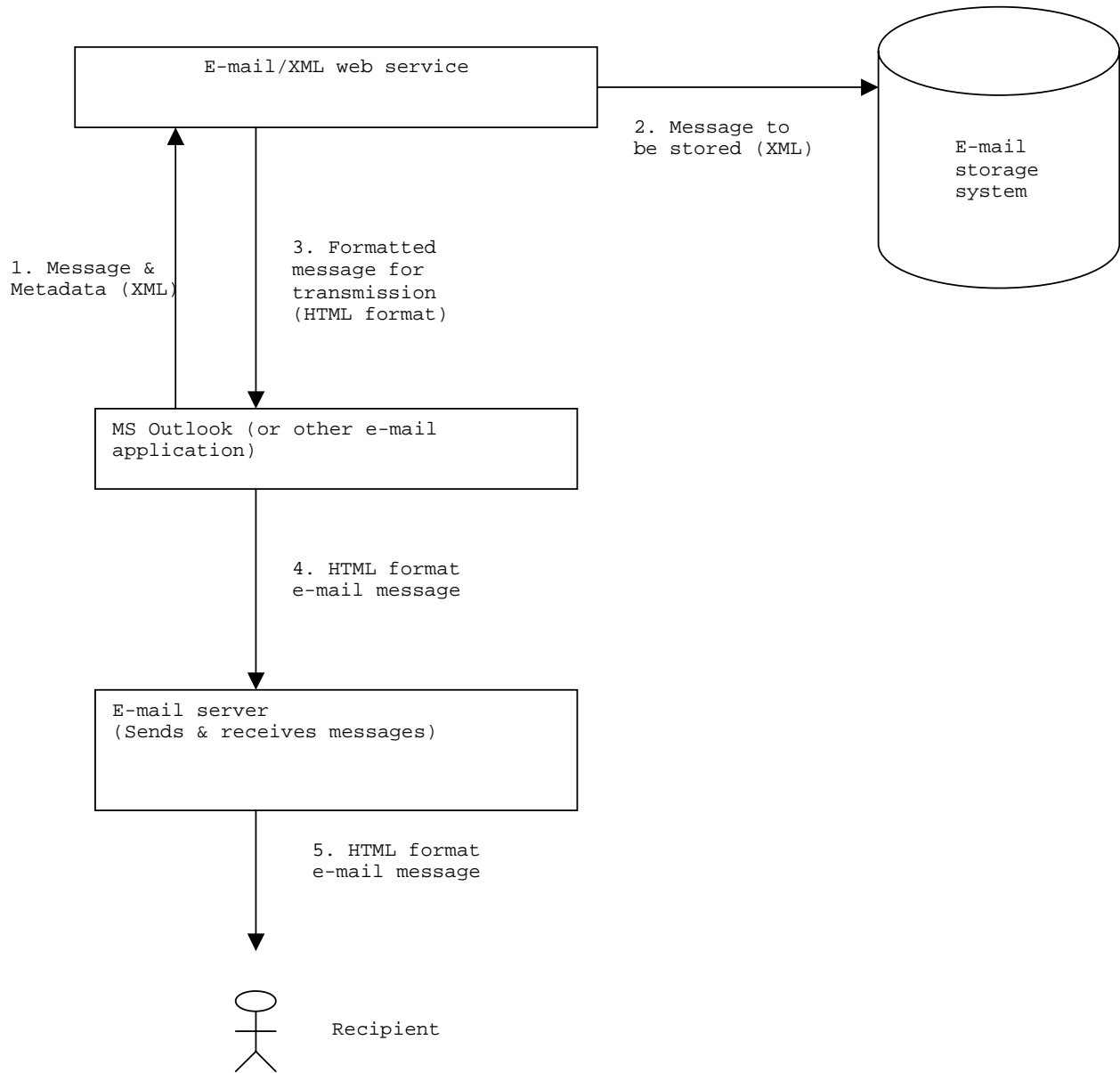
- E-mail is now becoming more and more widely used for official communications, but in some cases people still use quite an informal approach to composing their e-mails. We wanted the content and style of official messages to resemble more closely a letter written on official headed paper.
- If an e-mail needs to be filed, many organisations either print the message onto paper and store it in a paper file, or messages are stored in an ad hoc way by individuals in their personal e-mail folders. We wanted to set up a central filing system for e-mail messages, without involving the complexity of a full document management system.
- The e-mail messages must be stored in a way suitable for easy and reliable long-term preservation, so that they continue to be accessible and understandable for as long as required: that could be 10 years or 100 years or more.

We wanted a solution that would be simple to use and relatively simple to implement.

Our solution

We based our approach around Microsoft Outlook: this is because this is the most widely used e-mail application in the government and because we wanted our solution to be integrated into the familiar working environment of the users and also to be able to make use of many of the facilities provided by Outlook, without having to re-create them for ourselves. Although we chose Outlook for our demonstration, a very similar approach could be taken with other e-mail applications and we do not intend to imply that Outlook is better or worse than any other choice.

Our approach involves a customisation of Outlook, using an ActiveX DLL written in Visual Basic. This adds a number of additional features to the standard Outlook and also prevents access to some of Outlook's usual functions. The overall system follows a client-server architecture. The clients, that is the instances of the customised Outlook running on users' desktops, communicate with a central server that takes responsibility for archiving the messages and for applying the house style to outgoing e-mails. We refer to this as the "archiving server", to distinguish it from the usual e-mail server, which is still required in the normal way.



Metadata

The standard format for e-mail messages (as defined by the Internet Engineering Task Force) contains a lot of useful information, but for meaningful long-term preservation, it is beneficial to provide additional information. In our demonstration, we collect the following additional information:

- Context information such as dossier, handling, programma (or onderdeel etc.)
- Information about the sender of the message, such as full name, job description, postal address, telephone number
- Information about the recipients, such as full name and organisation.

An additional metadata entry form must be completed before a formal e-mail can be sent. However, this requires very little effort from the user: the personal information only needs to

be entered once and can then be filled in automatically by the software and the information about dossier, handling and so on is given by choosing from a list of predefined options.

The e-mail address alone is often not sufficient to identify a person from the point of view of preserving the context of a message. Therefore our demonstration makes use of the Contacts folder in Outlook, where additional information about the recipients of a message can be defined. Before the user can send a formal e-mail to someone, they must enter the recipient's full name and organisation details into the Contacts folder. This is then extracted by the software and stored in the message metadata.

Storage in XML format

In our demonstration, both the message contents and the metadata are stored in an XML document. There are two main reasons for this: one is because XML is a well-defined open standard that is widely believed to be a good format for long-term preservation; the other is that use of XML for the message content allows the use of eXtensible Style sheet Language (XSL) to define a transformation from XML to HTML. This takes the content of the message and the metadata and presents it in a formatted way, specifying the overall layout of the message, the choice of fonts and colours and the inclusion of a logo or other images. By doing this centrally, a common house style can be applied to all formal messages. Any change in the style only needs to be made in a single central place.

The Outlook user interface has been modified to guide the user through creating the different elements making up the content of the message. This is then converted to XML behind the scenes. The Outlook extension combines the message content and metadata into a single XML document. This is transmitted to the archiving server, encapsulated in a SOAP¹ message. The archiving server stores a copy of this XML file in the archive. It applies the XSL style sheet to create the formatted HTML message and sends that back to Outlook, also as a SOAP message.

Another function of the archiving server is to verify that the XML produced by Outlook satisfies the XML schema defined for the message. By checking that the XML document obeys the schema, then we can be sure that the XSL transformation will work correctly.

Attachments can be added to messages in the normal way. These are transmitted from Outlook to the archiving server. Each attachment file is stored separately in the archive and a link to the attachment and key metadata items are stored in the archived XML file. Because essentially any type of file can be attached to an e-mail message, the long-term preservation of attachments is a difficult problem, not directly addressed in our demonstration. However, our approach of storing e-mail messages as XML with metadata about the attachments, including information on the type of the file, allows easier control and monitoring of the type of attachment files in the archive and so is a first step to allowing a preservation solution to be applied.

¹ Simple Object Access Protocol. See www.w3.org/TR/SOAP for more information.