XBRL and Office 11: A Field of Dreams

"IF YOU BUILD IT, HE WILL COME," INTONED
the mysterious voice to the Iowa farmer in the 1989
Kevin Costner movie *Field of Dreams.* Costner's character,
Ray Kinsella, proceeds to build a baseball field right in
the middle of a patch of corn. Friends and neighbors
alike tell him he's crazy to give up good farmland to build
a baseball diamond in the middle of nowhere. Soon,
word spreads that strange and wonderful things are hap-
pening on this special field, and people begin to arrive
from all over. They aren't disappointed by what they see.

Microsoft is about to finish building its own field of
dreams. Its world-leading office productivity suite,
Microsoft Office, will soon undergo a major upgrade,
changing the field and the game of business documents
forever. The Microsoft dream is to build a set of smart
documents that can take advantage of all the positive as-
pects of eXtensible Markup Language (XML) while re-
taining the ability to look and act like standard Office
output. Coded Office 11 and due for release in mid-2003,
this next version of Microsoft Office has native XML built
into Word 11, Excel 11, Access 11, FrontPage 11, and
Visio. Native XML means that the XML files created by
Office 11 documents will be based on the XML standard
published by W3C (the World Wide Web Consortium).
Office 11 documents can be saved either as XML docu-
ments or in traditional Office file formats (.doc for Word,
.xls for Excel, etc.).

Office 11 also has a new tool called XDocs to capture
information according to a customer-defined XML
schema. According to the W3C, a schema is a set of
shared vocabularies, and it allows machines to carry out
rules made by people. XBRL, the eXtensible Business Re-
porting Language, uses XML schema to define a set of
tags that describe all aspects of financial statements.

Benefits of using XML are:
- Platform (operating system, hardware) independent;
- Prepare once, use many times;
- Transmittable over the Internet;
- Keeps data separate from display;
- Users define their own tags.

Unlocking Data Inside Documents

The power of importing customer-defined schemas will go
a long way to freeing up the knowledge worker from data
isolation. Data isolation can occur whenever data an em-
ployee needs for a decision is located in a document that
the employee doesn't have quick access to or, worse, the
worker has no knowledge that the data exists. With XML
tagging, data that used to be locked up in an enterprise sys-
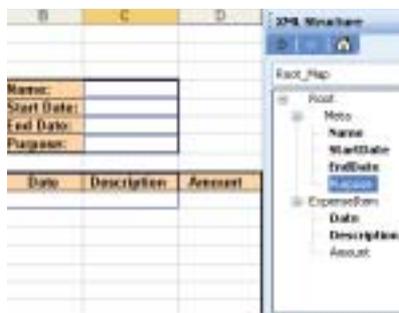tem becomes visible and usable within the company.

All documents created in Office 11 using the XML for-
mat can be shared easily, either within the company or
with trusted partners. When a document is created in
current versions of Microsoft Word, the business intelli-
gence created within the document is typically locked in-
side the hard drive of a PC or stored in a static format on
a network. Microsoft's dream is for each Office 11 docu-
ment to have the features and characteristics of a data-
base right in the document. This astonishing feat is possi-
ble because XML tags impart both content and contextu-
al information, permitting the use of search techniques
inside individual documents.

Let's look at an example of how Excel 11 might work.
Figure 1 is a small sample of an Excel form for an ex-

pense report. Note that the fields are already laid out in Excel and there's an XML dialogue window to the right of the spreadsheet cells. The XML structure on the right of the spreadsheet cells suggests the structure and element names for filing expense reports. Users will adopt this XML structure for the Excel worksheet cells by mapping the suggested structure to the appropriate cell on the worksheet.

Once the XML has been mapped to a particular cell, the data placed in that cell is automatically coded or tagged with XML code. The worksheet can then be stored as an Excel file in .xls format or as an XML document with an .xsd file format. ("These schemas are often referred to as XSDs because they are generat-

ed using a standard defined by the W3C called the XML Schema Definition Language."[1])

If stored as an .xsd file, additional reporting and analysis capabilities arise. For example, a simple query of the "purpose" field on the worksheet could yield all recent trips to industry conventions.

Only the cells that are "mapped" from the XML structure to the worksheet are now capable of im-

parting XML tags on the document. This creates semi-structured documents with fields or islands of meaning within otherwise ordinary documents. As a result, users are able to define the structure and the type of content that each data element in a document can contain.

The real beauty of Microsoft's XML support in Office 11 is that a user can get all the benefits of the eXtensible Markup Language, retain all formatting of the original application, and have to learn hardly anything about XML.

What about XBRL?

Microsoft has a strong history of supporting XBRL. Jean Pacoli, an XML architect at Microsoft and a co-creator of the XML 1.0 W3C recommendation, mentioned XBRL in a Microsoft PressPass interview November 12, 2002, about creating XML files in Office 11 applications. Pacoli said, "We want to give our customers the choice to decide what file format they want to use, down to the XML schema they employ…. Others may want to use a specific XML schema, like XBRL. We leave the choice up to them." ∎

Neal Hannon is an accounting lecturer for the Barney School of Business at the University of Hartford and serves as chair of the IMA's Information Technology Committee. Author of two books and numerous articles, Hannon is the IMA's representative to the XBRL International consortium. An elected member of the XBRL-US steering committee, he is the chair of the XBRL-US Education Work Group.

[1] http://www.microsoft.com/presspass/features/2002/nov02/11-12xmloffice.asp, ©2003 Microsoft Corporation