

# Making Terminology Work: Driving In Situ Use of Wicked Classification Schemes with Commoditised Software

**Werner van Huffel**

*Industry Technology Strategist (Healthcare)*

*Microsoft Asia Pacific*

*wvhuffel@microsoft.com*

## 1. Introduction

There is no denying that the area of healthcare IT implementation is an environment of Wicked complexity [1]. In order to create a semblance of order to IT implementation healthcare informatics has developed many standards such as HL7 and openEHR. However, these standards do not themselves facilitate the industry requirement for semantic interoperability, rather they provide a level of syntactic interoperability between integrating healthcare applications. Without the use of terminology standards such as SNOMED, ICD or LOINC et al, standards such as HL7 and openEHR are just nice ways to move data around.

It is also proving difficult to incorporate terminology standards into the work practices of clinicians as it corrupts the free flow of the patient-clinician interaction. The more time a clinician spends attempting to work to the way a computer is programmed rather than the way a patient trajectory develops could increase the risk to the quality and efficiency of care. Too many times clinicians are being constrained by the wicked nature of IT systems implemented in a monolithic model.

With the advent of Service Orientated Architectural guidance and development strategies [2], IT is able to offer healthcare implementation a new strategy – rapidly adapt and support clinical requirements and change. IT now has a way to better support, rather than subjugate clinical work practices.

Microsoft is working on SOA strategies and examples to demonstrate possible better avenues to support clinical work practices in situ of patient-clinician operations.

## 2. The problem

**Scenario:** A General Practitioner (GP) is seeing a patient and noting down their clinical profile using the SOAP (Subjective, Objective, Assess and Plan) pattern and submit them for a referral. They are using their Practice Management Software (PMS) to create electronic clinical notes.

In this scenario the doctor could create clinical notes in a several ways. The 2 most prominent are to use either:

- clinical narrative (essentially “free text”), entered into separate text fields, or
- a selection-list protocol

Hybrids of these 2 methods are also used.

The use of narrative text is more versatile in that it is not constrained by prebuilt design requirements. In narrative text it is possible for the doctor to construct any character sequence to represent the situation confronting them. The limitation of this method is that it can be cryptic to other clinicians or at a later date when the notes are reviewed.

A selection-list protocol allows greater control of the content of the note but may not allow the doctor to convey sufficient nuance – as could a narrative text input method. One other issue with select-list methods is that the content of the list may need to be defined at design time.

Both these methods may require the incorporation of codes for classification and/or terminology purposes. These classification and/or terminology purposes could relate to charge management or continuity of care requirements. The classification schemes themselves can have high organisational complexity (i.e. they can be considered to be “wicked” systems[1]) – for instance SNOMED-CT could be considered a wicked terminology system.

The incorporation of these requirements place an added burden on the clinician to conform to an external requirement which is sometimes counter intuitive to the way they work. Clinicians do not like to be told what to do, especially by software.

The use of classification/terminology standards in healthcare environment currently follows a few definable avenues:

- Ignore - No use of terminology (the most common case)
- Grudgingly adhere – “I’ll do it, but don’t what to.”
- Low rigor adherence – which lead to low levels of codification quality
- Post noting transcription – someone else’s problem – introduces transcription errors

For terminology/classification to work we need to have consistent high levels of quality codification and contribution. The IT question becomes: “What can be done to make it easier and simpler for the clinician to use terminology standards so that collection is more accurate and less of an operational burden?”

### **3. A possible solution**

Using the question “What can be done to make it easier and simpler for the clinician to use terminology standards so that collection is more accurate and less of an operational burden?” as a starting point Microsoft and HealthLanguage investigated several possibilities with clinical users

Using SOA guidance, such as that provided by the Connected Health Framework (CHF) [3], a possible hypothesis presented itself; “Is it possible to incorporate the terminology insertion service while the doctor creates the clinical note – in situ clinical encoding?”

### **4. Testing the hypothesis**

The concept of in situ clinical coding is not new, it can be delivered using many techniques. The creation of clinical notes is a mainstream practice in healthcare. Many systems on many different platforms, that incorporate clinical note taking, use one of the two methods or a hybrid of both. Microsoft, as a platform infrastructure, has built developer components to facilitate such application implementations.

#### **4.1. A brief History of Terminology services and Microsoft**

Microsoft and the NHS, with the help of HealthLanguage, initially created the SNOMED clinical noting control for the Common User Interface project [4]. This, however, was focussed at delivering web (AJAX) based components using a web service (SOA model) provided by HealthLanguage. The initial CUI based control was a hybrid of the 2 methods of clinical noting. These components were not commoditised software in the sense that they were developer controls and not readily usable by clinicians without considerable design and development of full systems.

At some time around 2007 Raphael Mastier (Microsoft, France) created a solution that used the Office Business Applications (OBA) development framework to incorporate a similar service into Microsoft Word 2007. This development was called “OfficeDoctor”. This implementation used a separation of functionality technique for code selection – which is similar to current mainstream use of select list methods. It was not very efficient use of a clinician’s time. However, this development took one very large step forward in recognising that the general operational practice of clinicians followed that of commoditised word processing software – such as Microsoft Word.

In 2008 the OfficeDoctor functionality was taken up by the Microsoft team developing the Common User Interface for the UK NHS CUI project. Using their extended capabilities in user interface design they were able to incorporate more advanced software controls that held greater functionality in the software – thereby reducing the operational burden on the clinician. The NHS UK team, led by Andrew Kirby, incorporated their CUI control capabilities into the OfficeDoctor solution developed earlier. At this point the Clinical Document Architecture (CDA) standard was also incorporated to facilitate semantic interoperable terminology encoding.

This in situ codification within a commoditized software application (Microsoft Word 2007) proves that it is possible to change the way software interacts with clinicians.


## 4.2. Working with the solution

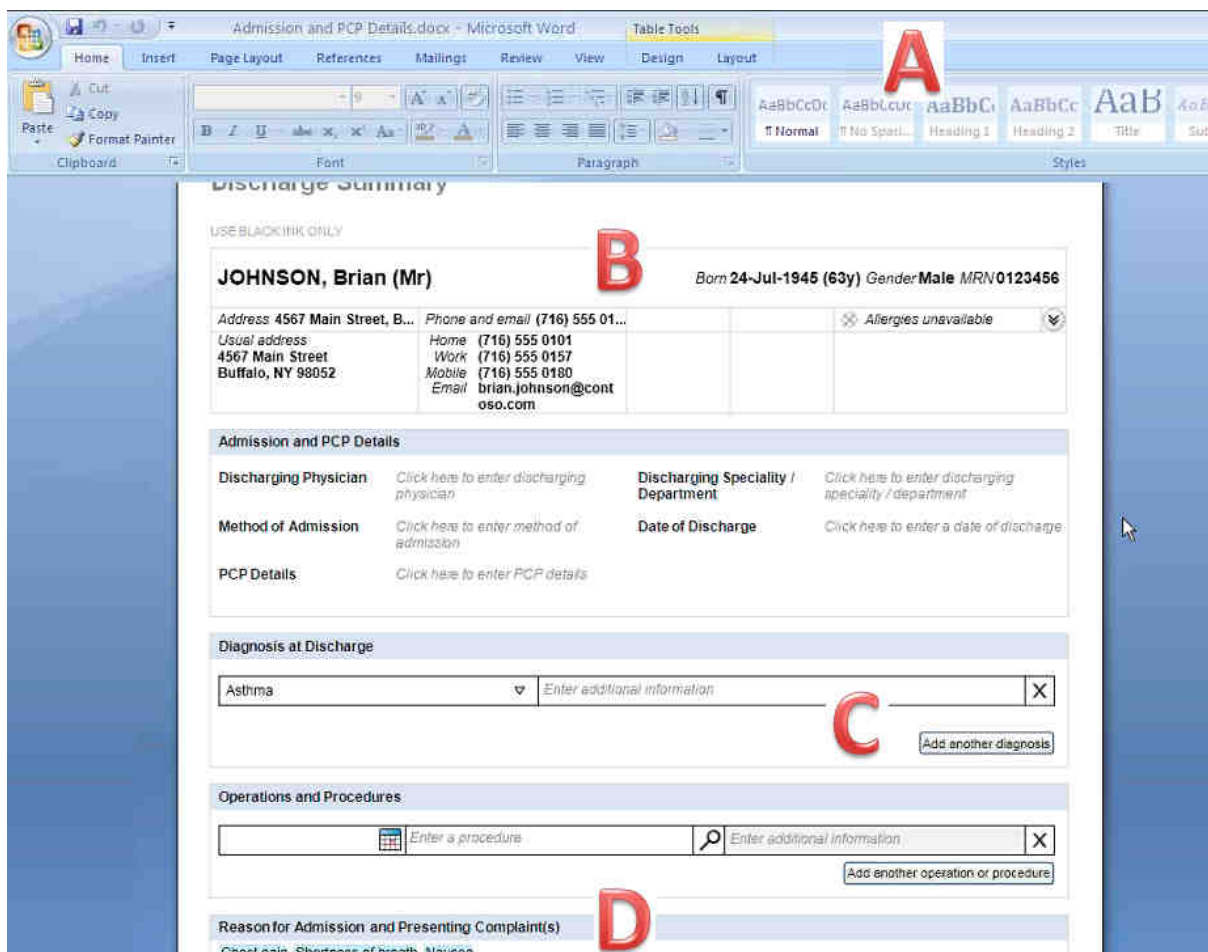
Consider the scenario of creating an e-Referral. A clinician creating a referral or discharge document, or any consultation note, would do so in a form that is familiar to many, that of a word processor. In starting the process the normal practice is to select a template which defines structure and function to the word processor. Only functionality contained within the template will be active during the creation of the clinical notes. It is possible to create a functionality which is available to all templates and place it in what is called the “Ribbon” (“A”) of the word processor (Microsoft Word 2007) – for this solution the template-based code approach was taken. The word processor creates a new document (“B”) based on the selected template and clinical noting can start.

As part of the process a PMS or HIS system may also be queried, using web services or HL7 messaging, to pre-populate fields in the created document (“B”). Another way to populate such information is to embed a CDA structure into the created document and reference the CDA data using XPath statements. Such integration was not in the scope of the demonstration solution as it has been proven in other solutions[5].

Finally the clinician is able to enter data in any form and the code-intelligence of behind the document and in accessed web services manages the complex terminology interactions. In the case of the example solution the clinician is able to use both select-list style (“C”) and narrative text style (“D”).

Let us first address the select list type functionality (“C”). In the example screenshot above the term “asthma” has already been populated. As the clinician wants to add more items they select the “Add another...” functionality (“E”) and type another entry into the table row created.

A specialised control – signified in this example by a “” - is linked to a terminology search web service that then takes the text and intelligently parses it to provide a contextualised list from which to select. The clinician is then immediately presented with an appropriate selection. (“F”).



The screenshot shows a Microsoft Word document titled "Admission and PCP Details.docx" with the "Table Tools" ribbon active. The document content is a "Discharge Summary" form. The form includes a patient information section with the name "JOHNSON, Brian (Mr)", birth date "24-Jul-1945 (63y)", and gender "Male". Below this is a table with contact information and a dropdown menu for "Allergies unavailable". The "Admission and PCP Details" section contains fields for "Discharging Physician", "Discharging Speciality / Department", "Method of Admission", "Date of Discharge", and "PCP Details". The "Diagnosis at Discharge" section has a dropdown menu with "Asthma" selected and an "Add another diagnosis" button. The "Operations and Procedures" section has a table with an "Add another operation or procedure" button. The "Reason for Admission and Presenting Complaint(s)" section contains the text "Chest pain, Shortness of breath, Nausea". Red letters A, B, C, and D are overlaid on the image to highlight specific features: A points to the Word ribbon, B points to the patient information section, C points to the "Add another diagnosis" button, and D points to the "Reason for Admission and Presenting Complaint(s)" section.

Figure 1 - Loaded Terminology enabled document

Diagnosis at Discharge			
Asthma	▼	Enter additional information	X
Myocardial infarction	▼	Enter additional information	X
Enter a diagnosis	🔍	Enter additional information	X

**E**

Figure 2 - Accessing the terminology service

Diagnosis at Discharge			
Asthma	▼	Enter additional information	X
Myocardial infarction	▼	Enter additional information	X
hypertension	🔍	Enter additional information	X

**Hypertension** synonym of  
 Hypertensive disorder, systemic arterial (disorder)  
 is a Disorder of artery  
 and a Systemic arterial finding

**F**

Figure 3 - Interacting with the terminology

Reason for Admission and Presenting Complaint(s)
Mr. Johnson complains of chest pain, nausea, vomiting, and shortness of breath.

**G**

Figure 4 - Using narrative text controls

Reason for Admission and Presenting Complaint(s)
Mr. Johnson complains of <span style="border: 1px solid blue; padding: 2px;">chest pain</span> , <span style="border: 1px solid blue; padding: 2px;">nausea</span> , <span style="border: 1px solid blue; padding: 2px;">vomiting</span> , and <span style="border: 1px solid blue; padding: 2px;">shortness of breath</span> .

I

Figure 5 - Positive reinforced feedback

Once selected, the terminology is embedded into the background CDA, Archetype or some other clinical document interchange structure.

In the narrative noting functionality the clinician is able to enter text as they would normally, as free flowing prose in a text section (“G”).

When completed the clinician selects “Find...” and the terminology web service finds all associations and the control highlights the text for the clinician.

All this is done very rapidly, based on local environment Service Level Requirements. It is possible to achieve a very high level of service access and in situ terminology usage in this manner.

While all this is happening terminology encoding data is being presented to the clinician. This is non-intrusive positive reinforcement of the operations required of the clinician to maintain correct levels of quality and control over the terminology.

A by-product of the implementation on a word processor is that we are able to contain both machine readable (semantically interoperable) data, in the form of CDA or Archetype or some other containment structure, with human readable content; simultaneously. Previously it has only been possible to do one or the other. This is made possible by the OOXML document structure, “customxml” part [5].

From the point of view of the clinician, all that is seen is a normal word processor screen.

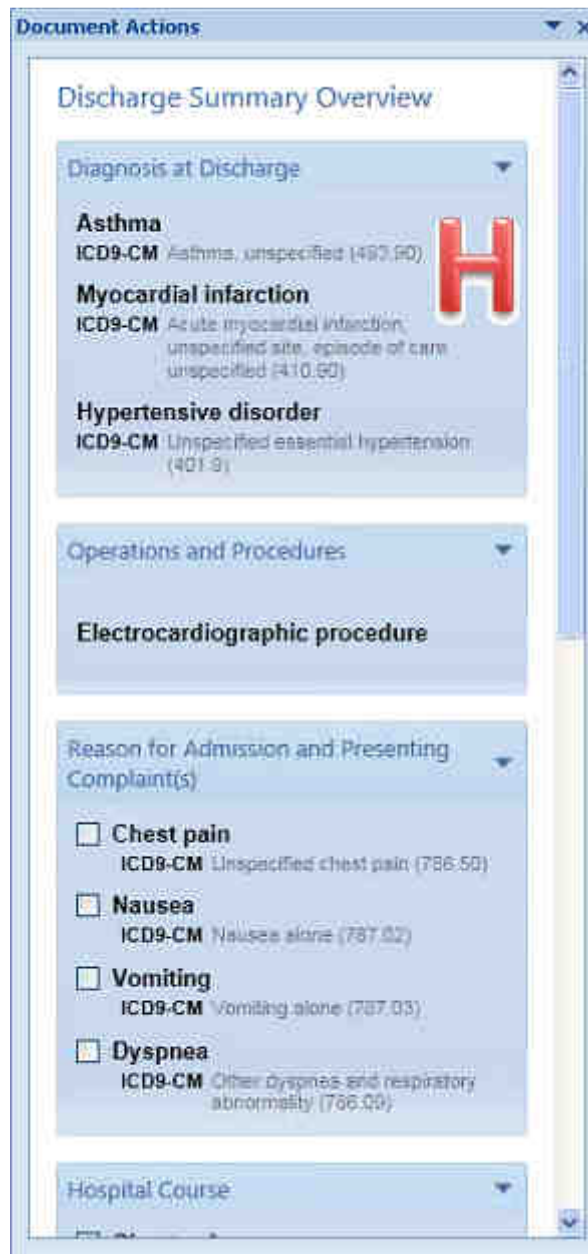


Figure 6 - Quick review and select pane

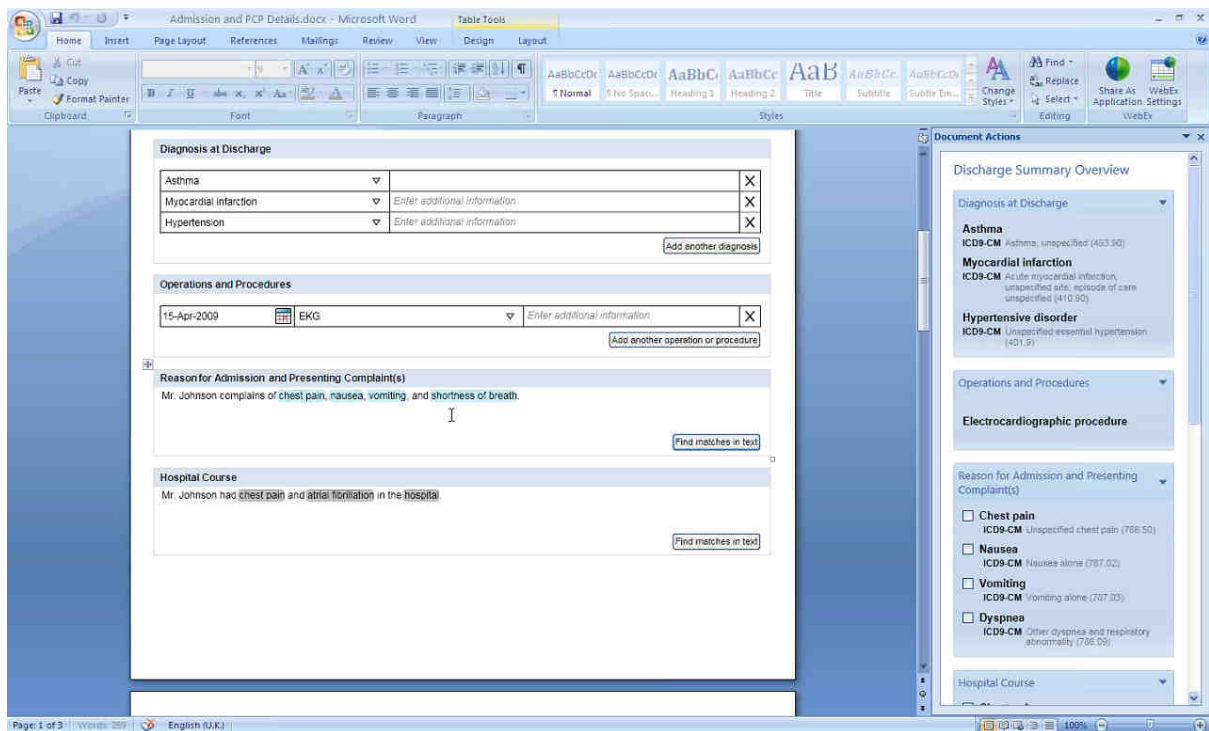


Figure 7 - Full Screen view

## 5. Generalising the approach

Though the scenario for this paper focuses on the use of such technology centred upon a Discharge/Referral-interaction framework, there is no technical obstacle to incorporating such functionality into other areas of clinical operations. Take for instance a need to drive terminology encoding within the surgical field. In such a case similar actions could be undertaken with a different template structure.

The possible use cases for such an approach could include (but would not be limited to):

- GP to Specialist ( for instance “eReferral”)
- GP to GP - or “Clinician to Clinician” – for patient exchange
- Hospital to GP (such as in care plan transfer or discharge summary handover)
- GP to Hospital (possibly in automating the admissions process)
- GP to Patient Employer (for absentee notices – with appropriate links to HR policy)
- GP to Patient Insurer (perhaps using Insurance codes for charging purposes)
- The creation of court requested documents (using court required terminology)
- Third-party coding of clinical notes
- Written Clinical exams or training (to train clinicians in such things as SNOMED)

The generalisation of approach hinges on the appropriate creation of a document template with which to generate the contextualised clinical document and corresponding appropriate storage structure (which, as in the case of this example, may be a Clinical Document Architecture (CDA) structure).

## 6. Recognised issues

Many clinicians still prefer paper for note taking purposes. In this regard having a more streamlined method for codification could make it easier transcription efforts. By making it easier it could also lower the cost and error rate of codification. The real issue is how to assist such clinicians to use computerised tools, not necessarily just terminology.

This is a demonstration of a possible avenue of development – there are many other methods to achieve this result.

In this effort we are concentrating on the commoditised software extension and not in bespoke application development. As such we are using only part of the full functionality of the word processor. The ability to create such structures may not be important to the clinician.

Not many other document formats have been investigated for this exercise. It may be that the same construction is possible in other structures.

It should also be noted that the document presented currently displays the ICD code while records, within the CDA construct, the SNOMED and ICD codes. The reasoning for this is that there is a terminology implementation complexity in creating such capabilities and it was thought that showing such a capability in a demonstrator such as this would be beneficial.

## 7. Conclusion

It is possible to create software using SOA guidance and development strategies to provide clinicians with software which works in situ with their operations. There is now demonstrable-capability of this under the Microsoft-HealthLanguage OfficeDoctor example.

## 8. Acknowledgements

Raphael Mastier (Microsoft, France)

Marc Horowitz (HealthLanguage, Global)

Andrew Kirby (Microsoft, UK)

Neil Jordan (Microsoft, Global)

Brian Levy (HealthLanguage, USA)

## 9. References

- [1] Westbrook JI, Braithwaite J, Georgiou A, Ampt A, Creswick N, Coiera E, Iedema R. Multimethod Evaluation of Information and Communication Technologies in Health in the Context of Wicked Problems and Sociotechnical Theory. *J Am Med Inform Assoc.* 2007;14(6):746-755
- [2] W3C. Web Services guidelines, [cited January 2009]. Available from: <http://www.w3c.org>
- [3] Microsoft Health. The Connected Health Framework, March 2009 [cited 3 April 2009]. Available from: <http://www.microsoft.com/industry/healthcare/technology/HealthFramework.aspx>
- [4] Microsoft. Common User Interface. 2008 [cited January 2009]. Available from: <http://www.mscai.net>
- [5] Van Huffel W. Document-centric clinical care using ECMA XML to document healthcare for machine and human readability. [cited 12 December 2008]. Available from: [http://hl7.org.nz/images/documents/ihic/presentations/Document%20Centric%20Clinical%20Care\\_Ecma%20XML%20and%20CDA.pdf](http://hl7.org.nz/images/documents/ihic/presentations/Document%20Centric%20Clinical%20Care_Ecma%20XML%20and%20CDA.pdf)