XML guidelines in ISO 20022 messages: explanations and improvements

Antero Mäenpää

University of Tampere
School of Information Sciences,
Interactive Technology
M.Sc. thesis
Supervisor: Erkki Mäkinen
4.6.2015

University of Tampere
School of Information Sciences, Interactive Technology
Antero Mäenpää: XML guidelines in ISO 20022 messages: explanations and improvements
M.Sc. thesis, 52 pages
May 2015

ISO 20022 XML guidelines for financial transactions contain large amount of information and their format vary greatly from bank to bank. Banks provide Handbooks for guiding their corporate customers to produce proper transactions, contained in XML payment messages. Quality of a Handbook directly affects the amount of resources needed for the customers to successfully implement a system generating payment messages. In this thesis I will investigate what information is crucial to include in an XML payment message Handbook and what is the best way to visualize it.

Key words and terms: ISO 20022, XML, guidelines, information visualization

Contents

1. Introduction

Financial transaction is an agreement between buyer and seller where an asset is exchanged for a payment. This can be as simple as buying chocolate at a market, where a customer (debtor, the one who gives money) is giving cash money to the person selling the products (creditor, the one who credits money).

In electronic transaction the goal of moving funds from one party to another remains, but the amount of underlying processes and participants are increased as bank accounts are involved and a secure payment is required. Figure 1 shows an example of a simple "Customer-to-bank" payment order flow as described by ISO 20022 (ISO 20022 Registration Authority, 2009)

Figure 1: Participants in customer-to-bank payment order

In a customer-to-bank payment order the Debtor is not giving money to Creditor, instead Debtor's agent (financial institution servicing Debtor's account, i.e., Debtor's bank) moves funds to Creditor's agent, where they are accessible by Creditor. An example of actual processes happening between participants is described in Figure 2 (ISO 20022 Registration Authority, 2009).

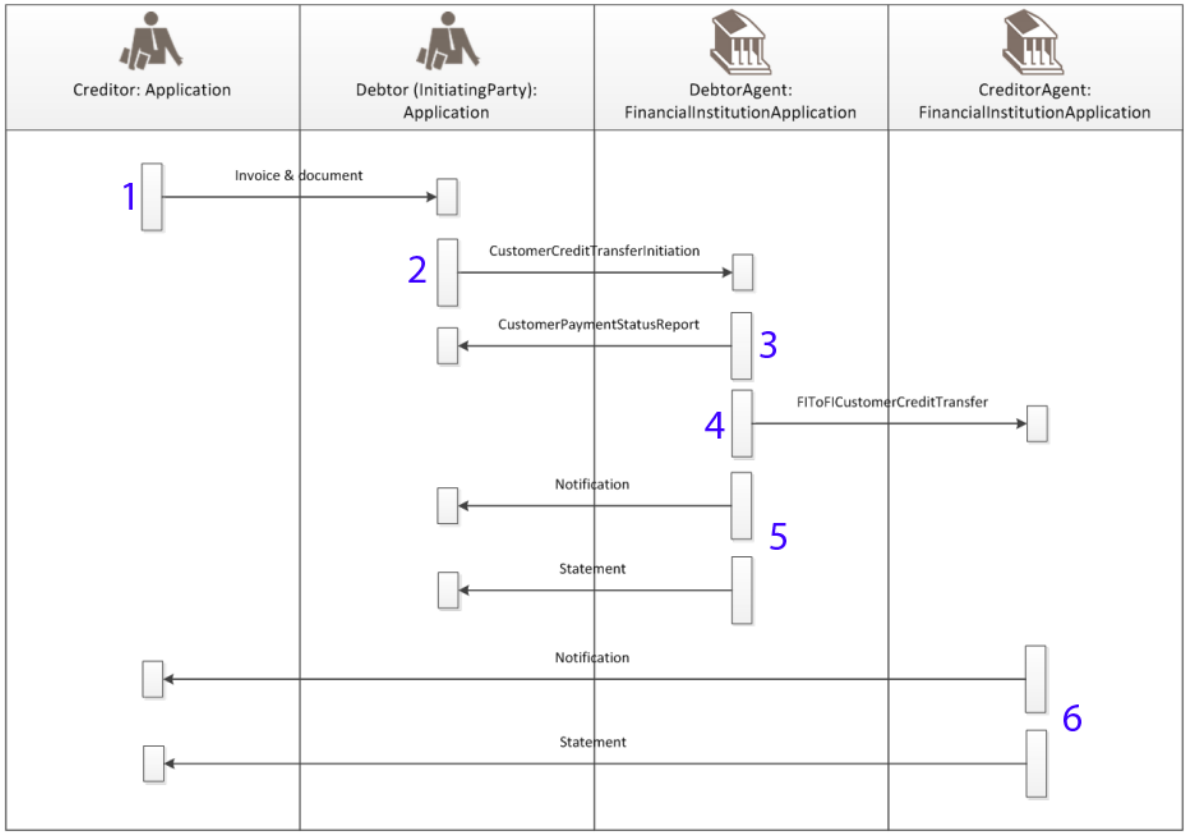

Figure 2: Typical exchanges of messages in a successful credit transfer business transaction

The "conversation" between parties, as depicted by arrows, is done via different electronic payment and status messages and in most cases is fully automated. In the example of Figure 2, before initiating payment Creditor sends an invoice to Debtor (1). To initiate the payment, Debtor has to deliver crucial payment data (2) to Debtor's agent. This message includes the amount of transaction and bank account numbers of participants, among other mandatory data. Debtor Agent then analyses this data and returns a feedback message to Debtor (3), indicating current status of payment. The example in Figure 2 describes a successful payment and thus moving of funds is authorized (4) and notification and statement messages are sent to both Debtor (5) and Creditor (6).

In a failure state the status report message (3) would indicate the reason for the failure and further processing of the payment would cease. The failure state may be due to insufficient funds in Debtor's account, for example.

These messages between parties contain structured information of the payment. The names and formats for each of the messages used in this example are standardised by ISO, International Organisation for Standardization (ISO 20022 Registration Authority, 2009). The ISO standard, however, is not the only standard describing payment messages. The method and format to transfer payment information between parties can vary greatly

from bank to bank and from country to country. Thus, there is a challenge that in order for Debtor and Creditor to exchange payment information, their banks need to understand each other's payment message format.

For the sake of clarity it is worth noting that Figure 2 describes a business transaction, as opposed to a personal transaction made by a bank's private customer where a person may login to one's account via online banking portal. Banks offer enhanced services for corporate customers requiring handling of vast amount of transactions. One of these enhanced services is the ability to send payments directly to bank, without the need to login via web browser. In practice, this direct communication is often done by company's ERP software.

In addition to banks having to know each other's payment format, bank's corporate customers have to know the messaging format and specifications their bank is using. For this reason banks publish "Implementation Guides" (often also referred to as Handbooks, XML guidelines, Message Implementation Guides, MIG's or IG's) targeted for corporate customers. These guides include the necessary information needed for bank's customer to send an automated payment to bank. Figure 3 shows the relation of bank and its corporate customer when it comes to preparation of sending payment messages.

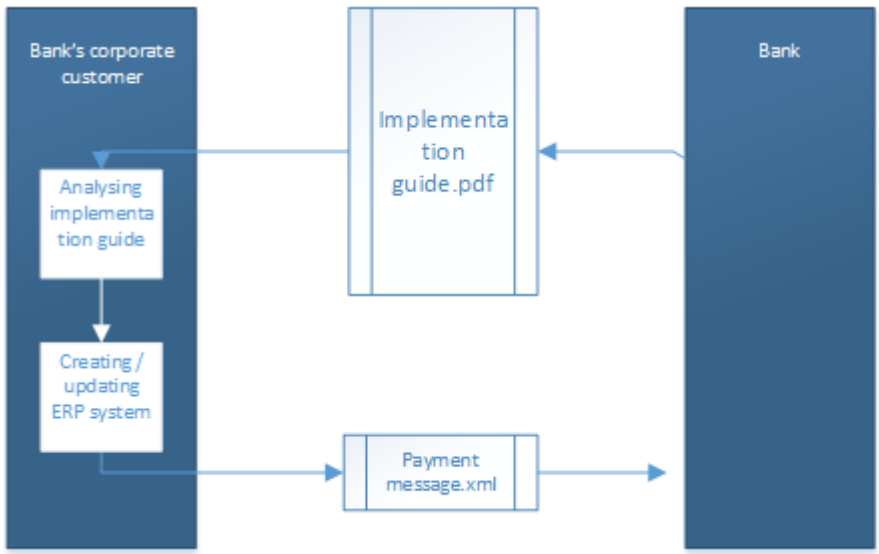

Figure 3: Relation between bank and its corporate customer

XMLdation is a company offering a service for payment message validation by allowing bank's customers to access "validation pipes" made by XMLdation (XMLdation, 2015). Validation pipe contains rules described in bank's Implementation Guide for a specific payment type. A customer of XMLdation can send payment messages into XMLdation validation service instead of bank's system. XMLdation returns an automatically generated report of the payment message, which includes information on whether the payment message would be accepted by bank, and if not, the reason why it would fail. This removes the need for a bank to develop testing services

themselves, so banks can remain in areas where their core expertise lies. For this reason, interestingly, the main customers of XMLdation are not individual companies but banks, who buy XMLdation's service and let their customers use it without a charge. Figure 4 shows the role of XMLdation between banks and their corporate customers.

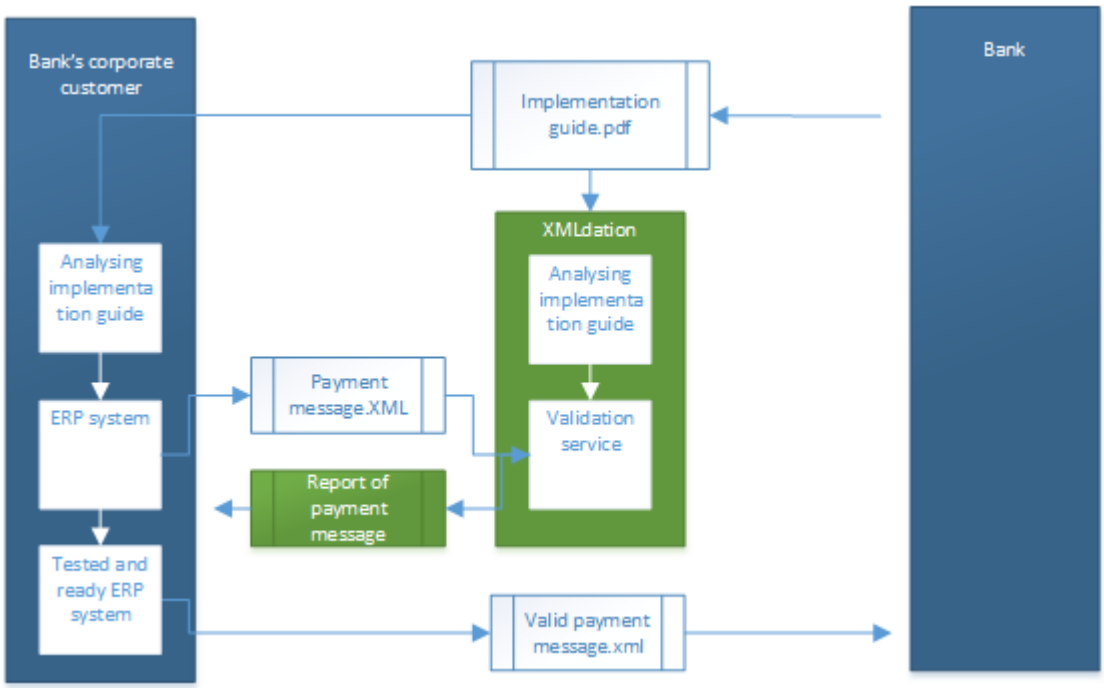

Figure 4: XMLdation, bank and a corporate customer

My role in XMLdation over the past years has been to analyse Implementation Guides of different European banks and incorporating rules they mandate into XMLdation's service. These bank Implementation Guides define XML-element usage and their restrictions, specific for each element within the payment message. We have found out in XMLdation that the information described in bank's Implementation Guides does not necessarily align with the rules banks actually have. For this reason a common practice in XMLdation is to set workshops with banks, in which the goal is to discuss unclear items in Implementation Guide, additional rules and differences in bank's real practice versus the practice published in Implementation Guide. In addition, bank Implementation Guides present information in a non-standardized manner, sometimes in an unclear way, making their analysation unnecessarily difficult and error prone.

In this thesis I will investigate Implementation Guides made for ISO 20022 payment messages by banks and national agencies. More specifically, my goals are to:

1. describe payment messages and the need for payment messaging standards
2. introduce the role of Implementation Guides
3. inspect Implementation Guides from user experience perspective
4. state the importance of well made Implementation Guide and offer insight on what to keep in mind when creating them.

My hypothesis is that improving the way information is presented in Implementation Guides would promote the unification of different payment messaging standards. Even when a specific standard is well thought and does what it is supposed to do, it cannot work as intended if its Handbook is ambiguous and people reading it struggle understanding the content. Unclear Handbook leads to additional work to figure out confusing items. If the Handbook in question is released by a standard agency, the importance of a clear guide is even strengthened, as this is the guide banks will base their own Implementation Guides. If there is an unclear item in this kind of Implementation Guide, it will lead to different practices being used by banks, and thus, by their customers.

Analysation of Implementation Guides strictly from usability perspective is fairly straightforward. However, in order to know what type of data is needed to create a payment message and what needs to be present in Implementation Guide, it is required to have deeper insight into financial messaging and standards used within financial industry. In Chapter 2, Preliminaries, I explain the role of financial messages in money transfers and investigate why XML specifically was chosen by ISO 20022 to relay payment message related data.

In Chapter 3, I describe the important payment standards currently used by the financial industry and investigate why there is such a great number of them, and what type of issues that has caused and may cause in the future.

In Chapter 4, I will list the mandatory and optional data required to create an ISO 20022 payment message. In addition, I analyse current practices used by financial agencies and banks in their Implementation Guides. Finally, based on the research for earlier chapters, in Chapter 5 I am able to suggest improvements and offer insight for people creating Implementation Guides. I offer suggestions for traditional message Implementation Guides and, in addition, argue that there is a need for a new type of interactive Implementation Guide.

I feel that due to possibilities granted by XMLdation and my educational background in Interactive Technology I’m in a unique position to analyse this topic.

2. Preliminaries

The purpose of this chapter is to explain the usage of Payment Messages in financial transactions transfer and the technology behind them.

Payment message is a term used for a file containing information about money transfer. In theory, payment message could be in any format imaginable as long as it contains all information required to make a successful payment and it is understood by the recipient. In practice, of course, it makes sense to have the payment message in a format so that it can be understood by parties receiving it, without the need to convert it or to have a non-automated process analysing it. This is where standard organisation ISO steps in with ISO 20022, a collection of standards developed for financial messaging (ISO 20022, 2015). In order to fully understand what ISO 20022 is doing and how, basic XML technology should be comprehended.

2.1. XML and financial transactions

XML, Extensible Markup Language, was developed in 1996 under W3C, to allow storage of data as well as markup, which describes the layout and logical structure of XML document. Design goals of XML include that it must be usable over Internet, be human readable and XML processing programs must be easy to implement (Bray, et al., 2006). It is interesting to note that W3C's XML version 1.1 document does not specify usage scenarios or any instances where XML should be used according to W3C. The document is only a specification of XML standard. XML was originally developed for large-scale publishing, but XML also plays an important role in exchange of data (W3C, 2015).In other words, W3C specifies the standard, not where and in what situations it is recommended to be used.

Jeff Atwood (2008) argues that programmers and programming tools have standardized on XML, and asks the readers reconsider whether XML is the right tool for the job. It is an extremely interesting question whether XML is the most fitting syntax for financial messaging and if not, what are the alternatives. International Organization for Standardization, ISO, states that when payment messaging standard ISO 20022 was under development in 2004, XML was the "de facto" open technical standard for e-communication, which is why XML was selected to be the format to transfer payment messages (ISO 20022, 2015).

Now more than 10 years later after first publication of ISO 20022, committees such as Common Global Implementation (CGI-MP) and European Payments Council (EPC) promote ISO 20022 based messaging (Swift, 2015) (EPC, 2015). EPC is the developer of Single European Payments Area (SEPA) Credit Transfer and Direct Debit messaging schemes. It can be argued that XML in financial messaging will continue to play a crucial role in the following years. According European Central Bank, on August 2014, 99.38 % of all inter-bank Credit Transfer transactions within SEPA countries were SEPA Credit

Transfers, which means 798.98 million XML files were transferred in that month alone (European Central Bank, 2015). In addition, SEPA Credit Transfers are only one type of message and one part of payment messaging chain, so the actual number of transferred financial related XML files is considerably higher. Figure 5 shows the gradual adaptation of SEPA Credit Transfers, where columns describe months, from March 2008 to August 2014 (European Central Bank, 2015).

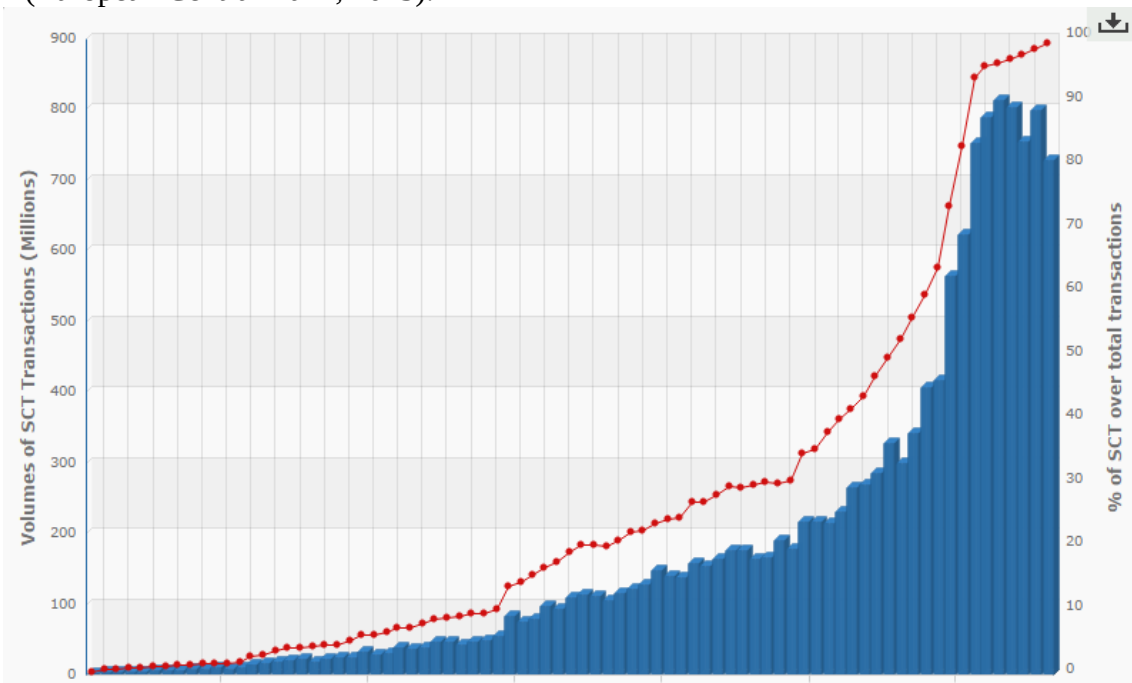

Figure 5: Credit Transfer transactions in SEPA format, volume and percentage

2.2. XML syntax

ISO 20022 Payment Messages are contained in XML files. In order to understand these, XML itself should be understood. In this section I will briefly describe the XML syntax.

XML document forms a tree-like structure, where a root element contains any number of child-elements, sometimes referred to as leafs. These child-elements may contain data or act as a container for other elements. Figure 6 shows an example of XML structure with some of the elements used in a Payment Message, more specifically in pain.001.001.03 Credit Transfer file.

```
<CdtTrfTxInf>
  <Amt>
    <InstdAmt Ccy="EUR">20</InstdAmt>
  </Amt>
  <Cdtr>
    <Nm>Antero Mäenpää</Nm>
  </Cdtr>
</CdtTrfTxInf>
```

Figure 6: Example of XML tree

First line in the example of Figure 6 contains text <CdtTrfTxInf>. This is a start tag of the root element “CdtTrfTxInf” (acronym for Credit Transfer Transaction Initiation). The root element contains children Amt and Cdtr, both of which contain another element holding actual data. Element InstdAmt, in addition to containing data “20”, also contains attribute Ccy which is set to “EUR”. Even without knowing any further context of the Payment Message, based on the example in Figure 6, it can be deduced that “Antero Mäenpää” is supposed to credit 20 Euros. This human readability is one of the strengths of XML format.

Figure 7 shows the terminology of XML element.

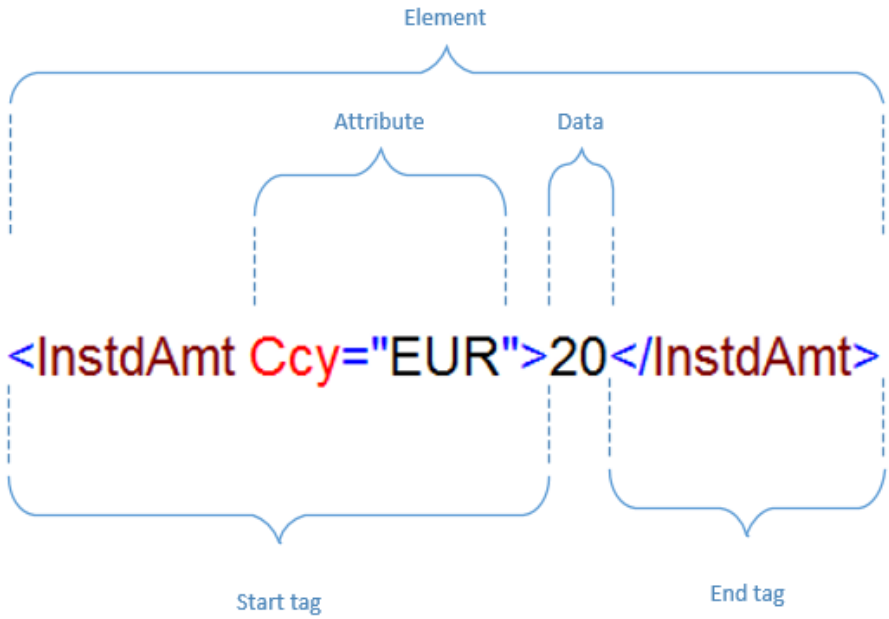

Figure 7: Terminology of XML element

XML element consists of start tag, end tag, optional attributes and data or child elements. Figure 7 contains data (“20”), but XML syntax would support data to be replaced with other elements. Attributes within element are optional and their amount is not limited. Data in XML element and within attributes can be textural, with certain limitations for allowed characters. Using characters relevant for XML syntax itself is not allowed. Therefore, these characters have to be replaced with other representations of them, called character entity references. This is done so that the XML parser, software which reads and interprets XML data is able to do its task correctly. For example when data includes a smaller than character (“<”), that character has to be replaced with its entity reference, <.

Figure 6 contained an XML document with 8 lines and indentation. However, indentation and lines are not important from technical point of view and exist only for making the XML document human readable. Lines and indentation can be erased and the document would still be well formed, meaning that it has a correct syntax.

2.3. XML schema

Regardless of the format, there often are different ways to present the same information. This applies in XML as well, exactly the same data can be represented by a vast number of different ways. Element names, XML structure and usage of elements and attributes all make it possible for the creator of XML file end up with a unique combination of data representation. For example, representing transaction data including the name, value and currency can be done as shown in Figure 8.

```
<Transaction>
  <Receiver Name="Antero Mäenpää">
    <Amount>20</Amount>
    <Currency>EUR</Currency>
  </Receiver>
</Transaction>
```

Figure 8 Representing transaction data in XML

Alternatively, all elements could be named differently and by using acronyms. Currency could be chosen to be presented as an attribute of the amount, instead of its own element. The data itself would remain to be the same, but the way it is represented would be different. In some cases, and especially in Payment Messages, it is crucial that the data is always represented in a fixed way, so that it can be understood similarly by both the sender and the receiver of the message. So how can this be achieved?

The answer is broad, and the first part of it is related to XML schema, which I will explain in this chapter. Payment standards and Implementation Guides are involved as well.

Purpose of XML schema is to define the structure, content and semantics within XML document. XML schema does not tell what exact information to give in an XML document, but instead, it defines how to give the data and what format it may be in. An XML file made according to the rules defined in XML schema can include a reference to the said schema. This allows the reader of the XML file to know that the file is supposed to meet a pre-defined structure and content. Since schema defines a structure, it is possible to visualise this with an external program. Figure 9 shows an example schema as visualized by XMLspy.

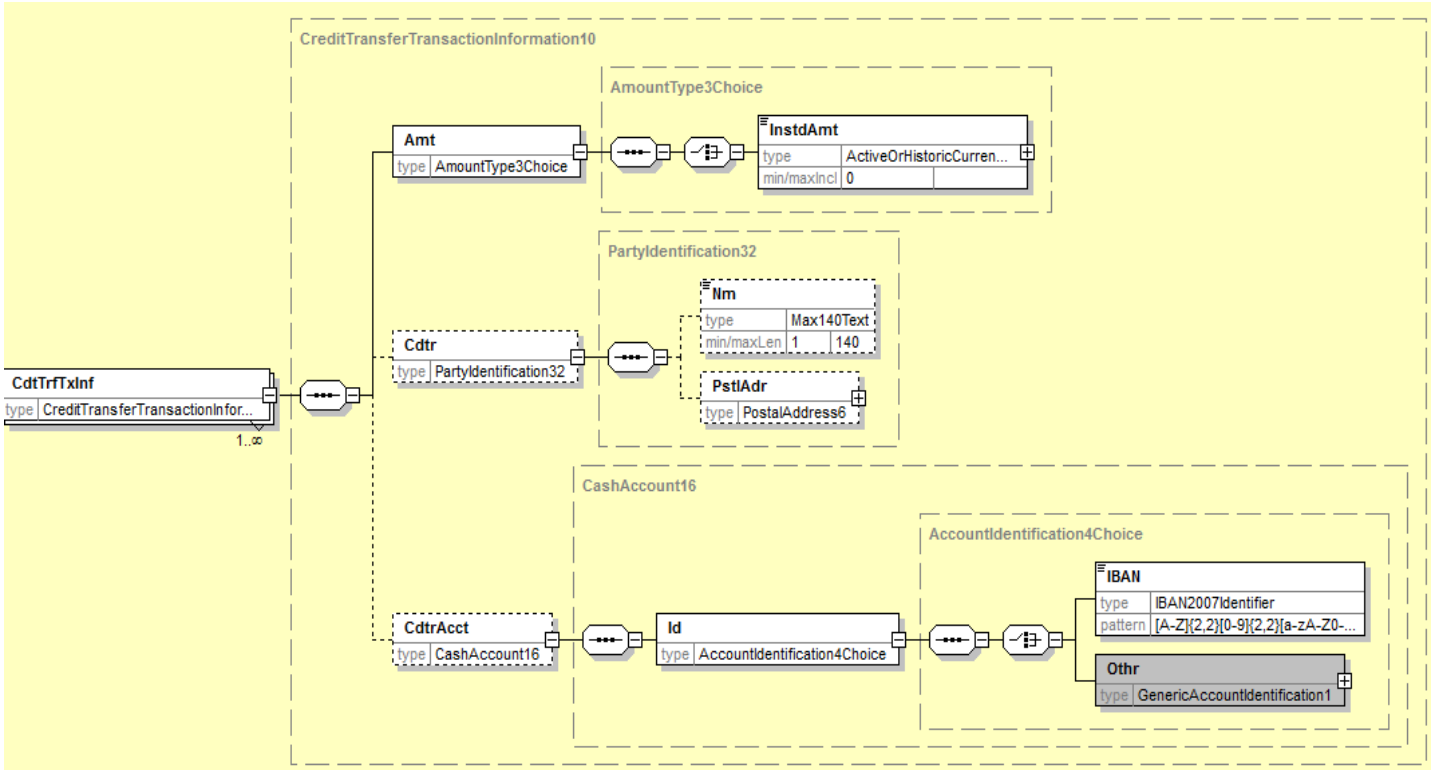

Figure 9: Visualisation of a schema as shown in XMLspy

Content in Figure 9 is based on payment message pain.001.001.03, defined by ISO 20022 and opened in XMLspy, a program specialized for editing and viewing XML and schema files (XMLspy, 2015). However, for the purpose of this example large amount of elements are omitted to only include three child elements of “CdtTrfTxInf”: “Amt”, “Cdtr” and “CdtrAcct”. Elements in schema are all either complexTypes or simpleTypes, based on whether they may contain textual data or other XML-elements. In Figure 9, ComplexTypes are “Amt”, “InstdAmt”, “Cdtr”, “PstlAdr”, “CdtrAcct”, “Id” and “Othr”, therefore these elements contain child-elements. SimpleTypes are “Nm” and “IBAN” and they hold textual data. Based on this visualisation alone it is possible to know how to present the transaction related information for amount, creditor and creditor account. Since schema itself is an XML document, it can be viewed in a text form as well. Figure 10 shows complexType “CdtTrfTxInf” in XML format.

```
<xs:complexType name="CreditTransferTransactionInformation10">
  <xs:sequence>
    <xs:element name="Amt" type="AmountType3Choice"/>
    <xs:element name="Cdtr" type="PartyIdentification32" minOccurs="0"/>
    <xs:element name="CdtrAcct" type="CashAccount16" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Figure 10 Example of a schema as XML format

Added benefit of a schema is that it makes technical data validation possible. This means that when a schema restricts an element value to be in a specific format and an XML file is assigned to this schema, an external program can be used to automatically validate the XML element. In a scenario where XML file shown in Figure 12 is validated using schema in Figure 11, an error in Figure 13 is returned in XMLspy.

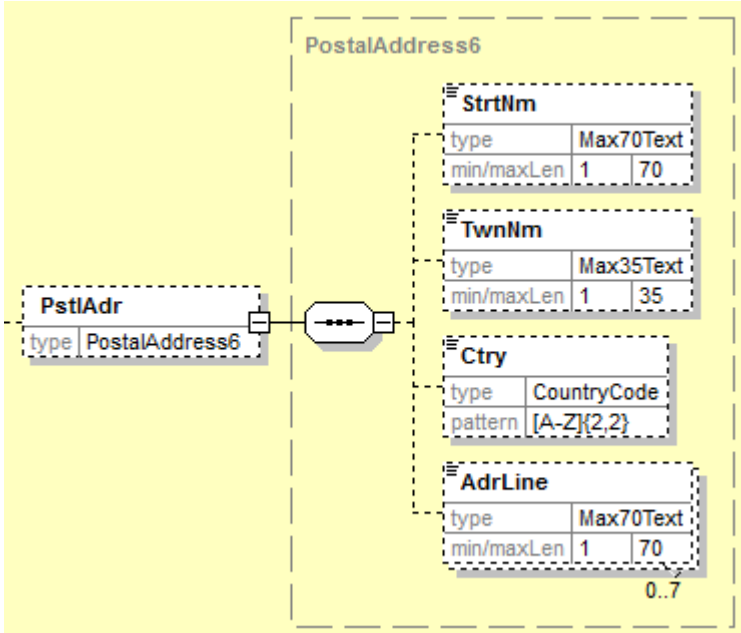

Figure 11: Schema depicting PstlAdr elements. Note the pattern set for “Ctry”

```
<PstlAdr>
  <StrtNm>a</StrtNm>
  <TwnNm>a</TwnNm>
  <Ctry>AAA</Ctry>
  <AdrLine>a</AdrLine>
</PstlAdr>
```

Figure 12: XML-file containing PstlAdr elements

```
Value 'AAA' is not allowed for element <Ctry> .
  Reason: it does not satisfy any of the defined patterns (see below)
    '[A-Z]{2,2}'
  Error location: Document / CstmrCdtTrfInitn / GrpHdr / FwdgAgt / FinInstnId / PstlAdr / Ctry
  Details
```

Figure 13: Error given by XMLspy when XML-file in Figure 12 is validated against the schema in Figure 11

In the schema of Figure 11, value of “Ctry” is limited with pattern “[A-Z]{2,2}”, meaning that exactly two capital letters from A to Z are allowed in the element. This is why an error shown in Figure 13 is returned when three letter value is given. However, although helpful, technical validation often is not enough when it comes to validation of payment

messages. Country Code “AA” would pass the limitation set by schema in Figure 11, but “AA” is not a valid ISO 3166-2 country code to be used in ISO 20022 payment messages (ISO 20022 Registration Authority, 2009). Technically it is possible for a schema to include a group of possible values for an element, so why is this not done in the case of country codes? This is because country codes may change over time, which would mean that every time there is a change in the global list of country codes, a new payment message schema would have to be published with the updated list of country codes. This is why ISO 20022 only limits the pattern of the element “Ctry” and supplements the element description with a rule that it has to contain a country from the most up to date list of countries. This standard, ISO 3166, is published and maintained by ISO as well (ISO 3166 Maintenance Agency, 2015).

It should also be noted that XML schema is not the only way to define structure and semantics of an XML file. A DTD file may be used for this purpose as well. However, ISO 20022 defines Payment Messages only with schemas.

3. Payment Standards

The reason why payment standards exist is fairly simple: to allow parties to send information in a way which is understood by the receiving party. In this chapter I will elaborate the above definition and explain the purpose of certain standards currently relevant within the industry and what the relationship between different standards is.

3.1. ISO 20022

ISO is a standardization organization and world's largest developer of voluntary International Standards (International Organization for Standardization, 2015). The acronym ISO comes from the words International Organization for Standardization, but as *isos* in Greek means equal, it is fitting that the letter order in acronym is ISO rather than IOS. ISO defines a myriad of standards, from country codes to food safety management. ISO 20022 is the name selected for their financial messaging standards. But a grand name and history in standardization is not enough to become relevant in message standardization field, so is ISO 20022 important in the financial industry and, if yes, what is the reason?

Within Europe, the SEPA initiative mandates that all Credit Transfers have to be SEPA credit transfers by the end of February 2014 (with some expectations) (European Commission, 2014), and as SEPA uses ISO 20022 messaging schemes as base, this consequently means that ISO 20022 is very relevant. Swift (2010) states that the reason for ISO 20022's specialty lies in the fact that ISO 20022 is distinguished into three separate and distinguishable layers,

- (1) key business processes and (2) concepts
- (3) logical messages
- (4) syntax.

The terms themselves are vague but by opening them further they start to make sense. The first layer is a business process (1). This means that in order for there to be a financial message, first it has to be known what the message is used for, what prior information is required to implement the message and what parties are involved when the message is transmitted. Concepts (2) means that the key actions and parties are introduced with common terms and definitions. For example, Credit Transfers include concepts such as Creditor (receiver of money) and Debtor (party that pays), and the limitation that they may either be persons or organizations. Both of these are associated with their respective agents (banks) and all of these parties contain a common set of information, such as name and postal address. These terms are unified with other business processes, therefore all the terms in Credit Transfers mean the same as in Direct Debits. An alternative way to use concepts would, for example, be to always use the term Payment Originator, instead of Debtor and Creditor. Payment Originator as a term tied to the context, therefore in Credit Transfers it is the Debtor and in Direct Debits it is the Creditor. As there are a vast

amount of different business processes, getting familiar with semantics and different contextual terms would require a considerable investment of time.

Logical message (3) is the description of all the information available and required in a business process. It contains message components and their hierarchical structure, but not the syntax. In other words, logical message may state that payment has to include Debtor, Creditor, payment amount and Creditor's Agent BIC number.

The last key layer defined by ISO 20022 is the syntax (4), which is the physical representation of the logical message; in practice this is the XML message. However, because the logical message is a separate layer from physical message, based on logical message it would be possible to create the physical message in whichever format desired.

As of June 2015, ISO 20022 has 364 different business processes (and thus financial messages) catalogued, and, in addition, about the same amount archived as old versions. At minimum, each of them contain a schema and a message definition document (ISO 20022, 2015).

3.2. SEPA

SEPA is an acronym Single Euro Payments Area. Unifying previously fragmented national markets within Europe was seen as a potential to reduce the cost of moving capital within the area, thus European Union initiated the SEPA project (EPC, 2015). Expectations of SEPA were grand: to resolve issues in cross-border payments and to unify domestic payments into a single format. Domestic transactions were to be equal with foreign transactions (Bott, 2009). European Commission estimated potential savings and new gains related to SEPA to be at around €122 billion per year, mostly due to e-invoicing. €10 Billion from the savings and gains were estimated to be due to standardisation (Bott, 2009). Among other requirements, SEPA mandates banks to accept XML messages aligned to ISO 20022's schemas with enriched rules defined by European Payments Council (EPC) [2015]. This eliminates the need for message conversion and allows banks within SEPA to be aligned with each other, therefore banks enabling banks to know they are able to transfer transactions with other SEPA banks. However, European countries had adopted business models and infrastructure specific for each country. Therefore, SEPA did not only mandate a usage of a different messaging standard, but it also meant that process designs and business models had to be rearranged (Bott, 2009).

European Commission mandates banks to be able to send and receive SEPA compliant payment messages by the year 2014 (European Commission, 2014). After the first quarter of 2015 the conversion into SEPA is complete in most countries involved in the project (European Central Bank, 2015). Transition to SEPA had a major impact on bank's corporate customers as well, as when banks changed the accepted standard, companies had to adjust. Companies had to start sending payments which aligned to SEPA rules and XML format, while taking into consideration the individual rules their

bank may have. This was considered to be a major project (Tossavainen, 2011). As these individual rules have to be stated and informed to bank's corporate customers, even individual banks within SEPA area create Implementation Guides. These Handbooks inform the reader how data is to be represented in a payment message in order for it to be considered valid.

In practice, when SEPA mandates Payment Messages to be in a specific format it means that the software generating payment messages has to be built in a way so that it outputs the content as instructed in the Implementation Guide.

3.3. CGI-MP

Common Global Implementation – Market Practice is an initiative by Swift to provide a forum for both financial and non-financial institutions to enhance and progress Customer-to-bank ISO 20022 payment messages. The goal of this is to promote the usage of ISO 20022 XML messages world-wide (Swift, 2015). In practice this means that CGI-MP manages Implementation Guidelines focusing on global, multi-bank and multi-payment type implementation. The information for these guidelines is gathered and updated accordingly based on information from corporate partnerships, bi-weekly meetings and national financial communities (Swift, 2014).

The goal of CGI-MP is admirable and it is very understandable that an institution like CGI-MP exists. ISO 20022's message definition alone as a guide for implementing a payment messaging system is not enough to verify that two parties are able to exchange payment transactions with each other. ISO 20022 payment messaging scheme allows this, but does not instruct a specific usage. The attempt of CGI-MP is to allow exactly this, to enable successful transactions between parties who follow rules set by CGI-MP. For Credit Transfer, CGI-MP defines a common set of rules for non-urgent transactions, urgent transactions and Cheque payments. An example of a rule set by CGI-MP is to mandate a presence of Creditor and Creditor name in a payment message, whereas ISO 20022 only states what these elements are and where they may be used. This allows banks following CGI-MP's market practice to know that a payment message they receive from a customer is always going to include certain set of information. However, it should be noted that the rules CGI-MP sets are on a very general level, and CGI-MP itself states that banks may have additional, bilaterally defined rules.

In a way, comparison between CGI-MP and SEPA shows certain similarities. They are both meant for multi-country markets and attempt to enable successful transactions between the parties who follow their standards. However, GI-MP exceeds the borders of Europe and does not define rules as strictly as SEPA payment schemes.

3.4. National Agencies

In this section I explain agencies which publish country specific Implementation Guides, from Payment Messaging point of view. Defining and explaining these agencies is difficult. Their operation is dependent on the country and they do not necessarily share similar purposes. European Central Bank lists these organizations/agencies on its web-page, but does not introduce a common name or definition for them (European Central Bank, 2015). A shared trait for these agencies is to represent a common interest of banks and financial companies in their country. These agencies are not banks, but banks may co-operate with them (Dutch Banking Assiociation, 2015) (Finanssialan Keskusliitto, 2015). In addition, these agencies play a key role developing and maintaining cashless payment models and practices, especially before the introduction of SEPA (European Central Bank, 2015).

For the sake of simplicity, in this thesis I will refer to these financial bodies, organizations or associations as National Agencies. By National Agency I mean country specific institutions which have made publications of ISO 20022 payments. In these publications they may mandate or recommend a specific usage of ISO 20022 payment message for their country. See Table 1 for examples of National Agencies.

Name	**Abbreviation**	**Country**
Federation of Finnish Financial Services	FK	Finland
Nederlandse Vereniging van Banken	NVB	Netherlands
Eesti Pangaliit	EBA	Estonia
Consorzio CBI	CBI	Italy
Die Deutsche Kreditwirtschaft	DK	Germany
Irish Payment Services Organisation	IPSO	Ireland

Table 1 Examples of National Agencies

One purpose of National Agencies is to represent a common interest of banks. From payment messaging perspective this means that certain country specific rules are handled by National Agencies instead of individual banks. A country may have certain regulations and practices which have to be defined, and as these rules are country specific, it is natural that they are not defined by multi-national standard agencies, like ISO 20022 or SEPA. Banks could individually list these country specific common usages in their own Implementation Guides, but as the regulations and usages are the same for all banks within the country, it is logical that countries generally have an outside agency in charge of maintaining them. This does not mean that banks within a country share exactly the same set of rules, but instead, that the shared rules banks have are listed in an external source released by National Agency.

For example, country may have a law that all transactions exceeding a certain amount have to be logged for statistical purposes. In order to log the transaction, a specific code has to be used from set of pre-defined codes. This raises the question who maintains this code list and who decides the location where this code should be inserted in an XML payment message. In theory, banks could do this by themselves, but without co-operation this would result in a situation where banks use different codes and different locations where this information is presented. In addition to being time consuming for banks, it would lead to corporate clients of banks using different ways of giving the same information. Instead, a separate National Agency is in charge of maintaining these rules, and banks within a country may refer to the usage guide of National Agency to meet the regulations set by law. This is beneficial for banks as well as their customers. When banks have common rules, it is easy for customers of banks to change the bank where they send their payment messages.

3.5. Relation of standard agencies and current status

When there are many standard agencies, just for ISO 20022 payments, it is interesting to see how they are related to each other and what implications these relations may have. ISO 20022 can be defined as the main, or root agency which other agencies use as the base for their implementation. These agencies create subsets of ISO 20022 payments to meet a specific need their users have or to fulfill laws set by countries or areas. Figure 14 shows the relation of ISO 20022, CGI-MP, SEPA and the hypothetical usage National Agencies and banks may have with them.

Figure 14: Standard agencies and their relation with banks

In Figure 14, ISO 20022 is on top to define the structure of a payment standard. CGI-MP and SEPA specify how the Payment Message elements defined by ISO should be used according to them. Banks 1, 2 and National Agency 2 create their Implementation Guides based on CGI-MP's subset. Banks 3 and 4 make their country specific Implementation Guides based on the guide by National Agency 2. Would the customers of these banks be able to send Payment Messages to all banks using CGI-MP as base? It is highly likely that they would not. Reason is that, out of the 448 elements defined by CGI-MP, 162 are stated to be bilaterally defined (Swift, 2014). This means that it is up to banks to decide how to treat those elements. Banks may require some of them or may state that specific usage rules have to be applied on certain elements. However, it is worth noting that the purpose of CGI-MP is not to allow all banks using it to be in complete harmony with each other (Swift, 2014).

For SEPA the end-goal is different. The purpose of SEPA is to harmonize payment messaging standards within its area so that banks have the same messaging standard. So, would SEPA banks be able to accept and understand Payment Messages with one another? The answer should simply be yes, and based on EPC's definition that is the only possible outcome as well. However, the reality in certain cases may be different. Some National Agencies have defined their own schemas based on ISO 20022. These modified schemas are called variants, and their usage is accepted and anticipated by ISO 20022. ISO 20022 even includes a dedicated a part in its Payment Message naming convention with a variant number. However, this means that the namespace within those schemas is not the same as it would be in a vanilla ISO 20022 schema. Figure 15 lists the namespaces of pain.001.001.03 Payment Messages (or equivalent) of ISO 20022 (1), Stuzza (2) and CBI (3) schemas.

```
1:
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
2:
<Document xmlns="ISO:pain.001.001.03:APC:STUZZA:payments:004" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3:
<CBIPaymentRequest xmlns="urn:CBI:xsd:CBIPaymentRequest.00.04.00" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

Figure 15: XML Credit Transfer Namespaces of ISO 20022, Stuzza and CBI

All these National Agencies follow the rules set by EPC for XML elements, but their namespaces differ. In addition, CBI uses a different name for the root element. Unless these differences are handled by the banks accepting payments, processing files outside of individual National Agency will lead to a processing error.

4. Implementation Guides, purpose and data

In this chapter I will explain the description and purpose of Implementation Guides. I will investigate what data is required in order to make an XML Payment Message and how the requirements for Payments Messages are communicated in Implementation Guides. For this reason I have studied 13 Implementation Guides released by Standard and National agencies as well as from banks.

In addition, I will investigate what are the impacts for banks and for their customers when information in Handbook is unclear.

4.1. Description

In Payment Messages it is crucial that data is always represented in a pre-agreed way, so that the data is interpreted correctly by the sending and receiving parties. First part in achieving this is to know the format and the usage of format in question. When the format is selected to be ISO 20022's XML, its description is defined in XML schemas released by ISO 20022, along with exhaustive description of elements in the XML schema in a Message Definition document (ISO 20022 Registration Authority, 2009). Second part in achieving the common understanding of Payment Message usage is to further define what parts of the Payment Message defined by ISO 20022 is to be used and what data to include. For this reason, Implementation Guides are released.

Implementation Guides can roughly be divided into three main categories based on the party creating them. They can be created by

1) Standard Agencies
2) National Agencies
3) Banks.

Standard Agencies either write their interpretation of ISO 20022 standard, in order for it to meet a specific need. In these guides they introduce how each possible element in their standard is to be used. This makes the Handbook very thorough, as the Implementation Guides have to cover all possible scenarios the message standard supports. Implementation Guides from National Agencies have a similar goal as guides from Standard Agencies, but they distribute their release only at a national level, as opposite to standard agencies supporting multiple areas and ignoring country specific information. Both Standard and National agencies are generally targeted for banks. Examples of standard agencies are CGI-MP and EPC which apply certain limitations and rules for the usage of ISO's standard. Examples of National Agencies are NVB (in Netherlands) and FK (in Finland).

It is up to banks to individually decide what the relevant items in standard are specifically for them. This information is distributed in a bank Implementation Guide, meant to be read by bank's corporate customers. In this type of Handbook the reader does not need exhaustive information on all possible elements allowed in Payment Messaging

standard. ISO 20022 refers to Implementation Guides from banks as Message Implementation Guides and acknowledges that these are not official ISO 20022 documents, instead they are created to describe specific way of using ISO 20022 messages in a particular context (ISO 20022 Registration Authority, 2009). However, it should also be noted that in practice, even though guides from Standard and National agencies are mainly written for banks, there are cases where banks' corporate customers are also expected to read them.

4.2. Data in Implementation Guides

In this section I describe what knowledge is required to implement a system producing ISO 20022 Payment Messages, and thus, what data is mandatory and recommended to be present in Implementation Guides. I also give examples of what the information may look like in Implementation Guide.

First, it should be mentioned what is the purpose of the Implementation Guide. What type of payments will the Handbook guide the reader to produce?

Implementation Guide document related information is not mandatory, but including it will help to keep track of changes within Implementation Guide. Therefore possible future updates and iterations of the Handbook are easier to understand. Document related information is

- document version and release date
- changelog
- abbreviations.

Document version may be a sequence based numbering scheme keeping track of unique states of the document. Changelog is a list stating the changes of current version versus the data in previous version. Reason to include this is to give the readers a choice to read only changelog when an update of the Handbook is released, without having to go through entire document. Examples of document version history and changelog in NVB:s Implementation Guide are given in Figure 16 and Figure 17 (NVB, 2012).

Version	Date
2.0	October 2010
2.1	November 2010
2.2	February 2011
5.0	January 2012
5.0.1	February 2012
6.0	March 2012

Figure 16 NVB Direct Debit version history

The following additions and modifications have been made in version 6.0 compared to version 5.0.1:

- Addition of 'COR1' to 2.12 Code as part of choice 2.11 LocalInstrument and modification of Usage Rule concerning "The mixing of different local instrument values is not allowed in same message".
- Addition of AT-12 : BIC of Creditor Bank to ANNEX C "Rulebook attribute equivalent"
- Addition of reasoncode AC13: Debtor Account is a consumer account to ANNEX E "ISO reasoncodes used"

Figure 17: NVB Direct Debit version 6.0 changelog

The following XML message related information is needed to meet guidelines set by XML standard and ISO 20022(SOURCES):

- XML encoding
- XML version
- schema type
- schema version
- schema namespace.

Character encoding is a unique binary code for each character. In order for XML message to be able to be interpreted in correct way, file must be read using the same encoding as was used to save it. ISO 20022 restricts encoding to UTF-8 and version to 1.0 (ISOENCODINGSOURCE), to avoid end-readers finding this out themselves. This should be stated in the Implementation Guide as well. In addition, to avoid possible errors W3 recommends encoding to be specified within XML file (W3ENCODING SOURCE). Encoding and XML version are attributes within the optional first line of XML file, prolog. See Figure 18.

```
<?xml version="1.0" encoding="UTF-8"?>
```

Figure 18: Example of prolog

In ISO 20022's naming scheme, schema type and version are combined. This information is needed to know what schema and message definition document to use as a base for Payment Message. Figure 19 shows the information included in ISO 20022's Payment Message naming scheme, with Credit Transfer version 3 as an example (pain.001.001.03).

Figure 19: Information within ISO 20022's payment message name

In ISO 20022's naming scheme, message type is described with four letters. In the example of Figure 19, the message type is "pain", abbreviation of payments initiation. Three-digit number following it is a message sub-type. In the case of payments initiation, the number may, for example, be" 001" to depict Credit Transfer or "008" to depict Direct Debit. The number after sub-type represents a variant number. Variant number is reserved to Standard Agencies or banks which use a modified version of ISO's payment message, to clearly indicate that a message is based on ISO's implementation but is still a different message (ISO 20022, 2015). In ISO 20022's official messages the number is always "001". The last number indicates version number of the message, first one being "01" and increased by one whenever a new version is released. Together this information forms the name for the message and for XML schema. In ISO 20022 XML file, namespace is mandatory information to be included as an "xmlns" attribute of the root element and schema name is included in the namespace. See Figure 20 for an example of two first lines in XML Payment Message.

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
```

Figure 20: Two first lines in a Payment Message

The following data is required in Implementation Guides to describe usage of XML elements and what data they may contain:

- charset supported by receiving party
- required usage rule for each relevant XML element

- treatment of elements not described
- optional example XML file(s).

Encoding of the file may be one which allows myriad of characters, like UTF-8, but receiving party's internal system may not be able to support them all. If this is the case, it should be stated what characters actually are supported and inserted to receiving party's backend system. Usage rule for each XML element forms the core content within Implementation Guide. Each ISO 20022 XML element may have the following information attached to it (Example of these in an Implementation Guide can be seen in Figure 21 (DNB Banka, 2014).):

- ISO 20022 index number
- occurrence multitude
- element name
- full ISO 20022 name
- element type
- depth in XML tree
- description
- format
- usage rules.

ISO Index No.	Or	Message Item	Tag Name	Structural Sequence	ISO Type	Mult.	Comments on the information to be entered herein
		Customer Credit Transfer Initiation	<CstmrCdtTrfInitn>	-			
1.0		**-GroupHeader**	<GrpHdr>	+		[1..1]	
1.1		--MessageIdentification	<MsgId>	++	Text	[1..1]	Accepted but not used at the moment.
1.2		--CreationDateTime	<CreDtTm>	++	DateTime	[1..1]	Accepted but not used at the moment.

Figure 21: Example of presenting element data in DNB's pain.001.001.03 Implementation Guide

ISO 20022 message definition may include several hundred different elements and their descriptions. For example, generating a sample file with all elements for Credit Transfer pain.001.001.03 message will produce a file with 949 pretty printed lines. This means that the way element data is presented in Implementation Guide will have a huge effect on its clarity and length.

It is important to not only think what data is required in an Implementation Guide, but also what data is not. Omitting certain items from the Handbook will help with keeping the document as short as possible. In addition, overpopulating the Handbook with non-beneficial data will lead to a confusing presentation. When a bank or standard organization has only a small number of usage rules, it may be beneficial to leave out

elements which have no specific usage rules. Alternatively, the Implementation Guide may only present elements which are actually taken into bank's backend system, and leave out elements which are ignored, even when ISO 20022 schema allows them. In this case it is very important to state how these non-presented elements are treated, whether they are allowed or not.

Sample XML message depicts one outcome of the final XML payment message, made according to the rules in the Handbook. Including this sample in the Handbook may be beneficial, as it offers a tangible data of the "real" outcome. Implementation Guide may depict multiple payment types, for example SEPA, foreign and domestic payments. Including a sample file for all payment types present in the Handbook is also good way to show the actual difference between them.

To conclude the information required to create a bank specific Payment Message, Table 2 shows what mandatory and recommended information can be present in an Implementation Guide.

Related to	Item	Presence
Implementation Guide document	Purpose	Mandatory
	Version & release date	*Recommended*
	Changelog	*Recommended*
	Abbreviations	*Recommended*
Technical XML information	XML encoding	*Recommended*
	XML version	*Recommended*
	Schema type	Mandatory
	Schema version	Mandatory
Payment data	Schema namespace	Mandatory
	Supported charset	*Heavily recommended*
	Treatment of elements not described	*Heavily recommended*
	Sample file	*Recommended*
Data for each relevant element	ISO 20022 index nb.	*Recommended*
	occurrence multitude	*Recommended*
	Element name	Mandatory
	Full ISO 20022 name	*Can be given*
	Depth in XML tree	*Can be given*
	Type	*Can be given*
	Description	*Can be given*
	Format	*Can be given*
	Usage rules	Mandatory

Table 2 Mandatory and recommended data in Implementation Guide

Table 2 shows most of the elements to be recommended and only six mandatory items. The reason is that technically it is possible to create bank specific Payment Messages with only knowing information related to those five items. Of course, creating Payment Messages do require more information to be known, for example, the description how each element is used. Implementation Guide in this case would rely that the reader finds other necessary information elsewhere. I would estimate that omitting information regarding charset and treatment for non-described elements will most likely lead to a situation where bank has to provide answers to these separately, for at least some of its corporate customers. When the supported charset is not known, error situations in future may occur if bank does not support all elements in an encoding. When creator of a Payment Message does not exactly know what charset is supported by the bank, some Payment Messages may contain unaccepted characters and lead to a processing error. Same applies for elements which are omitted from bank's Implementation Guide. Bank's customer may want to use some of them anyway, and it would not be known if they are actually taken into consideration.

Items which I have marked to be recommended I estimate to help the reader greatly when reading the Handbook and implementing system producing Payment Messages. I estimate that including them in a good manner will improve the clarity of the Implementation Guide, thus reducing the items which may be unclear for bank's corporate customer. Items marked as "can be given" are more complicated. In some cases it is beneficial to include them and in some cases it may not, depending on the format of the guide, number of elements and estimated audience. I will describe these more thoroughly in Chapter 5.

4.3. Impact of unclear items

Impact on ambiguous information in Implementation Guides can be large, as it will lead to usage of invalid and undesirable practices. Severity of an unclear item depends on the type of Implementation Guide it was published on. Unclear item in a Handbook by a bank to its corporate customers means that the corporate customers will potentially send invalid payment data to bank. As the payment data cannot be processed, this will lead to extra work by a bank and its corporate customer to fix the issue, so that the payment data produced by the customer is valid and processed correctly by the bank. However, the effect of an unclear item in a Handbook released by a standard agency is more drastic. It means that there is a risk for a bank to implement its backend system against the standard they are attempting to use as the base. Potential invalid practice is then being guided further by the bank in its own Handbook, which leads to the invalid practice used by clients of that bank. Figure 22 shows a simple example of Banks 1 and 2 using CGI-MP as the base in their standard, which the banks specify further in their own Implementation Guides, Bank 1 to clients A and B, bank 2 to clients C and D.

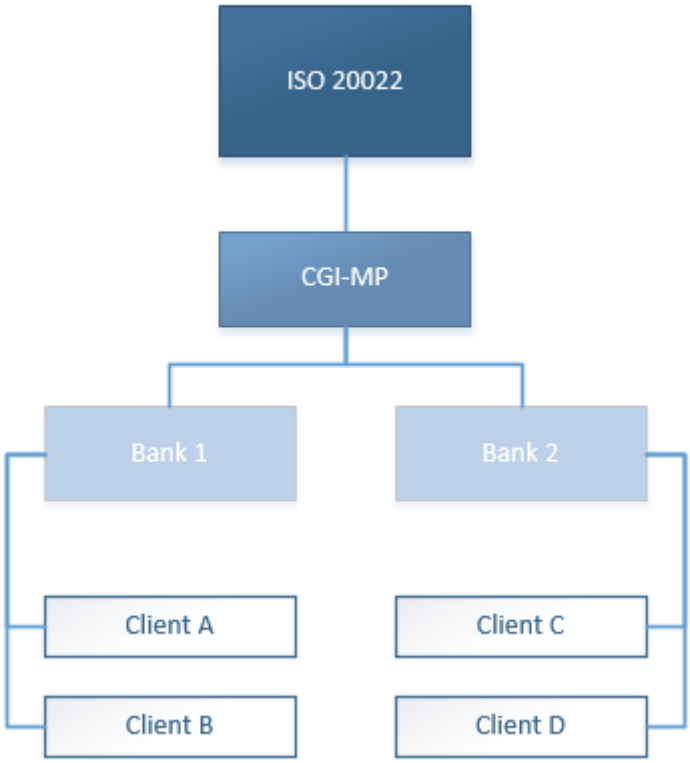

Figure 22: Example of scenario where two banks use CGI-MP as payments standard

All clients use ISO 20022's payment messaging schemes, with further rules set by CGI-MP. An unclear item in CGI-MP's Handbook will lead to clients of Banks 1 and 2 to implement payment messaging system which potentially produces different data based on the unclear item. Therefore banks would not be able to accept payments from each other's clients.

Scenario of what an unclear item in a Handbook can be in practice is from CGI-MP's pain.001.001.03 Implementation Guide version FEB 2014. It has an unclear PostalAddress usage specified for several elements, including Creditor and Debtor. See Figure 23 (Swift, 2014).

PostalAddress	<PstlAdr>	[0..1]	C	RECOMMENDATION IN ORDER OF PREFERENCE: 1. Use only structured address. 2. When using combination of both structured address and Address Line, must use structured tags for post code (if applicable), country subdivision (if applicable), town name and country and only 2 Address Lines (to include street address). 3. Use only Address Line (up to 7 lines; instrument by instrument limitations may apply) NOTE: PO Box should only appear in Address Line.
Department	<Dept>	[0..1]	BD	See recommendation above.
SubDepartment	<SubDept>	[0..1]	BD	See recommendation above.
StreetName	<StrtNm>	[0..1]	BD	See recommendation above.
BuildingNumber	<BldgNb>	[0..1]	BD	See recommendation above.
PostCode	<PstCd>	[0..1]	BD	See recommendation above. Country dependent
TownName	<TwnNm>	[0..1]	BD	See recommendation above.
CountrySubDivision	<CtrySubDvsn>	[0..1]	BD	Country dependent
Country	<Ctry>	[0..1]	R	See recommendation above.
AddressLine	<AdrLine>	[0..7]	BD	See recommendation above.

Figure 23: CGI-MP's instructed usage of PostalAddress

Characters in the middle column (C, BD and R) indicate the usage of element in that row. They stand respectively for Conditional, Bilaterally Determined and Required. Country is indicated to be required. However, the third point in the text field for PstlAdr states that usage of only address lines is possible. Based on this it can be argued that leaving Ctry out is valid. But also it can be argued that it is not, as it is marked with R. In XML-file the impact of these two different usages can be seen from Figure 24 and Figure 25.

```
<Cdtr>
  <Nm>Creditor's name</Nm>
  <PstlAdr>
    <StrtNm>Example address street 1</StrtNm>
    <PstCd>22339</PstCd>
    <TwnNm>Hamburg</TwnNm>
    <Ctry>DE</Ctry>
  </PstlAdr>
</Cdtr>
```

Figure 24: Usage of “structured address”

```
<Cdtr>
  <Nm>Creditor's name</Nm>
  <PstlAdr>
    <AdrLine>Example address street 1</AdrLine>
    <AdrLine>22339 Hamburg, DE</AdrLine>
  </PstlAdr>
</Cdtr>
```

Figure 25:Usage of AdrLines

When a bank implements its internal system assuming Ctry to be mandatory, a processing error will be caused when it encounters payment message content where Ctry is not given.

It is worth keeping in mind that CGI-MP in the first place is not meant to be a strict standard. Banks using CGI-MP would not be in harmony with each other in any case as CGI-MP leaves myriad of elements to be “bilaterally determined”, meaning that it is up to banks to decide whether they want to use a specific element or not.

5. Common characteristics and evaluation of Implementation Guides

Analysing Implementation Guides by National Agencies and banks show that they share certain similarities but still are vastly different, especially when it comes to the format. For the purpose of this thesis, I read and analysed 13 Credit Transfer pain.001.001.03 Implementation Guides, seven written by banks and six by National Agencies. The Implementation Guides I chose are those of which I encountered first when searching for guides online. While the selection is not completely random, I feel that they represent the overall field of Implementation Guides well. Implementation Guides I analyzed from banks are by:

- DNB
- ING
- Osuuspankki
- Rabobank
- Handelsbanken
- RBS
- Bank of Ireland,

and from standard/national agencies:

- CGI-MP (Common Global Implementation)
- CBI (Consorzio CBI, Italy)
- DK (Die Deutsche Kreditwirtschaft, Germany)
- FK (Finanssialan Keskusliitto, Finland)
- Stuzza (Austria)
- BSK (Norway).

My goal is not to review these Implementation Guides themselves, but instead to study the practices they use and analyze whether the practices are fitting for presenting Payment Message data.

5.1. File format

The format of an Implementation Guide has the largest impact on how the information is presented and can be read. File format directly affects what tools can be used by the creator and how much area there is to present data. Out of my selection, the formats used were PDF, XLSX (Excel) and XSD (XML-schema). Table *3* shows the amounts of each format.

Format	Number	Percentage
PDF	9	69
Excel	3	23
XSD	1	8
Total	13	100

Table 3: Distribution of formats used in Implementation Guides I analysed

As shown in Table 3, most Implementation Guides in my selection are in the PDF format. All expect one present the XML related information in a table, most likely due to the fact that each XML element has many possible pieces of information attached to it. I believe this is the reason Excel is used as well. The odd format here is XSD, meaning that XML schema is used as a guide as well as the definition of XML. This means that in order to read the Implementation Guide in a non-XML format, a program containing a schema viewer is needed to view the Implementation Guide.

5.1.1. PDF

PDF format in Implementation Guides have certain advantages. They

- can be opened in most Windows / Mac operating systems, as long as PDF reader is installed
- always show the data as the creator intended it
- cannot be easily modified by readers
- are print friendly.

However, in the context of presenting large amount of data in a table form, PDF format comes with certain limitations. Disadvantages of PDF's are that they:

- are limited to A4 or A3 format
- are divided into pages, thus breaking tables spanning on multiple pages
- may not fully support copying of data.

Especially the limitation to A4 or A3 is painful when it comes to presenting large amount of data on screen. This also means, that by default, they are presented as pages and thus each page is separated with whitespace. PDF format itself supports both horizontal and vertical layout of a page, but due to margins in a document, whitespace is always present on both sides of a document. On vertically oriented documents, amount of unused space on each side of a document is highlighted even more if the monitor viewing it is horizontally orientated. Distribution of horizontal and vertical layout in my selection of PDF Implementation Guides is shown in Table 4.

Layout of table	Number
Vertical	4
Horizontal	4
No table	1
Total	**9**

Table 4: Division of table layout

The amount of guides using vertical and horizontal layout is equal in the Implementation Guides using a table for presenting XML element rules. One PDF document does not show table at all, and instead lists rules in bullet points. I believe that the reason for using horizontal layout is to increase the amount of information on X axis on screen, which is beneficial when information is shown in a table format, when amount of columns exceed five and information in cells can contain text in full sentences. The increased amount of pixels in X axis can be seen when comparing Table 5, showing table wideness of vertical layouts with Table 6, showing table wideness in horizontal layout.

IG	Table width (px)	Amount of columns
BSK	656	9
FK	597	4
DK	622	6
OP	726	6
Average	650,25	6,25

Table 5: Table width in vertical orientation

IG	Table width (px)	Amount of columns
Bank of Ireland	1034	6
DNB	1034	9
Handlsbanken	1434	9
Rabobank	1041	8
Average	1135,75	8

Table 6: Table width in horizontal orientation

Implementation Guides using horizontal layout, on average and based on Implementation Guides in my selection, have almost 500 pixels more area to use for each row in the table defining an XML element. In percentages turning a vertical orientation to horizontal increases the amount of space in each row by almost 75 %. This is a considerable increase. Layout also affects the average amount of columns used in the table, as horizontally aligned pages have almost two columns more than vertically aligned. Furthermore, horizontal layout gives more freedom for the creator of the guide to select what data is presented for each XML element. In addition, on columns containing text, it decreases the amount of rows required to present the text, as the cell is wider. I have not identified any negative impacts on a horizontal layout. It could be argued that horizontal layout increases the length of the table because less rows can be fitted into a page, but in practice this is not true. Horizontally aligned guides can fit more rows per page, as the increased width allows each row to take less space compared to vertically aligned page. Figure 26 and Figure 27 show an example of both orientation style. Dark areas on both sides of the images mark the borders of the page. Due to the vertical layout format of this thesis document, the horizontal table has to be zoomed out

for it to fit. It is worth noting that text within several cells span on multiple rows in Figure 26, depicting vertical layout of a page.

Name	XML Tag	Occur-rences	Definition	Type	EPC-/DK-Rules
OrganisationIden-tification	<OrgId>	[1..1]	Unique and unam-biguous way of iden-tifying an organisa-tion.	Organisa-tionIdentifi-cation-SEPAChoic e	Either „BICorBEI" or „Other must be allocated

Figure 26: Table within Implementation Guide, page aligned vertically

Name	Index	XML-tag	Level	Presence	Type	Length	Description
Ultimate Debtor	2.23	<UltmtDbtr>	1	[0..1]			This field and its subfields are ignored for Foreign, Multibank and urgent Euro Payments. This element can be on BATCH/PI level or on transaction level (field 2.70), but not both.

Figure 27: Table within Implementation Guide, page aligned horizontally

It can be said that PDF format allows some form of interactivity with its support for hyperlinks. However, when the target of the hyperlink is within the document, the user jumps to the target location without the possibility to return to the place where hyperlink was clicked. The fact that it is not possible to return to original location makes the hyperlink within PDF not suitable for exhaustive usage. For example, if a XML element table spans over 10 pages in length and hyperlink explaining the definition of specific XML element is clicked, user is taken to a page outside from the table and it is tedious to find the original location again.

5.1.2. Excel

The second most popular style for Implementation Guides in my selection is Excel format. It is created for tables, and as XML elements contain multiple different pieces of information, table is a fitting format to present the data. In addition, by default, Excel does not contain page breaks separating the content of a table.

A clear advantage of Excel format is that it allows easy hiding of data and offering a possibility to adjust the size of columns. This means that reader is able to tailor the visible view to his/her purposes, to use the available screen space as efficiently as possible and hide non-relevant information. In addition, Excel supports filtering data based on desired criteria. I identify these as a highly important aspects of the format, as this has potential to increase the readability of the document. However, these changes are up for the reader to make.

A potential negative aspect in the Excel format is the ease of filling Excel sheets with too much data, making it hard to find relevant pieces of information. One Implementation Guide in my selection depicts pain.001.001.03 Payment Message content and contains 223 columns and 945 rows. Different columns contain data for different payment types in different countries, so the information is relevant but clearly not for targeted for each

reader simultaneously. In this case filtering the table to suit the needs for a specific implementation case is tedious.

It could be argued that since Excel contains data in cells, it is not fitting for information in sentences and paragraphs. This claim can be countered with the fact that the creator is able to tailor the outlook of the document so that this type of information is shown clearly. In conclusion, Excel is the format which allows the most variance to the way information is presented. Usage of different sheets for different types of data, native support for columns and rows and color coding of cells all support this. This also means that it is largely up to the creator to make use of the format in the most efficient way possible.

5.1.3. XSD

The most innovative way to present Payment Message data in my selection of Implementation Guides is schema format, XSD. Programs made for viewing schemas highlight the hierarchical structure of schemas, making the depth of an XML tree clear to see and showing information what child-elements are included within elements. Figure 28 shows the usage of <Documentation> in Stuzza's Implementation Guide, viewed in schema editor XMLspy (Stuzza, 2014). The content between documentation tags is shown as gray text in the schema view mode.

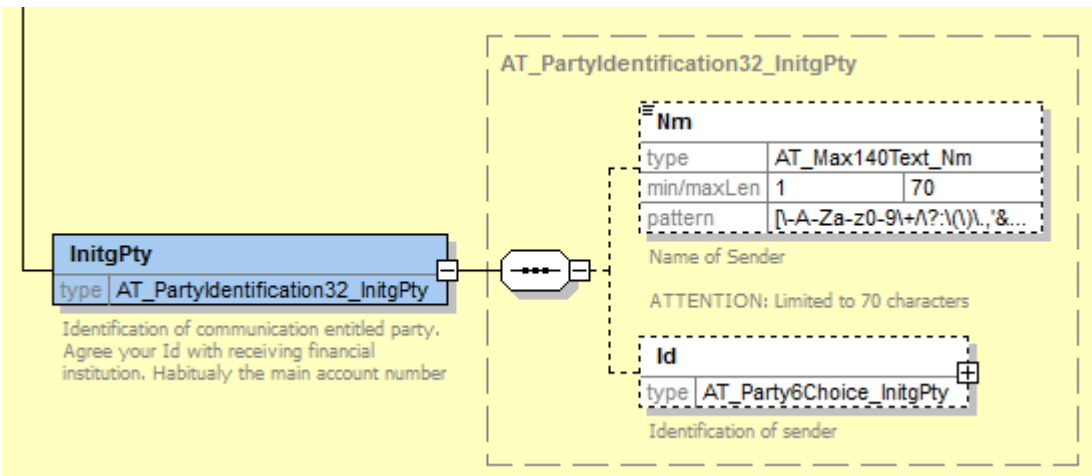

Figure 28: Example of the instructed usage of element "InitgPty" in Stuzza's SEPA Implementation Guide

In addition, viewing schema in a graphical form also makes it possible to see the type of an element clearly, indicating the possible values element may hold. Even though the amount of information visible at once is less than in the PDF and Excel formats, it is fairly straightforward to find the required element at any time. This is because element may be opened or closed in an XML-tree. I find this to be the biggest strength of Handbooks in schema format, as XML element depth is a crucial point in XML files, it is also crucial that the depth is indicated clearly in a Handbook. In the PDF and Excel

formats this may not always be the case, as the elements are listed in a vertical column. XSD makes use of horizontal space as well when depicting elements.

Negative side of XSD Handbooks is that they require more technical knowledge than their PDF and Excel counterparts. A special program is required view the visualization of the tree, and opening the Handbook in a normal text editor would only show it as an XML document. In addition, including general information about the payment would have to be done in another file, or alternatively within the XSD Handbook as an XML comment. Including information as a comment leads to it being easily missed, as the XSD file may contain close to a thousand lines. In addition, no formatting options for this text is available, so usage of headings, bolding and tables is not possible. This would often be beneficial due to the nature of the general information.

5.2. Data population

With data population I mean the extent of how thoroughly information is presented in an Implementation Guide. In the guides I analyzed I identified three different approaches, largest differences being in the way XML-element specific information is presented in a table form.

The first approach is to list absolutely everything necessary for creating payment messages, without the need for external documentation. This means that no prior information of Payment Messages is needed and no outside documentation is required to create a Payment Message (or a program generating Payment Message to be more precise). This sounds ideal, but the major drawback for reader is that the document will potentially be massive, especially if multiple payment types are to be covered. And even for covering single payment type, the amount of information presented for each element is large. The biggest impact is to define the XML element usage as ISO 20022 defines it, and this will take several sentences in a single cell of a table. More complexity is added when bank's usage differs or adds to ISO 20022's element usage, therefore this information has to be presented in the table as well. Second drawback is that when reader either already knows certain information beforehand or is prepared to acquire knowledge outside of the document, listing absolutely all information will result in an unnecessarily complex document. Presented non-relevant information takes away attention from actually relevant information.

Second approach is to accept the fact in some cases, most likely even on most of the cases, it is not sensible to list absolutely everything in an Implementation Guide. In this case Implementation Guide assumes that certain information is either known or reader is trusted to find it elsewhere. In the Implementation Guides I analysed this was the most common approach for the data population. In practice this means that the Handbook is made from bank/standard agency perspective, instead of the perspective of ISO 20022. In this type of Handbook it is assumed that the reader knows how to generate XML

messages, and the Handbook exists only to guide the reader to implement bank specific payment messages. This is done by listing the elements the bank accepts and adding a bank-specific rule for elements requiring it.

The third, the most minimal approach, is to only list the absolute minimum needed to create a valid payment message for the bank releasing the guide. In this type of guide only the difference to another standard is stated. This means that the bank specific rules are listed and that is all. So, no information on how bank would describe an element, what elements are mandatory and what may not be used. In the Implementation Guides I analysed there was only a single occurrence of this type of Handbook. This approach may be very efficient if done correctly, and only for certain user-type. The reason is that if reader already knows all other information beforehand, he/she does not need it to be listed in bank specific guide. Only the bank-specific rules are needed. Listing the rules of one payment type could be fitted into a single page in a PDF document.

An example of these three different styles to populate information in a Handbook can be seen the usage of tables in three different Handbooks, as shown in Figure 29 (Handelsbanken, 2013), Figure 30 (Finanssialan Keskusliitto, 2012) and Figure 31 (ING, 2012).

ISO Index No.	Struct. Seq.	Message Item	Tag Name	Mult.	Status	Type	Definition	Handelsbanken Special comments
	-	CustomerCreditTransferInitiation	<CstmrCdtTrfInitn>	[1..1]	M			
1.0	+	GroupHeader	<GrpHdr>	[1..1]	M		Set of characteristics shared by all individual transactions included in the message	
1.1	++	MessageIdentification	<MsgId>	[1..1]	M	1-16 Text	Unique identification, as assigned by the initiating party, and sent to the next party in the chain to unambigouously identify the message.	The first 16 characters should be unique.

Figure 29: Example of exhaustive table in Handbook

Indeksi	Mult	Message Element	Soveltamisohje
1.0	[1..1]	+ Group Header	
1.1	[1..1]	++ Message Identification	Maksusanoman yksilöivä tunniste, jonka maksusanoman muodostaja antaa.

Figure 30: Example of “normal” amount of information in Handbook. Column “Soveltamisohje” depicts National Agency specific information

1.8 Initiating Party <InitgPty>
Either Name <Nm> or Identification <Id> must be specified and must contain other characters than only spaces.

2.3 Batch Booking <BtchBookg>
Mandatory
Only "false" is allowed.
The word "false" must be in lower case!

Figure 31: Example of minimal information in a Handbook

5.3. Gestalt laws and Implementation Guides

Evaluating Implementation Guides from the perspective of Gestalt principles offers some data on how human readable Handbooks are in presenting information. In this section I will explain what Gestalt principles are and evaluate how well Handbooks generally obey them.

Gestalt principles originate from gestalt psychology, or gestaltism, which is an area of psychology trying to understand how humans acquire and maintain perceptions. Gestalt as a word originates from German, simply meaning *pattern*. The gist of gestaltism is often condensed in a quote from Kurt Koffka, "The whole is other than the sum of the parts" (Tuck, 2010), meaning that multiple different aspects together affect how a certain thing is understood, instead of there being one major aspect in a key role. Individual aspects, or design choices, might not work when used separately, but when used together, the whole becomes its own independent aspect.

Gestalt laws translate into a clear set of design principles, which are

- Proximity
- Similarity
- Connectedness
- Continuity
- Symmetry
- Closure
- Relative Size
- Figure and Ground.

These principles can be used as rules on how humans understand and see patterns (Ware, 2004). As Handbooks are static representations of information I believe that this is a good way to objectively attempt to analyse the tables used in Handbooks representing XML element information. It is interesting to see what gestalt laws can generally be identified from the tables used in Handbooks, how the gestalt laws work together and especially whether there are some occurrences of table usage which work specifically well.

Figure 32 (Rabobank International, 2014) depicts a common format of a table in a PDF document aligned horizontally.

Name	Index	XML-tag	Level	Presence	Type	Length	Description
PaymentInformation	2.0	<PmtInf>	Top	[1..n]			
PaymentInformationIdentification	2.1	<PmtInfId>	1	[1..1]	Alphanumeric	35	Unique reference number of the Batch/PI. For normal Euro Payments, your PmtInfId is overwritten by a PmtInfId generated by RCM. For Foreign, Multibank and urgent Euro Payments, this value is no longer relevant after import and is not reported on your account statement.
PaymentMethod	2.2	<PmtMtd>	1	[1..1]	Alphanumeric	3	Fixed Value 'TRF' for credit transfers.
BatchBooking	2.3	<BtchBookg>	1	[0..1]	Boolean	4 or 5	For normal Euro Payments, value 'true' indicates you want all transactions in the batch reported as a single item on your account statement, no details of the transactions will be available. Value 'false' indicates each transaction is reported as an item on your account statement including all relevant details. If no value is present, value 'true' is assumed. For Foreign, Multibank and urgent Euro Payments, this value is no longer relevant after import. All transactions are reported as a single item on your account statement.
NumberofTransactions	2.4	<NbOfTxs>	1	[0..1]	Numeric	15	The number of transactions within this Batch/PI.
ControlSum	2.5	<CtrlSum>	1	[0..1]	Amount	18	Total amount of all transactions within this segment. Decimals are separated by a point ".".
PaymentTypeInformation	2.6	<PmtTpInf>	1	[0..1]			Must be present here or on transaction level (2.31).

Figure 32: Table as presented by Rabobank

In Figure 32 it can be seen that the description column contains large amount of information, which increases the height of certain rows more than others. Evaluating the table from the point of gestalt law proximity, there are three distinct groups of element names: indexes 2.0 and 2.1 forming the first and 2.2 and 2.3 forming the second. Proximity is countered with connectedness, by distinct blue borders for both rows and columns within the table. It is argued that connectedness is the fundamental gestalt organizing law (Palmer & Rock, 1994), and the effect is indeed powerful. As each column contains similar style of information, it could be said that they are symmetrical. However, in this example, only focusing on visual cues it is not clear whether it is the rows together or columns together which form the important group of information, as they are both weighted equally. In the example of Figure 32, similarity could have been used for rows by using alternating background color, to guide the viewer to group rows together, thereby helping in seeing the important information. In addition, as each row contains exactly the same style as other rows, complex and simple types are not separated in any way, making them look similar. Similar rows, or types, could have been highlighted. One option would have been to use different background color only on rows containing complex types instead of different background color on alternating rows. Example of this type of usage can be seen in Figure 33 (Handelsbanken, 2013).

ISO Index No.	Struct. Seq.	Message Item	Tag Name	Mult.	Status	Type	Definition
	-	CustomerCreditTransferInitiation	<CstmrCdtTrfInitn>	[1..1]	M		
1.0	+	GroupHeader	<GrpHdr>	[1..1]	M		Set of characteristics shared by all individual transactions included in the message
1.1	++	MessageIdentification	<MsgId>	[1..1]	M	1-16 Text	Unique identification, as assigned by the initiating party, and sent to the next party in the chain to unambigouously identify the message.
1.2	++	CreationDateTime	<CreDtTm>	[1..1]	M	DateTime	Date and time at which the message was created. YYYY-MM-DDThh-:mm:ss.sss
1.6	++	NumberOfTransactions	<NbOfTxs>	[1..1]	M	Numeric	Number of individual transactions contained in the message.
1.7	++	ControlSum	<CtrlSum>	[0..1]	O	Numeric	Total of all individual amounts included in the message, irrespective of currencies.
1.8	++	InitiatingParty	<InitgPty>	[1..1]	M		Party that initiates the payment.

Figure 33: Usage of similarity by using different background colour in rows

EPC's usage of tables in SEPA Credit Transfer Implementation Guide version 8 is made from the perspective of containing exhaustive amount of information, so that the Handbook alone contains sufficient amount of information for generating a payment message. This can be seen in Figure 34 (EPC, 2015), where each element contains ISO definition, element depth, full ISO 20022 name and element data restrictions.

#	SEPA Mult	Message Element	SEPA Core Requirements
2.5	0..1	Customer Credit Transfer Initiation V03 +Payment Information ++Control Sum	*SEPA Usage Rule(s)* *The fractional part has a maximum of two digits.* ISO Name Control Sum ISO Definition Total of all individual amounts included in the group, irrespective of currencies. XML Tag CtrlSum Type DecimalNumber *SEPA FractDigits* *17* TotalDigits 18
2.6	0..1	Customer Credit Transfer Initiation V03 +Payment Information ++Payment Type Information	*SEPA Usage Rule(s)* *If used, it is recommended to be used only at 'Payment Information' level and not at Credit Transfer Transaction Information' level. When 'Instruction Priority' is to be used, 'Payment Type Information' must be present at 'Payment Information' level.* ISO Name Payment Type Information ISO Definition Set of elements used to further specify the type of transaction. XML Tag PmtTpInf Type PaymentTypeInformation19
2.7	0..1	Customer Credit Transfer Initiation V03 +Payment Information ++Payment Type Information +++Instruction Priority	*SEPA Usage Rule(s)* *If present, pre-agreed customer-to-bank conditions apply.* ISO Name Instruction Priority ISO Definition Indicator of the urgency or order of importance that the instructing party would like the instructed party to apply to the processing of the instruction. XML Tag InstrPrty Type Priority2Code

Figure 34: Example of a Handbook containing exhaustive amount of information

Interestingly, as each element in Figure 34 contains large amount of information, proximity supports grouping of elements together by making it clear that there are three distinct elements, grouped with the amount of text on the column "Message Element". However, similarity makes the distinction to columns, completely dividing the table into two sections: white and yellow. In addition, only four columns are present, and the rightmost column contains multiple pieces of information, so similar pieces of

information in that column are not connected with each other. Some grouping is, however, done by similarity, by italicizing the “SEPA Usage Rule(s)” section.

5.4. Conclusion of the evaluation

In this section I analysed the practices used by banks and standard agencies in their Implementation Guides. I found out that the formats and ways to present information differ from bank to bank and that there is no standard way how XML element information is presented in a table. Especially the amount of information given per each XML element varied greatly. I suspect one of the reason for this is that the Handbooks did not share a similar design purpose – some guides were made to be complete documents containing sufficient information to generate payment messages by their own, but most Implementation Guides relied on other documents for the reader to get certain type of information. Especially regarding the ISO 20022 element definition.

Vastly different documents visually and in different formats suggest that banks and standard agencies did not use a shared program generating Handbooks. That is why I suspect that each document was built individually from a scratch, or by on top of a previous version of the Handbook.

I found gestalt laws to be present in Handbooks, but they were used differently in many Implementation Guides. I suspect that in the creation process gestalt laws were not being paid attention to in many parties releasing Handbooks.

6. Creating Implementation Guides

In this chapter I will use the information gathered in Chapters 4 and 5 and apply it to the creation process of Implementation Guides. I will attempt to explain what aspects to keep in mind when creating Implementation Guides to make them as easily understandable as possible and help estimate how much data to introduce in the Implementation Guide.

In addition, I will share my own experiences from a project where the goal was to produce a Handbook. Therefore, while focusing on the end-product, the Handbook, in this chapter I will explain that the tools and practices used in the creation process of the Handbook play the biggest role affecting the quality of the final product.

Finally, I pill present interactive Implementation Guides and argue that interactivity within Handbook would be greatly beneficial for the reader. In addition, I will explain that a program specialized for generating Handbook would be helpful for the people responsible for creating them.

6.1. Audience

The challenge when creating Implementation Guides is to estimate how much prior knowledge the future readers will have, while keeping in mind that expertise of readers varies. Unless the Handbook is tailored for certain user type (XML and payments message experts vs beginners), compromises have to be made on how thoroughly XML elements are introduced. This is because including absolutely all possible implementation will easily make the Implementation Guide long and populated with unnecessary data, which is not beneficial for all readers. However, it is known that the goal of end-users is the same, to implement a system which produces payment messages as they are defined by the Implementation Guide.

Audience plays the most important role on how exhaustively should the information be presented and thus defines the basic structure for presenting information. Depending on how widely the Implementation Guide will be circulated, it would make sense for creator of Implementation Guide to try to find out the audience. For example, in a scenario where the desired Implementation Guide will be distributed only to a small number of clients, it would absolutely make sense to find out what is the existing expertise of those clients. When clients already have exhaustive experience in XML payment messages, the ideal case would then to create a minimal Implementation Guide. When audience has prior knowledge, it is not necessary to present that same knowledge again in an Implementation Guide. In addition, if (and only if) bank is making a transition from a legacy format into XML, it should be considered whether it to tailor the Implementation Guide for this transition by including information how an item in legacy format is treated in XML. In this kind of tailored approach, the legacy information is going to be useless for all readers who do not need to make the transition from a specific legacy format into XML.

I would argue that at least in Europe at the beginning of year 2015, bank's corporate customers already have a good understanding of XML payments. I base this opinion on the fact that majority of European countries and banks within them had to be SEPA ready (and thus XML ready) by the start of year 2014 (European Commission, 2014).

6.2. Creation process

The challenge when creating Implementation Guides is that there is large amount of ISO 20022 elements which are visually similar and potentially hard to distinct from each other. Human errors in creation process lead to unclear items, and in more drastic cases, false information. Negating these are crucial. Thus, how to minimize human errors?

Raw knowledge of ISO 20022 helps to minimize some type of errors in the creation process and helps quicken the process. If the creator knows and remembers which ISO 20022 elements hold actual textual data and which elements are containers for child-elements (complexType vs simpleType), a separate document is not needed to cross-check elements constantly. Similarly, it is a large help if the multiplicity of elements is remembered without checking them from elsewhere.

The biggest impact affecting creation process is the desired output format, as this determines what program can be used to create the Handbook. Second largest item to address is to decide how thoroughly XML elements are presented. However, if it is decided that the Handbook must work as an individual document, so that reader is able to view the Handbook individually without the need of external documents to produce a payment message, it should carefully be considered whether PDF is the right format for this.

Due to limitations in bank's internal system the case may be that some mandatory elements within XML message cannot be taken into account at all. Therefore, it is an interesting question whether this should be mentioned in the Implementation Guide for the relevant XML elements. This would be useful information for the sender to know, as the data in those mandatory elements will be lost and cannot be retrieved.

When deciding what data to actually include within the Handbook, Table 2 in Section 4.2 can be used as a reference and adjusted to the intended audience of the Handbook. The example table layout in Figure 35 presents a scenario where the goal of the Implementation Guide is only to present the rules bank / standard agency has.

Schema: pain.001.001.03.xsd		ISO 20022 format	CGI-MP rules	Example data
	R 1 2 3 4 5 6 7 8 9 10			
1..1	Document			
1..1	CstmrCdtTrfInitn			
1..1	GrpHdr			
1..1	MsgId	Max35Text		Id_101
1..1	CreDtTm	ISODateTime		2015-12-17T09:30:47Z
0..1	NbOfTxs	Max15NumericText		1
1..1	InitgPty			
0..1	Nm	Max140Text		Name of initiator
0..1	Id		Mandatory	
XOR	OrgId		Mandatory	
1..1	Othr			
1..1	Id	Max35Text		Initiating party's Org. Id.
1..n	PmtInf			
1..1	PmtInfId	Max35Text		PmtId_101
1..1	PmtMtd	PaymentMethod3Code	{TRF}	TRF
0..1	PmtTpInf		External Rule 1	
0..1	SvcLvl		Conditional	
1..1	Cd	ExternalServiceLevel1Code		NURG
1..1	ReqdExctnDt	ISODate		2015-12-31
1..1	Dbtr			
0..1	Nm	Max140Text	Mandatory	Name of debtor
1..1	PstlAdr		Mandatory. External Rule 2	
0..1	Ctry	CountryCode		FI
1..1	DbtrAcct			
1..1	Id			
1..1	IBAN	IBAN2007Identifier		FI7429501800000014
0..1	Ccy	ActiveOrHistoricCurrencyCode	Mandatory	EUR
1..1	DbtrAgt		External Rule 3	
1..1	FinInstnId			
1..1	BIC	AnyBICIdentifier		POPFFI22
1..1	PstlAdr		External Rule 2	
1..1	Ctry	CountryCode	Mandatory	FI
1..n	CdtTrfTxInf			
1..1	PmtId			
1..1	EndToEndId	Max35Text		Id_101_endtoend
1..1	Amt			
1..1	InstdAmt	ActiveOrHistoricCurrencyAndAmount		100
1..1	CdtrAgt			
1..1	FinInstnId			
0..1	BIC	AnyBICIdentifier	External Rule 3	NDEAFIHH
0..1	PstlAdr			
0..1	Ctry	CountryCode		FI
0..1	Cdtr		Mandatory	
0..1	Nm	Max140Text	Mandatory	Name of Creditor
0..1	PstlAdr		Mandatory	
0..1	Ctry	CountryCode	Mandatory	
1..1	CdtrAcct			FI
1..1	Id			
1..1	IBAN	IBAN2007Identifier		FI3329501800008512

Figure 35: One example on how to present rules for a payment message. This example covers the rules CGI-MP sets for pain.001.001.03 normal payments, shown with the least amount of elements

With the information presented in Figure 35, it would be possible to create a valid payment message according to rules it sets. However, it should be noted that this table alone would most likely not be clear enough for a user who is not familiar with

pain.001.001.03 payment messages. Additionally, information not presented in the table has to be known, namely version number of the Handbook, accepted charset and treatment of elements not depicted in the table.

The leftmost column in Figure 35 shows the multiplicity of an XML element, with different colors indicating presence. Repeating elements are indicated with red, mandatory elements with black and optional elements with gray. Element depth is seen on columns “R” to “10”, with element “Document” being the root. By knowing the depth it is possible to determine what are the parent and children of a specific element. Different colors are used as a background for elements to depict whether they are simple or complex types, e.g., whether they may include textual data or other elements. In the example of Figure 35 light blue indicates a simpleType element, which may contain text. Type of data is included in column “ISO 20022 format”. Using this format as a reference it is possible to find the exact data limitations element has, but the actual limitations are not listed in this table as it would require more space than only including the type. Furthermore, it is possible deduce the data limitations based on the type name alone, although this may not always be the case for readers who are not familiar with ISO 20022 pain.001.001.03 payment message. Rules are listed in the “CGI-MP rules” column. Here it should be noted that I chose to only use one row of text for each rule to keep the row height uniform. Whenever a rule does not fit within the cell, a tooltip is shown when user selects the cell. Finally, example data is included in the last column. Here the background color is not used to differentiate it from the rules. The example values would produce a valid payment file when inserted into XML document according to CGI-MP rules.

Gestalt laws identifiable from the example shown in Figure 35 are proximity, making rows distinct from columns as rows are always uniform in height, which is smaller than the width of columns. Connectedness is achieved with row borders and different background colors of simple and complex type elements. Continuity is most visible in the multiplicity of elements, result of shared row height, as well as in the rows sharing color. In addition, arguably also in the element names, even though depth is also contained in the column. One interesting difference between the table in Figure 35 and the tables I studied in Chapter 5 is that my table does not use proximity in a meaningful way. However, I do not believe this is a problem nor is it the goal of gestalt laws to include each of them – they just have to work well when used together.

In my opinion, complexity within Handbook is increased almost exponentially when the Handbook has to cover multiple payment types for a single payment message. The information for multiple payment types either has to be included in a single cell for each XML element, or multiple tables have to be made, each to cover an individual payment type. If the table presented in Figure 35 had to contain four payment types, for example CGI-MP, SEPA, domestic and foreign, it would be challenging to present them. ISO 20022 format would stay the same, but the rules and example data would have to be

added. In this case it could be considered to remove example data altogether, so that only three additional columns would have to be added. However, this would lead to a situation where a certain element is used in one payment type, but ignored in the others, and it is very likely that there would be multiple of these elements. This would then increase the total element count which would have to be included in the table.

6.3. Personal experiences

I worked in a project where the main goal was to produce an XML Handbook covering over 15 different payment messages for pain.001.001.03 Credit Transfers and Camt.053.001.02 Status Messages. Requirements were that the Handbook has to function as an individual document, without the need for the reader to rely on other documents to create XML messages. Handbook was to be in PDF format, pages oriented vertically and single document was to include all Payment and Status Messages. As this is the only project I have been on where the goal is to create a handbook, I cannot compare it to other creation processes. However, my educated guess is that this project was bigger than a "normal" Handbook release by a bank. I base this to the fact that the amount of messages we had to cover was bigger than in a scenario where, for example, European bank creates a Handbook for SEPA and domestic payments. However, I believe that the processes involving the creation of the Handbook most likely are similar in other banks. This project gave me insight into the actual creation process of Handbooks, which interestingly made me see Handbooks I had read prior the project from a new angle.

This project can be divided into two overlapping but distinct parts: gathering information and writing Handbook. The nature of gathering information in our case was that the handling of elements is ever changing and that the amount of information is vast. For each payment type, there had to an internal document with payment type related information and a place for each individual "piece of payment information" to store comments and internal discussion for. Piece of payment information in this case means a field in the internal format bank is using, where data from an appropriate XML element is taken from. This document is then modified and improved in online meetings and exchanged via email with us and the customer. Therefore, the requirements and purpose of the internal document were to

- store payment type information
- store bank's internal payment field information
- modified and shared during meetings
- contain ongoing discussion for each payment field in payment type
- sent to bank's experts for analysation and correction.

This type of task would benefit from having a specialized program made for it. We settled with using Excel, which worked adequately, but the fact remains that Excel is not designed for these purposes. Drawbacks of Excel were that controlling different versions

is difficult when content is changing and separate copies of documents are shared with customer. In addition, storing the ongoing discussion for each element will easily lead to great deal of textual data in cells.

Considering there were multiple people working on the project (three) and to make versioning easier, we wrote separate Word documents for each individual Payment and Status Messages. These acted as individual sections in the Handbook. This allowed each individual document to be updated individually from the other documents. One dynamically updating document was created to contain a reference to all other documents, so this dynamical document acted as the Handbook which was returned once the project was complete.

The key aspect during the project was that source information and the Handbook were kept in separate places. The importance of the "internal" document proofed to play a major role on how easy or difficult it was to write the Handbook. This is because when the source document is unclear, then the Handbook, the output document, will be unclear as well unless the unsettled item is resolved in the internal document. In one Handbook section containing one payment message there may be several different unclear items pending simultaneously, which means that Handbook is updated as we got more information from the customer. In other words, in order for us to implement a clear and correct Handbook, the internal document containing the source information had to be clear and up to date.

Another challenging aspect was to create example XML-files based on the data we had in Handbook. The purpose of example files in Handbook is to allow the reader to have actual data as a reference to support the instructions in Handbook, so their need is clear. However, implementing and keeping them up to date proofed to be tedious. Whenever the source document is updated, relevant Handbook item had to be updated as well as the relevant section in the example XML file. As they are in XML format, reading and changing a value directly to XML document is tedious, so VBA (Visual Basic for Applications, natively supported programming language in Excel) macro was created to read and produce XML files. This allowed us to have a primitive user-interface in Excel to create and modify XML-files and to easily see the actual data XML elements hold in the example files we produce.

6.4. Interactive Implementation Guides

Interactivity in Implementation Guides means that information not only flows from the Handbook to user, but in addition from user to Handbook, as the Handbook responds to user's input. This means that the user is able to change parameters within the Handbook to best suite his/her current usage case. If done correctly, the advantages of this added layer of information flow would be great. Advantages would include that the user is able to view more information on the XML elements relevant to user, and other, non-relevant

information can be selected to be hidden. This means that visible information in screen is as non-cluttered as possible. For example, interactive Handbook could show tooltips of the ISO 20022 element usage when an element is selected. In addition, if elements are shown in table format, columns like ISO index number and full ISO 20022 name can be hidden whenever they are not needed (full name means the non-abbreviated form of an ISO 20022 element, for example “NumberOfTransactions” for element “NbOfTxs”).

Due to the fact that the knowledge and needs of bank’s corporate customer’s varies, it is not possible for one static Implementation Guide to be as viable as possible for every customer. This would require a tailored guide for every expertise level of reader and for every payment type. The differences of XML payment message experience is large between an ERP provider catering all possible payment types supported by bank and a small company implementing a system sending SEPA payment messages to bank.

How can interactivity be achieved in Handbook? There are at least two viable ways: HTML Handbook or a Handbook made and viewed in a specialized program or platform.

Advantage for HTML Handbooks is that they do not require any specialized programs to create them and they can be viewed in a web-browser. In HTML Implementation Guide series of web pages would transfer the information on how to successfully implement a system producing XML payment messages to a specific bank. Certain features of web-pages transfer well to Implementation Guides, especially the support for hyperlinks and tooltips. However, interestingly I have not encountered a single HTML Implementation Guide, either in my research for Chapter 5 or in any work environment. I am unsure what is the reason banks and financial organizations do not release HTML Implementation Guides, but I suspect it has to do with the fact that they are more tedious to implement than PDF and Excel formats, and their versioning is not as clear as in other formats. PDF and Excel Handbooks can be implemented in WYSIWYG editors as opposed to creating web pages in HTML and CSS. Once they are released and sent forward, I speculate banks might consider it more official to send a PDF/Excel document than a link to a web page.

The negative aspect of HTML Implementation Guides, being tedious to create, can be overcome if the Implementation Guide is made with a specialized program. The output Handbook could either be in HTML format, or alternatively, in a format specialized to show Handbook related information. Advantages of a specially made program for Handbook creation is that they can help the user to create the Handbook, can be more easily maintained, can be made less error prone and somewhat negate the needed expertise to effectively produce a Handbook.

A specialized program made for creating interactive Implementation Guides could have an XML schema uploaded into it, which is then parsed by the program and creator can use the schema as a base when creating the Handbook. This fact alone helps tremendously, as writing/copying element names is not needed and it isn’t possible to

make human errors in writing element names incorrectly, mixing complex / simple types in XML elements or making errors indicating XML element multiplicity.

7. Conclusion

In this thesis I explained the role of standard agencies, payment messages and Implementation Guides in the financial industry. I investigated what type of data is required to be included in an Implementation Guide in order to create a valid payment message based on it. In addition, I gave insight on what type of items should be considered when creating Implementation Guides for them to be as easily understandable as possible.

I explained the need for utmost accuracy when it comes to Implementation Guides. It is crucial that the Handbook is factually correct and in an easily understandable format. Incorrect information in a Handbook released by bank leads to increased costs for both the bank and its corporate customer, as the issues within the invalid payment messages have to be resolved before payment messages can be processed by the bank. Ambiguous data in Handbooks is not as drastic as invalid data in a Handbook, but it too leads to increased resources used by both the bank and its corporate customer. Information in Handbook has to be fully understood in order for bank's corporate customer to produce correct payment messages. Factual data in an easily understandable format in Handbooks promote a smooth on-boarding for bank's corporate customers.

During my research for the thesis I identified that that the industry is lacking a service generating and maintaining Implementation Guides, as the guides I analysed did not follow any specific pattern and contained human errors. Human errors could be negated by a system checking the validity of data in an Implementation Guide.

I identified that interactivity is not used in Handbooks currently. I estimate that end-user and the creator both would benefit from the features interactivity allows, as it would promote more control for the user to view the guide in a way he/she desires. The benefit of interactivity for the creator would be it that it removes the need to tailor the Implementation Guide for a specific user-type.

For me the goal of this thesis was to focus on the end-result of Handbook, to evaluate the final version which is read by the customers of banks. Therefore, the biggest epiphany for me was to realize that the tools and practices used in the creation process of the Handbook are directly linked with the quality of the end-result. Tools and practices used in the creation phase are the foundation Handbook will be based on. In order for the Handbook to be excellent, the base has to be excellent as well.

References

Atwood, J. (2008). *XML: The Angle Bracket Tax*. Retrieved from Coding Horror: http://blog.codinghorror.com/xml-the-angle-bracket-tax/

Bott, J. (2009). The Single Euro Payments Area - New Alliances Required to Tip the Market. *ECRI Research Report*.

Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., Yergeau, F., & Cowan, J. (2006). *Extensible Markup Language 1.1*. Retrieved from W3C: http://www.w3pdf.com/W3cSpec/XML/2/REC-xml11-20060816.pdf

DNB Banka. (2014). *Specification for ISO 20022 Payment Initiation Message, Version: 1.0.* Retrieved from https://www.dnb.lv/sites/default/files/pictures/specification_for_iso_20022.pdf

Dutch Banking Assiociation. (2015). *Mission.* Retrieved from NVB English: https://www.nvb.nl/en/association/272/dutch-banking-association.html

EPC. (2015). *SEPA CREDIT TRANSFER SCHEME CUSTOMER-TO-BANK IMPLEMENTATION GUIDELINES.* Retrieved from http://www.europeanpaymentscouncil.eu/index.cfm/knowledge-bank/epc-documents/sepa-credit-transfer-scheme-customer-to-bank-implementation-guidelines-version-8/epc132-08-c2b-ctig-v80-approvedpdf/

EPC. (2015). *SEPA ISO 20022 Message Standards*. Retrieved from European Payments Council: http://www.europeanpaymentscouncil.eu/index.cfm/sepa-direct-debit/iso-20022-message-standards/

European Central Bank. (2015). *SEPA countries.* Retrieved from European Central Bank: https://www.ecb.europa.eu/paym/retpaym/paymint/sepa/html/index.en.html

European Central Bank. (2015). *SEPA Migration indicators*. Retrieved from European Central Bank: https://www.ecb.europa.eu/paym/retpaym/paymint/indicators/html/index.en.html

European Commission. (2014). *Single Euro Payments Area (SEPA): Commission introduces an additional transition period of six months to ensure minimal disruption for consumers and businesses.* Retrieved from European Commission: http://europa.eu/rapid/press-release_IP-14-6_en.htm?locale=en

Finanssialan Keskusliitto. (2012). *ISO 20022 PAYMENT GUIDE.* Retrieved from Finanssialan Keskusliitto: https://www.fkl.fi/en/themes/sepa/sepa_documents/Dokumentit/ISO20022_Payment_Guide.pdf

Finanssialan Keskusliitto. (2015). *About the Federation of Finnish Financial Services.* Retrieved from Federation of Finnish Financial Services: https://www.fkl.fi/en/about_us/Pages/default.aspx

Handelsbanken. (2013). *ISO 20022 CustomerCreditTransferInitiation pain.001 version 3.* Retrieved from http://handelsbanken.com/shb/inet/icentsv.nsf/vlookuppics/a_filmformatbeskrivningar_fil_iso_20022_xml_pain_001_ver_3_credit_transfer/$file/iso_pain_001_001_03_credit_transfer.pdf

ING. (2012). *ING FTP Service - Appendix to "XML Message for SEPA Credit Transfer Initiation" version 6.x.* Retrieved from https://www.ing.nl/media/ING_nvb_european_credit_transfer_sepa_en_tcm162-45682.pdf

International Organization for Standardization. (2015). *About ISO.* Retrieved from http://www.iso.org/iso/home/about.htm

ISO 20022. (2015). *Frequently asked questions*. Retrieved from ISO 20022: http://www.iso20022.org/faq.page

ISO 20022. (2015). *Full catalogue of ISO 20022 messages*. Retrieved from ISO 20022: http://www.iso20022.org/full_catalogue.page

ISO 20022. (2015). *Variant - Definition and characteristics.* Retrieved from http://www.iso20022.org/variants.page

ISO 20022 Registration Authority. (2009). *ISO 20022.* Retrieved from http://www.iso20022.org/message_archive.page

ISO 3166 Maintenance Agency. (2015). *Country Codes - ISO 3166*. Retrieved from http://www.iso.org/iso/home/standards/country_codes.htm

NVB. (2012). *XML message for SEPA Direct Debit Initiation Implementation Guidelines for the Netherlands.* Retrieved from http://www.betaalvereniging.nl/wp-uploads/2012/07/XML-Message-for-SDD-Version-6.0-March-2012.pdf

Palmer, S., & Rock, I. (1994). Rethinking perceptual organization: The role of uniform. *Psychonomic Bulletin and Review*, 29-55.

Rabobank International. (2014). *Rabobank Format description CT-XML*.

Stuzza. (2014). *Kunde-Bank XML Schmeadata.* Retrieved from https://www.stuzza.at/en/download/xml/kunde-bank/rb-7-1/xml-schemata.html

Swift. (2014). *Credit Transfer Base Template MIG.* Retrieved from CGI resource centre: http://corporates.swift.com/en/common-global-implementation-0

Swift. (2015). *CGI: mission and scope*. Retrieved from Swift: http://corporates.swift.com/en/cgi-mission-and-scope

The SWIFT Standards Team. (2010). *ISO 20022 for Dummies.* John Wiley & Sons, Ltd.

Tossavainen, T. (2011). *SEPAn tuomat haasteet yrityksille.* Tutkielma. Hämeen ammattikorkeakoulu.

Tuck, M. (2010). *Gestalt Principles Applied in Design*. Retrieved from http://sixrevisions.com/web_design/gestalt-principles-applied-in-design/

W3C. (2015). *Introduction.* Retrieved from http://www.w3.org/XML/

Ware, C. (2004). *Information Visualization, Perception for Design.*

XMLdation. (2015). *XMLdation*. Retrieved from Portfolio: http://www.xmldation.com/en/services/portfolio

XMLspy. (2015). *XMLSpy XML Editor*. Retrieved from XMLspy: http://www.altova.com/xmlspy.html