Public Process Management: a method for introducing Standard Business Reporting

Joris Hulstijn
Remco van Wijk
Delft University of Technology
The Netherlands
j.hulstijn@tudelft.nl
r.vanwijk@tudelft.nl

Niels de Winne
Nitesh Bharosa
Delft University of Technology
The Netherlands
n.bharosa@tudelft.nl
c.p.i.dewinne@tudelft.nl

Marijn Janssen
Yao-Hua Tan
Delft University of Technology
The Netherlands
m.f.w.h.a.janssen@tudelft.nl
y.tan@tudelft.nl

ABSTRACT
Businesses have to file many reports to show compliance with rules and regulations. Regulators try to reduce the administrative burden, by providing a standardized representation format and agreements about reporting procedures and the use of technical infrastructure. However, developing and managing such a standardized reporting scheme is hard. It involves interdependencies between processes, data and technology and the interests of many stakeholders. Drawing on existing practice this paper presents Public Process Management (PPM): a general method for process management in the public sector. In this paper we apply PPM specifically to the problem of introducing a standardized reporting scheme in an application domain. The method is driven by quality management and process redesign approaches, but deals with unique characteristics of compliance reporting: legal data requirements, provenance, process compliance and multiple stakeholders. In particular, PPM stresses strict adherence to an iterative development schedule, and shared conceptual models of processes, data definitions, technological infrastructure and governance agreements. The usefulness and adequacy of the method are illustrated by a case study on Standard Business Reporting, a standardized reporting channel in the Netherlands for both public and private agencies.

Categories and Subject Descriptors
H.1.0 [Models and Principles]: General. J.1 [Administrative Data processing]: Administrative Data processing -- Government

General Terms
Standardization, Legal Aspects

Keywords
e-government, financial reporting, compliance

1. INTRODUCTION
In response to incidents, society often calls for more or more stringent regulation. Also in business there are many corporate governance guidelines and auditing standards. Power [1] calls this the audit society. Compliance to the rules must be shown. Companies must therefore file an increasing amount of compliance reports. Consider for example tax reports, annual financial statements, reports about compliance with food and health regulations, applications for a loan, or evidence of corporate governance. A compliance report contains data, which serves as evidence that the business complies to some regulation or internal guideline. Preparing reports takes a lot of effort. Governments try to reduce the administrative burden for businesses by standardizing business reporting formats and procedures, and providing shared information technology support. There are at least three possible scenarios to achieve this: (1) One stop shop government [2]: data only needs to be provided once to a single entry point. Different government agencies can reuse the data. (2) Store once, report many. Companies only have to store their data once in a particular data representation format; from this data various reports can be generated. (3) Continuous control monitoring: data for reports is sent from the company’s information systems on a continuous basis, to monitor compliance [3]. This approach may be called ‘piggy-backing’ [4]: compliance rides along on commercial data.

An open data representation standard like XBRL with a taxonomy to provide a shared semantics could facilitate these approaches. In principle, data elements of the company’s information systems only need to be mapped once onto the data elements for reporting. Not only will this improve efficiency, but it will also improve reliability, as the original commercial data is used for reporting and there is less opportunity for manipulation. To accomplish these objectives, the complete information processing chain needs to be re-configured. Standards, procedures and information technology infrastructure need to be developed or selected, adopted and maintained. This process can be hard to manage. It involves complex interdependencies between processes, data and technology and the interests of many stakeholders. Standardized reporting has four specific challenges:

1. Legal requirements: data definitions are determined by law.
2. Provenance: the origin and processing history, as well as the quality of different kinds of data need to be assured.
3. Process compliance: processes for handling report data are restricted by rules, such as archiving or due diligence, and
4. Multiple Stakeholders: setting up a reporting schedule needs involvement by many stakeholders and end-users.

Moreover, in general information technology projects in government often turn out to be more expensive than estimated, require more time than planned, and do not deliver intended results [5-7]. In the literature there is much knowledge about the redesign of processes and workflow, originating from approaches like Business Process Redesign [8] and Total Quality Management [9]. Both have a generic nature and are not focused on the public domain. To meet these challenges, therefore, specific management techniques are needed.

In this paper we present Public Process Management (PPM): a general method for process management and information systems development in the public sector. The method is based on existing practices, which have been developed over several
information system and process redesign projects in the Dutch government. See for example [10, 11]. We have documented this approach and apply it specifically to the introduction of standardized reporting in the Netherlands, and test whether it meets the above mentioned challenges.

The PPM method provides an iterative approach to the analysis, design, implementation, deployment, and maintenance phases of a development project, focusing on an architecture with a separate process layer, data layer, and technical infrastructure layer, as well as the governance issues. Deadlines are strictly maintained. Issues which cannot be addressed can be postponed till the next development round. Like in business process management, PPM makes extensive use of graphical modeling tools, in particular BPMN and BPEL for process modeling [12], as well as ontology and taxonomy description languages like XBRL for data modeling. Models are understood by both domain experts and system developers, and help to lessen the implementation gap. Detailed process models help to decide which tasks can be automated, which tasks need to be supported by information, and which tasks can be dropped. Moreover, a shared representation of the data, process and technology layers gives stakeholders an overview of the needs of other stakeholders, and an increased willingness to share required information. Explicit and shared models may also facilitate a predictable decision making process in the network.

From the experience of standards development for international trade facilitation at the United Nations it appeared that process modeling plays two important roles [4, 13]. First, process modeling is a prerequisite for developing a standard in the first place. Data definitions can only be reliably agreed on when the processes and decisions for which the data will be used are known. For example, the definition of income used to calculate taxes is different from that used to calculate social benefits. Second, process modeling is essential for harmonizing processes across different agencies to handle the reports. Standardization itself is again a prerequisite for process simplification and harmonization. As our research framework we adopt the inter-organizational systems (IOS) paradigm [14]. We focus on ways to manage the IOS development and adoption process.

The working hypothesis of PPM is that (1) strict adherence to an iterative development schedule with clear deadlines and deliverables, and (2) developing shared conceptual models of the process, data and technical infrastructure layers and governance agreements, will make the development, adoption and maintenance of standardized reporting schemes easier to manage and result in a more effective solution.

To illustrate the plausibility of this working hypothesis, we discuss a case study of the Standard Business Reporting program in the Netherlands. Standard Business Reporting (SBR) is a program to reduce administrative burden for businesses by providing a standardized data representation format (XBRL), shared semantics (Netherlands Taxonomy) and reporting platform (Digipoort) for filing official reports. The approach was developed in the Netherlands Taxonomy Project (NTP) and later adopted by Australia, after which it was re-branded SBR, see [11]. We study the development and adoption of the taxonomy and the technical infrastructure, but also the governance structure and management processes underlying current operations.

The paper is structured as follows. In Section 2 we characterize our current application domain for PPM: standardized reporting. Section 3 contains a detailed exposition of PPM. In Section 4 we describe the SBR case study.

2. STANDARDIZING REPORTING

Public process management is a set of practices, which have been developed over several years of implementing XBRL-based reporting solutions in the Netherlands. Currently, the method is used for financial applications, including agencies such as the tax office, bureau of statistics and chamber of commerce. In addition, a similar XBRL-based reporting platform is being developed for banking applications. However, there is no reason why XBRL could not be used in other application domains too, such as health care or food processing. In fact, a pilot project in the meat processing industry has recently been completed [15]. Here, XBRL reports could be used to continuously monitor compliance with food safety regulations. Despite the apparent diversity, there are common elements to these applications.

- **Reporting**: all applications are about reporting. A report is defined here as a formal statement that certain facts apply.
- **Compliance**: reports serve either as evidence or as declaration that the citizen or company is compliant with a business rule, guideline or legal regulation.
- **Actors**: citizens or companies filing reports, institutions (public or private) verifying compliance, and possibly intermediaries (consultants, accountants, bookkeepers) and software or technology infrastructure providers.

Institutions requesting reports can be regulating authorities (national bank), government agencies (tax office), professional organizations (chamber of commerce), or commercial parties (e.g. banks). Institutions can demand reports because they are supposed to offer a public service, i.e. to reduce risks to society concerning safety and security, or maintain financial stability [16].

A generic reporting situation is shown in Figure 1. On the left we find the company; on the right the institution. The company needs to report about primary processes, but typically this is done indirectly. The company collects data about primary processes for management and control purposes. For instance, when there is an incident, it must be analyzed, dealt with and resolved. This is an example of internal control. Aggregating over internal control information, the company can generate reports for demonstrating compliance to external stakeholders. There is a transfer of information from company to institution (the report), followed by a verification from the institution that the company fulfilled its reporting duty, possibly followed later by an official decision.

![Figure 1. Generic reporting situation](image)

This generic reporting situation may exist for many reports and institutions simultaneously. Simplification can be implemented in different ways. In each scenario the division of labour (decoupling point) between company and institution is shifted.

1. **One stop shop government**: various public services are delivered through a single entry point, sometimes called a single window, focused on the needs of the company or citizen [2]. Consider for example trade facilitation, such as the establishment of a single entry point for all cross-border regulatory paperwork (customs declarations, VAT report, veterinary check, etc.) [4].
However, in many settings the single-window concept is both legally and practically impossible. Legally, data collected for one purpose may not be used for another purpose without written consent of the data owner. Practically, the content of a report may differ because of the underlying purpose. In a tax report, a company will try to report as little revenue as possible within the rules. In the annual financial statement the company may want to show as much revenue as possible to please shareholders. Finally, the quality level of data may differ between reports. For example, for a statistics report the raw business data are sufficient, whereas annual financial statements must be consolidated and verified by an external accountant. The following scenario is more plausible.

2. Store once, report many. Generally there is a lot of overlap between the contents of various compliance reports. For instance, corporate tax declarations contain much the same information as the annual financial statements. Moreover, different reporting institutions require different formats, authentication mechanisms, or software support systems. Companies need to learn, install and maintain all these separate reporting mechanisms separately. Ideally a company will store the required information only once, using a shared data representation format and semantics, after which reports for many stakeholders can be generated on the basis of the same software system. Although the role of the company is larger than in scenario 1 or 3, it does have economic benefits to adopt this scenario. The data representation standard and underlying software support may be shared and reused. This will also eventually also reduce regulatory burden on the part of institutions. For businesses, further benefits may result from innovations based on standardization of financial services. How such innovations will affect the market is an open question.

3. Continuous Control Monitoring [3]. When the frequency of reports is increased, the report – as a document – is replaced by a continuous stream of control monitoring data, from company to institution. Provided that reliability of the reporting stream can be guaranteed by a system of internal control measures, continuous monitoring has the advantage that assurance can be provided continuously on the basis of current events, instead of incidentally on the basis of past events, as in the case of traditional auditing. There are two variants. The institution pulls the control data out of the company’s information systems directly (known as piggy-backing [4]), or alternatively it obtains them from a validated copy of the company’s information systems. Compare the architecture of [3]. The latter method has less legal and practical obstacles.

Wimmer [2] has also analysed the various stages of e-government projects. The more advanced the stage, the more complex the project. A project can provide (1) information about public services, (2) the possibility to contact people (communication), (3) the possibility to download and fill in forms to apply for public services (interaction or contracting), and (4) the possibility to handle a complete service (transaction). The reporting situation is of the fourth kind: a company completes a transaction, because it fulfils its legal duty to report. In particular when alternative reporting channels are discouraged, this means that reporting services need to fulfil the highest standards with respect to reliability, accessibility, usability and confidentiality.

Projects in the standardized reporting domain have some interesting challenges, which make it difficult to use more general project management techniques.

1. Legal Data Requirements. The contents required by compliance reports are determined by law. Therefore, data definitions most conform to legal requirements. This may hinder interoperability.

2. Provenance. The actual contents of reports may differ in quality level (verified or raw data), precision, origin, reliability level, aggregation level and processing history. Process Compliance. There are regulations about how processes of institutions must be conducted related to aspects like archiving, protection of copyright, good governance, legitimacy of decisions, audit trail, and so on.

3. Multiple Stakeholders. The reporting domain involves many public and private parties, possibly with diverging interests. No agency can build an information system by itself. Solving interdependencies often results in a tedious and political decision making process [17].

The Public Process Management method has developed in practice to meet these challenges, among others.

3. PUBLIC PROCESS MANAGEMENT

This section contains an exposition of the various techniques and practices which make up PPM. In general, it makes sense to partition a project into smaller parts. PPM has two partitioning principles. Firstly, development proceeds iteratively in phases with clear deadlines and deliverables: analysis and design, implementation, execution and monitoring. This provides a temporal partitioning. Secondly, there is a separation between the different layers of a projected solution: process layer, data layer, technological infrastructure, and governance aspects. We start by explaining the development phases.

3.1 Iterative Development Phases

PPM is designed to operate in a political arena, in which many stakeholders need to collaborate in a project, even though they may have diverging interests. In such circumstances, requirements creep is likely [18]. One must accept that some demands of some stakeholders will never be met, and settle for what is feasible given the limited budget and time [17]. In particular, it makes sense to enforce deadlines after relatively short intervals (two or three months), so that a sense of progress is maintained. This is called time-boxing. It is a well-established guiding principle of rapid software development [19, 20]. Under time pressure, disputes about what would be nice to have will be postponed and replaced by concerns about achieving the most needed functionality. Typically, a project like this develops incrementally over several implementation rounds. Requirements from stakeholders are prioritized according to the MoSCoW scheme: ‘Must’, ‘Should’, ‘Could’ and ‘Won’t have in this round, but would like to’. Remaining issues and additional requirements can then be scheduled for the next development round of the project.

The development phases themselves are fairly standard (Figure 2). They correspond to the traditional phases (or activities) of the engineering life cycle: analysis, design, implementation and testing, adoption (or deployment) and maintenance (and evaluation) [21]. Each phase or activity produces one or more deliverables. The analysis phase results in a requirements specification: what objectives should the IT solution fulfill? The design phase results in a blueprint of what the solution should look like. Here, many of the design choices have to be settled. The implementation phase results in a working solution, which conforms to the requirements and has been tested in a restricted setting. During the adoption phase, the solution is rolled out over all remaining application areas and taken into production. The maintenance and evaluation phase, finally, involves the endless process of monitoring performance, evaluating and improving the solution. Only during this stage the full benefits of a controlled solution will be reached.
3.2 Layers

Like in many approaches to Business Process Management [22] and Enterprise Architecture [23], the central philosophy of PPM is that redesign projects involve changes at several layers simultaneously. The relevant layers for the introduction of a shared reporting format are: processes, data, and technical infrastructure. Governance aspects relate to all layers (Table 1). Similar layer structures are used in enterprise architecture [23]. Another relevant analogy is with NORA 2.0, the standardized Dutch government ICT architecture, which distinguishes a business architecture (compare process layer), an information architecture (compare data layer) and a technical architecture (compare technical infrastructure layer).

| Processes: actors, activities, control flow, messages, coordination |
| Data: data model, taxonomy, inference, integrity constraints |
| Technical Infrastructure: network, storage, processing, message delivery |
| Governance: organizational structure, funding, requirements, SLAs |

A lower layer delivers functionality to be used by a higher layer. Conversely, a higher layer must rely on functionality of the layer below it, although the details may be hidden behind an interface. This is the well-known principle of abstraction. Layers are often interconnected. For example, for each choice-fork in a process analysis there are certain data elements in the data layer which determine the choice condition. Similarly, when a process analysis shows a message being exchanged between parties, this triggers the question what data elements this message should contain, and how this message should be securely delivered.

The following section describes each of the three layers and the governance aspects in more detail.

3.3 Process Layer

We start with the process layer. Thinking in terms of business processes became popular in the 1990’s with the advent of Business Process Reengineering [9, 24]. The idea was to re-design organizations using information technology, and focus on the added value of activities to some specific customer. In case of public processes, the customer will usually be the citizen, or some other recipient of the outcome of the process. We use the following description, taken from Davenport [24] p5: “a process is simply a structured, measured set of activities designed to produce a specific output for a particular customer or market”. To redesign processes, one must first analyze the existing situation (system-asis), derive the underlying objectives (including compliance) and measure current performance with respect to aspects like costs, through put, lead time and quality of outcomes. It is important to consider the actual way the work is being done; not how the work is supposed to be done. Once the current situation is known – to all stakeholders -- the process is redesigned to meet the objectives, but in a simpler way to improve effectiveness, efficiency, control and compliance. For instance, processes can be improved by reducing the number of steps, reducing the number of dependencies between steps so parts of the process can be executed in parallel, or by reducing the number of parties involved, thereby reducing overall complexity. The redesign produces a clear vision of the projected new situation (system-to-be). After that, the organization needs to change and gradually adopt the vision, and the corresponding information technology. Measuring initial performance is important, so that improvements during the change process can be made visible.

**BPMN Modeling**

In order to make detailed models of both the existing situation and the envisioned new situation, we make extensive use of business process modeling tools, in particular Business Process Modeling Notation (BPMN) and Business process Execution Language (BPEL) [12]. BPMN makes the work flow of a process explicit, indicating the order of activities, roles or actors who execute activities, choice points, parallel execution, message exchange and coordination. Given a BPMN model, it is relatively straightforward to analyze a business process. BPEL in turn makes it possible to execute process models in a simulation environment, for instance to test the system-to-be against requirements. In general, graphical models like these (e.g. also activity diagrams or interaction diagrams as in the Unified Modeling Language ) have been shown to be useful for business process documentation, and therefore facilitate knowledge sharing among stakeholders [25].

**Figure 3. Simplified BPMN model of Reporting Process**

Once a detailed BPMN model has been constructed, it can be used and analyzed in various ways.

- **Complexity** is determined, for instance by counting the number of steps, the number of actors involved, the number of choice points or the number of data items needed to make choices. In general process complexity is an indicator for other performance measures, like lead time, number errors, capacity etc. Brooks [26] distinguishes the essential complexity of a process, which must be faced, with accidental complexity, which can be reduced without loss of quality in the outcomes. Complexity reduction forms an important part of the business case of a process redesign.
Tasks, roles and information needs are determined. For each data element, one can for example specify in which activity the information is created, and in which activities it is subsequently used (create-use matrix). Such a matrix can also be adapted to register access control rights for users in different roles, or in different stages of the process.

Process Compliance is analyzed. Does the process as described adhere to rules and regulations? Consider for instance archiving. All messages, data used in decisions and outcome data, must be archived in such a way that the process can be traced back. This is called an audit trail. For example, in Figure 3 we count eleven activities, although some are composed of sub-activities. Choice points are hidden inside processes (validation, internal processing). There are exception handling links (indicated by an electricity sign), and time outs (indicated with a clock sign), which can be logged. Message exchange is indicated by a dashed line with an envelope sign. In this case, information needs that can be easily discovered are about the causes of exceptions and delays, and prerequisites for validation. Compliance with archiving regulations for the Gateway process can for example be verified, by making sure that the Provide Response step will always store the audit trail of a session into a reliable read-only device.

Kinds of Processes
In PPM processes are distinguished according to their functions (compare also Figure 1).

1. primary processes directly add value, i.e. help to achieve the main objectives of the organization
2. recording processes collect, register, store and retrieve evidence about the primary processes, for management, control and accountability purposes
3. control processes should detect, prevent and repair possible defects (not reaching objectives) in the primary process
4. accountability processes report to external stakeholders whether objectives are being met concerning primary processes, current situation and future expectations.

Typically these processes are executed by different departments within an organization (segregation of duties, see below). These processes are interlinked, but they can be executed in parallel, although in different time frames. For instance, accountability processes repeat on a quarterly basis, while control is active on a daily basis. Note that for some organizations, information processing is their primary business (e.g. tax office) while for others it is only secondary (e.g. meat processing firms).

In order to define recording and control processes, primary processes must be specified in terms of performance indicators. In management accounting one traditionally recognizes three different ways in which a process can be specified [24, 27].

1. behavior: specify sequence of activities to be executed,
2. outcome: specify the intended results of the activities, or
3. qualification: specify competencies for executing activities.

Separation of Duties
In accounting theory the principle of separation of duties is considered very important [28]. Some roles or functions in an organization may not be played by the same person. Separate functions create independent registrations, which can later be compared to test for accuracy and completeness of the records. In public processes, there is often a separation between the front office, which is supposed to support the citizen, and the back-office which is supposed to uphold rules and regulations. In information systems, separation of duties can be maintained by role-based access control (RBAC) [29]. Employees have a personal login, which is linked to the organizational roles they play and the tasks they are authorized to perform. This in turn determines which information they should have access to. In other words: organizational structure prescribes access control. A common security principle is that employees should only have the right to access or manipulate data when this is absolutely necessary for the execution of their tasks (principle of least privilege) [28]. This principle is opposed to the principle of transparency found in many open source initiatives where reliability is maintained by transparency, rather than restriction.

3.4 Data Layer
The essential component of the reporting situation sketched in Section 2, is the report itself: a message from citizen or company to regulator, stating that they are compliant with a specific rule or regulation, because certain facts hold about the business. The required content of such a message must be specified. Such a specification consists of representation format (syntax) and a specification of the meaning (semantics). In the reporting domain, it makes sense to use an open standard as a representation format. In PPM the default choice is XBRL, although other variants of XML can in principle also be used, such as for example HR-XML for the exchange of data about Human Resource management. For semantics the default choice is the Netherlands Taxonomy (NT).

Extensible Business Reporting Language (XBRL) is a platform-independent language based on Extensible Markup Language (XML) for formatting business information in a way that can be read across different applications [30]. The fundamental idea of XBRL is to allow for a separation of reporting facts from reporting meta-data [30, 31]. Facts are grouped and categorized by tags: labels which designate the beginning and end of data elements. For example, this data specifies that the net turnover is €12.030: &cbs-bedr:NetTurnover unitRef="U01_euro">12.030</cbsbedr:NetTurnover>. The intended meaning of the tagged values is specified by means of so called meta-data: data about data. Taken together all meta-data forms a so called taxonomy. The connection is made by means of link-bases: documents specifying typed links between the referenced elements in XML or XBRL documents. In practice these links look like URLs referring to a shared definition repository. In our example, we have defined a namespace, cbs-bedr, to refer to the taxonomy at http://www.nl-taxonomie.nl/2010/domein/cbs/bedrijven/cbs-bedrijven. (NT part about Statistics reports for companies).

A specific piece of XBRL with data is called an instance. An instance is composed of various parts (Figure 5). In addition to data (e.g. numerical values or text fields), reference is made to the concepts or attributes which are specified (e.g. net turnover), the context (e.g. entity number and year to which the data refer), the currency (e.g. Euro), and additional footnotes or comments. Also on the conceptual level, a concept (meaning of a tag) is composed of various parts: the taxonomy providing the definitions of the concepts contains legal references to laws or standards motivating the concept definition, calculations or rules, so called dimensions, and a human readable label, possible in various languages. Calculations and formulas (rules) are used to specify business rules, which apply to the data in a XBRL instance. They are ideal for specifying and automatically verifying formal reconciliation relations, as used in accounting. For example, the summation of turnover calculated over October, November and December separately, should equal the net turnover calculated over the final quarter. If not, someone made an error and the XBRL instance should not be accepted as valid.
An instance contains basic data elements. These can be grouped, structured and presented in different ways, based on the presentation format chosen. This mechanism can be compared to style sheets or DTDs in HTML. Data in a single instance can in principle be used to generate different reports. For example, it makes sense to present financial data to human users by a traditional T-shape balance mirroring debit and credit.

![XBRL Instance](image)

**Figure 5. Example of the Structure of an XBRL Instance**

Generating an XBRL instance from given accounting data is relatively straightforward. Many commonly used accounting software applications are already adapted for XBRL. Writing or adapting a taxonomy is more complex. Taxonomies are written by experts using special tools. In the case of XBRL, experts must have a background in accounting or taxation, in addition to understanding the use of meta-data and taxonomies. Because a taxonomy is essentially a large set of links (linkbase) a tool is needed to relate the various cross references and keep an overview. A list of tools is maintained by XBRL International (http://www.xbrl.org/Tools/). A particularly important aspect of developing taxonomies is the traceability of the data definitions. For example, the way in which net turnover or profit should be calculated for a statistics report is specified by an administrative decision (Besluit Gegevensverwerving CBS, 2003). The way in which net turnover or profit should be calculated for a tax report may differ. Therefore the taxonomy itself should make reference to the authority from which the definition promulgates.

XBRL is an extensible standard. When required, the standard can be enhanced with additional definitions for specific domains. For example, those aspects of the Netherlands Taxonomy which relate specifically to statistics reporting are addressed in a specific section of the taxonomy, which is maintained locally, by CBS. This ability to extend the taxonomy may seem like a great benefit, because it allows one to develop definitions locally, where expertise is located. However, allowing extensions is a danger too: they damage the universal applicability of a standard. A standard is useful precisely because it removes the need of individual users to adjust to different circumstances. When taxonomies are extended for new domains, the individual user will not be able to follow and use all these extensions. Therefore some central form of governance and coordination is necessary (see Section 3.6) For more about the trade-off between extension and standard, we refer to [32].

### 3.5 Technical Infrastructure Layer

The main component of the technical infrastructure layer is the Gateway, which serves as an interface between citizen or company and regulator. In doing so it provides a public service. The technical infrastructure layer should seamlessly integrate the data and process layers. It should provide mechanisms for users to identify and authenticate themselves, to establish a secure connection, to provide reports and store them in a dependable way, and to receive a response whether the report is acceptable and validated according to the XBRL syntax.

A gateway infrastructure is a way of exchanging information and must therefore meet certain requirements [33]. Here we mention: confidentiality, integrity, authentication, non-repudiation, availability, reliability, scalability, adaptability, maintainability and usability. Confidentiality means that reports should not be accessible to any person, apart from the person filing the report and the people authorized by the regulator to handle the report. Integrity means that the report should be delivered in exactly the same state as it was filed. Authentication means that the sender of a report must be uniquely identified (e.g. by the chamber of commerce number) and authenticated (e.g. by encryption key). Once a report has been delivered the sender should not be able to deny having sent it (non-repudiation). Availability is crucial for a service on which citizens depend: the service should be accessible at all times except for previously announced service windows. Reliability (dependability) means that the system can be relied upon: it should behave the same in similar circumstances. Finally, the system should be relatively easy to use and understand in order to not discourage potential users.

A common way to develop a gateway infrastructure is by a Service Oriented Architecture [34, 35]. SOA represents a style for configuring and implementing information system architectures. A characteristic of SOA is the idea of wrapping functions provided by different applications as individual services and preparing them for multiple use. The services correspond to building blocks, which can represent various functions. Among the various possibilities of implementing a SOA, using XML in the form of Web services is most common [36]. Like XBRL, the main components of Web services are based on XML.

The World Wide Web Consortium (W3C) suggests that three components are needed for the implementation of web services. Firstly, a suitable carrier or packaging protocol is required to enable the exchange of messages between applications. This is accomplished using the Simple Object Access Protocol (SOAP). SOAP prescribes the structure of a message, and organizes the possible function calls between applications [34]. Underneath, SOAP employs various transport protocols, for example http, ftp and smtp. Secondly, to use Web services, access to the service must be ensured by an interface description. It is necessary to specify which methods and functions are part of a Web Service, so a call can be processed and answered. For this purpose the W3C developed one of the most established XML standards, the Web services Description Language (WSDL) [35]. Third, a directory service for finding relevant Web services is required. The Universal Description, Discovery and Integration (UDDI) standard is such a building block. Web services are registered in an UDDI with the characteristics of the services offered [34].
3.6 Governance aspects across the layers

SBR is more than technology. It is a set of agreements about standards and procedures. For this reason, there must be clear ways of making sure the agreements remain up to date. IT governance refers to the “patterns of authority for key IT activities” [37 p. 261]. Weill and Ross [38] identify three main governance mechanisms.

1. decision-making structures: organizational committees and roles that locate decision-making responsibilities [29].
2. alignment processes: techniques for securing widespread and effective involvement and.
3. formal communications: two-way communication and a good participation/collaboration relationship

Consider for example updates of a taxonomy. Because the circumstances change and because laws and regulations are frequently altered, the taxonomy must also be regularly updated. How can the update process best be organized? Given the Weill and Ross model, we have to decide up-front: who has the authority to decide about changing the taxonomy, how to make sure updates are aligned with internal processes of the demanding stakeholders and changes do not have irreparable consequences, and how can the new version of the taxonomy be announced to end-users?

Regarding decision making and responsibilities: often it makes sense to have a steering committee with representatives of the gateway infrastructure, the stakeholders, intermediaries and end-users. A steering committee has the authority to make changes and should guard uniformity of the standard.

Note that governance aspects change during development. Initially, in the analysis and design phases, the process is best seen as a project, with a clear end-point: delivery of a taxonomy. But after the taxonomy has been taken into use, its maintenance is better characterized as a standardized work-flow. Therefore it makes sense to end its project character and put it under responsibility of the ‘line’ organization, including the budget.

4. STANDARD BUSINESS REPORTING

In the Netherlands, the Standard Business Reporting Program (SBR Program) is a set of innovative projects in the area of business to government information exchange. In the SBR Program several government agencies and industry partners collaborate to simplify and standardize (financial) reporting. This collaboration is enshrined in an agreement (covenant) that was signed by over eighty parties, both public and private.

4.1 Data Collection

The research is still in an exploratory phase, so we use a case study method [39], rather than questionnaires. The purpose is to identify potential success factors for information system development and process redesign projects in government, especially concerning compliance. In particular we want to evaluate the use of iterative development and strict time-boxing, and the focus on specific architecture layers, i.e. working hypotheses (1) and (2). Data for this case study was collected by various means. We have interviewed and closely examined many of the issues in the paper with leading experts on the SBR program in the Netherlands. In addition we used publicly available information about the SBR program and Netherlands Taxonomy (see http://www.sbr.nl.nl). Results were validated against earlier research on SBR [10], which was conducted independently.

4.2 Case Description

The program started in 2004 as Netherlands Taxonomy Project (NTP). In 2006, a generic infrastructure project was carried out drawing up requirements for the a new process infrastructure for financial reporting based on XBRL. In 2007 the first versions of the technical infrastructure developed for exchanging the data were ready. Stakeholders decided that the technical infrastructure should be maintained by the government IT maintenance agency Logius (previously GBO.overheid). In 2009 the taxonomy project was handed over to Logius altogether and a steering group consisting of senior representatives of all Ministries involved was appointed. As of 2009, NTP continues under the international name Standard Business Reporting (SBR). Similar approaches have been adopted by Australia, and later also New Zealand, China and Singapore.

Figure 6 gives an overview of the SBR landscape. On the left we find a company, and possibly an intermediary (accountant, bookkeeping, tax consultan etc), who are both supported by software providers. In the middle we find the various taxonomy variants chosen for the different reporting streams, and the gate ways. Institutions demanding reports are shown on the right.

SBR has chosen the ‘store once, report many’-scenario. That means that although the data definitions and the infrastructure may be re-used over different reporting chains, the actual act of reporting remains specifically addressed to one agency. The one-stop-shop scenario (single window) or the continuous monitoring scenario would be too far reaching. Firstly because it is legally not allowed to re-use data collected for one purpose, for different purposes. Secondly, because reports may have a different function and may therefore have different contents. Thirdly, because data for different report may have a different quality level, aggregation level, precision or source.

The figure shows the core SBR program with reporting streams for CBS (production statistics, investment statistics and short term statistics, i.e. revenue per period), Chamber of Commerce (possibility to file the annual financial report) and Tax Office (revenue taxes (OB), corporate taxes (VpB), income taxes (IB), intra-EU performance (ICP), and short versions of VpB and IB).

In addition to the core SBR program, there are similar initiatives developed at UWV and several large banks. UWV is the government agency responsible for social benefits. They have a project to standardize the sick-leave reporting process. Employers must announce to UWV when an employee reports ill at work, and when they have recovered. This project does not use XBRL.
but HR-XML, a variant of XML for Human Resource management. UWV will use the same technical infrastructure (Digipoort) as the core SBR program. There is also a project related to Electronic Purchasing and Invoicing, which should streamline purchasing at a national scale. This project makes use of electronic invoices represented in XBRL. All invoices which are being sent to a national agency can currently be presented in XBRL. Finally, a consortium of banks are involved in a similar project to standardize the loan approval process. They have chosen to develop their own gateway, called Rapportage Portaal. The banks have developed an extension of the NT, but have agreed to follow all updates and releases of the official NT, so efficient reuse does remain possible (in principle). The technical infrastructure for identification and authentication of business parties is currently being streamlined in a project called E-herkenning (E-recognition), which should in the long run replace and standardize existing identification and authentication solutions. This is not listed in the figure.

4.3 Issues and Dilemmas
Below we will discuss several issues and dilemmas in the context of the development phases and layers, and how they have been dealt with. This provides interesting insights in SBR.

Development Phases
The development phases are enforced quite strictly. Figure 7 shows a development schedule as it has been used in several roll-out projects in the SBR domain. There are two crucial go/no go decisions. One after the analysis and design phase, when commitment is needed that the project will go ahead as specified in the blueprint. Note that analysis and design are merged. This does not mean that a requirements specification (analysis) and a design (blueprint) should not be separate deliverables, but rather that determining requirements and developing ideas about what is feasible should be intertwined. Another reason is that these phases involve similarly skilled people: visionaries and architects, with an eye for unforeseen possibilities. By contrast, the implementation phase needs project managers who get the job done. In the third phase the implemented process and technology components are deployed in practice. Initially this is done in a small application area. Only after evaluation and acceptance of the working solution, possibly after alterations to the analysis and design documents, and with an enriched business case, a roadmap can be drawn up to scale up deployment in other application areas. This also involves a marketing plan to make sure external parties (companies, intermediaries) will adopt the new way of reporting.

Figure 7. Development schedule chart in use within SBR

Process layer
We discuss a dilemma concerning feedback reports. Consider once again Figure 3. Suppose you have just filed a tax report for revenue tax. Shortly thereafter you receive a feedback response from Digipoort that the message was received in order. You conclude that you have satisfied your legal duty to report. However, later the tax office discovers that the tax report you submitted is faulty. Although it was validated as correct XBRL, economically it makes no sense. For instance the revenue reported was 10 times as much as could be expected from other key data, like sales and salary costs. You will have to re-submit. But what if the fault only appears much later? Can we expect a company to debug and re-submit a tax declaration concerning data which are no longer current? And what about the feedback response? The response is sent by a technological infrastructure, Digipoort, on the basis of syntactic validation. What is its legal status? Giving no response at all would be bad from a usability point of view: users want to know if they succeeded in filing a report. But a response appears to come from the tax office directly.

Data layer
Concerning the data layer we discuss the dilemma of allowing extensions, versus uniformity of a standard (Section 3.4). As was explained above, the general policy in the SBR program is to prefer the NT taxonomy, but to also allow other open standard data formats for specific domains (UBL, HR-XML). In the case of the banks, an intermediate solution is chosen. Banks use their own extension of the taxonomy, but in the release schedule they follow updates of the NT. Therefore users can still expect to be able to re-use the common data part.

Another issue concerns the possibility of XBRL generating different reports from the same data, by using presentation formats. This leads to a legal problem. By law, an accountant verifies whether the annual financial statements present a ‘fair image’ of commercial reality. When the metaphor of an image is taken too literally, this means that the accountant can only take responsibility for the reliability of a document in its actual presentation; not for the underlying data elements. After all, presentation formats could leave some relevant data elements out, and not only accuracy but also completeness of the reported data are testified by the accountant. This issue still needs to be settled by experts of the Dutch accountants association Royal NIVRA.

Technical Infrastructure layer
In the SBR program, the role of gateway is played by Digipoort. SBR uses open technology standards were possible, like XBRL, XML, SOAP, WSDL and UDDI.

An interesting dilemma concerns authentication. Originally, intermediaries who were authorized to file reports on behalf of their clients, needed to prove this using a complicated authentication procedure, with credentials provided by commercial Authorization Service Providers (AuSP). However, this process is complicated and difficult to understand for end-users. Also, AuSPs are commercial parties who charge a fee for the authentication service. This has lead to resistance among end-users. Recently therefore, the user platform decided to drop the AuSP requirement in favour of usability and accessibility. Compensating measures exist, because all parties logging into Digipoort must already authenticate with usual PKI government credentials; the report itself identifies the party.

Table 2. Annual release schedule of taxonomy versions

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Available for</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st of May</td>
<td>Alpha</td>
<td>Taxonomy partners</td>
</tr>
<tr>
<td>1st of November</td>
<td>Beta</td>
<td>Taxonomy partners</td>
</tr>
<tr>
<td>15th of November</td>
<td>Beta</td>
<td>Market parties</td>
</tr>
<tr>
<td>1st of December</td>
<td>Version 20xx</td>
<td>All parties</td>
</tr>
</tbody>
</table>


**Governance issues**

A strict release schedule is maintained for different stakeholders. In this way, partners can test and use the taxonomy – so possible defects are found – before troubling market parties (Table 2).

According to the Weill and Ross model of information technology governance we need to determine three things. First, responsibilities are laid down in the following organization chart (Figure 8). SBR is governed by a council, in which all major stakeholders have a say. User groups are represented in the SBR platform. They can give feedback on the way the program develops. The platform is supported by three expert groups, one for data, one for processes and technology and one for marketing and communications. Expert groups are meant to initiate, discuss and solve current issues. This structure ensures that all major stakeholders have a say, while also guaranteeing enough expertise to reach workable solutions. Second, we need to ensure alignment among stakeholders. The actors in the SBR domain form a network, who share a data standard (NT) and provide a service: an information processing chain for reporting applications. Therefore there are frequent meetings (e.g. platform meetings; expert group meetings) to make sure parties know of reported issues and scheduled changes. Regarding adoption by end users, a professional marketing and communications plan is maintained. Third, formal communication procedures must be strictly followed. For example, before releasing a new version of the taxonomy, it must be tested by all stakeholders. Now suppose one party did not perform the test and the release has to be postponed. This needs to be communicated in a uniform way.

![Figure 8. Organization structure of taxonomy management](image)

**4.4 Discussion**

Now that we have seen the development phases and layers, we are in a position to review how PPM deals with the challenges we identified in Section 2 of the case of SBR.

1. **Legal requirements.** Legal requirements and additional requirements from stakeholders are dealt with in the data layer. They are part of taxonomy definitions themselves. Changes to these definitions are carefully managed by a release schedule, and frequent meetings (expert meetings, platform meetings).

2. **Provenance.** Because SBR has chosen the ‘store once, report many’ approach, the result being differences in content and quality of information are no problem. Users still prepare each report as a case study of the SBR program in the Netherlands. In particular, we explained that several combinations of the XBRL format, the Netherlands Taxonomy or its extensions and the Digipoort infrastructure can be reused in different configurations. Also we have shown that a clear organizational structure and governance procedures are necessary.

An interesting observation concerns the balance between restrictive and flexible project management strategies. In a political networked environment, project management methods like PPM need to balance between restricting participants’ behavior and providing flexibility for participants to develop solutions which fit their needs [17, 40]. For example, time-boxing is restrictive; it fixes the duration of a project phase. Therefore other variables, such as the content to be provided or the priorities of issues must remain flexible. Similarly, the possibility to develop an extension to the Netherlands Taxonomy, as in the case of the banking portal, provides flexibility in the data layer. That is only possible given the fact the banks have restricted themselves to the duration of a project phase.

As a general conclusion, we could say that although none of the techniques which make up PPM are by themselves very new or innovative, the fact that these concepts are used together in a cohesive way is novel, at least in government information technology projects. Reports have shown that the successful adoption of XBRL and the objective of reducing administrative burden, is still a long way ahead [10]. Mistakes have been made. For instance, an initial focus on technology while neglecting business models for adoption, may have lead to a delay. But the PPM methods we have reported on here, demonstrate that there are ways of dealing with these issues. In the end, best practices are just that: practices which have been shown to be useful. We hope this paper will help others avoid some of the pitfalls and mistakes that are inevitably made in a complex project like this.
REFERENCES