

# Workflow Modeling Language Evaluation for an Archival Environment

Stephan Heuscher, ikeep Ltd.  
Bern, Switzerland  
stephan.heuscher@ikeep.com

18th April 2005

## Abstract

Although archives today are more and more urged to preserve digital data, the processes in archives are still predominantly bound to the handling of traditional paper-based records and strongly depend on implicit knowledge, i.e., the processes through which records are being transferred to and stored and maintained in the archives are only roughly sketched and rarely explicitly documented (a few exceptions like, e.g., paper de-acidation or climate conditions in storerooms, aside). This is not a problem with paper-bound records, because the goal of preserving the records' integrity, originality, and authenticity, is achieved by keeping carrier or substrate of the information (e.g. paper, polyester film etc.) physically and chemically as inert as possible. As digital records do not rely on a specific carrier or substrate whose intactness or inertness could guarantee integrity, originality, and authenticity, but only on the information content itself as well as the description of the formal structural aspects of its coding (like, e.g., data formats, data semantics, rendering techniques etc.), which are subject to change. These changes must be unambiguously defined, documented, and included into the description of the record. To allow for this, the processes first have to be described. This paper addresses the question of how to best describe an archival process by taking into account the main archival paradigms outlined. We conclude that a language having high-level Petri net semantics best meets the requirements of an archival workflow language.

## 1 Introduction

Workflows are ubiquitous in our days. They are used to describe and formalize the business processes to make them repeatable, auditable, and automatically executable. Archives have always had processes when receiving, handling, restoring, or making available records and documents given into their custody. The records and documents created in these processes should be kept and conserved like other holdings of the archives to enable a future researcher to better recreate the original form and environment of the records and documents (for more information on the purpose of archives, refer to e.g. Keller 2004).

While the archival processes were conducted using paper-based reporting, because the processes and personnel rarely changed and holdings needed little attention, there was no apparent need to formalize these processes, although there were some attempts (Ministerrat der Deutschen Demokratischen Republik (1984)). As more and more records and documents are born-digital and archives integrate them into their holdings, change has become a constant. The processes depend more on the technology used and they change frequently. This might be due to the rapid advances in processing, storage, and communication inherent to information technology or even merely an upgrade of a system component. At the same time, the holdings have become heavily dependent on their infrastructure and they themselves need to be treated (i.e. checked, copied, converted, migrated etc) more often to adapt to the aforementioned changes. Preservation in the earlier sense meant that the

records or documents should be kept as original as possible on their original carrier, while the approach when archiving digital records and documents is to keep them intelligible and accessible using the current technologies. The “computer museum” approach featuring the collection of old hardware and software to preserve documents and records in their original form and environment, has been dismissed for obvious financial, space, personnel, and maintenance reasons.

Digital documents and records are fragile (Rothenberg, 1995: “digital information lasts forever - or five years, whichever comes first”). One flipped bit could render useless a document and all documents depending on it. The fragility is not only one of bits, but also lies in the need for the bits to be interpreted to make them meaningful and understandable to humans. It is impossible for a single human today, to comprehend all the transformations of digital data. E.g. the task of “simple” home computer displaying a single character stored on its hard-disk cannot be described in this paper. To show the complexity of the task, we will describe just a few steps in the processing chain. First the representation of a character is read from the hard-disk, processed by the controller on the hard-disk, sent to the controller on the motherboard, then sent over the PCI (Peripheral Component Interconnect) bus into the main memory, where it is rendered (e.g. to a raster format) and sent to the graphics card. Then it is transformed into analog signals, which are digitized and sent to the LCD (Liquid Crystal Display) controller so that the character is displayed on the monitor. We only described the process in low-level, overly simplified steps to show that even in these basics over-strain the technical knowledge of more than 99% of the people. In the digital world, it is not possible to reason by physical analogy, as e.g. on a shellac record, the depth of the groove is equivalent to the amplitude of the membrane of a loudspeaker.<sup>1</sup>

---

<sup>1</sup>As an example of something more intuitively accessible to humans, we will use a shellac recording. To render the recording only a needle and a postcard are needed. If you stick the needle into the postcard, spin the record, and put the needle in groove, you will hear the recording because of a simple physical analogy: The depth of the groove represents the amplitude of the microphone that recorded the sound. In practice, the changing depth of the groove leads to the vibration of needle, which is then amplified by the postcard, the sound being transmitted to the air and thus to the ear

An additional complexity lies in the need of the computer to interpret or render digital information to make it accessible to humans. The specification of these interpretations of digital files and their formats and these implementations can be complex, ambiguous, undefined, or even erroneous. This is the reason why some archives impose restrictions on the type of files they accept. The archives try to keep the complexity at a minimal (manageable) level because it is up to them to keep the contents of the transferred files understandable and usable. Some file formats allow for many forms of content, e.g. the portable document format (PDF) can contain scripts, images, sound, and video, which makes files of this format very difficult to render in their original form, as the format of all encapsulated objects must also be known and the possibility to interpret them must be ensured into the undefined future (Adobe Systems Incorporated, 2003). Looking at the many formats the computer revolution produced and that are no longer in use and the exponential growth in formats in use that can not be made intelligible to humans any more, this is a task no archives can accomplish. So the formats the archives support to keep their holdings interpretable, need to be restricted.

For the archives to demonstrate that the facts mentioned above have been considered and the corresponding actions have been taken, the processes in archives holding digital material have to be formalized and themselves preserved. The vast amount of electronic records is only manageable with a high degree of automation, so the conclusion to be drawn is that the high degree of formalization facilitates the automation of the processes with computers; it is even a prerequisite!

The main questions discussed in this paper are: “Are there special considerations when modeling archival workflows?” and “How should archival workflows be modeled?”. It is clear that the answer to the first question has an influence on the answer of the second, as the considerations identified therein serve as basis for the grounds of the result.

The remainder of this paper is structured as follows. Section 2 outlines the archival principles that must be considered in the specification of a

---

of the listener. With digital data, there is always the need for digital components; there is never a direct physical path.

modeling and identifies the rating principles (with focus on archival workflows). Section 3 provides three instances of workflows on which the modeling languages will be tested. Section 4 provides an overview over the possible modeling languages for workflows and a rating of these languages according to the rating principles. In Section 5 we summarize the results and draw the conclusion. Section 6 outlines future work.

## 2 Archival Paradigms

In this Section we will establish a basis by describing the main archival paradigms and inspecting them from a workflow-modeling viewpoint. The paradigms are based on those presented by Gilliland-Swetland (2000).

### 2.1 The Sanctity of Evidence

The preservation of the evidentiary value of records and their associated context is the creed of every archivist (Jenkinson 1947). Records passively document the events that led to their creation. The evidence is intrinsically embedded in the records and their context and may not be immediately recognized by the record creators, nor the archives keeping them in their custody. For this reason, an unbroken chain of custody and a precise documentation of the changes to the records is of such paramount importance. Only then, the integrity of the evidential values can be assessed retrospectively. Because, in contradiction to the integrity of records within archives, every alteration of the record or its context may embed new or erase old evidence, and if there is no documentation of the treatment of the record, the original evidence is impossible to ascertain.

Traditionally, archives were trusted to keep their holdings authentic mainly because they were archives. The idea that an archives must legitimate its actions is relatively new. Even the ISAD(G) standard for archival description in its first version (1996) had not deemed necessary means of describing how and by whom the changes in the description are made. This shortcoming was corrected in the second version (International Council on Archives 1999) of the standard.

However, the constant advances in information technology eliminate the need for transferring records away from their creators as they can continue to reside in the information systems at little extra cost and free of the storage problems inherent to paper records. This has led to a new, additional model for the record life cycle, the records continuum model. Here, the records' creators guarantee the integrity of the records over a much longer period of time. The archive's role in this model is much more active. They are now involved in the record-creating process even before the records are created, they aid in designing the record-creating system. Moreover, they define requirements for that system and monitor the compliance in operation in order to guarantee the integrity of the records. For digital records, the problem is especially acute if the archives choose to implement a migration strategy for ensuring the long term intelligibility of their digital holdings. This strategy calls for a conscious and repeated violation of the integrity on the bit level and will therefore render useless any method for ensuring authenticity based on the bits. We are not aware of a technical mechanism that ensures authenticity on another level than the bit level. Therefore, the archives that performed this violation of integrity must provide proof of the authenticity of the migrated records. A result of this strategy is the shift from original to archival authenticity, meaning that the original files are no longer available for authenticity assessment, only the files and audit information generated by the archives. If an archives chooses emulation as preservation strategy, the authenticity problem is passed down to the emulator.

Much research has gone into the question of how to determine and ensure the authenticity of electronic records, e.g., InterPARES (2002) and Cullen et al. (2000), there are no definite answers to this problem. It all comes down to trust in records and processes and their documentation (Cullen et al., 2000, pp. 32-50).

From a workflow-modeling viewpoint this means that the design and enactment of the workflow has to have precise and unambiguous semantics. To enable retrospective evaluation of the intrinsic elements of the original record, all alterations of the records or context must be recorded and preserved together with the records.

## 2.2 Provenance and Original Order

These principles focus on preserving the context and internal structure of groups of records and form the core of archival theory and practice. The principle of provenance states that the records of one provenance (i.e. creating agency) should not be mixed with records from another provenance. This principle, first described in the early 19th century, was adopted by most archives until the end of that century. The principle of original order dictated that structure of the records were to be kept, in the order they were maintained in when they were active, even after transfer to the archives.

The aim of these principles is to preserve the context of and evidence within the records and their relation to each other. The search within the records is simpler for the agency, as it has not changed; the researcher will find the records and those closely related to them more easily. Additionally, these principles allow an archives to handle massive amounts of records, because their transfer to the archives consists only for transport and storage and does not include rearrangement or other intellectual decisions and therefore only allocates little resources.

We did not name the principle of respect des fonds as an archival principle in this Section because it is not clearly distinguishable from the other principles in the literature (see Rousseau and Couture (1994, p. 31 and pp. 61-68), Papritz (1998, pp. 8-27), Schellenberg (1961, p. 14), and Schellenberg (1965, p. 41 and p. 90)).

From a workflow-modeling viewpoint, these principles do not easily translate into requirements. The merely underline the importance and of documenting and defining the priori and posteriori contents and structure of the records, i.e. the input and output of the workflow Papritz (1998, pp. 82-91).

## 2.3 The Life Cycle of Records

The life cycle of records models the functions of use of and responsibility for records. In the traditional archival perspective, every record's life has three stages. First, in the active stage, the records is being created and modified by the record-creating body. Second, in the semi-active stage, it is used as a reference for the documented actions, it is stable (i.e. it will not be modified). Third, in the inac-

tive stage, the record is no longer easily accessible by its creators (i.e. transferred to the archives or destroyed).

From a workflow-modeling viewpoint this means that the model must not be restricted to archival use, but must also be flexible enough to accommodate more generic functions needed in the workflow design outside of digital archives. The modeling language (or representation thereof) must also be simple enough to be understood by non-expert personnel. Because the IT environment of record-creators cannot be controlled, the modeling language must not be tied to a specific workflow engine or tool.

## 2.4 The Organic Nature of Records

Records are the byproduct of an activity of an organization or individual. The relationships between records, their creators and their legal, historical, and procedural context is complex and hard to preserve. The coherence of records with each other is determined by the activity leading to the creation and the body responsible for its creation. Archival practices exploit the organic nature of records by ensuring that records with a strong coherence are collectively captured and documented. This ensures that the legal, historical, and procedural contexts can be re-established much more easily than by the analysis of singular records.

This does not translate into workflow-modeling requirements directly.

## 2.5 Hierarchy in Records and Their Descriptions

Records are embedded in a hierarchy firstly defined by the position of the record-creating agency and secondly by its filing practices. Archival description reflects these hierarchies and describes the sets of records defining and providing sufficient information on each level of the hierarchy. The definition of this "sufficient" information may vary depending on the archives' perceived needs for intellectual control, collection management, and the foreseen user interest.

This approach yields many advantages, namely:

- It allows an intuitive access to information, specifically information not easily located by

subject or keyword (i.e., most archives materials) (Papritz, 1998, p. 185).

- It permits economies in description. The depth to which the records are described can be adapted to the foreseen use and current workload of the archives.
- It can be used on any types of collections and materials. No specialized knowledge is needed to understand the archival descriptions of special forms of materials (e.g. maps, scripts, or video tapes).

This does not translate into workflow-modeling requirements directly, however, it shows the familiarity with the concept of hierarchical structures.

## 2.6 Rating Criteria

The following Subsections provide the criteria by which the workflow modeling languages will be rated. The criteria summarize the discussion in Section 1 and the prior Subsections.

### 2.6.1 Formal Semantics

To enable the unambiguous design and enactment of a workflow, its semantics have to be well defined. The modeling language has to have formal semantics. The meaning of textually defined semantics will change over time. Therefore, we believe that only a language defined in mathematical terms can ensure its consistent interpretation for the long term.

This criterion is the direct interpretation of the sanctity of evidence, because a formal basis of the modeling language minimizes the need for elaborate descriptions in textual form, thereby lowering the margin of error.

### 2.6.2 Flexibility

Archival processes change over time. The modeling language has to demonstrate that it can be easily be extended to meet future, yet unknown needs. Preferably, it should already have demonstrated this ability in the course of its existence. Additionally, not every aspect of the workflow can be modeled, but some special cases may need an ad hoc extension (Ministerrat der Deutschen Demokratischen Republik (1984, p. 15)). As an example, an

archives could receive a CD-ROM full of WordPerfect files, even though the workflow expects Tiff files. The archives may opt to transform these files into Tiff files before continuing with the workflow. This decision and the conversion would have to be logged in the process, even if the conversion step does not occur in the designed workflow.

This criterion roots in the life cycle of records. Because the environment of the workflow cannot be controlled outside the archives, the modeling language must be easily adapted to new environments and uses. This criterion also ensures that the modeling language will not be subject to change on new requirements.

### 2.6.3 Decomposition

It should be possible to design workflows in building blocks and then connect them vertically and horizontally. Vertical decomposition allows the definition of parts of independent workflows that can be connected at designated points. Hierarchical decomposition means that a workflow can be defined at different levels of complexity. On the top level, a rough sketch of the workflow steps, while on lower levels, the interactions of the systems and technical interfaces need to be specified in detail. Both of these decompositions are useful to reduce the clutter in the diagrams, i.e. to keep the workflows comprehensible to humans, to allow for the reuse of previously defined and tested building blocks, and to speed up the workflow development process (parallel development of building blocks).

This criterion is derived from the hierarchy in records and their descriptions. Archivist are used to the top-down approach in describing and finding records in a hierarchical structure because they deal with it on a day to day basis. Hence, it is natural to begin with defining the task and then refining it on multiple levels of detail.

### 2.6.4 Simplicity

The concepts and the implementation of the concepts comprised in the modeling language have to be easy to learn and communicate. The basic principles should be grasped intuitively.

This criterion derives from a requirement stated in the life cycle of records. Because the modeling language will need to be understood by non-

experts, simplicity is a key factor for the acceptance of the language at the record-creating agency. Additionally, a modeling language based on easily understandable principles is better suited for long-term use, as simple principles are less prone to misinterpretation due to loss of subtleties in definitions.

### 2.6.5 Stability

To be suitable for long-term use, the modeling language must demonstrate a high degree of stability. This means that it should be mature and in active development and use. Additionally, the language should have backwards-compatibility, i.e. the interpretation of models created using an older version of the language must remain unchanged even when using a newer version of the language.

This criterion is derived from the general requirements towards long-term preservation. Frequent changes in the syntax and semantics of the ingested materials lead to an exponential effort in the data management of the digital archives, as all archived versions need to be kept in order to preserve the original evidence within.

### 2.6.6 Retraceability

The workflows should generate audit trails to enable their retraceability. Although this is a low-level requirement (defined in International Council on Archives (1996)), it is of major importance in the actual implementation and the archival value of the digital records and documents processed. This criterion is listed because features in modeling languages may aid or hinder in this respect.

## 3 Archival Workflows

This Section defines two examples of hierarchically connected workflows within digital archives to test the suitability of the language. Since we have found no consensus on standardizing archival workflows besides Consultative Committee for Space Data Systems (2002*b,a*), two connected workflows on different hierarchical levels were taken from those (forthcoming) standards.

In the following Subsections we will present example workflows from a digital archives. The workflows presented here should illustrate the most im-

portant requirements for archival workflows described above. The terminology and naming in this paper (“Ingest”, “Data Management”, “Archival Storage”, “Access”, etc.) follows ISO 14721:2003 (Space data and information transfer systems – Open archival information system – Reference model) that is equivalent to Consultative Committee for Space Data Systems (2002*b*).

The workflows proposed here are one possible definition of the workflow to achieve the goal of the workflow, i.e. ingest an SIP (Submission Information Package), that is, the transfer and accompanying checks of archival data from the Producer to the OAIS in the form of an SIP. Other workflows are also possible and may even be better suited in a given environment, but the ones shown here should provide a solid foundation for the comparison of the workflow modeling languages.

### 3.1 Ingest Workflow

The ingest workflow is defined by the Open Archival Information System (OAIS) by the Consultative Committee for Space Data Systems (2002*b*) at a conceptual level. Additionally, a separate standard for a more detailed specification of the producer - archive interaction is currently under development (Consultative Committee for Space Data Systems, 2002*a*). It is the result of agreements between the digital archives’ management and the producers. The main task of the ingest workflow is to integrate the records provided by the producer into the archives holdings.

This is achieved by first testing the SIP against the submission agreement, transforming it into an Archival Information Package (AIP), including the metadata generation for the package descriptions, followed by the transfer of the AIP into archival storage and that of the metadata to data management. The steps are as follows:

1. Producer delivers SIP
2. Ingest checks SIP against contract (validation)
3. Ingest creates AIP(s) and Package Description(s)
4. Ingest stores Package Description(s) (metadata) in Data Management
5. Ingest stores AIP(s) in Archival Storage

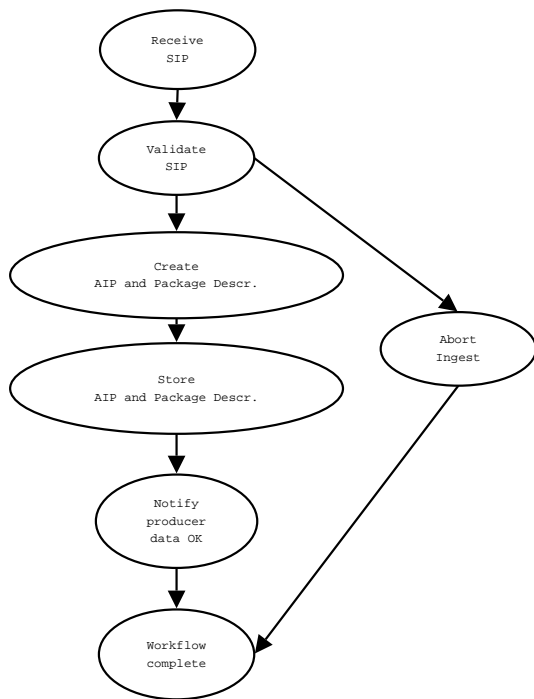


Figure 1: Ingest Workflow

6. Ingest notifies the producer, that the data has been stored and that the may delete his data
7. Workflow complete

Figure 1 on page 7 shows a graphical representation of this workflow.

### 3.2 SIP Validation Workflow

The SIP validation workflow is defined by the Producer-Archive Interface Methodology Abstract Standard (Consultative Committee for Space Data Systems, 2002a) building on the OAIS Reference Model (Consultative Committee for Space Data Systems, 2002b). It is a more precise definition of the “Validate SIP” step in the submission workflow discussed in Subsection 3.1. This workflow was chosen to see how well the modeling languages tested handle hierarchical refinement, a method often used in the design of processes.

The workflow is described in Consultative Committee for Space Data Systems (2002a, pp. 3-46). The steps are as follows:

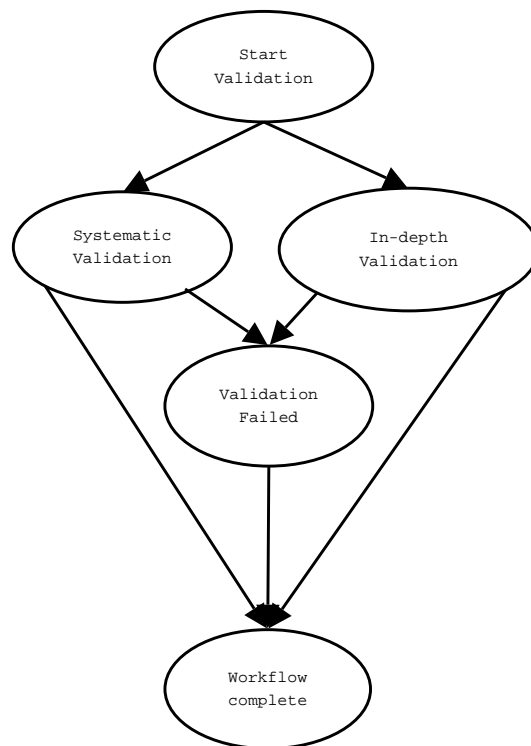


Figure 2: SIP Validation Workflow

1. Apply the validations (Check the conformity of the delivered objects with respect to the model of objects to be delivered and validate their contents)
  - Systematic validation (These validations are carried out after each transfer session)  
*or*
  - In-depth validation (These validations are only carried out if a coherent set of data objects have been delivered)
2. If the validation has failed, the producer is notified.
3. Workflow complete

Figure 2 on page 7 shows a graphical representation of this workflow.

## 4 Workflow Modeling Languages

In this Section the modeling languages are introduced shortly, followed by a mapping of the three example workflows, some thoughts on the mapping process, and an evaluation of the language based on the rating criteria described in Subsection 2.6.

### 4.1 High-level Petri Nets

The Petri net modeling language is widely used for the modeling, simulating, and verifying systems (International Organization for Standardization, 2002, p. 5). C.A. Petri first proposed it in 1962 for modeling interactions between systems.

A Petri net is composed (in its graphical representation) as a set of places (circles) and transitions (boxes), which are connected by arcs (arrows). The arcs only connect places and transitions and they have a direction (pointing from or to a transition). The Petri net is the union of places, transitions and arcs. A Petri net also contains tokens (dots); their location symbolizes a state of the net (they can only reside in places). A transition becomes active, if all places, which have an arc pointing to it, contain at least one token. When an active transition “fires”, it removes (consumes) one token in each of the aforementioned places and creates a token in each place with an arc pointing from the firing transition to the place. Whether or not an active transition fires, or which transition fires if there are multiple active transitions on the same token, has to be determined externally.

The beauty of Petri nets lies in their having a mathematically equivalent form to the graphical representation, which can be used to verify certain properties of a net, e.g., safeness, boundness, reachability, deadlock. The mathematical representation of Petri nets puts the modeling language on a solid formal base, a requirement, which seldom holds for other modeling techniques.

Basic Petri nets would not suffice for the modeling of workflows, because the concepts of hierarchy and time are alien to them Van der Aalst and Van de Graaf (2003). In contrast, high-level Petri nets offer this functionality on top of the same well-researched foundation. A more detailed description of hierarchy, time, and color (value) is available in

Van der Aalst and Van Hee (2002, pp. 41-48). Additionally, there are ongoing efforts to standardize high-level Petri nets with the International Organization for Standardization (2002).

The Figures ( 3 on page 11 and 4 on page 11) show the transformation of the generic workflows to a Petri net representation. The transformation is straightforward, since the states of the generic workflows directly represent the Petri net transitions, because the actual work is done in the transitions. The arcs of the generic workflows represent the places.

The Petri net representation provides additional possibilities for workflow simulation, as it can easily be animated to check that all use cases have been thought of. This animation is not only pretty to look at, but helps non-technical users to a better understanding of the sequence of steps.

The evaluation in Table 1 on page 9 clearly shows that high level Petri nets are very well suited to model workflows for digital archives and that the sole problem lies with its complexity.

### 4.2 Statecharts

Statecharts extend the conventional finite state machines with hierarchy, orthogonality, broadcasting, and history. Finite state machines are modeled using state diagrams, which are directed graphs with nodes (states) and arrows (transitions; labeled with events, conditions, and/or actions). The additions provided by the statechart formalism allow for a hierarchical structure of states and sub-states, while minimizing the number of states within the statechart.

Statecharts are widely accepted and used in the industry for specifying the behavior of complex reactive systems (Ackad, 2000, pp. 4-6) (Kyeyune, 2000, pp. 4-8). There are a few proposals for applying statecharts to workflow modeling motivated by the perceived inadequacy of other modeling languages to model events, e.g. Wodtke and Weikum (1997) and Lüttgen et al. (1999). Due to its origin, the modeling of events is one of the strengths of statecharts.

The formal foundation of statecharts was initially developed by Harel (1987) and has been adapted to the various needs, which has given statecharts a reputation of highly varying semantics; Van der Beck (1994) has listed more than twenty variants.

Criteria	Compliance	Remarks
Formal Semantics	1.0	Strong mathematical backing
Flexibility	1.0	Has been extended for many modeling purposes
Simplicity	0.5	Simple core, more complex extensions for workflow modeling
Decomposition	1.0	Not provided by basic Petri nets
Stability	1.0	Very stable and ongoing research and use
Retraceability	1.0	All execution paths visible
Total	5.5	

Table 1: Evaluation for high-level Petri nets

Figure 5 on page 11 shows the transformation of the two generic workflows into one statechart representation. The transformation was straightforward, since the states of the generic workflows directly represent the states of the statecharts. The arcs of the generic workflows represent the arcs of the statecharts. Since hierarchical decomposition is a central element of the statechart formalism, this was used to merge the workflows into one single workflow.

The visual statechart formalism is well suited to map states and reactions to events. Statecharts model hierarchy very intuitively, by allowing additional statecharts to be represented by a single state. However, there are no events (besides the end-event of processing) within the workflows portrayed. Together with the mixing of state and processing in this formalism, it there is little incentive to opt for this modeling language.

### 4.3 Activity Diagrams

Activity diagrams are part of the Unified Modeling Language (UML) by the Object Management Group (2003) and therefore part of the de-facto industrial standard in object-oriented modeling. It is a graphical language with textually defined semantics, thus has no formal semantics. These semantics can be modified to suit workflow modeling (Eshuis and Wieringa, 2002), but no commonly accepted semantics yet exist.

The notation of activity diagrams is based on a mixture of flow charts, statecharts, and Petri nets, having circles as start and end states, a diamond shape for decisions, rounded rectangles for action and subactivity states, and a heavy bar for synchronization and forking. They support the notion of subactivity states, i.e., a hierarchical structur-

ing of the activities. In the next version of UML the activity diagrams are supposed to obtain more Petri net-like semantics.

The following Figures ( 6 on page 12 and 7 on page 12) show the transformation of the generic workflows to an activity diagram representation. The transformation was straightforward, since the states of the generic workflows directly represent the states of the activity diagrams. The arcs of the generic workflows represent the arcs of the statechart, with added diamond shapes for routing decisions.

The examples show that activity diagram are a hybrid of flow charts, statecharts and Petri nets. They do not bring anything new to the subject, see also Table 3 on page 10.

### 4.4 More on Modeling Languages

Other modeling languages were initially considered, but dismissed, mainly because their semantics are not well defined, e.g. XML Process Definition Language (Workflow Management Coalition, 2002), or are not stable or well-researched, e.g. communicating flowcharts (Dong et al., 1999) or Petri charts Holvoet and Verbaeten (1995).

## 5 Conclusion

As we have shown it is possible to model our example archival workflows in all the portrayed modeling languages. However, the requirements of an archival workflow language are best met by a language having high-level Petri net semantics (Tables 1 to 3).

Another reason to choose high-level Petri net semantics is that they represent a very well re-

Criteria	Compliance	Remarks
Formal Semantics	0.5	Varying semantics
Flexibility	0.5	Constrained to reactive systems
Simplicity	1.0	Intuitive
Decomposition	1.0	Built-in
Stability	1.0	Mature, but varying semantics
Retraceability	0.5	Broadcasts are hard to trace
Total	4.5	

Table 2: Evaluation for statecharts

Criteria	Compliance	Remarks
Formal Semantics	0.5	Not defined by UML, see Eshuis and Wieringa (2002) for semantics for workflow modeling.
Flexibility	0.5	Has not been retrospectively demonstrated
Simplicity	0.5	Mix of two semantics plus special rules
Decomposition	1.0	Subactivity states supported
Stability	0.0	Semantics will change with UML 2.0
Retraceability	1.0	All execution paths visible
Total	3.5	

Table 3: Evaluation for activity diagrams

searched theory of concurrency. Petri nets are very easy to understand and they can be simulated to further aid the understanding. Petri nets are a very good at modeling workflows because the Petri net does not perform any of the tasks modeled, the workflow management system (WMS) also only manages the execution of the tasks, while specialized software or humans actually perform the work. The actual execution of the tasks is hidden to the WMS as it is hidden from the designer of the Petri net. They only see a box that consumes something and produces a result (data or token). Petri nets are ideal to involve the non-IT-personnel in the design process, because they are easily comprehensible and blend into the common model of workflows that most people have (start - work - result - work - end result).

## 6 Future Work

In future work we plan to verify and further explore the findings presented in this paper in practice, where the main focus will lie on finding suitable means for expressing workflows in machine-readable form.

Additionally, more workflows need to be designed. This should be done in cooperation with the archival community. We hope that this will raise awareness for problem of workflow description and standardization.

## Acknowledgments

We would like to thank the ARELDA project team of the Swiss Federal Archives for their support, namely Peter Keller-Marxer (discussions and review) and Thomas Zürcher (discussions on archival matters).

## References

- Ackad, J.-C. (2000), Optimierte automatische Statechart-Implementierung im Software- und Hardware-Entwurf eingebetteter Systeme, PhD thesis, Technische Universität Braunschweig.
- Adobe Systems Incorporated (2003), *PDF Reference*, 4 edn. Online: <http://partners.adobe.com/asn/acrobat/sdk/>

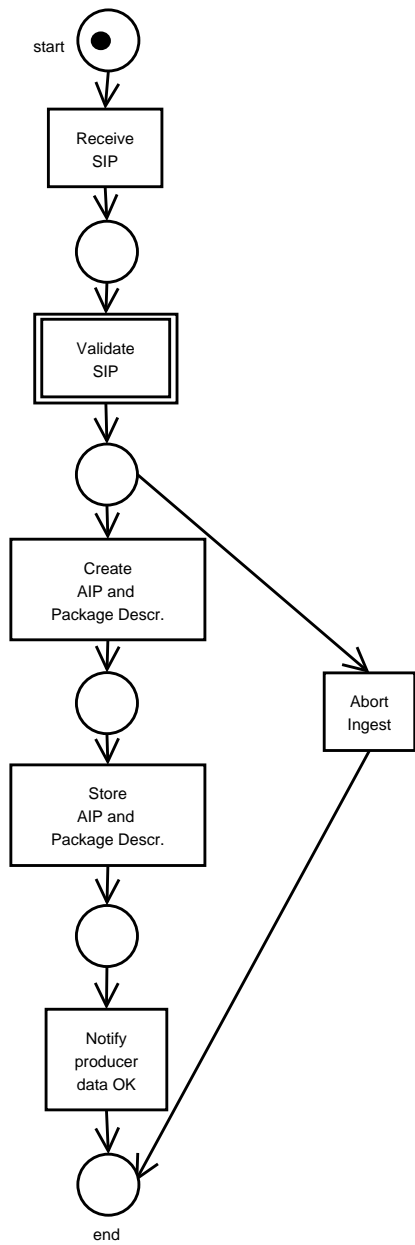


Figure 3: Ingest Workflow (Petri net)

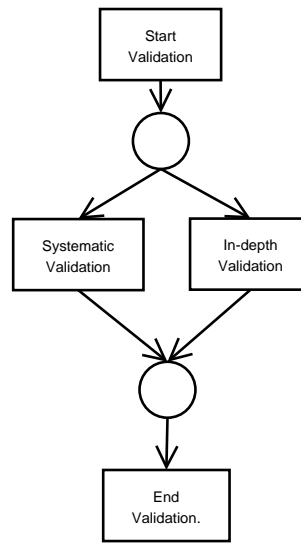


Figure 4: SIP Validation Workflow (Petri net)

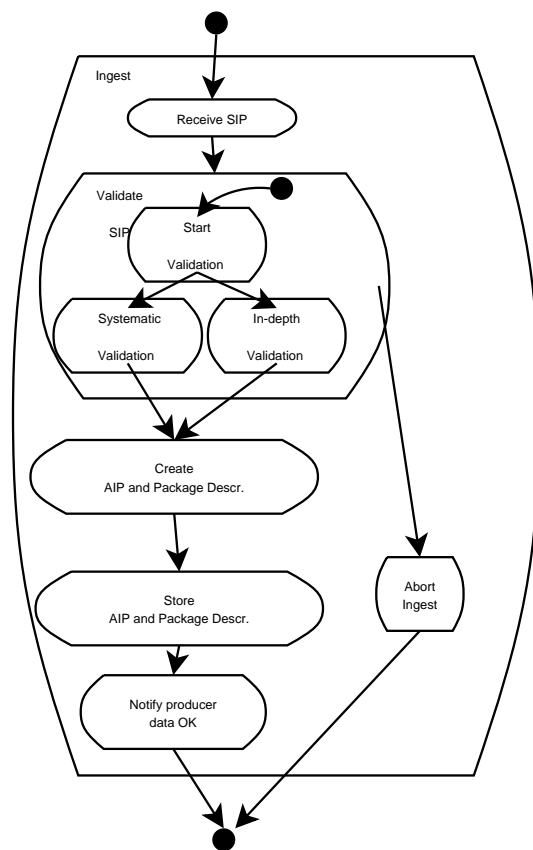


Figure 5: Ingest Workflow (Statecharts)

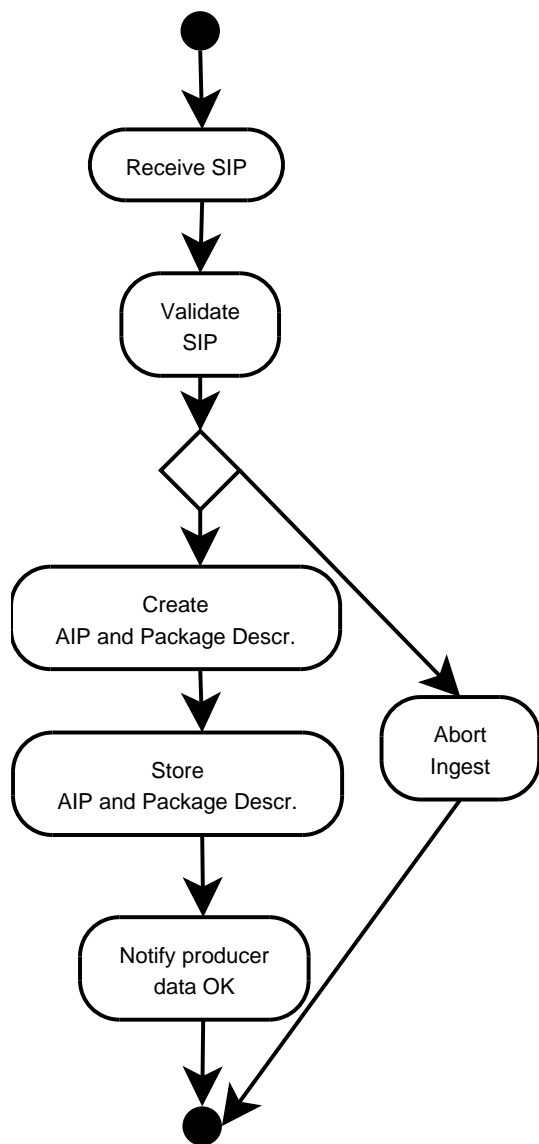


Figure 6: Ingest Workflow (Activity Diagrams)

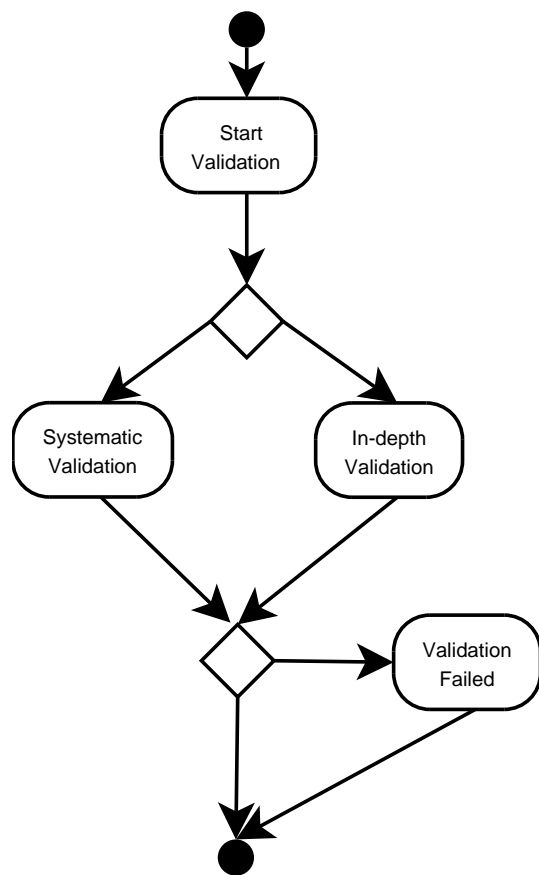


Figure 7: SIP Validation Workflow (Activity Diagrams)

- public/docs/PDFReference15\_v5.pdf (2004-10-08).
- Consultative Committee for Space Data Systems (2002a), ‘Producer-archive interface methodology abstract standard, red book (draft)’. Online: <http://ssdoo.gsfc.nasa.gov/nost/isoas/CCSDS-651.0-R-1-draft.pdf> (2004-10-08).
- Consultative Committee for Space Data Systems (2002b), ‘Reference model for an open archival information system (oais), blue book, issue 1. january 2002’. Online: <http://www.ccsds.org/documents/650x0b1.pdf> (2004-10-08).
- Cullen, C., Hirtle, P., Levy, D., Lynch, C. and Rothenberg, J. (2000), Authenticity in a digital environment, in ‘CLIR Reports’. online: <http://www.clir.org/pubs/abstract/pub92abst.html> (2004-10-09).
- Dong, G., Hull, R., Kumar, B., Su, J. and Zhou, G. (1999), A framework for optimizing distributed workflow executions, in ‘International Workshop on Database Programming Languages’, Lecture Notes in Computer Science, Springer, pp. 152–167.
- Eshuis, R. and Wieringa, R. (2002), Verification support for workflow design with uml activity graphs, in ‘International Conference on Software Engineering’, ACM Press, pp. 166–176. online: <http://tmitwww.tm.tue.nl/staff/heshuis/icse.pdf> (2004-10-09).
- Gilliand-Swetland, A. (2000), ‘Enduring paradigm, new opportunities: The value of the archival perspective in the digital environment’. online: <http://www.clir.org/pubs/reports/pub89/pub89.pdf> (2004-10-09).
- Harel, D. (1987), On the formal semantics of statecharts, in ‘Symposium on Logic in Computer Science’.
- Holvoet, T. and Verbaeten, P. (1995), Petri charts: an alternative technique for hierarchical net construction, in ‘IEEE Conference on Systems, Man and Cybernetics’, pp. 1–6.
- International Council on Archives (1996), ‘Code of ethics’. online: [http://www.ica.org/biblio/code\\_ethics\\_eng.html](http://www.ica.org/biblio/code_ethics_eng.html) (2004-10-09).
- International Council on Archives (1999), ‘Isad(g): General international standard archival description, 2nd edn’, ICA Standards. Online: [http://www.ica.org/biblio/isad\\_g\\_2e.pdf](http://www.ica.org/biblio/isad_g_2e.pdf) (2005-02-10).
- International Organization for Standardization (2002), ‘High-level petri nets - concepts, definitions and graphical notation, final draft international standard iso/iec 15909’. Online: <http://www.Petrinets.info/docs/pnstd-4.7.4.pdf> (2004-10-09).
- InterPARES (2002), ‘Authenticity task force final report: Establishing and maintaining trust in electronic records’. Online: [http://www.interpares.org/book/interpares\\_book\\_d\\_part1.pdf](http://www.interpares.org/book/interpares_book_d_part1.pdf) (2005-04-13).
- Jenkinson, H. (1947), ‘The english archivist: A new profession’, Inaugural Lecture for a New Course in Archive Administration delivered at University College London.
- Keller, P. (2004), ‘On the purpose of archives’. Online: <http://heuscher.ch/PurposeOfArchives> (2004-10-09).
- Kyeyune, Y. (2000), Developing Concepts and Methods for Module and Integration Tests of Reactive Systems, PhD thesis, Universität Dortmund.
- Lüttgen, G., Von der Beeck, M. and Cleaveland, R. (1999), Statecharts via process algebra, in ‘Proceedings of the 10th International Conference on Concurrency Theory’, number 1664 in ‘Lecture Notes In Computer Science’, pp. 399–414.
- Ministerrat der Deutschen Demokratischen Republik (1984), *Archivarbeit rationell - Arbeitsabläufe*, Staatsverlag der Deutschen Demokratischen Republik, Berlin.
- Object Management Group (2003), ‘Unified modeling language specification version 1.5’. online: <http://www.omg.org/cgi-bin/doc?formal/03-03-01> (2004-10-09).
- Papritz, J. (1998), *Archivwissenschaft*, Vol. 3, 2 edn, Archivschule Marburg.

- Rothenberg, J. (1995), 'Ensuring the longevity of digital information', *Scientific American* **272**(1), 42–47.
- Rousseau, J.-Y. and Couture, C. (1994), *Les Fondements de la Dicipline Archivistique*, Presses de l'Université de Québec.
- Schellenberg, T. (1961), *Akten- und Archivwesen in der Gegenwart*, Karl Zink Verlag.
- Schellenberg, T. (1965), *The Management of Archives*, Columbia University Press.
- Van der Aalst, W. and Van de Graaf, M. (2003), Workflow systems, in C. Girault and R. Valk, eds, 'Petri Nets for Systems Engineering', Springer, pp. 506–540.
- Van der Aalst, W. and Van Hee, K. (2002), *Workflow Management: Models, Method, and Systems*, The MIT Press.
- Van der Beck, M. (1994), A comparison of state-chart variants, in H. Langmaak, W.-P. De Roever and J. Vytupil, eds, 'Formal Techniques in Real-Time and Fault-Tolerant Systems', Vol. 865 of *Lecture Notes in Computer Science*, Springer, pp. 128–148.
- Wodtke, D. and Weikum, G. (1997), A formal foundation for distributed workflow execution based on state charts, in F. N. Afrati and P. Kolaitis, eds, 'International Conference on Extending Database Theory', Vol. 1186 of *Lecture Notes in Computer Science*, Springer, pp. 230–246.
- Workflow Management Coalition (2002), *Workflow Process Definition Interface – XML Process Definition Language*. Online: [http://www.wfmc.org/standards/docs/TC-1025\\_10\\_xpd1\\_102502.pdf](http://www.wfmc.org/standards/docs/TC-1025_10_xpd1_102502.pdf) (2004-10-09).