



HAL
open science

A Case Study on Process Modelling – Three Questions and Three Techniques

Olivier Glassey

► **To cite this version:**

Olivier Glassey. A Case Study on Process Modelling – Three Questions and Three Techniques. Decision Support Systems, 2008, 44 (4), pp.842-853. 10.1016/j.dss.2007.10.004 . hal-00724747

HAL Id: hal-00724747

<https://hal.science/hal-00724747>

Submitted on 22 Aug 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Case Study on Process Modelling — Three Questions and Three Techniques

Olivier Glassey

Swiss Graduate School of Public Administration (IDHEAP)¹

Abstract

In this case study we compare three process modelling techniques in order to find common concepts and to identify significant differences. We base this comparison around three general questions:

- What are the objectives of the organisation?
- Who is doing what with which resources?
- How does the organisation work?

The answers to the third of these questions (“how does the organisation work?”) are quite similar for all three of the modelling techniques we examine here. The main differences, at the modelling level, appear when considering the answers to the first two.

Keywords

Process modelling, graphical representation, description levels, comparison.

Biographical note

During the research phase of this case study, Olivier Glassey held a senior research post at the Swiss Graduate School of Public Administration in Lausanne. He was subsequently awarded a fellowship post at the Swiss National Science Foundation, and was a visiting researcher at the Fraunhofer Institute for Open Communications Systems (FOKUS) in Berlin, where he was a member of the Competence Centre for Electronic Applications in Government (ELAN). Olivier Glassey holds a PhD in Business Information Systems from the University of Lausanne and an Entrepreneurship diploma from the Swiss Federal Institute of Technology in Lausanne. Today, he provides consultancy and training in knowledge management, and in the composite domain of process modelling and knowledge management.

1. Introduction

Process modelling techniques are widely used in both business and academic organisations where research on process modelling or workflow management has been conducted for several years ([5], [11] or [13]). Process modelling requires workflow models, along with techniques for capturing and describing processes [9], with activity-based workflow modelling (in which a workflow consists of a partial or total set of ordered tasks, that is, partial or total sets of ordered operations, or descriptions of human actions) amongst these techniques. In other terms, a process model or view is an abstraction of an implemented process [7]. We deliberately use the word “techniques”, as these models can be made up of methods, modelling languages or integrated software applications (some open and free, others proprietary and commercial). This variety of techniques can be somewhat confusing, and the choice of an adequate technique for a given project might be difficult.

Between 2001 and 2003 we participated in a European research project on e-government (see Acknowledgement), whose main goal was to analyse processes in public administrations, and to design new, improved processes for providing online electronic services. This research project, “An Integrated Platform for Realising Online One-Stop Government (eGOV)”, was realised within the “Information Society and Technology” fifth

¹ IDHEAP, Route de la Maladiere 21, 1022 Lausanne, Switzerland, Tel. +41 21 694 06 00, Fax. +41 21 694 06 09, email oglassey@yahoo.fr

framework. A consortium of eleven commercial companies, public administrations, and academic partners developed, deployed, and evaluated an on-line governmental portal which offered advanced features to the citizens: personalisation, multilingualism, authentication, accessibility, etc. This platform was based on content and service repositories (both at the national and local levels); a service creation environment was used to administrate the repositories. The project also include the development of a governmental mark-up language (GovML) to describe public services and life events in order to search, locate, and retrieve governmental digital resources. A number of administrative services were created and provided through platforms deployed in Austria, Greece, and Switzerland at the national and local levels. For more on the integration of electronic services and XML-enabled repositories we recommend [23].

One of our tasks was to analyse and optimise the processes underlying these administrative services. [16] and [25] provide an excellent introduction on process analysis and reengineering and on how these can be used to improve organisational performance. Process modelling tools and methods support capture, representation, organisation and storage of knowledge on the state of an organisation. Indeed, process descriptions can be written in natural language, but formal process modelling with standardised semantics provides a bridge between process analysis and design made by people, and technical process implementation. Formal process models show, amongst others, actions together with constraints on execution order between them [24]. Descriptive models represent the current state of processes and prescriptive models are used to show how these processes could be optimised, i.e. how workflows could be better organised to reach given goals. These goals can be strategic, organisational, notably in terms of collaboration, and operational.

In order to design and implement these electronic services, we followed an approach inspired by [6] and wanted to create models of the actual system, of an ideal system, and of the system to implement. In order to design process models, i.e. a series of diagrams capturing the dynamics of a system [12], we needed to use a modelling technique. Partners with a technical background were in favour of UML, those experienced in organisational science supported OSSAD, and people coming from the field of business process insisted on using ADONIS:

- Adonis is a **software tool** for modelling operational processes and it has its **own proprietary modelling technique** and description language. It is one of the many commercial methodologies that are tightly integrated with a modelling environment, such as Aris Toolset (from IDS Scheer AG), Mega Process / Mega Designer (from Mega International, Inc.) or Bonapart (from Pikos GmbH). These tools are widely used in large firms and public administrations.
- OSSAD is an **open and standard modelling method** for organisations and information systems [4]. It was developed within a European research project and is supported by only one or two commercial software tools. Such public domain methods are not very common, the only comparable methodology is OPEN (Object-oriented Process, Environment and Notation), developed and maintained by a not-for-profit consortium [2].
- UML is a **standardised graphical description language** that can be used in the domain of process modelling although it was specifically developed for information systems. UML is a **de facto standard for modelling** and there are plenty of software tools, both commercial and free, that integrate this notation.

As the schedule was very tight, project partners decided to use ADONIS without in-depth analysis, but within in our own research work we later realised a detailed survey of these three techniques. We did not conduct this study in order to choose the “best” technique; we rather tried to find common concepts and to identify major differences in order to provide a comprehensive framework for process analysis and improvement. We analysed the concepts, the models and the application domains of Adonis, OSSAD and UML and made a detailed comparison [19]. In addition to this detailed survey we globally examined the underlying

concepts of three well-known and widely used modelling methodologies with very different backgrounds:

- MOKA is a methodology for developing Knowledge-Based Engineering applications, in particular in the fields of aeronautical and automotive industries, and it is used for designing complex mechanical products [17].
- CommonKADS is a methodology developed to support structured knowledge engineering within the European ESPRIT IT Programme. It now is a European de facto standard for knowledge analysis and knowledge-intensive system development, used by many companies and universities [10].
- ARIS is a very successful method and toolset for modelling business process, used in many universities throughout the world for research and teaching activities [1].

Each method provides between four and six types of models and although they are rather different we compared these forms of representation in order to identify the “views” that we would use for our study on process modelling. We decided to leave aside the technical layer of modelling, for example the physical (geometrical) representation of a product in MOKA or the (software) implementation diagrams in UML. These modelling techniques all provide a representation of reality:

- At the **abstract** level: functions of a product, missions of an organisation, or tasks of a given piece of software.
- At the **organisational** level: structure of an organisation and available resources, either technological or informational.
- At the **operational** level: step-by-step description of how a product is manufactured or how a given result is obtained.

In the following sections, we will study these representation levels in process modelling by answering the following questions:

- **What** are the strategic goals of the organisation?
- **Who** is doing **what** with **which** resources?
- **How** does the organisation operate?

In this paper we will not present the software tools we used, neither will we detail their respective functionalities in terms of analysis, simulation, documentation or code generation, neither will we look at technical or financial aspects. We will specifically focus on representation concepts and categories of diagrams for business process modelling. However, readers will find below a brief summary of the three techniques in terms of coverage, richness of representation, and ease of use:

- The main focus of ADONIS is on business process modelling; the graphical notation is very structured but not flexible; it is rather simple to use and fully integrated with the companion software tool.
- The goal of OSSAD is to support IT management within organisations; it has a limited number of generic concepts but these are not extensible; OSSAD is very simple but lacks good tools for graphical representation.
- UML is mainly used in the field of information systems; it provides many concepts that are very flexible and extensible, but it is rather complex for non-IT people; many tools integrate the UML notation.

2. Three modelling techniques

As stated above, we will not explain how to build a model with Adonis, OSSAD or UML neither will we try to describe all the underlying concepts and terminology. There are several complete reference books and articles (for example on UML see [8], on Adonis [22] and on OSSAD [14, 21]) and users manuals ([3] for Adonis and [4] for OSSAD) that do so quite well. However we will show that they are based on very different approaches and that they have quite different backgrounds, because we believe it makes the comparison all the more interesting.

Adonis is both a modelling environment and a proprietary modelling technique integrated with the software tool. It was developed by the Austrian company Business Object Consulting (www.boc-eu.com), a spin-off of the Business Process Management Systems group of the University of Vienna. Adonis is widely used in financial services and public administrations, notably for process optimisation and documentation, for quality management or for ISO certifications. The standard version of Adonis supports three types of models, but users can buy extension modules, offering for example IBM Lovem or UML support:

- *Process maps*: they give a general idea of processes and sub-processes that take place within an organisation.
- *Working environment models*: they show the structure of an organisation in terms of units, responsibilities and roles, as well as resources.
- *Operational process models*: they follow a process from beginning to end, showing all the activities that are to be done, actors that are responsible for a designated activity and resources linked to the realisation of an activity.

OSSAD (Office Systems Support and Analysis Design) is the result of a European research project conducted from 1985 to 1989 within the ESPRIT (European Strategic Program for Research in Information Technology) program. The goal of this open and non-proprietary method is to manage organisational problems induced by the massive introduction of technology in offices. OSSAD offers two levels of modelling and several types of diagrams or graphs:

- The *abstract model* shows the strategic goals of an organisation in terms of functions (e.g. marketing, finance, production) and information packets that circulate between these functions (e.g. contracts, statistics). Functions can be decomposed into cascading sub-functions, as many as necessary to describe a given organisation, a final node of sub-function being called an activity.
- The *descriptive model* represents human means and technological resources used within an organisation. These are defined in terms of procedures (how to realise an activity) and operations (the steps that are to be followed to accomplish a procedure), as well as in terms of roles (who participates in a given activity), resources and tools. The descriptive level consists of three types of graphical formalisms:
 - o The *activity-role matrix* provides formal links between activities and roles.
 - o The information circulation graphs describe the communication between roles (*role diagram*) and procedures (*procedure diagram*).
 - o The *operation diagram* shows the chronological sequence of elementary operations accomplished within a procedure.

UML (Unified Modeling Language) is an object-oriented graphical notation language that was developed and standardised by Rational Software (www.rational.com) and the Object Management Group (www.omg.org). As its name says, UML was born in 1997 out of the unification of three object modelling techniques: Booch, Object Modelling Technique and Objectory (OOSE) Process. Grady Booch, James Rumbaugh and Ivar Jacobson were the fathers of these techniques and now all work for Rational Software. In a few years UML became a de-facto standard for specifying, visualising, developing and documenting software [8]. This language covers the different cycles of software engineering (analysis, conception and implementation) with 9 types of diagrams:

- *Use case diagrams* represent the behaviour of a system from the user's point of view.
- *Class diagrams* show the static structure of a system (classes and their relations) without temporal information, a class being an abstract representation of a set of similar elements.
- *Object diagrams* represent objects and their relations, an object being a particular element of a class.

- *Sequence diagrams* describe interactions between objects along a temporal line.
- *Collaboration diagrams* represent interactions between objects in a structural form. Sequence and collaboration diagrams are called interaction diagrams and they are isomorphic, i.e. it is possible to transform one into another.
- *State diagrams* show the dynamic behaviour of an object in terms of states, transitions and events.
- *Activity diagrams* describe the flows circulating between activities within a system.
- *Component diagrams* show the physical implementation of a system, in terms of software components.
- *Deployment diagrams* describe the configuration of executables and related components.

For this work we selected use case, sequence, collaboration and activities diagrams because they are relevant to process modelling.

3. Comparison of Adonis, OSSAD and UML

To summarise the previous section we could say that Adonis, OSSAD and UML were created in different fields of research: business process (re-)engineering, office (re-)organisation and information system development. Furthermore they originated in different manners: commercial spin-off from an academic institute, European research project and fusion of three well-known object-oriented methods with the support of a software company. They nevertheless all have a common goal: to offer a representation of reality at different levels and from different points of view. They also share a common idea: cascading models with “zooming” possibilities. Thus we think that comparing them makes sense, however different they might seem, as long as we provide comparison “angles”, i.e. the three general points of view mentioned in the introduction:

- **What** are the strategic goals of the organisation?
- **Who** is doing **what** and **which** resources are available?
- **How** does the organisation operate?

Our comparison shares the common goal we identified for these techniques, as we want to study them at different levels and from different points of view. These are categorised formally below:

- *Abstract level*: strategy-centred
- *Structural level*: organisation-based
- *Operational level*: scenario-based (i.e. based on sequences of elementary operations or activities)

Table 1 shows the classification of the different types of models or diagrams that we used in this work. OSSAD activity-role matrices (ARM) create formal link between the abstract and the structural representations, therefore they are shown at the border of these two levels.

Table 1: Adonis, OSSAD and UML Models

	ADONIS	OSSAD	UML
What?	Process maps	Abstract models	Use cases
Who and what?	Work environment models	Activity-role Matrix	- Sequence diagrams - Collaboration diagrams
		- Organisational unit models - Role models - Procedure models	
How?	Operational process models	Operation model	Activity diagrams

For this comparison we use a simple example of business registration:

- The general idea is to create a new business.
- Basically, at the organisational level, an entrepreneur, some type of public administration office and a notary are involved in this process.
- The procedure to follow in order to register a business, along with specific requirements, is defined by law.

In order to illustrate the different graphical concepts we will provide examples for most models, but we will not explain in details the graphical notation as we think these examples are relatively straightforward and easy to understand.

3.1 Abstract representation

Process maps in Adonis, abstract models in OSSAD and use cases in UML have a common goal, modelling the objectives or the functions of an organisation. They are however relatively dissimilar in their conception and do not show the same type of information.

Adonis process maps [Fig. 1] only show processes in general terms. Their graphical semantics is basically the same as a bullet-list in a word-processor, with the addition of zooming capabilities: one can click on a process to see sub-processes or operational process models.

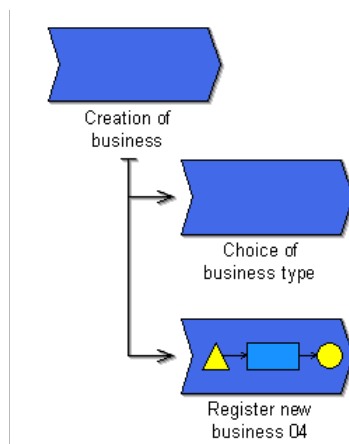


Figure 1: ADONIS Process Map

OSSAD also shows processes or functions, but it adds the idea of information packets and graphically shows the circulation of information packets between processes [Fig. 2]. Moreover this technique integrates the concept of external processes (or entities), which allows the representation of the interactions of an organisation with its environment. A process can be decomposed into sub-processes [Fig. 3] and sub-processes that are not decomposed are called activities in OSSAD.

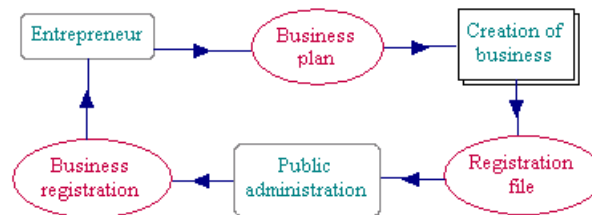


Figure 2: OSSAD Abstract Model

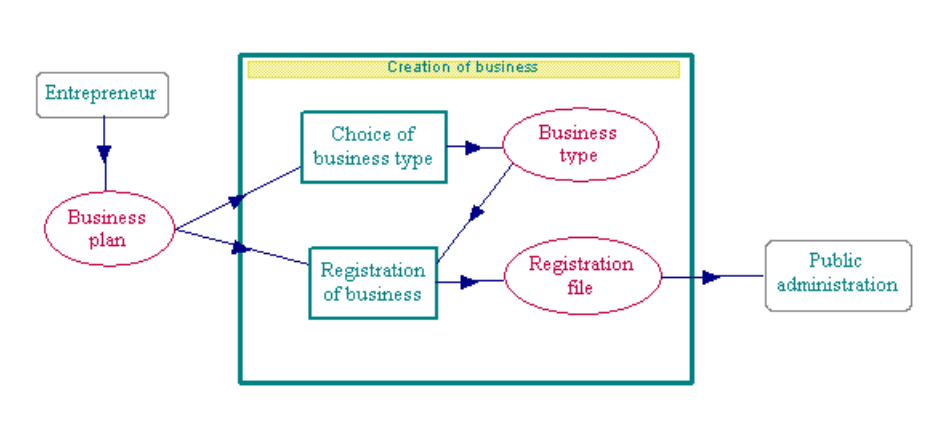


Figure 3: OSSAD Zoomed Process

Although the terminology is different, we think that UML use cases are equivalent to processes in Adonis or in OSSAD. They can also be decomposed with several levels of zoom [Fig. 4 and 5]. The UML concept of actor is furthermore very similar to the external process in OSSAD, apart from the graphical representation. However there is a big difference between UML and OSSAD: where the former only shows simple associations between actors and uses cases, the latter specifies what type of information circulates between the processes. This appears very clearly when one compares Figure 2 and 4 or Figure 3 and 5.



Figure 4: UML Use case

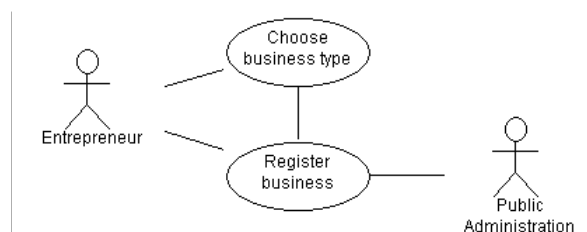


Figure 5: UML Detailed Use Case

To conclude this overview on the abstract level, let us point out that the concept of process is present in the three techniques: Adonis uses it as is, OSSAD makes a difference between internal and external processes and adds the idea of information packets circulation, UML relates processes and actors. From a modelling point of view, we think that Adonis models do not really bring any added value whereas UML and OSSAD integrate more representation semantics. The most detailed method is OSSAD, as it can show exactly the same information as UML and adds the representation of information circulation. In our opinion these differences are only logical: OSSAD is organisation-centred and it was intended for offices where the “raw material” is information, while UML was developed for software engineering with a focus on data and messages exchanged between classes of objects and it is not necessary to represent these at the abstract level.

3.2 Structural representation

The most notable differences between Adonis, OSSAD and UML are found at this level and we will go over them below. Again we believe they can be explained with the initial conception and application domains of these techniques:

- UML was not designed for representing the structure or the hierarchy of an organisation, with the direct consequence that it does not make a difference between physical actors and the roles that they are assigned within an organisation.
- Adonis and OSSAD are specialised process modelling techniques, thus they integrate the representation of the structure of an organisation and differentiate physical and conceptual actors from their roles or responsibilities. Figure 6 shows such an organisational chart, with units, workers and roles attributed to workers. This was made with Adonis and we will not show the OSSAD equivalent, which is similar (only the graphical symbols are different). This type of model can be useful to describe an organisation and its structures but it is not directly in relation with process modelling. Anyway, if needed it would be possible to develop UML class models showing the structure of an organisation although they were never intended to be used in that way.

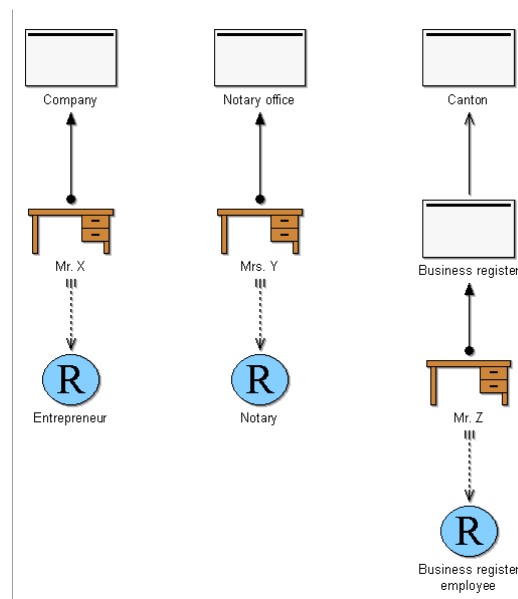


Figure 6: ADONIS Working Environment Model

The working environment model is the only Adonis model that allows structural representation. Yet we have to add that in the Adonis tool it is possible to assign roles or physical actors to a given elementary activity within an operational process model and to visualise all these assignments in a recapitulating table. However OSSAD and UML offer specific models in order to show “Who does what?” The OSSAD role model [Fig. 7] and the UML collaboration diagram [Fig. 8] present some similarities: the former shows the circulation of information resources between roles and the latter describes the exchange of messages between actors under a structural form. Yet in UML collaboration diagrams it is possible to number the messages in chronological order and thus there is the possibility to give temporal information that does not exist in OSSAD role models.

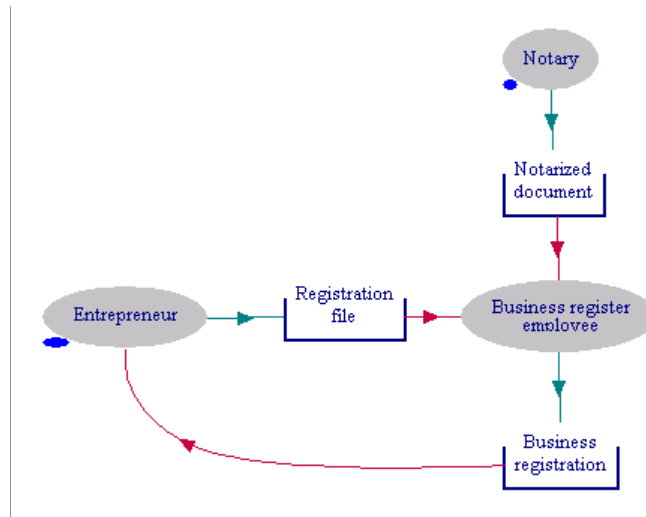


Figure 7: OSSAD Role Model

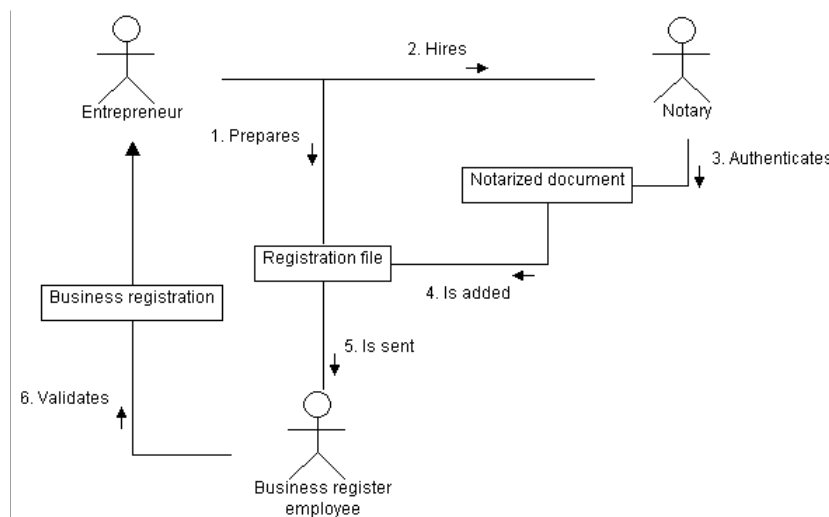


Figure 8: UML Collaboration Diagram

The procedure model of OSSAD [Fig. 9] provides a link between the abstract and the operational levels, as each procedure represents an activity from the abstract model and an operation model must be linked to a procedure. Moreover an information packet of the abstract level is formally made of one or several information resources in role or procedure models.

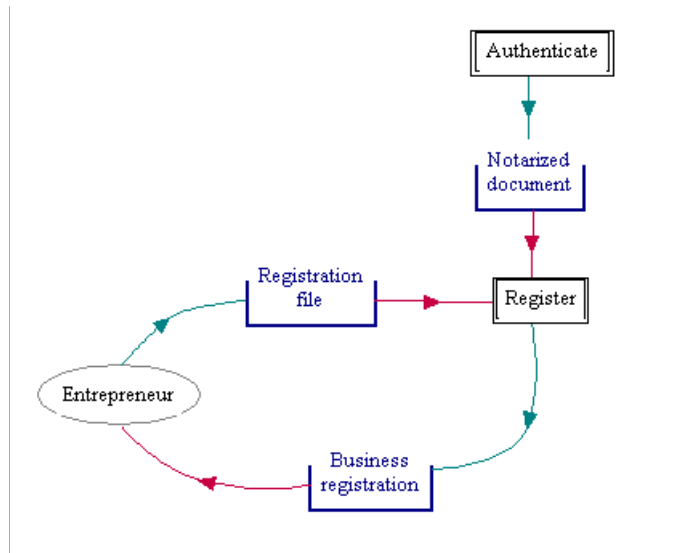


Figure 9: OSSAD Procedure Model

In UML there is no formal link between the abstract and operational levels, however there is a semantic link: a sequence diagram [Fig. 10] should be the representation of the scenario defined for a given use case. Detailed explanations on use cases and scenarios can be found in [20].

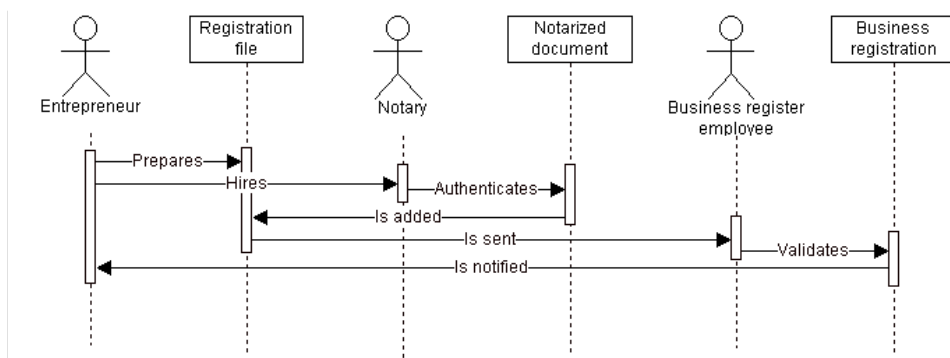


Figure 10: UML Sequence Diagram

It is interesting to note that procedure and role models are linked in OSSAD: they must show the same information resources, with the emphasis on circulation between procedures in one case and between actors in the other case. This symmetry can also be found in UML, as we already mention that sequence and collaboration diagrams are isomorphic, one with a temporal focus and the other with a structural one.

OSSAD is the only technique that provides a model in order to establish a formal connexion between the three levels: the activity-role matrix [Fig. 11] defines which roles are responsible for a given activity and each activity of the abstract model is represented by a procedure at the structural level and detailed in an operation model at the descriptive level. In other words an activity at the abstract level (the finest level of decomposition) is mapped onto a procedure at the descriptive level, i.e. a sequence of operations carried out by given roles. There may be several activity-role matrices, as there may be many ways of carrying out a given procedure and as the same activity may be carried out by different roles. These activity-role matrices prove very useful for reorganisation, as they show different ways of undertaking a procedure and potentially allow the definition of more efficient procedures. As we wrote above, the Adonis software allows its users to assign roles to elementary activities, but this is not formalised in terms of methodology and cannot be used for such

reorganisation purposes. On the other hand, such a matrix would not make much sense in UML, where the actors are already integrated at the abstract level.

Activities-Roles			
	Entrepreneur	Notary	Business register employee
Creation of business	X	.	.
Choice of business type	X	.	.
Authentication of documents	.	X	.
Registration of business	X	.	X

Figure 11: OSSAD Activity-Role Matrix

3.3 Operational representation

In our opinion, the representation of control and information flows is clearly identical in Adonis, OSSAD and UML. At this level, each process is viewed as transforming as set of inputs, modelled by incoming flows, into outputs or outgoing flows [12]. Indeed the same concepts are used, even if some graphical symbols differ:

- Elementary activities or operations are performed in a chronological manner.
- Swimlanes show which actors or roles are responsible for elementary activities or operations.
- Conditions, parallel operation and start/stop points are used to control the execution flow of the elementary activities or operations.
- Information resources or tools are linked to elementary activities or operations.

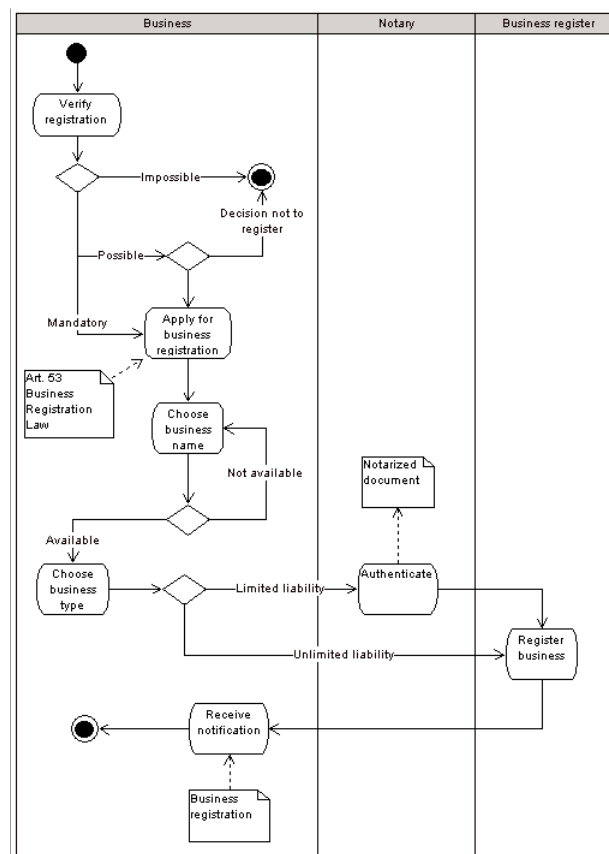


Figure 12: UML Activity Diagram

As operational process models (Adonis), operation model (OSSAD) and activity diagrams (UML) are very similar; we only show one example here: the execution flow for a business registration [Fig. 12]. Still, there are a few differences, both in terms of semantics and of graphical representation:

- Adonis differentiates roles and physical actors, OSSAD only takes roles into account and UML does not make any formal difference between these two concepts.
- Adonis provides many predefined symbols for resources (document, file, database, mobile resource, financial resource, computer, etc.), with UML one can model any type of resources as classes of objects, and OSSAD only uses three predefined concepts: information resources, documents and tools.

This parallelism is not surprising, for the representation of a chronological sequence of elementary operations and of resources has a low level of abstraction and must therefore be close to reality, without much of a margin of interpretation. These models are directly inspired from flowcharts and are found in many other techniques based on Petri net representation (see [15] for more on modelling systems with Petri nets).

4. Conclusion

Up to this point we used rather freely some terms such as operation or activity, role or actor, etc. We wanted the readers to get a general picture and not to be confused with identical terms that have different meanings in Adonis, OSSAD or UML. Nevertheless, these techniques formally link given terms and concepts. We grouped them in Table 2 in order to provide a synthetic view of terms and concepts.

Table 2: Adonis, OSSAD and UML Terminology

	ADONIS	OSSAD	UML
Abstract level	Process	Process	Use case
	---	External process (or entity)	Actor
	---	Activity	---
	---	Information packet	---
Structural level	Aggregation	Zoomed process	Environment
	Organisational unit	Organisational unit	---
Operational level	Responsible	Actor	Actor
	Role	Role	Actor
	Resource	Resource	Object
	Activity	Operation	Activity
	Decision	Post-condition	Branch
	Parallelism	Parallel operation	Fork and join
	Swimlane	Role	Swimlane
Resource	Resource, document and tool	Object	

This table allows us to make a transition towards our conclusions: with a simple look one can realise that OSSAD provides more concepts, especially at the abstract level. Looking back at Table 1, one can also see that OSSAD has more models than UML and Adonis. Of course, more does not automatically mean better and we announced in the introduction that our work was not intended in order to choose the “best” technique. We will only try to determine which technique might be more adapted in given situations.

Let us briefly summarise the characteristics of Adonis, OSSAD and UML. At the operational level, they are equivalent and can be used indifferently. At the structural level there is a difference between the pair Adonis-OSSAD, the business process and organisation oriented methods, and UML, from the information systems field. The choice of one technique would then be dependent of the domain to be modelled. If it is a hierarchical organisation where

persons can have different roles within the execution of process, OSSAD provides more precise semantics and a better integration with the abstract level, whereas UML might provide a somehow “blurry” representation of reality. However the latter is probably more flexible and would be adapted for ad-hoc or virtual type of organisations. Last, at the abstract level we should say that Adonis process maps are almost too “abstract” to be of any use. This is not inevitably a weakness, as it might not be necessary to represent graphically the strategic goals of an organisation in order to develop a good process model. Adonis is quite good to model the structure of an organisation and especially the available resources, and in some cases this might be more important than having a good abstract image. UML use cases do a very good job to that regard, while OSSAD concepts of information packets might be a bit confusing for those that are not used to this technique.

As a global conclusion, we will give our general appreciations on these techniques:

- Adonis is a good and complete integrated tool for process modelling, however it lacks some conceptual hindsight: that is not a problem when it used for simulation or analysis, but in our opinion that is a drawback for reengineering projects.
- OSSAD is probably the most complete technique and its models are very well connected with each other. On the other hand it is not widely spread and there are only two tools that support it.
- UML is the most generic and flexible technique and there are plenty of tools and resources available on the market. The flexibility of UML can sometimes be a disadvantage, as it does not formalise some concepts used in process modelling, which can cause losses of information or lead to the creation incomplete models. It is recommended to support conceptual, logical and physical design phases for information systems [18].

Acknowledgement

This work would not have been possible without the support of Prof. Jean-Loup Chappelet and was partly realised within the research project “eGOV: An Integrated Platform for Realising Online One-stop e-Government,” funded by the IST Programme of the European Commission (Contract Number IST-2000-28471, complete references available at: <http://www.egov-project.org/>).

5. References

- [1] A.-W. Scheer, ARIS - Modellierungsmethoden, Metamodelle, Anwendungen (Springer-Verlag, Berlin, 2001).
- [2] B. Henderson-Sellers, T. Simons and H. Younessi, The OPEN Toolbox of Techniques. (Addison-Wesley, MA, 1998),
- [3] BOC, Formation de base à Adonis, Training Manual (Business Object Consulting, Austria, 2002).
- [4] C-Log, Tutorial OSS@D Process Design (C-Log International, Geneva, 2001).
- [5] D. Wastell, P. White and P. Kawalek, A methodology for business process re-design: experiences and issues, *Journal of Strategic Information Systems* Vol. 3 No. 1 (1994).
- [6] D.A. Marca and C.L. McGowan, SADT Structured Analysis and Design Technique, McGraw-Hill (1987)
- [7] D.-R. Liu and M. Shen, Business-to-Business workflow interoperation based on a process-view, *Decision Support System* 38 (2004), 399-419.
- [8] G. Booch, J. Rumbaugh and I. Jacobson, The Unified Modeling Language User Guide (Addison-Wesley, MA, 1999).
- [9] G. Mentzas, C. Halaris and S. Kavadias, Modelling Business Processes with Workflow Systems: An Evaluation of Alternative Approaches, *International Journal of Information Management* No. 21 (2001).

- [10] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde and B. Wielinga, B. Knowledge Engineering and Management: The CommonKADS Methodology. (MIT Press, Cambridge, MA, 2000).
- [11] J. Herbst and D. Karagiannis, Integrating machine learning and workflow management to support acquisition and adaptation of workflow models, *International Journal of Intelligent Systems in Accounting, Finance and Management* No. 9 (2000).
- [12] J. Lee and G.M. Wyner, Defining specialization for dataflow diagrams, *Information Systems* No. 28 (2003).
- [13] J.E. Cook and A.L. Wolf, Discovering Models of software process from event-data, *ACM, Transactions on Software Engineering and Methodology* (1998).
- [14] J.-L. Chappelet and J.-J. Snella, Un langage pour l'organisation : l'approche OSSAD, 3rd Edition (Presses Polytechniques et Universitaires Romandes, Lausanne, 2004).
- [15] L.A. Cortes, P. Eles and Z. Peng, Modeling and formal verification of embedded systems based on a Petri net representation, *Journal of System Architecture* No. 49 (2003).
- [16] M. Hammer, Beyond Reengineering: How the Process-Centered Organization Is Changing Our Work and Our Lives, HarperBusiness, (1997).
- [17] MML Working Group, MOKA User Guide - MOKA Modelling Language Core Definition (MOKA Project - ESPRIT 25418, Deliverable D1.3, available at: <http://web1.eng.coventry.ac.uk/moka/mml.htm>, 2000).
- [18] N. Prat, J. Akoka and I. Comyn-Wattiau, A UML-based data warehouse design method, *Decision Support System* 42 (2006), 1449-1473.
- [19] O. Glassey and J.-L. Chappelet, Comparaison de trois techniques de modélisation de processus : ADONIS, OSSAD et UML (IDHEAP Working paper, Lausanne, 2002).
- [20] O. Glassey, One-Stop Government Prototype based on Use Cases and Scenarios, in R. Traunmüller and K. Lenk, Eds., *Electronic Government : First International Conference EGOV 2002, Aix-en-Provence, France, September 2-5, 2002* (LNCS 2456, Springer Verlag, Berlin 2002).
- [21] P. Dumas and G. Charbonnel, *La Méthode OSSAD, pour maîtriser les technologies de l'information.* (Les Editions d'Organisation, Paris, 1990).
- [22] S. Junginger, H. Kühn, R. Strobl and D. Karagiannis, Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation - ADONIS: Konzeption und Anwendungen, *Wirtschaftsinformatik*, 42(5) (2000).
- [23] S. Swamynathan, A. Kannana and T.V. Geethaa, Composite event monitoring in XML repositories using generic rule framework for providing reactive e-services, *Decision Support System* 42 (2006), 79-88.
- [24] S.S. Pinter and M. Golani, Discovering workflow models from activities' lifespans, *Computer in Industry* No. 53 (2004).
- [25] T.H Davenport, *Process Innovation: Reengineering Work through Information Technology*, Harvard Business School Press (1993).