

What every manager should know about SOA

(Eric Kluijfhout)

The nine brief chapters in this brochure, all taking one page or less, will inform management in Higher Education about SOA.

SOA stands for Service Oriented Approach as well as for Service Oriented Architecture: the difference between the two will be explained on the next page.

Those who do not want to spend ten minutes of their time reading the full brochure, and would rather spend 10 seconds to get updated, may turn directly to the last chapter!

Contents

1. SOA - what is it?	2
2. SOA - why do we want it?	3
3. SOA - the law of preservation of misery?	4
4. SOA - the neighbours got it already?	5
5. SOA - the distributors inside the institution	6
6. SOA - am I ready for it?	7
7. SOA - the pitfalls.....	8
8. SOA - a peek into the future.....	9
9. SOA - in the evening at the bar	10

Authored by Eric Kluijfhout of the SURFfoundation, Netherlands, 2008.

The views expressed in papers and contributions are the views of the contributor and contributing organisations and may not reflect the views of the e-Framework Partnership.

1. SOA – what is it?

Service Oriented *Approach* or Service Oriented *Architecture*? Both are SOA and are a method to finally achieve flexible, transparent, user-friendly and manageable ICT services. The 'approach' takes business processes as a starting point, where as the 'architecture' takes the perspective of ICT services.

SOA is nothing new, but combines the existing principles of:

1. Business Process Modelling (BPM) and Business Process Redesign (BPR) – both hypes in the nineties
2. System integration technologies or 'middleware' – a hot topic in the past years
3. Web-technologies and standards – currently becoming increasingly popular with the IT front runners

SOA is a container of concepts, in business as well as in Higher Education.

- ad.1. People with a background in organisation and processing, like organisation and policy advisors, prefer to talk about the service oriented *approach*. A 'service' here means simply a service that the institute provides to students, to staff or to another institute, which may or may not involve IT support. This 'approach' is characterised by taking a critical 'service-view' at business processes to improve user friendliness, flexibility and transparency of the information provision.
- ad.2. Starting from the system integration challenge, the talk is about a 'service oriented *architecture*'. 'Work under architecture' according to formal agreements and 'IT governance' are the key principles here. The service architecture indicates how ICT systems can provide 'services' to each other. An example is a Student Information System (SIS) that automatically transmits mutations of student data to the VLE. Reduction of the number of applications and maintenance of all interfaces in between is an important driver for the implementation of a service oriented architecture.
- ad.3. Finally there are the pragmatists, usually originating from the World of the World Wide Web. On the web they see numerous functions that with small adjustments could be applied in higher education, without complicated process analyses or architecture studies. Their mantra is: Just do it! To them a 'service' is a tool - MSN, a weblog, YouTube, a wiki, GoogleEarth, GoogleDocs, etc. – that one can use always and everywhere using a browser.

The multitude of perspectives explains not only the confusion of tongues but also the popularity of SOA: many can identify to at least one aspect of the concept, and the idea of a 'service' makes it possible for policy workers and process owners, IT staff and students to enter a dialogue. The challenge is of course to consequently reach common agreement about the meaning of SOA in the context they are in.

2. SOA – why do we want it?

On the previous page a number of advantages of the services concept are highlighted. Here we explore the six most important ones for higher education.

The user up front. On the basis of one's personal selection of 'best of breed' software services, the end user - student, faculty, researcher, administrative staff - constructs a customised personal learning and working environment on the desktop. The environment can be accessed everywhere – outside the institute as well – through a single log-on via the browser.

Work processes are leading. In the services approach the work processes direct the appliance of ICT, and not the other way around. Because software services perform relatively small and autonomous tasks, they can be applied in work processes in a flexible way. This in sharp contrast with current monolithic applications in higher education!

Collaboration while preserving identity. Software services from different suppliers and based on different technology platforms can easily collaborate – as long as the exchanged messages are understood. This increases the possibility for collaboration between institutes with different technology platforms considerably. Of course the condition is that clear agreements exist about the precise *meaning* of the messages: for instance, do we both define the term 'student' in the same way?

Chain integration. Chain integration - the coupling of work processes and IT services crossing the border of the higher education sector – gets easier. Think about the information exchange between secondary and tertiary education; between the various economic sectors; exchanges within the government and the market of potential students. By developing *portfolio services*, instead of a portfolio module within the SIS, such a portfolio will play an integrating role in all phases of life long learning and working.

Share cost. By developing software services that can be applied in a number of processes, in a number of institutions, and even in whole sectors, cost reductions are possible. Think about existing services like authentication, authorisation, and mail, but also MSN. Because services are able to communicate over the internet, the coordination and management could be organised in Shared Services Centres. Using this model for their 'standard' ICT services, the institutes can save considerably on the costs of ICT.

More efficient IT management. In SOA each ICT-function exists only once in an institute, because a single service-component can be used by several work processes. This will lead to an enormous 'un-doubling' of IT functions compared to the current situation. Think about log in, search and computing functions that usually appear in many applications. Additionally the maintenance of several customised interfaces between applications will disappear using SOA.

3. SOA – the law of preservation of misery

The advantages offered by SOA such as flexible work process design, easier collaboration, sharing of costs, more efficient management of IT facilities and user friendliness do come at a cost!

Flexibility by standardisation! Crucial to SOA are agreements on how services should communicate between each other. The required technical (web) standards are in place. But probably more complex are the agreements about the exact data that will be exchanged between the services. Within one's institute the situation may be manageable – it should be possible to unambiguously define the term 'student' for the whole institute. In collaboration with other institutes the challenge becomes more complex – often the neighbours have a slightly different definition of what a 'student' is. In the case of chain integration the situation grows even more involved – what we understand to be a student, may be an employee to the employers, and a citizen to the government. In these circumstances, how do we all agree on a reliable definition of a 'person', and how will this 'person' compare to our 'student'?

Shifting complexity. In current ICT systems processes usually are 'baked in'. This causes inflexibility and makes adjustments to the system a costly and time consuming affair. At the same time the systems show a considerable overlap in functionality. In SOA the actual services are usually relatively simple 'black boxes' that can be applied in a flexible way in a variety of work processes and as such avoiding duplication of functionality. The cost however is a significant complexity in the correct *fitting and direction* of the numerous service components into a logical process ('orchestration'), and in the 'translation' of the dataflow between services that may interpret a term, for example 'student,' slightly differently.

Extremely slow! It is obvious that the coupling of numerous independent services that communicate between each other over the campus network, or even over the internet, will produce an enormous amount of messages. The smaller – and therefore the more flexible to use – the services are, the heavier the traffic of messages. On top of the actual data other information about the data (metadata) is sent. At the current state of technology this will lead to unacceptable delays.

Security. The use of several services owned and sometimes managed by third parties, and increasing the traffic of messages on the internet, causes considerable security problems. The required standards are still under development, but even these solutions will eventually add to the 'overhead' and lead to slow response times.

Where are they? Last but not least: only a few services exist that are developed specifically for education! Work is in progress to change this situation, among others by SURF within the e-Framework collaboration. Beside these efforts almost all IT suppliers are making their products 'service-enabled'. Implementation of SOA will be a gradual and time consuming process.

4. SOA – do the neighbours have it already?

Many institutes are involved with SOA, although not always under the SOA flag, and sometimes unconsciously. Due to the different backgrounds SOA originated from - BPM, middleware, web standards - the types of SOA-projects are of a wide variety.

Order in the chaos. Many institutes lost the overview of the systems they apply and how these systems interact - often this is the result of organic growth over the years. The mapping of the current systems often is the first step on the road to a more effective management and un-doubling of functionality. Many such projects consequently adopt SOA-principles as a long term perspective to avoid these kinds of problems in the future, or at least diminish their occurrence.

ICT governance. In response to the ad-hoc development of the current system landscape in many institutes, many came to internal agreements about the use of standards, choice of technologies, ICT project organisation and operation, etc. This type of agreement is a first step towards – and a condition for – working towards service oriented architecture.

Reference architecture. Beside mapping the current situation and agreeing to the IT principles, a number of institutes are addressing the question “How can we apply IT *strategically*?” Derived from the strategic goals of the institute clear priorities and conditions for future IT services are formulated, and a future reference architecture drafted – usually with SOA as the leading principle.

Portal project. In order to increase user friendliness of the information services (specifically for students), last year a variety of portal projects were launched. Delivering services by means of browsers is a common method, used in SOA as well. At the same time these portal projects did not deliver the expected glorious results: the underlying challenge of doubling of functionality and of data that ‘behind the portal’ are still ‘locked in’ could not be solved, but became much more visible. That’s one of the reasons that many institutes are looking at SOA now!

System integration. Many IT departments connect existing systems on the basis of web technologies and web standards. Through this approach a lot of experience is gained that will be useful later on, when SOA will be more structurally applied.

Webservices and self-services. Inspired by the motto ‘Keep It Simple!’ many are of the opinion that the SOA approaches of ‘architecture’ and ‘process design’ are too heavy (see ‘SOA – the law of preservation of misery’). Many current web applications can be used immediately – or can be tweaked to work with little effort – for education: MSN, weblogs, YouTube, wikis, GoogleEarth, GoogleDocs, etc. Additionally, many existing applications can be ‘service enabled’ relatively easily, using web technologies that enable students and staff to access (parts of) applications via a web-interface.

5. SOA – the distributors inside the institute

Within the institute one has to deal with supporters, enemies and the undecided with respect to SOA, all with private stakes and rationales from their own perspective.

Policy staff: will want to talk about SOA in relation to 'business', 'positioning of the organisation', 'increased flexibility of work processes', and a 'user centred approach'. Above all SOA is considered to be a service oriented *approach*, or 'a perspective' to look at organisation, processes and information provision. Quotes as 'SOA is not a technical issue' and 'SOA is mainly about organisational change' can be heard often.

Head of IT. This manager is usually plagued by budget cuts, users that know better all of the time, and budget-keepers that under the pretext of 'academic freedom' purchase new systems on their own account. For the Head of IT SOA is as much an opportunity as a threat. An opportunity to get 'architecture' and 'IT Governance' finally on the agenda, and a threat because seriously working on SOA will have implications this manager cannot foresee completely. "Without clear agreements the SOA approach will only lead to a bigger mess" and "Let's first start with a small pilot project" are the not very motivating but indeed justified comments from this corner.

IT-staff: They see SOA as a logical step in the development of the so called 'middleware-technology' that allows systems of different suppliers to work together. Volunteering comments as "We already have SOA, we just installed a portal" or "SOA works as good without SOAP¹" IT-staff indicate that, in according to their opinion, SOA is not that special and above all is – and should remain – the business of IT-staff!

The web-fan. Often this is the proverbial academic, always searching for the 'latest and newest features' in IT – mostly during working hours. Favourite hobbyhorses of the web-fans are web services and the 'plug and play infrastructure'; mash-ups (connected web services, such as railway schedules projected as riding trains on GoogleEarth); 'self-services' (student and staff getting direct access via a browser entrance to part of a system); and 'opening the IT services' of the institute.

The architect. This position seems to exist thanks to SOA. Previously the architect had difficulty explaining his work, and consequently the perceived added value of his work; but now he surfs on the popularity of SOA. But appearances can be deceiving: with so many visions of SOA an identity crisis lurks! Nowadays bringing together all stakeholders, each with their own vision of SOA, may be the most important function of the architect!

¹ Simple Object Access Protocol

6. SOA – am I ready for it?

Whether SOA is a realistic option for the institute depends on the level of urgency of the IT problems to be resolved; of the ambitions and the impact of IT for the organisation; of the 'grip' that the management has on IT development; and on the available resources - especially staff.

Valid business case? SOA generates the fastest Return on Investment (ROI) in environments that have a high change rate and where short 'time to market' is essential to gain important strategic and/or financial benefits. Since this typology does not exactly match most Higher Education institutes, not many convincing business cases exist for SOA in Dutch Higher Education. SOA for collaboration with other institutions and chain integration requires a long(er) term perspective to establish the inherent coordination beyond the borders of the actual institutes.

The right level of ambition? The right level of ambition depends totally on the problem to be solved or the desired outcome. It would be going too far to fully explore the breadth of analyses possible, but, in summary, one may start 'small' with SOA (e.g. conducting a pilot project to integrate two systems as well as 'big' (e.g. starting from thorough architecture studies). In the latter case the organisation must meet a higher level of requirements (see above).

IT Governance in order? Although one can 'start small' with SOA, in principle the choice for SOA is all-embracing: a 'little SOA' will not produce strategic advantages – such as flexibility, manageability, and user-friendliness. A strategic choice for 'big SOA' can only be implemented effectively if the IT governance is in good shape – consider how un-doubling of functionality could mean cuts in current applications, a strict policy regarding purchase and development of applications and components, establishing agreements about, and consequently enforcing, data models and the use of standards.

Staff? In the end it's the people who have to work with SOA! Given the multitude of interpretations of SOA, a minimum requirement is a shared work definition of SOA, agreed to by all involved. This work definition will be determined more strongly by the problem to be solved than by the 'academic correctness' of the SOA concept. Because SOA starts from the perspective of business processes, heads of business operations of the primary and support processes must be involved as 'process owners'. Since basically SOA is about IT support, IT-staff should be at the table as well. The architect or information manager usually acts as initiator and 'bridge' between these two 'worlds': this person will be expected to meet high criteria concerning process steering and communication skills. Where technical expertise is required, external hiring should be considered strongly.

7. SOA – the pitfalls

An institute starting with SOA faces a number of unpredictable pitfalls.

Solution looking for a problem. The danger of all hypes: it is unclear what the problem is that should be solved using new technology. If this is the case, and there is a desire to have a taste of SOA, establish a well demarcated 'play ground'. In this area work can be done on small, conveniently arranged, but 'real' problems.

No business case. Especially when SOA is really considered to be an alternative for the current suite of monolithic systems, it is hard to get a good overview of future efforts and costs. Convincing and solid business cases for SOA in higher education are hard to find. This suggests that well defined, clearly demarcated, and not too complex problem areas should have priority.

Get lost in architecture studies. When mapping the current system landscape and configuring the future service oriented architecture there's a danger that too detailed architecture models and descriptions may be drafted, that a) are in not in proportion to the problem that needs to be solved, b) will be delivered with delay, and c) can only be understood by fellow architects. A clear problem definition, delivery of increasingly detailed architecture in short cycles, and feeding these back to the problem owners, may prevent this.

We know best. A number of benefits of SOA – chain integration, use of 'best of breed' services, shared services centres – by definition require collaboration. Adoption of SOA on your own is an expensive endeavour and many of the potential advantages will not be realised.

The Board needs balls. The most significant advantages of SOA can only be realised, if implemented institute-wide, and preferably sector-wide, by collaboration. Reaching and maintaining agreements - 'governance' - is an absolute prerequisite. If institutes don't strictly keep a hand upon cooperative agreements, the previously described advantages will not counter-balance (high) SOA costs!

8. SOA – a peek at the future

If we realise the full potential of SOA, what should we expect? Here's an example in the education area to envisage what would happen.

Scenario -

It has been decided that starting this week the procedure for course admission in the institute will change. The Head of Student Administration has overall responsibility – and not the IT department – and starts to redesign the admission processes according to SOA principles:

1. The different tasks of the new admission process are modelled in a graphical tool: tasks are drawn as blocks, and the process sequence is visualised by arrows between tasks. The result is a flowchart of the admission procedure, presenting the tasks in the order that they should be executed.
2. For each task in the flowchart the Head of Student Administration must then point out:
 - a) which data are needed for the execution of the task (e.g. to perform the task 'compare qualifications with admission criteria' the necessary data 'secondary education scores' and 'admission-demands')
 - b) which data must be presented after the task has been performed (e.g. after performing the task 'compare qualifications with admission criteria' the required outcome could be 'matches admission criteria' or 'does not match admission criteria')
 - c) the *rules* that define how input-data, while performing the task, should be transformed to output-data. *How* the rules are applied within the task is not really relevant.

The result is a flowchart that shows the tasks, the order of the tasks to be performed, and the messages that should be exchanged between the tasks - the 'services' that the tasks deliver mutually!

3. For each of the specified tasks a tool will search the Internet - via a 'Who Is Who for Services' - for software components that can deliver the related services. Such software components may originate from commercial suppliers, by open source initiatives, or by education institutes that at a point in time may have developed software components.
4. The flowchart of step 2 then automatically gets 'filled' with the discovered software components. The result is a 'Services Architecture' for the new admission procedure. If for specific tasks no software components could be found, the task can be performed manually, or a software component can be built after all.
5. The software components cooperate within the services architecture (mutually delivering services) over the internet, or will be installed on the servers of the institute. The new admission procedure is now accessible through a browser.

In this example it doesn't matter what programming language is used to develop the software components, as long as they can communicate with one another on the basis of agreed standards. Especially the area of communication standards has seen significant progression during the last years, and almost all software suppliers have committed to the standards and consider SOA as the road for the future.

9. SOA – in the evening at the bar

“SURF apparently lost its interest in wires and boxes. Now they want to decide how we should rule our institutes.”

“The Dutch StudieLink system was supposed to be an example of SOA. Look what that’s brought them!”

“This SAKAI thing was planned to be SOA, wasn’t it? Do you know anyone using it?”

“It seems that SURF is working on SOA related issues together with the British and the Australians. Oh, yes, and with New Zealand as well. Do you ever hear anything about it?”

“The commercial IT-boys will never seriously develop services that seamlessly connect to their competitors’ products!”

“Now that we are talking about these things: do you know how you really can save money using ICT