WHITEPAPER

Innovative Integration of Software Development Systems

MANAGEMENT SUMMARY

Software development tools often have been, and may continue to be, integrated with the help of inflexible point-to-point scripts that are difficult to maintain, however, there is now a much more promising solution. Developed specifically to provide a meaningful solution from a business and IT perspective - the goal is a more effective service oriented architecture that meets your needs - to give you flexibility and scalability, making child’s play of the simple integration of systems and external partners.

This White Paper provides background information on the technology, gives examples of its application and demonstrates the technical and commercial advances we provide in comparison to other approaches.

Authors:
Prof. Dr.-Ing. Peter Fromm, Hochschule Darmstadt
Ralf Klimpke, agosense GmbH

RESUMÉ

Prof. Peter Fromm
Dr. Fromm studied electronics in Aachen. After his PhD in mechatronics in 1999, he built up a Software and System Engineering department in VDO / Automobile Information Systems Division (later Siemens and Continental). His main duties centred around the successful introduction of CMMI Level 3 and the definition and implementation of a global standard engineering tool chain. As part of his duties he was responsible for the process areas of configuration and change management, requirements engineering and software engineering. In 2008 he accepted a post as Professor for Microcontroller and Information Technology at the University of Darmstadt.

RESUMÉ

Ralf Klimpke
Ralf Klimpke studied industrial engineering at the University of Technology in Esslingen. He has worked in various positions since 1995, including at MKS GmbH (which became PTC in 2011), a supplier for Application Lifecycle Management Tools. There he was responsible for the creation of an indirect operation channel for a product line. Later, as Key Account Manager, his acquisition of many well-known large companies, he was responsible for establishing the company as a strong presence in the automobile branch. Since 2009, Ralf Klimpke has worked as a co-founder and Director of agosense GmbH, bringing with him more than 15 years of industry experience in operations and marketing to the company.
The effective integration of software engineering systems and the automated exchange of development data are hotly disputed topics in IT and development departments. The focus of the discussion is the integration of systems from various process areas into one continuous tool chain, often across company boundaries - starting with requirements management, through test management, change and defect management all the way to configuration management. For process oriented industries, for example the automobile industry, medical technology industry or the finance industry, an integrated tool chain together with a comprehensible process provided by efficient software, has become indispensable.

The reality however shows that not all companies can or want to implement an integrated ALM Suite\(^1\) from a single software manufacturer as a standard. This creates process gaps, for example in the area of modelling, or if certain user groups apply highly specialised tools which are not part of an ALM Suite (Best of Breed Strategy)\(^2\).

Additionally, development processes do not stop at company boundaries. Client-supplier relationships, or development partnerships often involve the (electronic) exchange of information. This creates the challenge of automating and documenting processes, to ensure they are maintained and able to be monitored at any time.

Where these integration needs have been served by point-to-point integrations, we would like to present you with a more innovative and more promising approach.

**Traditional Approach: Point-to-Point-Integration**

How was (and is) integration being implemented? Usually, through the creation of synchronisation scripts and programs within the company or the department itself or by external consultants - with the advantage that these are generally fitted to the case at hand. But then what happens when the development process is modified or an integrated tool releases an updated version? Or when individual groups or departments use different work processes? These changes mean that integration scripts must also be changed, that means, new scripts must be created or commissioned, tested and perhaps even distributed in different configurations among a large number of users.

In addition, these integrations are often created and administered by the technicians themselves, usually developing into a cost intensive parallel IT department. While the reasons for this are clear - as individual development divisions are in a position to react more quickly than a centralised IT department - it also means that technicians must take time away from their core activities, namely product development and project work, for the administration, maintenance and adaptation of these integrations, usually with the result that they are unable to ensure an optimal operational concept.

Some software producers or third parties offer plug-ins for data synchronisation between development tools. However, with respect to the level of functionality and adaptability required, these generally provide more of a compromise than complete solution, with the level of reliance on the tool producer increasing dramatically when updates for existing tools need to be implemented. Compounding matters is the fact that integration technologies are often not yet sufficiently developed to support these new versions, precisely at the moment updates are (required to be) implemented.

The more linkages between tools there are, of course, the greater the level of dependency, thereby delaying or preventing necessary product updates.

The disadvantages of such strict integration, which usually only links two specific tools are clear. They are highly complex to create and maintain and therefore very susceptible to changes and updates of individual components. On top of this, an increase in the number of tools, therefore also in the number of point-to-point connections, also becomes increasingly difficult to maintain and administer.

Under this approach, the use of these tools with regards to operations and process security is complex: how well can this approach be scaled for use with larger quantities of data or larger numbers of end users? How complicated is the roll-out of scripts and plug-ins for every individual PC? Do potential conflicts have to be resolved manually by the user or an administrator? Who ensures that

\(^1\) ALM Suite – itself a highly integrated tool platform, which covers as many process areas in Application Lifecycle Management as possible.

\(^2\) Best-of-Breed Strategy: describes an approach that seeks out the most suitable or most specialised tools for each process area of the company.
the user has also synchronised their current data for use in the next step of the process? These are only some of the more common
issues that are usually only able to be addressed with a great deal of discipline and manual (re-)working.

Clearly, the level of operational effort for the maintenance of this approach to integration has been underestimated. But what
alternatives are there for integrating varied software tool environments in a way that is efficient and flexible?

**The search for an efficient alternative**

Before the question of alternatives can be answered, the most
common challenges for modern integration technology need to
be defined. In this, the most frequent challenges listed include:

- Flexible Infrastructure that is easily embedded into the
  existing IT environment
- Simple integration of further (and/ or internally developed)
  tools
- Potential integration of additional process areas, including
  product lifecycle management, project management, etc.
- Scalability and processing speed (real-time synchronisation)
- Automation of process and a reduction in manual errors
- Robustness, the ability to deal with errors and conflicts,
  reporting/ logging
- No additional effort or additional user interface for end users
- Centralised IT operation for simultaneous decentralised
  configuration for groups or projects
- A complete product, supported by an independent producer
- Easy to understand and use; i.e., based on standard existing
  technology

For data exchange with external partners:

- Potential automation for the collection and provision of data in company portals.
- Logging/Recording of data exchanges (both in and out) to ensure compliance with „external“ processes requirements
- Support for standardised data exchange formats e.g. XML, ASAM, RIF/ ReqIF, etc.
- Full product support - no individual programming or servicing

**Increased efficiency with Service Oriented Architecture (SOA)**

The bigger picture, beyond the world of ALM and software development, reveals that established standard technologies have long
been used in other areas, such as the area of business processes.

Service oriented architecture (SOA) provides a perfect example of the integration of business applications and business processes. It has been successfully used for years, and this has resulted in widespread acceptance and adoption. Ultimately, software engineering processes are nothing more than specialised business process, making the adaptation of these standard technologies for use
in software development a logical step.

But what exactly does this term, service oriented architecture, mean? To give one example: „A service oriented architecture (SOA) is a concept that is structured to suit the business and IT needs of a company, that is modular and flexible for use in business processes. Therefore, a service oriented architecture cannot be sold "out of the box"; rather, SOA is a concept that must be adapted to suit individual circumstances - both with regards to the organisation as well as the existing application landscape of a company.“

---

3 Source: http://www.soa-know-how.de
Basically, this involves the following components:

- Basic services for (web based) communication
- Unique services to map business processes
- Adapters to implement data interfaces with individual applications
- Applications that display business processes graphically
- Execution layer for business processes (process definition and mapping)
- Administration and presentation functions

Transferred into the world of software development, this structure can be created in the following ways: Figure 2 clearly shows the creation of a tool chain, integrated using SOA. The data is generally sent via a collaboration channel from project management, requirements management, change management and test management between external participants. This integration solution replaces the initial classic data exchange via paper, e-mail or data carrying device. The data is read from a source system, e.g., an FTP server or portal system, filtered where necessary, then syntactically and semantically interpreted, converted and imported into the target system. Internal tools are seamlessly integrated into a continuous tool chain in the same way, although with the key difference that the data does not need to be exported and imported via an interim format. In this case, individual tools are accessed directly via the integration interface - making point-to-point integrations superfluous.

According to this structure, the individual operations of a service oriented architecture can be organised into three levels, the process layer, middleware and adapter.

Figure 3 shows the components involved and the services illustrated in the „Integration Platform“ shown in figure 2.

The adapter layer, Data Exchange, is responsible for the physical exchange of data, and provides direct linkage to the individual tools. It interlinks closely with the lower middleware layer, Data Integrity, which ensures error free and complete data transfers and optimises performance.

The upper middleware layer, Data Conversion, is tasked with the semantic conversion of data, and therefore links closely with the process layer. The mapping of attribute values, data fields (e.g. status information) and the conversion of data sets into company specific formats is located here.

The standardised data is then processed in the Process Integration layer and transferred to the target system. This occurs by means of the „BPEL“ (Business Process Execution Language) graphical modelling language. This process modelling, together with relevant adapter functions allow defined processes to be documented simply, and carried out exactly.
Goals and Applications
An important goal of integration is the creation of a data exchange that is as automated as possible between the various participating systems extending beyond network and company boundaries. The efficiency gains to be had from automated exchange speak for themselves as a significant argument in favour of this solution, however the exceptionally flexible integration potential for a range of internal tools is should not be overlooked as a major positive. Using process modelling, this range of tools can be adapted exactly to suit the needs of the company. Automation reduces the level of manual involvement required, and thereby also reduces costs. Removal of the human error factor and the securing of synchronised accurate data also massively increases data quality, process quality (traceability) and performance.

The typical applications for cooperation with external partners include the exchange of requirements documentation, change requests and error reports between automobile manufacturers and suppliers. Standardised data formats such as RIF/ ReqIF or ASAM make it possible to automatically exchange data with external partners, regardless of the tools being used.

Application for the Marquardt GmbH, Automobile supplier and switching system specialist headquartered in Rietheim-Weilheim:
The customer (Daimler AG) is sent a trial product to begin the first function and integration tests for the corresponding automobile project. Errors or variations from the agreed specifications are collected by the Daimler engineers and entered into the supplier’s portal „DanTe” in the form of XML data.

Marquardt takes the relevant data from this portal using its own access. The portal is regularly checked for new messages at regularly configurable intervals by the agosense.symphony integration platform. Any new data is automatically imported into the corresponding development project’s internal change management (in this case Serena Dimensions) and assigned to the project team responsible. While the Marquardt development team is processing the notification, defined updates are sent back to the customer until such time as the problem has been resolved. Throughout this process, both the customer or Marquardt may also use the platform to report further errors or status updates generated by internal testing. This ensures that everyone involved in the project has constant access to the latest information regarding the progress of the project under development, providing a reliable basis for planning future activities.

To ensure that the data exchanged remains consistent, internal processing and exchange processes must be closely harmonised. This process is modelled and executed directly from the agosense.symphony platform, ensuring reliable automated synchronisation.

Application for a Swiss Bank headquartered in Zürich:
To manage application development in the private banking division, new product features and change requests from a variety of sources, including from business units, operations and quality management, were adapted using Atlassian Jira and Confluence to form the central project collaboration application. Each new product or product approval was organised, prioritised and assigned to the relevant application in accordance with defined processes by request management, with the result that the corresponding applications and requirement engineers could be automatically transferred to the project chosen for implementation.

Regardless of the development methods being used - iterative or the traditional
The development divisions of requirements engineering, solution architecture, software engineering, test engineering and the corresponding configuration management were constantly involved to adapt the applications at hand as part of the product lifecycle. This was the only way that the Bank would be able to install hundreds of applications (Java, Mainframe and others) and maintain them for the duration of their service life.

This was done by ensuring that requirements engineers are initially assigned amendments in the form of change requests from the requirements management system (Polarion Requirements) in order to create the specifications and applications necessary for further development. Corresponding UML models are created by the requirements engineer in Sparx Enterprise Architect and automatically synchronised with Polarion Requirements. New status information on the requests is regularly provided by the collaboration application in Atlassian Jira, keeping the user who originally requested the change informed on its progress.

The software architecture team use the change requests from Polarion Requirements along with additional information from the technical analysis in Sparx Enterprise Architect to validate the architecture of the application, and adapt it where necessary. This process ensures a consistent documentation trail throughout the whole lifecycle. Software engineers use the requirements specifications together with the architecture and design documentation for the actual process of software development. This allows the developer to see the project requirements and requests directly in Atlassian Jira, to change them, and link them with code changes. At the same time, requirements that have been released are transferred to the test area in Hewlett Packard Quality Centre (HP QC)/ALM. Test engineers use this tool to create new test cases or adapt existing test cases, and all planning for testing is synchronised with the project planning in Atlassian Jira. Upon completion of the coding, applications are checked against the test cases and test plans in HP QC/ALM. An error report is generated for any unsuccessful test, and automatically sent back to the project team for further development. This process of transferring the test results required by developers to allow them to analyse and correct errors continues until a successful test result signifies the end of the development cycle.

The different tools used in the software development process are continuously modelled and controlled using agosense.symphony, any errors or conflicts are dealt with as they arise. In addition, depending on the process definition, data can be either directly synchronised via user interaction (e.g. through change of status) or conducted according to a time schedule. This ensures that everyone involved in the process can spend the majority of their time working with their specialised software tools, retaining access to all necessary information without having to switch tools or user interfaces.

Further expansion of the tool chain by integrating Atlassian Confluence, CA Clarity or other applications via the integration platform is currently being explored.

The SOA provider market
Apart from companies like Oracle, SAP, Microsoft or IBM, who often place an emphasis on the integration of their own products, there are only a few development system specific applications like the agosense.symphony from agosense GmbH. In contrast with the preliminary definition of service oriented architecture, agosense.symphony is a complete product which already contains all the necessary components and services, and has already been introduced into many companies. Using a graphic process modelling method, the platform offers the greatest flexibility for implementing company specific rules for defining data flows between tools and participants. Using open standards such as BPEL, ASAM, RIF/ReqIF etc ensures that the platform is easy to integrate into the existing system and able to be connected to almost any external system.

Thus, agosense.symphony fulfils all the challenges presented here, while also offering some further advantages over alternative approaches.
By directly connecting the adapter to an application programming interface (API) it provides the broadest possible range of functions for the tool to be integrated. Other attempts to standardise APIs across software producers have often resulted in fundamental limitations to the individual capabilities of the tools. With agosense.symphony, the integrated execution layer for the process models allows processes to be defined and controlled by users without any programming knowledge. Additionally, as an independent company, agosense enjoys a high level of support from various system providers in the development and maintenance of a variety of adapters for all major software producers.

Figure 6 provides a schematic representation of the integration platform. The adapter framework - here only partially shown - shows some of the supported tools and formats. These are connected to a communication bus system via specific adapters, which are then controlled using process execution via the bus system. Data flow processes and actions can be defined and represented using the graphical modelling language BPEL. Functional models are integrated above the application, allowing the administrator to easily configure the system using a graphical browser user interface.

**Summary**

„Innovative Integration of Software Development Systems“ - As stated in the title, this White Paper provides an alternative innovative approach for the integration of development tools. Using service oriented architecture, the software development process is efficiently covered, creating a medium to long-term solution to traditionally inflexible and complex point-to-point integration that is essential from a business and IT perspective.

**About agosense**

As a provider of the integration and exchange platform agosense.symphony agosense has positioned itself in the field of “Development Process Integration” – integrating heterogeneous Application Lifecycle Management (ALM) environments in software and system development into a continuous process chain.

The product range is aimed chiefly at companies who seek sustainable long-term and independent integration of their development and management systems, or those who have close connections with their business partners.

The platform enables a process driven and automated data exchange within the development process – even with external partners. This makes use of a range standard formats including RIF/ReqIF, ASAM, and others to support the broadest possible range of development tools. The range of available adapters for popular tools ensures a swift implementation. The agosense.symphony architecture ensures that integration scenarios can be adapted to suit individual needs as set by customer requirements.

Numerous companies from a broad range of industries have already chosen agosense GmbH as their company solution:

- Amadeus Germany GmbH
- BMW AG
- Bury GmbH & Co. KG
- Continental AG
- Daimler AG
- Intel Mobile Communications GmbH
- Robert Bosch GmbH
- Volkswagen AG
agosense does not only work closely with customers, partnerships with software tool producers are also very important for the company. These relationships are the only way that agosense can develop high quality adapters for individual application and their versions and react quickly to changes.

In addition, agosense also offers a range of licensing options for software tool developers that agosense is happy to design to suit their specific needs with regards to packaging, branding and customising. As an OEM partner you can offer your customers a decisive added value advantage when their integration needs extend to external tools.

agosense partners also include consultancy firms. Expand your consultancy service portfolio by offering different business models, for example, solution delivery or service provision for agosense products. That means your customers can benefit from standardised solutions while retaining the flexibility to adapt to their specialised needs.

Contact
agosense GmbH
Stammheimer Str. 10
70806 Kornwestheim

telephone +49.7154.99951.0
fax +49.7154.99951.99

info@agosense.com

Get more information on our website www.agosense.com.