Integration of intelligentware in complex software: stratification aproach

Aleksey Senkov[a]

[a] *Smolensk branch of Moscow Power Engineering Institute, Energeticheskii proezd, 1, Smolensk, 214013, Russian Federation*

Abstract

Integration of intelligentware into complex software systems, accounting systems, ERP systems, as well as the integration of such tools among themselves is an urgent scientific and scientific-practical task facing large companies. The paper discusses the methodology of integration, based on the sequential construction of a single bus, designed to transfer data and knowledge between system elements. The approach is based on the best practices of electronic data interchange (EDI) in various industries, as well as on the enterprise service bus (ESB) pattern adopted as a standard in some large companies.

Keywords 1

Intelligent systems, intelligentware, integration, ERP, software systems

1. Introduction

The second half of the 21st century can be characterized as a stage of explosive growth of information and intelligent technologies. Serious results have been achieved by engineers and programmers of companies such as Tesla, Google and others.

Almost all success stories in the field of intelligentware and intelligent systems have been delivered by startups, i.e. companies that initially take an innovative path of development and aim to create new technologies. Such companies are very flexible in terms of choosing technical solutions and technologies. They often receive a new product release every week or even quicker.

At the same time, it is important to note the difficult situation of large international companies, where the renewal processes are much more laborious and, therefore, more time-consuming resulting in a significantly longer system renewal compared to launches of new systems by startups.

In practice, one can find many examples of large companies with a worldwide reputation, which have to work with technologies and systems developed in the 70s and 80s of the 20th century. The reasons for this historical legacy are different in each case. The consequences can be summarized in a company's inability to integrate new technologies into skeletal historical systems.

A natural desire of such companies is to use all the opportunities provided by modern technical solutions and technologies (including intelligent ones) in their daily activities. However, old technical solutions often do not support integration with modern intelligentware for several reasons.

- Outdated technologies do not support modern integration channels, especially protected.
- To upload data in the required format, it is often necessary to revise the system itself - the revision can be awfully expensive and time-consuming.
- Obtaining and using results of the work of intelligent systems also requires expensive and time-consuming revision of the original system.

Russian Advances in Fuzzy Systems and Soft Computing: selected contributions to the 8-th International Conference on Fuzzy Systems, Soft Computing and Intelligent Technologies (FSSCIT-2020), June 29 – July 1, 2020, Smolensk, Russia

EMAIL: a.v.senkov@mail.ru

Thus, large companies find themselves hostages of the existing IT infrastructure and deployed software and cannot ensure their prompt adaptation that would correspond to the high rates of intelligentware development set by modern IT communities.

Regarding intelligentware, the issues of its integration into accounting systems (ERP systems) are usually considered in two ways:

- in the formation of intelligent solutions and their integration into certain information systems, the peculiarities of such integration are described [1, 2];
- methodologies and technologies for creating software for intelligentware and intelligent systems "in general terms" without detailing the peculiarities of integrating such systems into existing software systems [3, 4].

This determines the relevance of developing approaches, ways and methods for prompt integration of modern intelligentware into existing accounting systems.

2. Peculiarities of building accounting systems in international holding companies

Figure 1 shows a model structure of a large international company. The company consists of many divisions, each responsible for its own type of the company business. The enterprises belonging to the divisions are located all over the world, which means that business processes taking place in them can vary greatly.

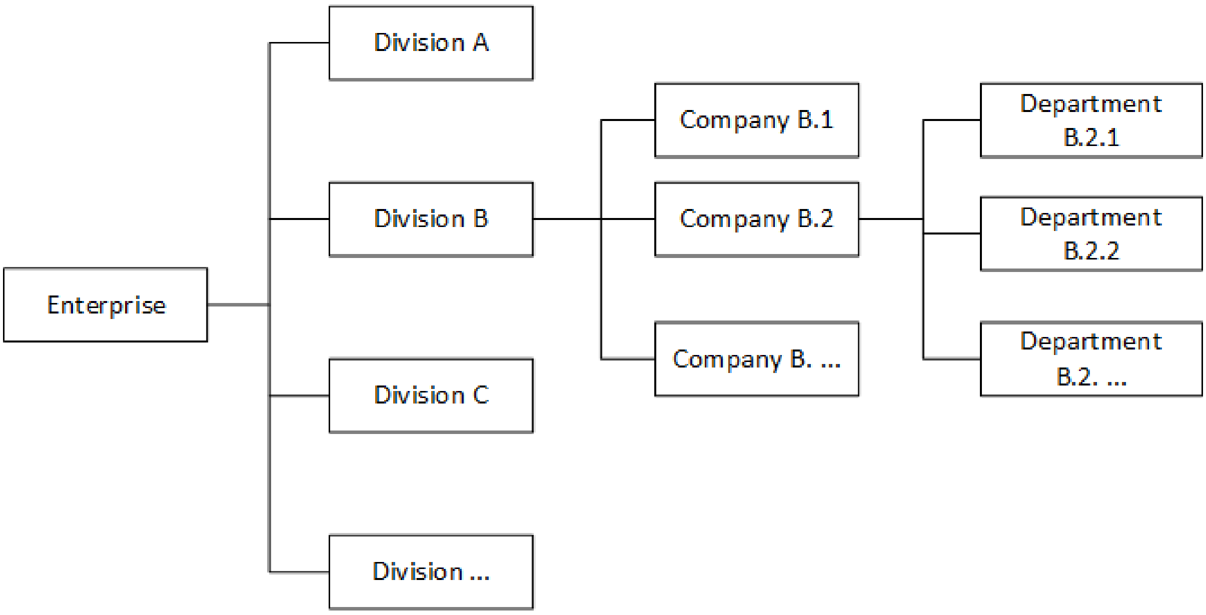

Figure 1: An example of a large holding company divisional structure

In the case of a complex implementation of a large accounting system such as SAP R/3 or another accounting system, as well as a proprietary accounting system, the problem of differences in business processes was usually solved as follows. A reference solution was developed that implemented average automation of business processes, which was further extended to the divisions. The divisions, in turn, performed business-specific revision of the reference solution and extended it to subordinate enterprises. The enterprises had to revise the solution to meet the requirements of local regulations (Figure 2). In addition, interfaces for interaction of systems at one level and interfaces for inter-level interaction were rather rigidly defined at each level. This unification was monolithic: both the integration channel and the business data to be transferred were fixed at the same time.

Of course, in different companies the approach to solving the problem of software adaptation could change, however, the result was usually similar: a large-scale software product appeared inside a large company, consisting of many versions of the reference solution that interact with each other.

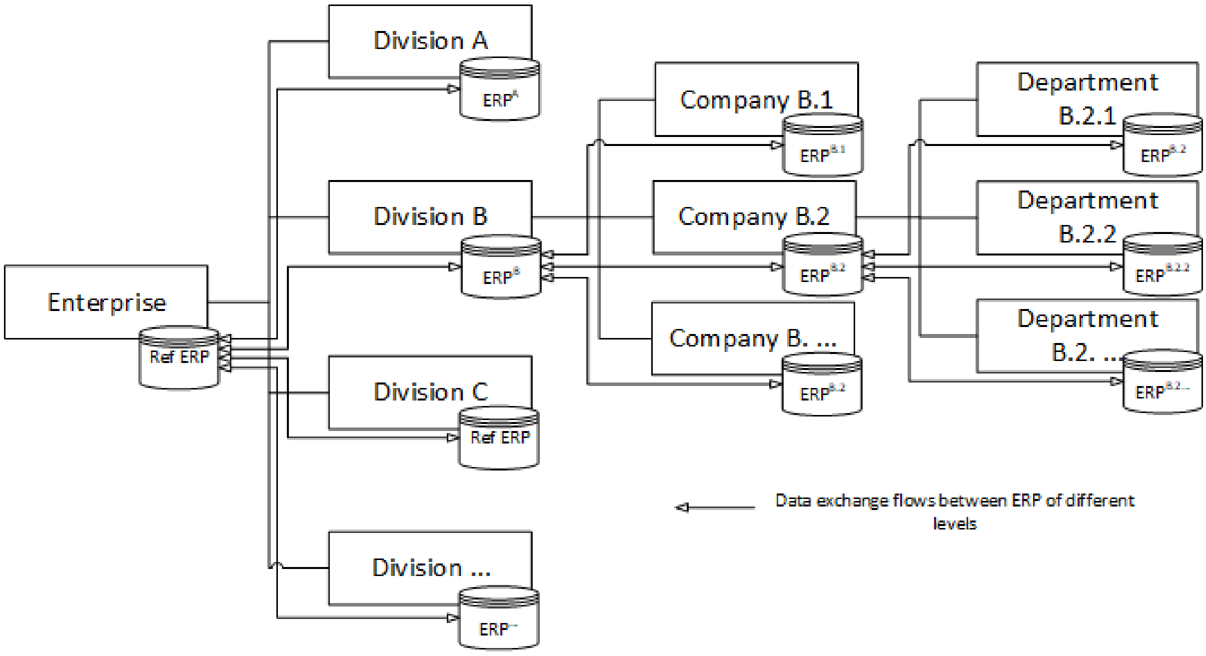

Figure 2: An example of building a complex accounting system within a large holding company

A similar technique for managing software products is adhered to by 1C Company (producer of most replicated Russian ERP) [5], which supplies "boxed" software products such as 1C: Accounting, which then undergo a business-specific revision for a particular company. Further on, 1C Company releases updates to the "boxed" product, however, each update requires involvement of a developer, who monitors changes in the boxed version and ensures that the enterprise-specific solution update does not decrease the enterprise automation level (within the framework of the specified 1C solution).

Many companies, which used modified 1C solutions, later either abandoned centralized 1C updates and made all the changes in their solution due to regulatory changes on their own, or abandoned adapted solutions in favor of a "boxed" product.

A similar situation can be observed in large companies. Over time, maintaining outdated solutions, both technologically and in terms of business processes, becomes an extremely difficult (time-consuming and expensive) task.

3. EDI-based approach to automated b2b communication between companies

EDI (electronic data interchange) is an approach to automated communication between companies. As part of the EDI implementation for various industries, various interaction standards as well as general interaction technologies have been developed.

The basic standard is the UN/EDIFACT [6] standard, which is the framework for developing standard implementations for different business areas. The most common are the following: SWIFT in the banking sector; EANCOM in trade; IATA in air transportation; EDICON in the construction industry; HL7 in healthcare.

Each of these standards is a complete description of a set of messages circulating in a certain area and rules for their processing.

In terms of organization, EDI implementation involves operators of electronic document management (EDM) - the third party, which is, in fact, a transport between two participants in the data exchange (Figure 3). Thus, it is enough for EDM participants to establish integration with the operator, after which they are able to send information to any counterparties, both connected to that operator and connected to other operators, by roaming between such operators.

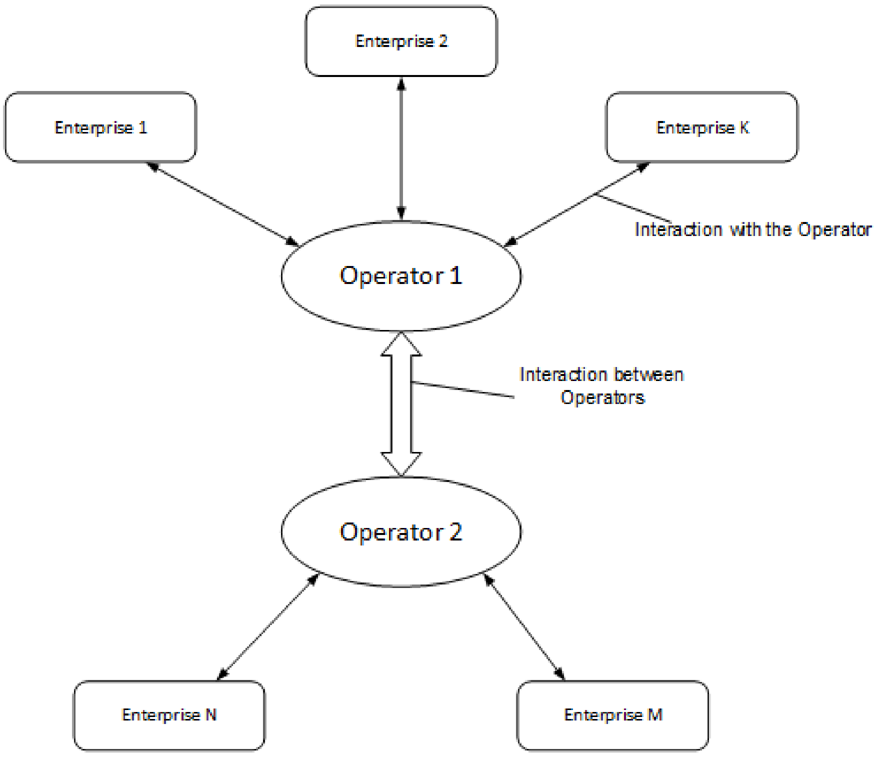

Figure 3: Symbolic scheme of interaction between organizations through EDI

EDI is a good example of an integration pattern that connects multiple parties to an exchange. It should be noted that the standardization of data and messages at the highest level of abstraction is the main condition for the success of such integration.

4. Stratification approach to integration of intelligent systems

The EDI approach can also be used to solve the problem of updating and intellectualizing information systems existing within large holdings. This approach consists in fulfillment of two significant conditions.

First, a single centralized data bus should be implemented within the holding. In the modern IT community, this is called ESB (enterprise service bus). Within this bus, all messages that can be transmitted between system elements must be standardized. From a technical point of view, the bus can even implement the same technical interfaces that were previously implemented in the deployment of a large-scale IT system. This implementation will allow to smoothly bring the bus into productive work.

To a first approximation, the bus should implement the interaction of reference information systems at one of the levels (Figure 4).

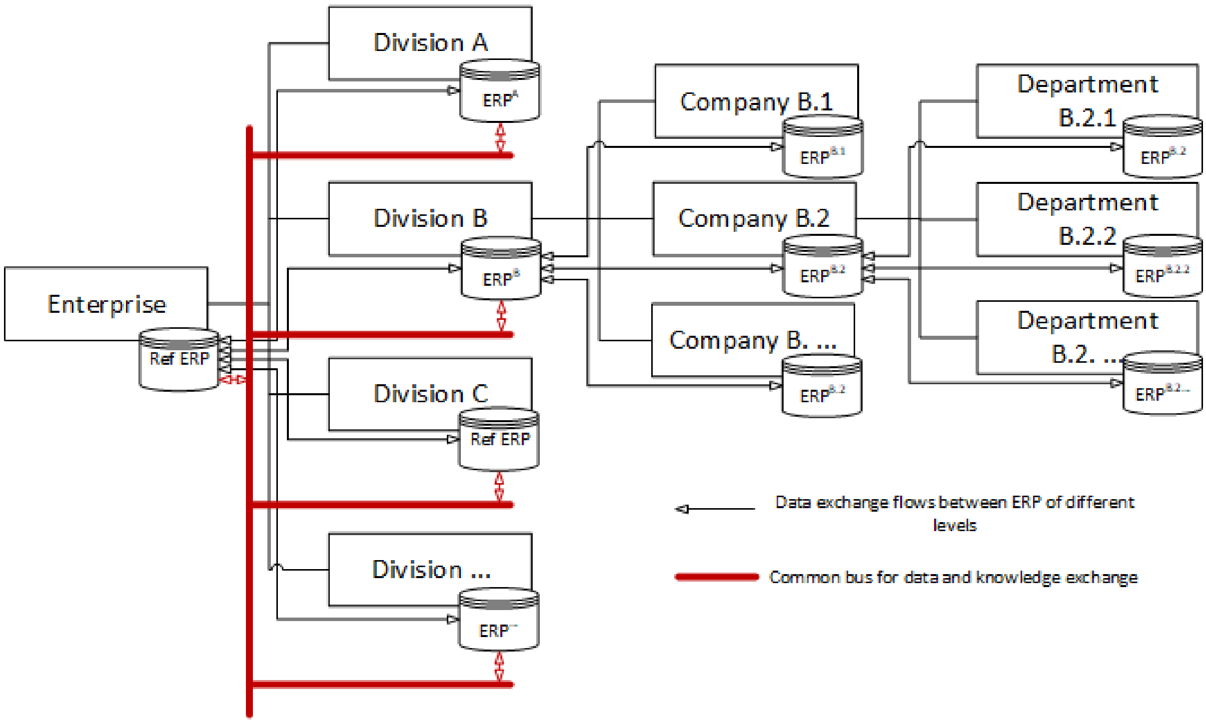

Figure 4: The result of building a single integration bus at the first stage, “top-down”

Second, a messaging standard should be developed. It should be noted that such a standard should include a description of information flows characteristic of information systems: data, regulatory reference information, and regulatory documentation [7]. This part of information may well be covered by one of the EDIFACT standards or their combination. The other part of the standard should concern the transfer of knowledge between intelligent systems. In [8] and [9], attempts were made to formalize such interaction for the implementation of an intelligent platform for managing complex risks. On the basis of these as well as other works, the following types of information characteristic of the exchange between intelligent systems can be distinguished:

- initial data for training intelligent systems;
- knowledge presented as facts;
- knowledge represented as rules or structures of intelligent models.

Based on this information, a set of messages can be developed, that are most commonly exchanged in the following types of interactions:

- intelligent systems with each other;
- intelligent systems with accounting systems;
- accounting systems with intelligent systems.

When the two stages indicated above are completed (the bus has been built and a set of messages has been developed), the bus and the rules of interaction can be brought into productive use. After that the stage of transition to the next level (stratum) can be started, where exactly the same bus will be built, but at a different level.

An important feature of this stratification approach is the possibility to start building the bus from any level of the system, i.e. both “top-down” and “bottom-up” options are available (Figure 5).

Of course, with this approach, collisions can arise when at different levels of the organizational hierarchy the messages identical at the time of building the bus are implemented by different types of messages. Such features of the bus can also be considered an advantage, since further development of accounting systems can lead to a significant discrepancy between such messages in their intended purpose.

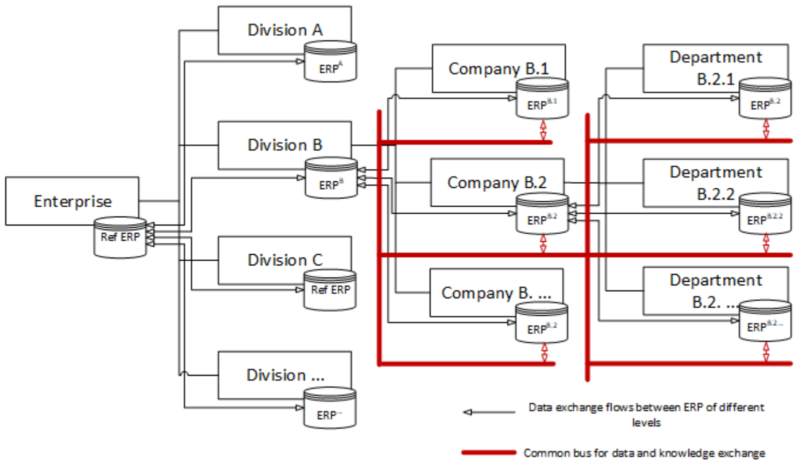

Figure 5: The result of building a single integration bus at the first stage, "bottom-up"

5. Conclusion

The proposed approach to the integration of corporate systems and intelligentware has several advantages.

First, it provides building of a single data and knowledge bus of an enterprise, which, in turn, having previously divided the monolithic solution with fixed integrations into many independent systems with unified interfaces, provides modernization of each system in parts.

Second, as a result of forming a unified standard for intelligent model messages, it provides not only intellectualization of individual systems, but also exchange of knowledge within the entire holding.

Third, if such a bus can reach not only the level of a separate system, but also the level of a separate function, this will allow the transition to modern microservice architectural solutions [10] not only in terms of the accounting system itself, but also in terms of intelligentware and intelligent systems. The next step in this direction is the transition to serverless solutions [11], which can become the next step in the evolution of information systems architectures.

6. References

[1] V. V. Sokolyanskiy (et al.), Architecture of modern intelligent transport systems and ways of its implementation in the economic infrastructure of cities, Problems of economic sciences, 4, 74 (2015) 82-89.
[2] M. A. Belyaeva, O. V. Buresh, T. N. Shatalova, Development of an integrated decision support system for project management in conditions of uncertainty, Bulletin of Orenburg State University, 132 (2011) 43-48.
[3] G. V. Rybina, Modern intelligent systems: integration problems and peculiarities of software development, Devices and systems. Management, control, diagnostics. 3 (2011): 12-34.
[4] G. V. Rybina, Intelligent systems: from A to Z, Nauchtechlitizdat, Moscow, 2014.
[5] M. G. Radchenko, Professional development in the 1C: Enterprise 8, 2nd ed. Piter, Saint-Petersburg, 2012.
[6] UNECE, UN/EDIFACT Specification, 1999. URL: https://www.unece.org/cefact/edifact/welcome.html
[7] V. V. Borisov, I. A. Bychkov, A. V. Dementyev [et al.], Computer support for complex organizational and technical systems. Hot line –Telecom, Moscow, 2002.
[8] A. V. Senkov, The format of the decision support system knowledge base for intelligent management of complex risks in complex organizational and technical systems. International Journal of Information Technology and Energy Efficiency, 2, 5 (2017) 23-34.
[9] A. V. Senkov, Intelligent software platform for complex risk management. Fundamental studies, 11, 2 (2017) 325-330.
[10] K. Richardson, Microservices: Development and Refactoring Patterns, Piter, Saint-Petersburg, 2019.
[11] M. Roberts, Serverless Architectures, 2018. URL: https://martinfowler.com/articles/ serverless.html