Integration of IT Systems with the Use of Microsoft BizTalk Server Platform

Tomasz Krawczyk

Abstract. The paper presents an approach to designing complex IT systems based on the EAI architecture. Enterprise Application Integration (EAI) is an integration framework composed of a collection of technologies, methods, and tools aimed at modernizing, consolidating, and coordinating IT systems. The work also gives a description of the possibility of using Biztalk Server as the software platform to implement systems integration infrastructure. An example of a system that can provide an important part of complex integrated system is the GRANICE system. Although the GRANICE system is fully functional environment, using the integration platform which is an intermediary in the communication with other systems,we can significantly extend its functionality.

Keywords: integration of IT systems, service-oriented architecture, enterprise application integration, Biztalk server, message, pipeline, orchestration, system GRANICE.

1 Introduction

Having comprehensive precise and sufficient information significantly determines the relevance and effectiveness of decision-making, understood as making an informed choice between different variants of problem solving. The information has to contain data processed into understandable form for people who use it in order to be effectively used. The use of information depends largely on possible transmission among particular entities. Nowadays, complex IT systems are responsible for collecting and transforming data set into information [0]. Usually a given organization uses a few or several different information systems and services of external

Tomasz Krawczyk
WASKO S.A., Berbeckiego 6, 44-100 Gliwice, Poland
e-mail: t.krawczyk@wasko.pl

A. Nawrat et al. (Eds.): Advanced Technologies for Intelligent Systems, SCI 440, pp. 195–210.
springerlink.com

companies. Each of these systems is responsible for processing data covering only a dedicated area which is a part of the activities of the organization. Very frequently encountered is a situation in which some data are duplicated across multiple systems, which in turn leads to the fact that these data are inconsistent (not sure which system stores the up-to-date data). Usually applications being an essential part of an IT system are located on different operating systems, they use different database engines, utilize different communication protocols for communication, they use different data structures characteristical for specifications of its operations and are developed in different technologies. From the perspective of the end user, it does not matter which technology was used to collect and transform data, he expects the information that will allow him to make the right decision. In most cases in order for information to carry useful content it has to come from several sources and be processed by many IT systems. Sometimes it is necessary in order to supplement information to refer to external data sources or the use of external systems (not being a part of the IT structure of the organization) to transform the information into the form desired by the user. Several years ago preparing by only one IT system expected and useful data, (which usually was designed to process information from only a dedicated area) required more and more time, which resulted in substantial increase in costs. With the increasing requirements as to information systems provided by users, the closure of the functionality in one system became more and more troublesome. Besides, it was noted that chosen functionalities of various systems fulfilled the same tasks and that a similar task is carried out by some systems more efficiently than others. Companies that specialize in providing dedicated solutions appeared.

2 Service-Oriented Architecture and Integration Techniques

The increasing challenge posed to information systems concerning the delivery of information meant that there was a need and indeed a necessity to ensure cooperation between different IT systems, both within a single organization and among organizations. This cooperation included both data exchange and the use of services offered by other systems for data processing. It became necessary to create infrastructure of an IT system in which different systems work together - therefore a need occurred to integrate systems. There also appeared the concept of distributed systems, consisting of independent and cooperating components, which are characterized by transparency. As part of such environment the data processing takes place in different systems, but from the perspective of the end-user this environment is seen as a consistent, complete entirety. The process of transformation data under distributed computer system is directly related so-called distributed processing, that is, information processing, in which discrete components (both hardware and software) may be deployed in different locations (geographical and / or organizational) [5] [2]. In the overall assessment the key features of a good distributed computer system are the following:

- Openness to extend and adding functionalities by attaching new elements,
- " Scalability - the possibility of using the services offered by a growing group of users without significant loss of performance.

One of the first approaches to integration of applications in distributed systems was architecture based on point-to-point, meaning direct connecting of components. This solution requires a thorough knowledge of integrated systems, i.e., their communication protocols and data structures. For each of the two systems there is a need for preparation of a separate application which is responsible for the exchange of information between these systems. The situation becomes more complicated if we want to integrate more than two applications. For n systems, the number of such connections amounted n x (n-1) /2 in the worst case and twice as many interfaces, since each pair of A-B applications required creation of two interfaces (one responsible for communicating with the application A, using communications protocol for this application and the second one for communication with the application B) [2][12].

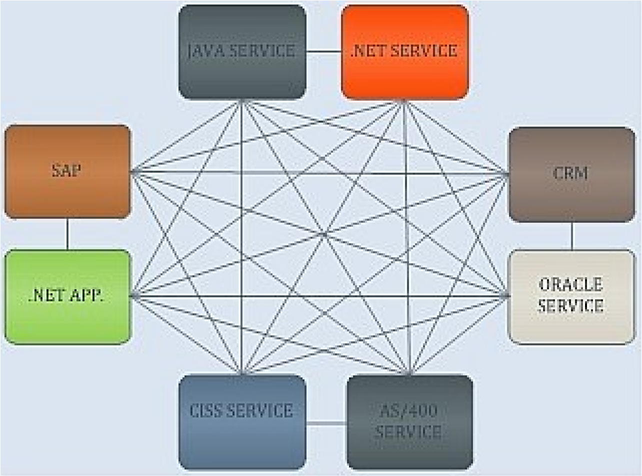

Fig. 1 Point-to-point architecture

The solution according to the original assumptions met its expectations, however, rising expectations of users and the need for cooperation with an increasing number of systems has not proved well. Currently placed demands impose for created distributed systems the need to use already existing components without the necessity to create a whole system from the start. Individual components of a complex system running on different platforms should be treated as indivisible components, which are the basis to create a business process. This approach forces heterogeneity, i.e. varied size of data structures, different functionality, different data models in place of harmonization or unification of all components, tools and communication protocols. Distributed systems are naturally heterogeneous. The creator of the system usually must use components with different properties performing different tasks on different platforms or using different data structures. The system consists of a number of tools: mainframe stations, SAP hosts, databases, services or applications [2]. Changing requirements make the need to implement a new element, different from hitherto ones. During this operation, new business functions are added and components are changed. This is possible if the underlying architecture tolerates heterogeneity and allows for flexible management - architecture point-to-point does not meet these requirements [2]. Problems associated with the capability (e.g., services) of entering the run-time interaction of two software artifacts with the aim of reaching common objectives (interoperability) appear when we want cooperation of services or data sets that are heterogeneous [3].

We can distinguish several levels of heterogeneity which result from the need of different IT systems to communicate within the distributed system. These include [7]:

- heterogeneity on the syntactic level such as conflict in encoding characters or serialization of data set,
- heterogeneity on the structural level - which results from the use of different types of representation of information about the same object,
- heterogeneity on the semantic level - due to discrepancies in the expected meaning of concepts used.

The modern process of integration assumes acceptance of the strategies, in which already existing systems will be an integral part of the newly created distributed system. Often, it forces solution to problems arising from heterogeneity on syntactic and structural level, and it leads to problems with stability and backwards compatibility. In response to new requirements for cooperation between different systems there came the concept of creating IT systems, which puts the emphasis on defining services - SOA (Service Oriented Architecture) for creating business processes with different owners. SOA based concept accepts heterogeneity of systems and places special emphasis on flexibility in implementation, which should be focused on the possibility of continuous changes, extensions and updates [2]. SOA does not constitute specific architecture. It is based on the assumption that business functionality is included in services. Focusing on the business value of interface, the service creates a bridge between business and IT environments. In the SOA paradigm we can distinguish three main concepts [6]:

- SOA Service aims at separating the business aspects of the problem from the IT software representation and placing implementation of this aspect in the service. Technical details should not be visible outside. Only the interface understood for business environment should be exposed,
- Interoperability - SOA shall enable interaction between two software artifacts, which in order to achieve their goals have to communicate with each other during the run,
- Loose connections - SOA reduce the links between systems, thus limiting the modifications or errors and greatly increase flexibility.

In SOA, we can also distinguish three main components, which include [6]:

- Infrastructure,
- Architecture,
- Processes.

Infrastructure defines the tools to achieve high interoperability, such as transformation (transformation of data), routing, service management or security provision. The architecture is necessary to reduce the specific implementations of SOA, so that they become functioning managed systems. The processes are responsible for the realization of tasks. The process involves the concept of business process modelling i.e. preparation and defining of services that constitute a business process. Service-oriented architecture proposes the implementation of business products through the execution by the instrumentation as atomic as possible and reusable services which are technological beings. In the mature SOA architecture service created for a new business product is a complex service benefiting from already existing services, and its execution time is relatively short [6]. One of the potential implementations of technical demands of SOA are Web Services. However, Network Services do not meet all relevant demands of SOA, these are in particular guarantees of interoperability and assurance of sufficient level of loose connections. In recent years there have been new concepts concerning construction of distributed systems, so that they met the assumptions proposed by SOA. In the most important ones we can include the systems based on Enterprise Application Integration and Enterprise Service Bus. EAI should be understood as middleware type application software (platform) to create architecture integrating heterogeneous systems, by ensuring the feasibility of business flows. It is based on a centralized the hub-and-spoke architecture, in which all communication takes place via a central hub. This solution enables data sharing between multiple information systems and automation of distributed business processes within the company [2]. In the hub and spoke architecture connections of (integration) applications occurs in the central hub. The connection to the application is done by using so-called adapters (spoke). They are responsible for the connection to a given application as well conversion of downloaded data from the application to data format in which hub is operating and vice versa, i.e. conversion of hub data into the data format used by applications [4].

The basic assumption of ESB assumes that each system connects only with the bus to be able to communicate with each system connected to the bus. ESB postulates the existence of native intelligent transport layer to which any system can be connected by plug-ins.

Services available within ESB are independent from the transport layer (they can be used by Web Service, JMS, MSMQ, etc.). Messages flowing within the ESB can be redirected and modified according to specific rules. Modern bus provides monitoring of operation of services as well as virtualization of services through load balancing, failover, mediation between transport protocols, often it has a BPM solution (Business Process Management) for implementation of long-lasting business processes. [4].

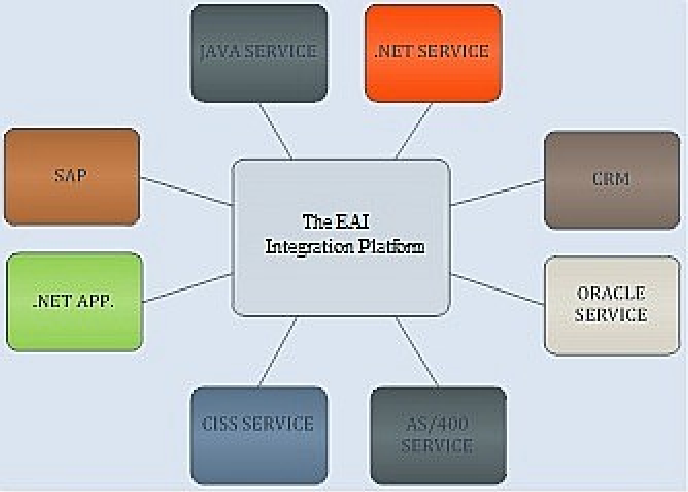

Fig. 2 The EAI Integration Platform

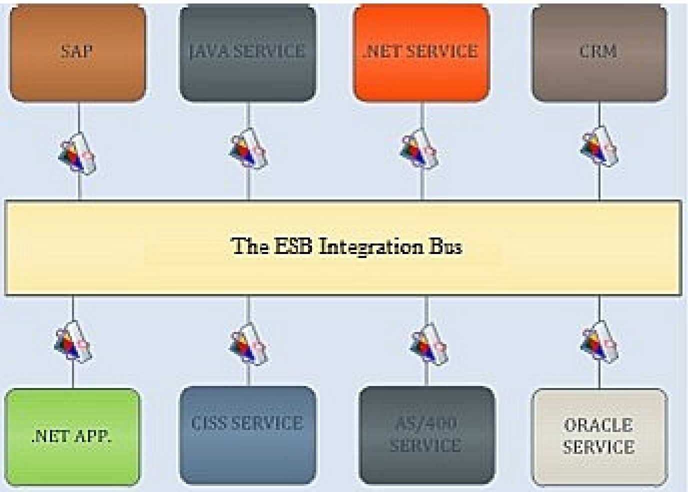

Fig. 3 The ESB architecture

3 Microsoft Biztalk Server - Integration Platform

One of the systems of this type (providing support for the architecture of EAI as well as ESB) is a Microsoft's BizTalk Server commercial product. It provides the infrastructure to the integration of applications as well as it provides an environment to connect and build together any environments, consistent with the vision of SOA. It provides the functionality of the integration of data exchange and integration on the level of business processes. Microsoft BizTalk Server provides environment for designing business processes responsible for the exchange of messages, and it has mechanisms allowing to define the acceptable definition of input and output data and patterns of their transformation. On the one hand it is the tool for connection and integration of previously independent applications which were running along with the establishment of business rules, standards and interfaces of the communication, on the other hand it can serve the same role in relation to entire systems, between which it is necessary to build mechanisms for data exchanges and joint services. Thanks to BizTalk server exchanging information between systems in a complex infrastructure is based on industry standards and architectural patterns such as: (SOA) Enterprise Application Integration (EAI), Business to Business (B2B) Complex Event Processing (CEP) and Enterprise Services Bus (ESB) [10]. The main elements of Microsoft BizTalk Server are Messaging Engines as well as so-called orchestration (Orchestration Engine), which form the basis for integration architecture of data exchange between different IT systems. The role of the messages engine is receiving incoming messages, the analysis of these messages in order to identify their format, content analysis to determine how a message should be processed, the delivery of processed messages to the destination and tracking the status of the processed message. The role of the orchestration engine is to perform the business logic according to the specific business process. Business Process is a set of activities that meet the desired needs of the end user (may take a minute, hour or even several months) [10]. Microsoft BizTalk Server performs tasks with the use of publishing and subscribing model (publish-subscribe). In this model there is used one-way synchronic broadcasting communication mechanism (which in some cases may require the use of an external program of infrastructure Publish / Subscribe). Input data are received by input port, which consists of two parts: an adapter responsible for the reception of data from the source system as well the input stream. Input - receiving stream consists of four elements, whose role is to perform the following tasks: Decode (Decode) - responsible for decrypting or decoding the message, disassemble - the breakdown (Disassemble) - which is transforming the message, checking (Validate) - responsible for checking the correctness message (if the message is the syntactically correct and that it meets the requirements specified by the XSD scheme) and determined the identity (Resove Party) used to establish the identity of the sender (e.g., signature verification if it is defined). The main task is converting the input stream received data into an XML format (the main format used BizTalk platform) - received data have a message form [10].

XML is extensible tag language, platform-independent, which allows easy replacement of documents between heterogeneous systems. Requirements for XML document structure can be determined using the XSD scheme. For the needs of searching documents in XML format XPath has been created, which provides an expression enabling to refer to any elements or attributes of the document satisfying the required conditions. For the transformations of the document described by a single XSD scheme output format compatible with any other scheme the XSL transformation is used [13]. Data received by the adapter and processed by the input stream at a further stage of processing have the form of a message in XML format

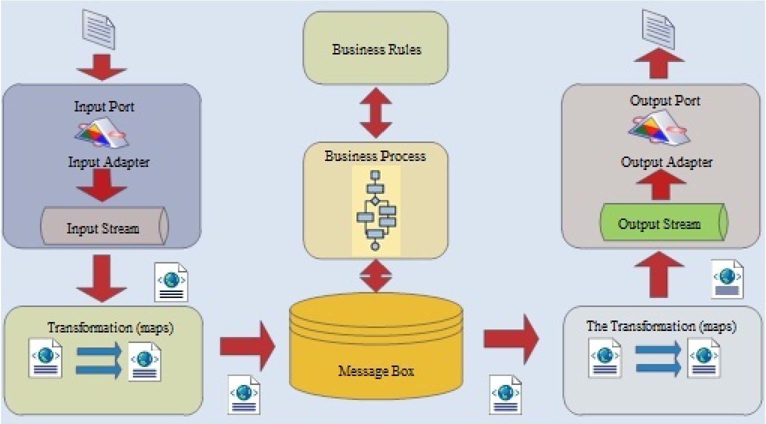

Fig. 4 The idea of the BizTalk Server integration platform

with strictly defined, described structure of the XSD schema. Sometimes it is necessary to change the structure of message to another. Transformations can be simple or complex and may involve combination (data grouping) or require additional calculations. BizTalk Server uses so-called maps for the description of the definition of such transformations. The maps contain business logic required for data transformation between two different schemes. Graphic editor provided by the manufacturer is used for building maps, which comes down to a graphical identification of relationships between elements (or attributes) of XSD schemes that describe converted messages. These relations can be extended with so-called functoids - (there are over 70 availoable as well it is possible to build one's own) i.e. special functions that allow to perform additional operations on data (e.g., a combination of two values of text type). Received and processed messages go to so-called inbox messages. Message box is a database called the MessageBox on the MSSQL Server, which ensures that the data will be sent to the sender - will not be lost in case of accident or error. Stored messages can be transmitted directly to the output port, which consists of the output stream and the adapter. The task of the output stream is to transform data from XML into the format expected by the target system, while the role of the adapter is providing data to the target system. Transmitting - output stream consists of three elements responsible for the following tasks: preparing message to ensure that the message can be created for sending (Pre-assemble); creating messages to send (Assemble) - by performing additional operations, such as the transformation of message from xml format to the flat file format; encoding (encode) - responsible for the encryption or (and) coding the message. The need to perform additional operations on the received message is often, that is the business process execution according to the established workflow. Workflow determines how the message flow between objects involved in the processing and defines what operations are going to be exdcuted at a given stage of processing. To describe the workflow in Microsoft Biztalk Server so-called orchestrations are used, which control the flow of business processes, and the orchestration engine is responsible for their run. Similarly to creating maps orchestration is done by using a dedicated graphical editor [8]. For descriptive purposes of orchestration dedicated BPEL language has been created. It is used for specification of the parties (partners) involved in the processing messages sent among the parties and describes the business logic of the process. In the BPEL language we can distinguish two types of actions: simple and complex. Simple actions are: service call, receiving and replying to queries and appending data to the message. Complex actions include the following structures: sequence, parallel execution, or so-called nondeterministic choice instructions (while, switch and pick) [9]. Environment of orchestration handles all the difficult issues related to implementation of long-term, dynamic processes, such as transactions, compensations, control flow, interconnection of messages. Orchestration engine uses business rules engine, which provides automation of rules and business intelligence, taking over all the tasks associated with sequencing commands and control flow. It is used to define and modify a dynamically changing business rules, written in the form of rules. The rules are expressed in natural language using dictionaries or with the use of describing elements of the rules by means of names easy to understand. A powerful inference mechanism determines the result of the implementation of rules and may deliver other rules so as to reflect a change that has taken place in the environment. The executive mechanism of rule engine finds the need to comply with additional rules depending on the state of the object changed by execution of the previous rule. Within processing orchestration engine retrieves messages from the inbox messages, converts them and re-writes the processed message in messages box, which then goes to the output port [10]. Adapters for BizTalk Server system play a very important role - they are responsible for communication with external systems. They provide support for data streams specific to a given technology (along with support for metadata associated with a given receiving or broadcasting protocol). BizTalk Server provides a large amount of ready-made adapters, which can be divided into adapters connected with given communication standards service (such as MSMQ, HTTP, SMTP, FILE, FTP, WS *, WCF) service of industry protocols (such as HIPAA, HL7, SWIFT) and application adapters - dedicated to communication with given applications (SAP, Siebel, Tibco, Oracle ESB / DB, EDI) and outsourced companies adapters [8]. Additionally, there is a possibility to create your own adapter based on the Adapter Framework. To the main advantages of using BizTalk Server as an integration platform we can include [8]:

- Effective communication both internally and with external entities,
- Automation of communication with other organizations' applications and between applications within the company,
- Ability to define business processes and event management, and fundamental in-process productivity ratios to ensure its effective and efficient action,
- Building dedicated solutions for every sector in any geographical region,
- The possibility of using a standard set of reusable services, which significantly reduces the costs associated with the creation of business processes,

- Modular - an extensible architecture that allows for adding plugins (adapters) of independent producers, facilitate changes in the structure and standardization of message,
- The use of standard - generally used technologies such as XML and Web services,
- The ability to build business processes graphically,
- The ability to monitor business activity - downloading information about the up-to-date state of the process,
- Integration with other Microsoft products such as SharePoint, Office, or products associated with Business Intelligence,
- Extensive programming environment - Visual Studio and Team Server or MS Project,
- Relatively low cost of implementation and maintenance.

Due to its extended possibilities BizTalk Server may be used for the following scenarios [8]:

- Integration of the application on the level of data exchange,
- Automation of processes in which the process requires the use of external services to perform a given activity, which is a part of the process,
- Infrastructure optimization, so that it will be possible in a simple way to join new systems to the existing infrastructure and providing efficient and effective cooperation between them.

4 Integration Capabilities of GRANICE System with External Systems

The system GRANICE consists of two basic elements into which we can include flying devices and software. The IT system established within the project contains many complex components, among which we can distinguish:

- software responsible for communication with a flying object, which includes downloading data from the object and transferring basic control commands,
- software for monitoring objects - including monitoring of primary flight parameters (including parameters related to the behaviour of objects such as fuel consumption, speed, etc., and characteristic parameters for the flight such as location or image from cameras), including applications enabling to visualize these parameters and in the case of data concerning the position showing the current location of an object on the map,
- software to organize and manage missions, including applications enabling to plan the mission in details, including the set operations to be executed on each stage of the mission,
- Software that lets retrieve flight information of an object at given time,

- Software that lets managing authorisations for both objects (through which applications can communicate with the object, and which applications have access to data transmitted from the object),
- A series of small applications that implement other functions (monitoring system activity, etc.).

This system is a complete entirety fully executing the tasks set for it. Due to its specificity and the fact that during performance of tasks it collects and processes large amounts of data, it can become an integral part of a complex system responsible for full service of flying objects. Therefore, essential issue is to ensure cooperation with other IT systems. The software developed under the project meets principal assumptions of SOA, and communication between particular system components is based on services, however ensuring cooperation with other IT systems, especially those that are not based on services, without using the integration platform is difficult. Therefore, it becomes justifiable to use such an integration platform based on BizTalk Server. The integration may involve both cooperation on the level of data integration - a source of the system can be both GRANICE system and any external system - as well as cooperation on the level of process automation. Ensuring cooperation with other systems shall significantly extend the capabilities of the system itself and improve processes associated with maintaining the entire infrastructure including both software, processes and flying objects. Access to the services available on the platform GRANICE in case of the use of integration platform based on Microsoft BizTalk Server is possible through the use of an adapter WCF (Windows Communication Foundation) and appropriate for the database engine adapter (such as MS SQL Adapter), because information collected during the mission are stored in a database. In the future it is planned to create a dedicated adapter BizTalk for communication with the system that shall unify access both to data and services offered within the system. The way to exploit the opportunities offered by the GRANICE system integration with external systems depends largely on the goal that we want to achieve, that is why it is important to be focused on business issues rather than on technical ones. Scenarios for using integration may include a number of areas of cooperation. The most typical one is the scenario in which an external application requests access to data collected during the flight for its use for example to analyze. The GRANICE system data flight parameters are sent in real time (with a slightly delay of not more than 1 second) from an object to authorized monitoring applications, they are also archived in a database. The data format within the system has a specific structure, well-known and acceptable for applications that are a part of it, the format of the data is not known to an external application, which expects data in its format to function properly. It is therefore necessary to convert data from the GRANICE system data format to external applications format. Another requirement posed by the external application is that it only expects certain data, such as object position data only. If you use the BizTalk platform the solution of the problem defined in this way is to prepare a map, in which we may specify what data from our system are to be passed to the external application. An example of an external application may be the reporting application whose task is to prepare reports on the mission performed. It expects data containing information about the mission start time, starting points, the level of fuel, the information about when the characteristics points were achieved and information about the end of the mission. GRANICE system provides the data to integration platform that runs the process responsible for transforming data into a format expected by the reporting application. Processed data are delivered to the reporting application, which generates the required reports. Another example of the use of an integration platform on the level of data integration is a situation in which the mission report is sent by e-mail to defined recipients. The list of recipients may be stored for example in a database or text file. Using available adapters, we can also solve a problem described in such a way. It is possible to publish a report created in SharePoint, because the network adapter is available for this system.

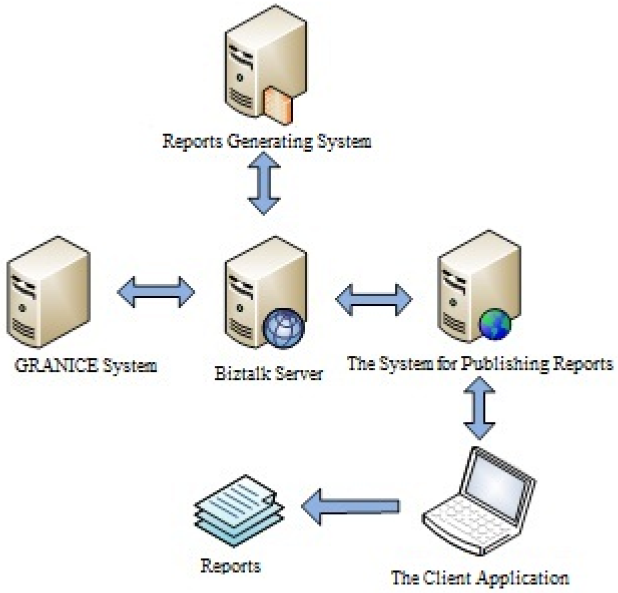

Fig. 5 The integration of the reporting system

Much greater opportunities for integration of applications using BizTalk Server platform are present on the level of process automation and optimization of infrastructure. A typical example of integration on the level of process automation can be a mechanism of building the optimal routes. As a part of the GRANICE system there is available a mechanism responsible for routing on the basis of defined waypoints. The point is defined by geographical coordinates (latitude, longitude and altitude). To plot a route between the points specified at the stage of building the route external services may be used. The role of external service will be appointing (or possible verification) the sets of points defined, the position between the characteristic points defined at the stage of creating the route. Designation of the optimal route may take into account limitations of terrain. An example here might be a situation in which a user specified a route from point A-B, but on the designated route there is a mountain, so the flight of a flying object can be difficult (may require raising the height of flight which could reduce the coverage, etc.). The role of the external software is to analyze all possibilities and choose the best option in order to accomplish a task. This may require adding a few or several new points between those determined by the user. Additionally, it is possible to integrate with weather forecasting system, which will constitute the source of information about atmospheric conditions. Atmospheric conditions have a significant impact on the missions performed by a flying object and can determine their success or failure. Knowing the time and the itinerary of the mission, using a weather forecast system, which is not an integral part of the system GRANICE, you can significantly extend the capabilities of the system. An example scenario is a situation when a given specific purpose of the mission (route, duration and the operations to be performed at each stage) can be supplemented with the data on weather conditions via the integration platform. Identified data may be the basis for deciding whether the mission is to be performed or not. An extensive system of approving plans for the mission may also be built on the base of integration platform.

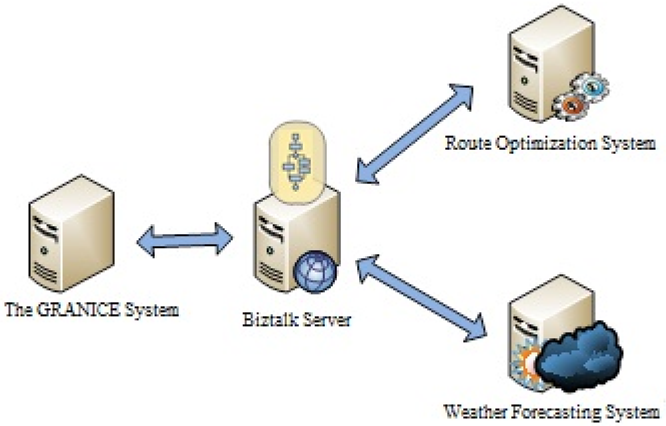

Fig. 6 The integration with routing and weather forecast system

After determining each mission so that it could be performed, there may be the need to obtain approval from a supervisor. Depending on the type of mission (e.g., if the mission itinerary includes a flight in the airspace of another country) it may require the approval of several people. Data concerning mission plan are received from the GRANICE system by the integration platform through a dedicated adapter, and then the workflow process starts, which is responsible for processing the received message and preparing response that includes information whether the mission can be executed or not. The process itself responsible for preparing the answers may be complex and last even several days. Depending on the nature of the mission (duration, monitored area, etc.) request for approval is sent to one or more persons.

This request may take the form of demand transferred to another system or a signed email message. Obtained response can also be in various forms (that is received by the adapter, then using the input stream converted to XML format). Depending on the information that represents the next steps of workflow path are executed. An example here might be a situation in which requires violation of another country space flight and may need the acceptance of minimum two persons entitled to this. Request for approval and details of the mission are sent to three people with such competence. In case of acceptance of two persons to the GRANICE system the feedback is sent that the mission has a positive opinion and can be done, otherwise the system gets information that the mission cannot be executed.

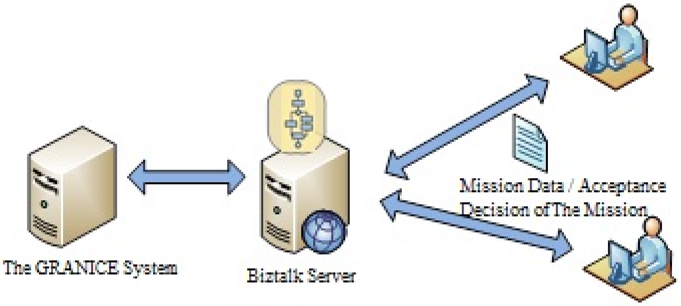

Fig. 7 An example of the integration- the mission acceptance

Executing missions in accordance with specified plan involves sending control commands to the object. These commands include both control of a flying object, as well as the commands responsible for operations carried out by the object (e.g., starting video recording, control of the camera, etc.). At the stage of mission planning there are designated only items necessary to execute the mission and the basic operations defining the tasks performed at a high level of abstraction (eg, start video recording.) For preparation of control commands for a given object we can also use external applications or services. The integration of external applications responsible for the preparation of these orders of GRANICE system can also be executed via the integration platform. Another example of using an integration platform on the level of process automation and optimization of infrastructure may be co-operation with the technical department. Important elements of the GRANICE system are flying objects, which are mechanical devices. Any mechanical device requires maintenance - needed are periodic reviews, repairs of any defects, or even refueling. All operations related to the technical maintenance require documenting in detail (information systems are usually used for maintenance of records). From the perspective of the user of GRANICE system user these operations have no meaning, a user requires only that a given flying object is ready for execution of the mission. Ensuring cooperation between people responsible for carrying out the mission and those responsible for the maintenance of infrastructure related to technical support of flying objects becomes one of the important issues. The mission execution may not be possible if there is no airworthy and prepared flight object. The user expects information about the disposal of the flying object at a time when a mission is to be performed. Technical workers expect information about planned accomplishment of the mission, how many objects will be necessary for its implementation and its duration estimatimation. With this information they can effectively organize their work and plan periodic reviews of equipment.

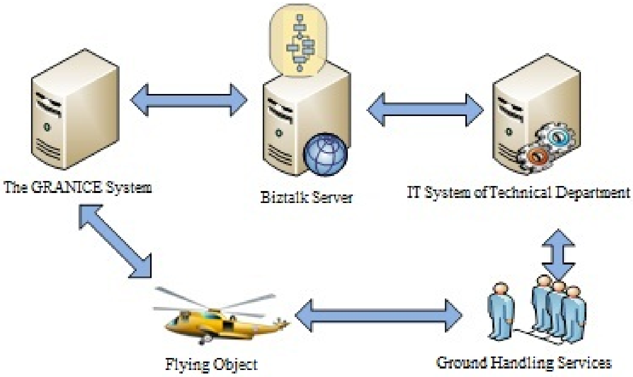

Fig. 8 Integration with the technical department

One of the possible scenarios for the use of an integration platform, may be a situation in which information about authorized missions (along with the planned execution time) go to the technical department information system, which allows the employees of this department preparing the flying object for the flight. The information that the object is ready to execute the mission from the technical department via the integration platform is sent back to the GRANICE system.

5 Summary

Although the GRANICE system is fully functional environment, nevertheless, using the integration platform which is an intermediary in the communication with other systems, we can significantly extend its functionality. Data collected during the flight, then made available through dedicated adapters, and transformed using the maps to the respective formats may be a source of input data for a number of external applications. Their use depends largely on the possibility of applications that use the data. The integration platform based on Microsoft BizTalk Server provides solutions that enable consistent, based on accepted and recognized standards, scalable mechanism for cooperation applications. As a part of the GRANICE system it will allow for creation of integration solutions in a simple way (most of the operations can be performed with the use of graphical editors). The integration platform will become the central element of the system in which there is concentrated information about the processes responsible for the tasks in charge of messaging among different systems. It will allow for creation of more and more sophisticated solutions that meet users' requirements. The cooperation of many different IT systems shall cause that certain processes are carried out more efficiently, and the end user receives valuable information, which is the basis for the reference of a measurable business advantage.

References

1. Banaszyk, P., Fimińska-Banaszyk, R.: Podstawy organizacji i zarzadzania. Wydawnictwo Wyższej Hanzeatyckiej Szkoły Zarzadzania, Słupsk (1997)
2. Brand, P.: Budowa usług z wykorzystaniem JCR (praca magisterska), Akademia Górniczo-Hutnicza im. Stanisława Staszica, Kraków (2010)
3. Brodie, M.: The promise of distributed computing and the challenges of legacy information systems. In: Proceedings of the 10th British National Conference on Databases, Heidelberg, FRG (1992)
4. Goel, A.: Enterprise Integration EAI vs. SOA vs. ESB, `http://ggatz.com/` (visited August 16, 2011)
5. ISO/IEC. Open Distributed Processing, Reference model (RM-ODP: Foundations, 1st edn. International Organization for Standardization, International Electronical Commission (1996)
6. Josuttis, N.M.: SOA in Practice. O'Reilly (2006)
7. Lemmens, R.: Semantic interoperability of distributed geo-services, Delft (2006)
8. Pelczarski, T., Widomski, P.: Microsoft Microsoft BizTalk Server 2010 - Integracja, workflow, monitorowanie procesów (2010), `http://www.microsoft.com/pl-pl/showcase/details.aspx?uuid=85e4391b-ca20-4bf5-a62e-f0bfe029b12a` (visited August 16, 2011)
9. BPEL4WS 1.1 Specification (2005), `http://xml.coverpages.org/BPELv11-May052003Final.pdf` (visited August 16, 2011)
10. Microsoft BizTalk Server, `http://www.microsoft.com/biztalk/en/us/default.aspx` (visited August 16, 2011)
11. Microsoft BizTalk Server Esb Toolkit, `http://www.microsoft.com/biztalk/en/us/esb-guidance.aspx` (visited August 16, 2011)
12. O potrzebie integracji aplikacji - BizTalk Server 2004 - Wprowadzenie (2004), `http://www.codeguru.pl/baza-wiedzy/aktualnosc,2109` (visited August 16, 2011)
13. `http://www.w3.org/` (visited August 16, 2011)