I. INTRODUCTION

One widely accepted definition of workflow comes from the Workflow Management Coalition [3]: “Workflow is the computerized facilitation or automation of a business process, in whole or part.”

Another definition is given in [1]: “Workflow management comes from office automation area, where all kinds of documents need to be digitalized and transferred among co-workers. Nowadays, the workflow management attracts a lot of attention due to its ability in modelling, executing, and monitoring processes. The processes can be not only the business processes, but also any procedures that need to be presented and managed. But the motivation for its booming is its promising usage in managing business processes.”

“The benefits of applying workflow technology to business process management are as follows:
- The business processes are explicitly defined, so that the responsibilities and the coordination relations are clearly determined.
- It is easy to optimize the business process because of their explicit definitions.
- Business processes are modularized and these modules can be reorganized by the Workflow Management Systems (WFMS) to form new business processes, so as to react quickly to unexpected changing business needs and conditions.
- WFMS can track daily operations.
- WFMS integrates applications on different platforms into a business process.
- WFMS provides personal workplaces.
- WFMS stem separates business logic of a process from the tasks themselves in the process. Therefore, the user in the workflow system does not have to deal with the route of the business process, but only concentrates on the task itself.” [1]

A general task for the development of the workflow system in the current business activities is the implementation of principles of the automatic control for business systems. These systems do not consist of pure technical components, but they integrate both human and human-computer activities and non-automatic interactions [5]. Thus, the implementation of the principles of the automation will benefit the exploitation behaviour of the business systems. The automation principles for the workflow system can be implemented, solving two general problems:
- To define a common framework for the different stages of the workflow system modelling, specification and execution of business processes and implementation workflows. This problem relates to the domain of the application of standardization approaches and methodologies.
- To develop and to use a set of software products, related to the main functional stages of the workflow system design and deployment. These problems concern the choice, utilization and integration of software products, related to the business model development, specification of business workflows, and software simulation of business processes and business execution of workflow tasks.

A set of solutions allowing modelling, specifying and executing business workflows using formal languages that can possibly be executed in specialized runtime environments is presented below. The diversity of these technologies and the existing workflow specification activities makes eBusiness very complex to understand and the choice of solutions very hard to achieve, more particularly for SMEs, which are small structures that, in general, cannot afford extensive researches and developments on their own. Here can be defined two key aspects that affect the entire life cycle of any eBusiness development: the choice of a choreography language and the choice of an orchestration language. These two groups of workflow technologies are central to the specification and execution of all workflows:
- Choreography is concerned with global, multiparty, peer-to-peer collaborations where business entities interact in long-lived state and coordinated fashion regardless of any programming model or supporting platform used. Choreography languages (e.g. BPSS, WSCDL, etc) cannot be directly executed (they are mainly descriptive) and require to be translated (mapped) to an orchestration language in order to be executed.
- Orchestration focuses on the behaviour of a single business entity, it is a hub and spoke model where a controller residing at a single location locally enforces...
the progress of a process by following its definition. Orchestration languages (e.g. BPLM, BPEL, XPDL, BPELJ, jPDL, etc) are executable languages and define a runtime environment for their execution.

Choreography and orchestration express the operational semantics of business entities involved in distributed services and complement each other. Choreographies translate global workflows between business entities while orchestrations translate local workflows to a business entity. Global workflows concern the exchange of messages between peers without any centralized control. Local workflows can be either external or internal to a given entity. External local workflows define the public external behaviour of a single entity and differ between entity’s roles while, internal workflows are hidden from the outside and implement external workflows.

Choreography and orchestration languages can be either graphical (e.g. BPMN, UML, etc) or textual (BPSS, WSCI, WSL, BPEL, etc). Alternative languages exist for both choreography and orchestration. Some can even be used for both, although their centre of gravity would be either around choreography or around orchestration.

Orchestration languages are typically high-level specialized programming languages although some languages or language extensions go much closer to general-purpose programming languages, e.g. like JSR207 and jPDL that facilitates the programming of business workflows directly in Java, or BPELJ that allows integrating Java code (snippets) directly in BPELJ code. Although they can be initially classified, these languages refer to different concepts according to their own creators. They are named for instance: execution language, modelling language, definition language, description language, etc. Understanding the exact differences between all these languages, their precise scope, their applicability to any problem and evaluating, which will emerge, is not an easy task.

Choosing the right language(s) is not the only challenge as many other technologies are also involved. The difference between the workflow technologies is not clearly distinguished sometimes. For instance, BPMN and UML are two graphical notations primarily used for orchestration although their embedded collaboration capabilities could also be used for choreography. Another example is BPSS for choreography that can be easily complemented for orchestration. Using multiple languages to specify the same workflows would allow to compare effective results and to trial different tools.

II. MODELLING TECHNOLOGIES

UML is a powerful modelling technology for developing business processes and information models. It is an open standard and it is developed specifically for the construction of business processes and information models. It is based on UML 1.x. and it is adopted by various industry use groups. The UMM is not well supported by tools and the knowledge about UMM is not widespread. Now it is considered complex and it requires a high level of expertise to use.

UML is an object-oriented modelling language from the OMG. It specifies and visualizes models of software systems. It has become the generic modelling standard applicable to any software development project, and knowledge of UML is widespread. In the area of business process modelling, the UML Activity Diagrams are of especial interest, particularly for modelling the choreography of business processes.

BPDM is an open standard providing an abstract model for modelling business processes that is being developed by the OMG. BPDM is still being developed and there are no tools supporting it.

BPMN is an open graphical notation standard for business process modelling [4] that was developed by BPMI, now merged with OMG. It is based on well-known flowcharting notations and thus intuitive for business analysts. The specification also provides a notation for more complex processes to be produced and mapped to BPEL (4WS), which enables it to be used as the first step in the business process development’s lifecycle. It has tool support, particularly for the mapping to BPEL, but also for XPDL. It can be recommended for use at the business process design’s stage, in particular for modelling the choreography of business processes.

Both UML and BPMN are recommended for e-business modelling. This is no contradiction as each language has its own merits, its own tool support and its own environment, in which it can best be used. While BPMN has a focus on business processes, UML has a focus on software design and therefore the two are not competing notations but are different views on systems.

III. CHOREOGRAPHY LANGUAGES

Choreography focuses on the composition of services. Every standard that is related to choreography has to specify how existing services can be composed and which protocols have to be considered between the participating services, during normal execution as well as in erroneous situations. The most developed choreography languages: WS-CDL, WSCI, WSL, and ebXML for BPSS are shortly described below.

WSCl and WSL have never reached the status of a mature standard. WS-CDL is a powerful choreography language but the standard has not been finalized yet, and its support from tools is poor.

BPSS is the choreography language in the ebXML family of standards. The family of BPEL specifications together with WSDL for service descriptions seems to be major trend in business process modelling. Considering that abstract BPEL processes only support the specification of the observable behaviour from the view of one coordinating process and do not allow the specification of the observable behaviour between n processes.

IV. ORCHESTRATION LANGUAGES

The creation of an environment where Web services, both internal and external, interact with each other is a very prospective and actual goal of the modern e-business systems. The term “orchestration” is introduced to describe the flow of such communication as a multi-step, long-lived business process from one party’s view. Languages used to write these processes in a way that software systems (work-
flow engines) can execute them are called orchestration languages.

4.1. Business Process Execution Language for Web Services (BPEL4WS)

BPEL can be a low-level language designed to describe business processes involving both: B2B and B2C relations. BPEL stands as a “bridge” between the programming-in-the-large graphical BPML language and the programming-in-the-small languages. Programming in the large can refer to programming code that represents the high-level state transition logic of a system. This logic encodes information such as when to wait for messages, when to send messages, when to compensate for failed non-ACID (Atomicity, Consistency, Isolation, and Durability) transactions, etc. Programming in the small, in contrast, deals with short-lived programmatic behaviour, often executed as a single ACID transaction. It allows access to local logic and resources such as files, databases, etc. BPEL’s development came out of a realization that programming in the large and programming in the small required two different languages. As numerous "small" programming languages already existed (e.g. C, C#, and Java), computer scientists felt no need to introduce another. BPEL code can be automatically generated from BPML diagrams. BPEL is still evolving and has strong support in the market.

4.2. The XML Process Definition Language (XPDL)

XPDL is used to define executable workflow descriptions, which can be executed on workflow engines. XPDL is as easy to read as an XML format. All attribute and element names are descriptive. There exist many workflow engines using XPDL as a process implementation language, including open source engines like Enhydra Shark, WiMOpen, or Open Business Engine. Modelling is also supported by many tools including Open Source Enhydra JaWE.

4.3. Java Business Integration (JSR 208/JBI)

JSR 208 was announced by SUN in 2005. JBI is a very promising technology. Some products are in stable versions and some of them are Open Source or are licensed under a LGPL. Its strengths are: Java based (reuse of existing components), only needs a J2EE 1.4 platform; Easy to integrate in projects; Possibility to build one’s own JBI compliant Service Engine; Lots of standards supported (SOAP, HTTP, JMS); Possibility to plug in third party components/upgrade them; Opened to Web services / external services.

4.4. ebXML Collaboration Protocol Profile and Agreement (CPPA)

ebXML CPPA provides modular process definitions as standard building blocks to develop more detailed process definitions, using the mechanism to include packages, and complex collaborations. This important step provides flexibility for interested user communities. They can add more capabilities and business functionality in an iterative fashion. The flexibility and constructionality are important aspects not only for the adoption of the used standards but their effective use and successful deployment into heterogeneous environments and across domains.

4.5. Business Process Execution Language for Java (BPELJ)

BPELJ is an open standard generally based on pure BPEL and has some additions enabling Java to be used in it. It is developed quickly because of the interest in it under the pressure of expectations of its improvement. BPELJ explores the new fields of business processes orchestration and no equivalent standard is being developed simultaneously.

4.6. Web Services for Business Process Design (XLANG)

XLANG is an extension of WSDL, the Web Service Description Language, introduced by Microsoft. XLANG is a relatively old language, thus well developed. It provides both the model of an orchestration of services as well as collaboration contracts between orchestrations. The goal of XLANG is to make it possible to formally specified business processes as long-running interactions. XLANG is an open standard, which uses WSDL to describe the service interfaces of each participant. The behaviour is specified with a control flow that choreographs the WSDL operations.

4.7. Windows Workflow Foundation (WWF)

WWF will be integrated in the next Microsoft Software Solutions such as Windows, Office, Sharepoint, and will enable “document-centric workflows for information workers” to be created. Another advantage is the graphical modelling/coding tool included in Visual Studio 2005. The major weakness is the platform and implementation technology dependency (Windows/.NET). WWF is not suited for wide-scale services orchestration. It should rather be considered as a workflow framework to build workflow-centred applications with abilities to address Web services.

4.8. Process Definition for Java (PD4J - JSR 207)

JSR 207 was submitted to JCP (Java Community Process) by BEA (2003). No product with PD4J support is available yet.

4.9. Java Process Definition Language (jPdl)

The main goal of jPdl is to describe the orchestration of services, like BPEL or XPDL, in an XML file. The logic part, consisting of states, tasks to be done, conditions or over-rides of behaviours must be done in delegated Java classes. jBPM provides a graphical process designer for jPdl, and a reference API for implementing tasks node, event handlers, etc.

4.10. The Web Services Flow Language (WSFL)

WSFL is an XML language for the description of Web services compositions as part of a business process definition. It is good for representing Web services as business processes and interaction between service providers. The WSFL description syntax is far too complex for
most users and it is necessary to use a graphical design tool to work effectively. This is an obsolete standard, replaced by BPEL4WS.

The set of the presented workflow standards is given in Figure 1.

WORKFLOW MANAGEMENT

- Modelling Standards
  - BPM
  - UML
  - BPD
  - UMM

- Choreography Languages
  - BPM
  - UML
  - ebXML BPSS
  - WS-CDL
  - WSC

- Orchestration Languages
  - BPEL1.1
  - XPDL
  - JSR208/JPDL
  - ebXML CPPA
  - BPELJ
  - XLANG
  - WWF
  - PD4J-JSR207
  - JPDL
  - WSFL

Figure 1. Workflow standards

5. CONCLUSIONS
The presented technologies define a set of solutions allowing modelling, specification and execution of business processes and implementation workflows using formal languages that can possibly be executed in specialized runtime environments.

6. REFERENCES

This research is partly supported by the European Commission, project NoFP6-027178 VISP, and National Scientific Fund of Bulgaria, project No BY-MH-108/2005.