INTRODUCTION

The field of business applications encompassing software for production planning or order management is dominated by some big vendors offering large standardized off-the-shelf application systems, e.g. SAP R/3, BAAN IV, or Oracle Applications. Despite all advantages of selling and buying these large integrated application systems, there are two major shortcomings: Usually only large enterprises can afford to buy, install, customize, and maintain these systems, and the complexity which is inherent to these still growing systems makes it harder and harder for vendors to enlarge and maintain them. Furthermore, markets for these large packaged application systems are almost saturated, because most of the large enterprises have covered their demand. On the other side, there are lots of small and medium enterprises (SME), which demand very specialized application systems together with a broad coverage of common business functionality, as the core of the related business tasks stay the same—indeedependent from a company’s size. However, nearly no SME can afford to purchase or even to maintain these large packaged application systems. This leads to two important problems: SMEs demand on business application systems can not be satisfied, and vendors of large standardized off-the-shelf application systems lose an important market. In addition, small and medium vendors of business application systems, which offer highly specialized applications, have to master high barriers to entry the software market, as they have to offer core business functionality as well, in order to sell their specialized products.

A way to avoid this emerging software (industry) crisis in the area of business application systems, is to achieve a goal which is followed up for a long time by the software engineering research community: to assemble software by using pre-defined reusable parts, so-called (software) components. Definitions of what exactly constitutes a component vary in the literature. For definitions and an overview of definitions given in the literature. (Szyperski, 1998, p 164-168). In common,
software components are mostly understood as self-contained units with contractually specified interfaces and functionality. They may be sold independently, and be composed by third parties (e.g. to form new application systems) in combinations not assumed by the component’s manufacturer.

Since we focus on the area of business application systems, we will use the term business component (BC) instead of component to emphasize our focal point. BC is a specialization of the term component, which names components that implement a set of functions out of a given business domain, e.g. finance or order management (Fellner, Rautenstrauch, & Turowski, 1999). For components in general, e.g. for graphical user interfaces or database access, some but not all of the assumptions made or conclusions drawn in the following may be valid.

Even though the combination of goals differ from each other according to a company’s size or role, e.g. large vendor or small customer, important driving goals behind systematic software reuse based on component-orientation are expected improvements concerning cost efficiency, quality, productivity, market penetration, market share, performance, interoperability, reliability, or software complexity.

Depending on the strategy a company follows up with component-orientation, different kinds of standards develop in variant standardization scenarios, e.g. a large vendor of off-the-shelf application systems trying to reduce his software’s complexity may define a proprietary “standard”, like Baan or SAP. On the other hand, industry standards may be (automatically) established by best practice, or open standards may be proposed by (independent) public organizations as the OMG (Object Management Group), not at last to expand markets and to avoid oligopolies. Keeping in mind that all mentioned scenarios have specific advantages for respective market participants, we will focus our contribution on discussing scenarios related to open standards, since it is common opinion that most improvements related to component-orientation will not prove true in this scenario if standardization fails.

After a short introduction to component-based business application systems, we address existing standardization approaches and the modeling layers to which they belong. E.g., some basic approaches which belong to the technical layer are already available, like CORBA (Common Object Request Broker Architecture), COM (Component Object Model), JavaBeans, or mediators. However, no single approach is generally accepted. On the domain-related semantic layer, a much bigger lack of standardization can be observed. One reason for this lack of standardization is because there are too many “standards,” and entities that try to establish their products as (industry) standards, often are just pretending to be component-oriented. Besides well-known approaches, e.g. from the OMG or OAG (Open Applications Group), contributions from standardization efforts in other areas are addressed, which do not originally focus on the task of standardizing BC, but are very well usable for this task, like UN/EDIFACT (United Nation’s Electronic Data Interchange for Administration, Commerce and Transport), BSR (Basic Semantic Repository), or business reference models as proposed by the German Business Information Systems research community.

Another reason for the lack of standards for BC is, that there is still a lack of
Related Content

Situated Discourse and Essential Terms
[www.igi-global.com/chapter/situated-discourse-essential-terms/29674?camid=4v1a](www.igi-global.com/chapter/situated-discourse-essential-terms/29674?camid=4v1a)

Virtual Worlds, Standards and Interoperability
[www.igi-global.com/article/virtual-worlds-standards-interoperability/46112?camid=4v1a](www.igi-global.com/article/virtual-worlds-standards-interoperability/46112?camid=4v1a)
Underpinning EISB with Enterprise Interoperability Neighboring Scientific Domains
www.igi-global.com/chapter/underpinning-eisb-with-enterprise-interoperability-neighboring-scientific-domains/125358?camid=4v1a

A Model for Building Trust in E-Government
www.igi-global.com/chapter/model-building-trust-government/43770?camid=4v1a