5TH INTEGRATED PRODUCT DEVELOPMENT WORKSHOP
IPD 2004 SCHÖNEBECK/BAD SALZELMEN b. MAGDEBURG
SEPTEMBER 23-24, 2004

KNOWLEDGE-BASED SHOE DESIGN PROCESS

Dr. A. Semenenko, Dipl.-Ing. R. Krikler

Keywords: Knowledge-Based Design, Custom-Made Shoes, Best-Fit Shoes, Rule-Based Approach, Knowledge Framework

1. Introduction

In the early 90's, JOSEPH PINE introduced the fundamental concept of mass customization [PIN–93], which in opposite to the traditional serial production, is a customer driven approach of producing industrial products. *Mass Customization* (MC) is defined by the synthesis of mass production and the manufacturing of customized products by combining the positive characteristics of both: cheap mass produced products and expensive customized products with high quality and short delivery times [PIN–94, TsJi-01]. The difference between MC and single-item production is defined by PILLER as a threshold value for cost per unit produced. If this value is not higher than 10-15 percent of a comparable mass product, then it is referred to MC, otherwise it is referred to single-item production [PIL-98].

Designing customized shoes is a knowledge intensive process. One of the crucial points in a modern shoe manufacturing lies in shoe design, which is a determining factor in selling best-fitting or custom-made shoes. That is why within the *Extended user Oriented Shoe Enterprise* (EUROShoE) project we put the main stress to the shoe design process (see Figure 1). According to the idea of mass customization, in the EUROShoE project a radical improvement of the concept of the shoe as an industrial and individual product is targeted to satisfy the demand of customer. Beside the individual shoe style and functional purposes mass customized shoes offer high comfort by using best-fitting or custom-made shoe designs [BoDu-02]. An optimal fitting shoe contributes to the maintenance of our feet and our healthy [OTT-84]. Nevertheless, shoes are often produced only in one shoe width per size to reduce the price of the shoe. As the size and width of shoes are oriented on average measurements, it is difficult or mostly impossible to buy optimal fitting shoes. In consequence, the demand for mass customized shoes that fits better is enormous [PFI-02].

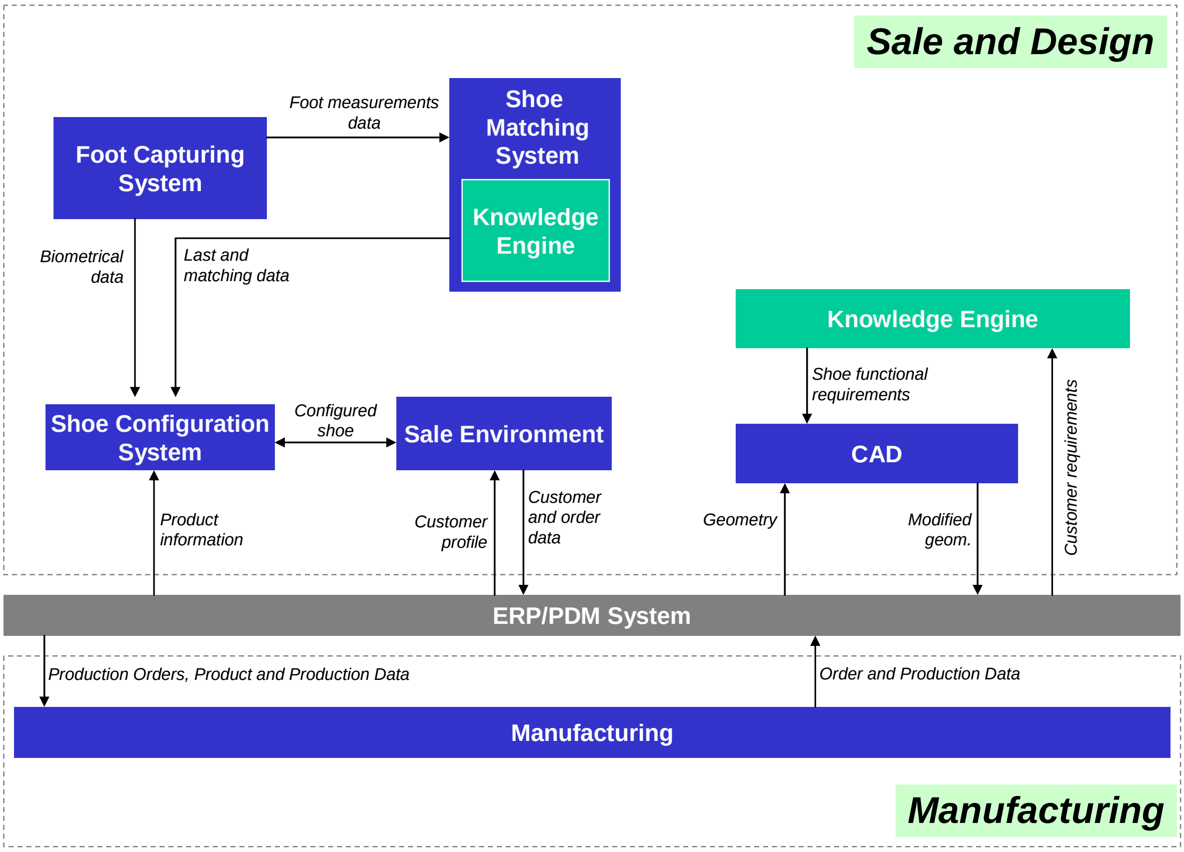

Figure 1: Overall architecture of the EUROShoE project

Generally speaking, there are two different approaches used in shoe manufacturing: best fit and custom-made. The main focus of this paper is set on the first approach. According to the first approach, an available last should be chosen to fit the customer foot. The next step is to order a shoe with a corresponding size and material. The custom-made method takes into consideration “quick adaptation functions” available in a new CAD system for last and shoe design. This shoe design must fulfill several following requirements: style requirements that depend on the model chosen by the consumer, foot size depending on the morphology of the consumer’s foot as well as biomechanical requirements that relate to the “function” of the shoe when used by the consumer. The output of the design phase for custom-made shoes is a complete product model specification of the shoe including all CAD data [GrLW-03]. The approach to support the shoe design process, described in this paper, is based on the *Knowledge Framework* developed within the project.

2. Knowledge Framework

In the following section, the EUROShoE’s *Knowledge Framework* and its components are described (Figure 2). *Knowledge Framework* with the used technologies and processes for defining, organizing and processing the design knowledge is the best way for designing best-fit and custom-made shoes. The aim of the *Knowledge Framework* is to gather and apply design knowledge, last fitting knowledge and last adaptation knowledge to simplify and support the design process. In particular, the aim of the *Knowledge Framework* is to store design knowledge in a way that it can be reused for future designs or quick modifications. With the application of last matching knowledge the process of selecting an appropriate last for a customer foot can be simplified and automated by a computer. The result of this process step will be the basis to produce best-fit shoes.

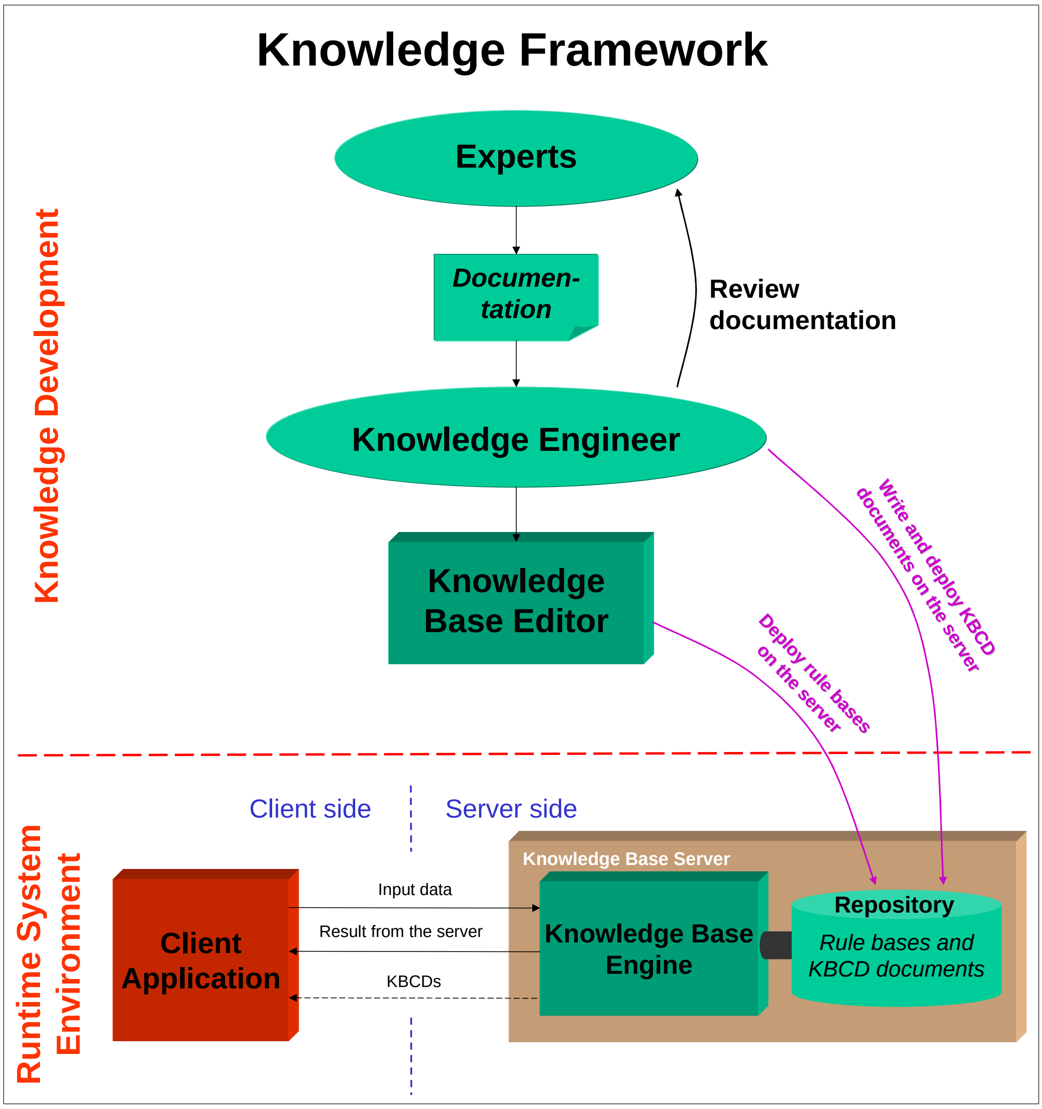

Figure 2: Knowledge Framework architecture

As the developed IT architecture is strongly distributed, the *Knowledge Framework* implements the client/server approach as described below:

- the server stores and processes the knowledge necessary for the shoe design
- the client collects input data for the *Knowledge Base Engine* and

- processes the results received from the server

The server application contains two main modules of the *Knowledge Framework*:

- *Knowledge Base Engine* to process the knowledge
- *Repository* to store the rule bases

In the following section a *Knowledge Engineering Process* will be described.

2.1. Knowledge Engineering Process

The *Knowledge Engineering Process* consists of the knowledge, obtained by the experts, modeled and prepared for further use by the *Client Applications.*

An essential role of the *Knowledge Framework* is to support simultaneous definition of design knowledge. For this reason one of the main requirements to the *Knowledge Base System* (as shown in Figure 2) is the flexibility concerning the rule bases. Thus, the developed system allows the user to define the rule bases independently from the application domain.

2.1.1. Knowledge Development

In the following section the *Knowledge Development* will be briefly outlined. *Knowledge Development* is a result of the execution of the *Knowledge Engineering Process* (Figure 3). The first step in the *Knowledge Development* is to collect and document all the information, i.e. rules, data models, etc, which is important for the following implementation of the *Knowledge Base*.

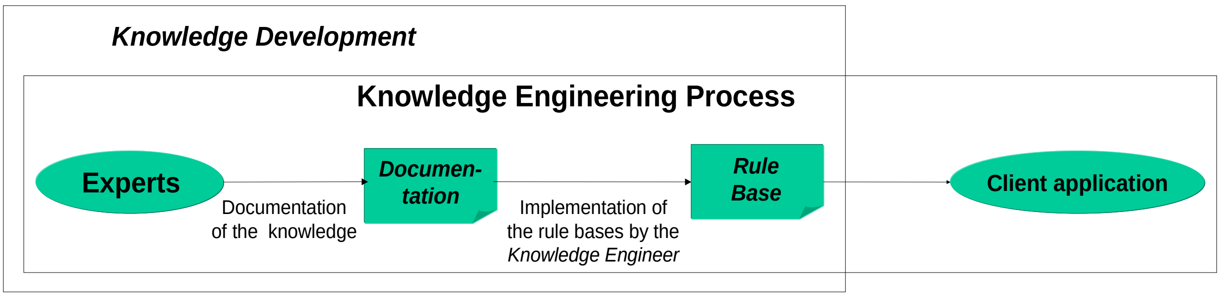

Figure 3: Knowledge engineering process

The implementation and exploitation of the knowledge is performed by a *Knowledge Engineer* with following tasks:

- reviewing each rule base concerning the documentation in terms of inner logic and the implementation within the input/output data model

- recommending changes for the documentation if necessary
- implementing each rule base with *Knowledge Base Editor* (as shown in Figure 2)
- collecting *Knowledge Base Content Descriptors* (KBCD) for each rule base

Knowledge Base Content Descriptors are provided to simplify the use of the *Knowledge Base system*. The KBCD describe corresponding objects for a certain rule base with the possible characteristics, which are used to create valid input data. Furthermore, the KBCD defines the input data model for a rule base in a platform independent manner as valid XML file specified through XML schema definition.

As a next step, the produced rule bases and KBCD are deployed on the *Knowledge Server*. After this, the deployed rule bases, combined with the KBCD documents, are available for the client applications in *Runtime System Environment*, described in the next section.

2.1.2. Runtime System Environment

The *Runtime System Environment* is a distributed support for the sale, design and manufacturing process (as shown in Figure 2). After the rule bases are deployed on the *Knowledge Server*, the client applications of the *Runtime System Environment* use the rules to produce results from the knowledge base. The additional deployed KBCD documents allow the input data model to be available in a platform independent form. A client application uses this document to produce input data for the *Knowledge Base Engine*. In Figure 4 the tasks a client application has to fulfill are shown.

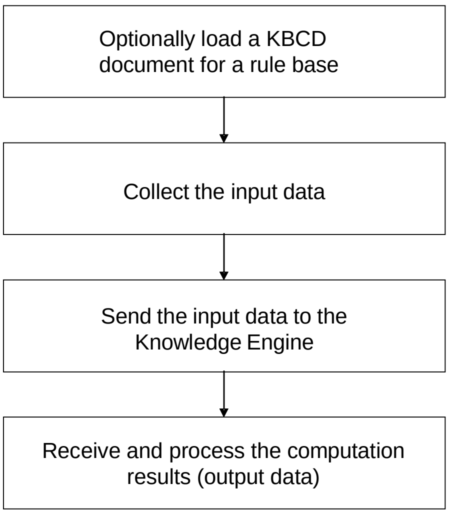

Figure 4: Tasks of a client application

2.2. Knowledge Base System

As described previously (see Section 2), the *Knowledge Base System* contains two main modules. These are a *Knowledge Base Creation Module* (KBCM) and *Runtime System Environment* of the *Knowledge Framework* (Figure 5). The KBCM consists of a set of applications called *Knowledge Base Editor*. These applications are of significant importance for the *Knowledge Developing* process.

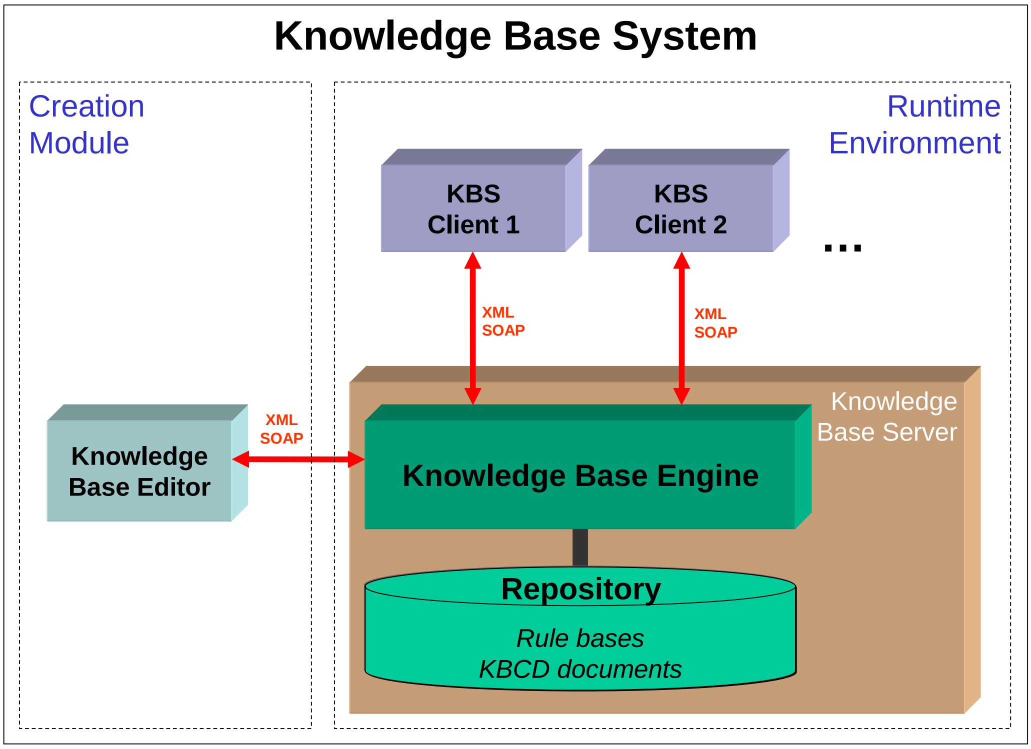

Figure 5: Knowledge Base System architecture

In this section the main blocks of the architecture will be described. The *Knowledge Base Editor* is a GUI-based (Graphical User Interface) software application to define and modify design knowledge in form of rules. This application allows to:

- fill the knowledge base with rules and facts using a user-friendly graphical interface
- test the information provided by the knowledge based adaptation system and finally
- manage the knowledge base, specifically inserting, modifying and deleting rules and facts from a rule base

As described in Section 2.1.2 the *Runtime System Environment* of the *Knowledge Base System* has three main parts:

- *Repository* to store and organize the rule bases of the *Knowledge Base System*

- *Knowledge Base Engine* to provide web service methods for the client applications
- *Knowledge Base System* clients to enable the communication with the *Knowledge Base Engine*.

These clients are the part of the IT infrastructure, as shown in Figure 2 and Figure 5. The main functions of these clients are:

  - Collecting input data, initial data and identifications of the knowledge base
  - Interacting with the *Knowledge Base Engine* web services
  - Receiving the output data generated by the *Knowledge Base Engine*

Communication and interaction between the different parts of the *Knowledge Base System* is based on XML and SOAP respectively on Web Service Methods. The *Knowledge Base Engine* processes the input data and the rule bases. The communication and data exchange process is shown on Figure 6.

The *Knowledge Base Engine* receives the input, as an initial data from a client application, and loads a rule base (identified by an ID) from a database. After that, it fills the inference engine (a part of the *Knowledge Base Engine*) object base with rules and input data and runs the inference engine. Finally, it returns the results of the process to the client. The input and output data interfaces of the *Knowledge Base Engine* are based on XML.

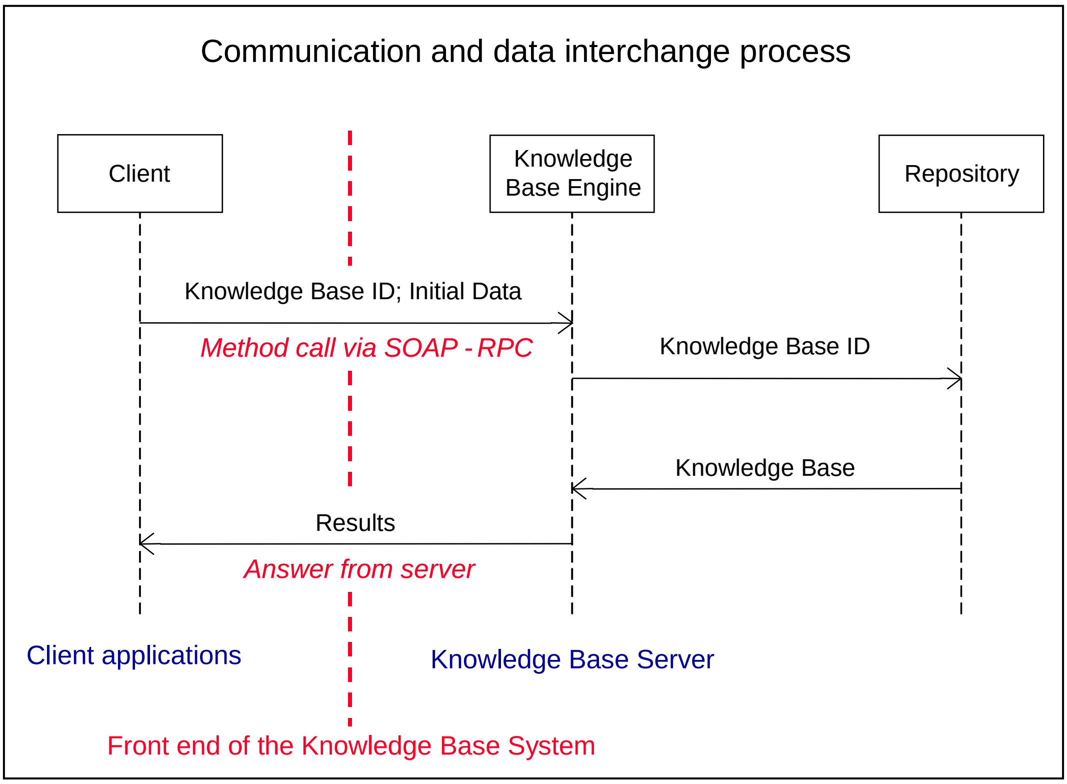

Figure 6: Communication data interchange process

It is important to mention, that the *Knowledge Base System* has no access to the client applications. As a consequence, the system is not dependent on any information about the post-processing of the produced results by the clients. As the communication and data exchange process are based on industrial standards like Web Services and XML, the platform independent communication and data exchange within the *Knowledge Base System* is warranted.

2.3. Validation of the Knowledge Base System

For test and validation *Institute of Applied Computer Science in Mechanical Engineering* (RPK) developed a client application as shown in Figure 7. This application uses all necessary technologies such as XML or Web Services that are required to build up a possible client application. Here, the *Knowledge Base Test Client* fulfills the following test purposes:

- network connection to the server
- implementation of the *Knowledge Base Engine*
- implementation of the I/O interfaces in the *Knowledge Base Engine*
- implementation of the rule bases

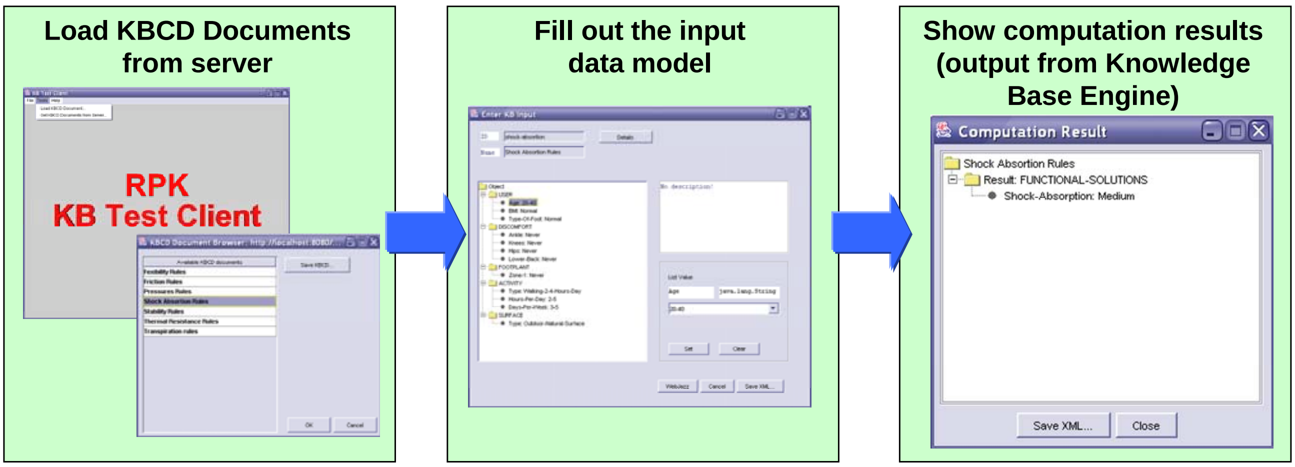

Figure 7: Validation of the Knowledge Base System

3. Conclusion

This paper outlined shortly the scope and objectives of knowledge-based shoe design process. Methodology and engineering solution introduced are based on a *Knowledge Development* process. The represented *Knowledge Base System* architecture for the mass customization of shoes aims to support an effective and economical design and manufacture of individual shoes. The shoe last is adapted by applying biomechanical design rules on characteristic areas of the last shape to obtain a custom made shoe with perfect fit. The implementation of the architecture proposed and a followed continuous validation in all test proved the feasibility of the concept as shown in Figure 7. The implementation and validation fulfilled all the expected results and benefits. The key innovations and benefits of the outlined approach are summarized as follows:

- *Knowledge Framework* as a distributed IT system architecture based on client/server paradigm
- platform independent data exchange and communication, based on industrial standards such as Web Services and XML
- flexibility of the *Knowledge Base System* concerning the content of the rule bases
- complete support of the *Knowledge Development* and validation processes in the *Knowledge Framework* through the client applications

We used an integrated product model that combines the customer needs and characteristics with geometry and functional properties of the shoe. That’s why our approach fulfills the requirements of the custom-made or best-fit shoes to ensure customer satisfaction.

References

[BoDu-02] Boer, C., Dulio, S.: The vision document, Project document of EUROShoE, 2002

[GrLW-03] Grabowski H., Lossack R.-S., Wunsch M.: Mass Customization of shoes by automatically finding best-fitting shoes, 2nd Interdisciplinary World Congress on Mass Customization and Personalization (MCPC 2003), München, 2003

[OTT-84] Otto, E.: Beitrag zur digitalen Beschreibung der Morphologie und Antropometrie des Fußes als Ausgangspunkt der Schuhleistenkonstruktion, Dissertation, Magdeburg, 1984

[PFI-02] Pfi: Schuhproduktion bei kleinen Losgrößen – Produktion von Individualschuwerk, Primasens, http://www.pfi-ps.de/aktuell/pfi/0101pfi11544.htm, abgerufen am 9.12.2002

[PIL-98] Piller, F.: Kundenindividuelle Massenproduktion – die Wettbewerbsstrategie der Zukunft; München, Wien, Hanser-Verlag, 1998

[PIN-93] Pine, J.: Mass Customization: the new frontier in business competition, Boston Harvard Business School Press, 1993

[PIN-94] Pine, J.: Maßgeschneiderte Massenfertigung : neue Dimensionen im Wettbe-werb, Wien, Ueberreuter, 1994

[TsJi-01] Tseng, M., Jiao, J.: Mass Customization, in Gavriel Salvendy (Hrsg.), Hand-book of Industrial Engineering, 3. Auflage, New York, Wiley, 2001

Dr. A. Semenenko
Institute for Applied Computer Science in Mechanical
Engineering (RPK), University of Karlsruhe
Kaiserstr. 12, D-76128, Karlsruhe, Germany

Phone: +49 – 608 – 8726
Fax: +49 – 66 – 1138
E-mail: semenenko@rpk.uni-karlsruhe.de

Dipl.-Ing. R. Krikler
Institute for Applied Computer Science in Mechanical
Engineering (RPK), University of Karlsruhe
Kaiserstr. 12, D-76128, Karlsruhe, Germany

Phone: +49 – 608 – 6364
Fax: +49 – 66 – 1138
E-mail: krikler@rpk.uni-karlsruhe.de