An Approach to Designing Distributed Knowledge-based Software Platform for Injection Mould Industry

MILKO MARINOV[1], NICOLA MAGALETTI[2], TSVETELIN PAVLOV[3], FABIAN GAUS[4], DOMENICO ROTONDI[2], PAVEL VITLIEMOV[5], SLAVINA IVANOVA[6]

[1]Dept. of Computer Systems & Technologies, University of Ruse, BULGARIA
MMarinov@ecs.uni-ruse.bg
[2]Corporate Research Unit, TXT e-solutions SpA, ITALY
nicola.magaletti@txt.it, Domenico.Rotondi@txtgroup.com
[3]University Computing and Information Services Centre, University of Ruse, BULGARIA
CPavlov@uni-ruse.bg
[4]Laboratory for Machine Tools and Production Engineering of Aachen University, GERMANY
F.Gaus@wzl.rwth-aachen.de
[5]Dept. of Manufacturing Engineering, University of Ruse, BULGARIA
pvv@manuf.uni-ruse.bg
[6]IT Project Management Department at Image Group, BULGARIA
slavina.ivanova@gmail.com

Abstract: - Nowadays, the globalization of the manufacturing enterprises requires collaboration across frontiers. In order to attain effective collaboration, the information about the product life cycle must be captured and administrated in a way that supports the decision taken during the product development. In this context, the manufacturing process information needs to be shared between manufacturers. The moulding process is quite complex involving many variable process parameters like pressure, temperature and time settings. These process parameters have to be optimally set in order to improve part quality and maximize the production capacity of the injection moulding machine. In this paper we propose an approach to designing distributed knowledge-based software platform for injection mould industry. The user requirements to the software tools and the key features of our design are defined. The basic principles and the architecture of the software platform and decision support tools are considered.

Key-Words: - Distributed systems; Service-oriented architecture; Rule-based knowledge representation; Rule engine tools; User requirements analysis; Injection moulding.

1 Introduction

The European tooling industry comes across a more and more competitive environment. With more than 10,000 companies and an average annual turnover of 13 billion Euros it represents a high value-added workforce. This workforce takes up a key position for the industrial value-adding chain of the production industries of Europe and all other Industrial Manufacturing Systems countries, as studies and best-practices reveal. This key position originates due to the fact that almost each ramp-up of a production line in every industry depends on the completion of the required tools and their integration into existing production facilities. Moreover, the stability and efficiency of the production process as well as the product's quality are inevitably a consequence of the tool's quality. These results are achieved as toolmakers take account of their European customers' demands by producing innovative, customised and unique tools. This key industry is therefore an enabler for high-value and high-technology production in Europe.

This work is a part of the ongoing project "The Tools for Innovative Product-Service-Systems for Global Tool and Die Networks (TIPSS)" (http://www.tipss-fp7.eu). TIPSS is an EU-funded Framework 7 project which main objective is to develop suitable tools to enable toolmakers to improve their local and global performance at internationally networked locations. These methods, techniques and technologies are based on so-called "smart tools", which represent injection moulds equipped with state-of-the-art sensor technology delivering real-time data from the production process. In doing so, the toolmaker gets an "insight" view into the tool and a broad knowledge of the actual condition of the latter can be gained. A cost-effective and rapid creation of both knowledge- and technology-based industrial services is made possible.

All activities along the value-adding chain can be supported by a Web-based platform to coordinate all forms of forward and backwards collaboration. At the same time, the platform accesses the data collected by the smart tools and the customers' production facilities. Smart tools are injection moulds equipped with state-of-the-art sensor technology that delivers real time data from the production process. The on-line analysis of the gathered data enables an optimized mix of both condition-based and preventive maintenance services, which directly leads to an increase of the overall operational availability of the production cell. In order to monitor the production process, algorithms and methods for the intelligent interpretation of the gathered data are developed within TIPSS, which allow the simulation of different scenarios forecasting the tool's further behavior in operation. Based on those forecasts, errors or downtimes can be interpreted in real time by the toolmaker, thus minimizing the time to respond and the repair time, according to the service level that has been agreed upon the customer.

The paper is organized as follows: Section 2 surveys some related works. The user requirements to the software tools and the key features of our design are defined in Section 3. The background of the service oriented architecture is described in Section 4. The architecture of the TIPSS software platform is considered in Section 5. The architecture and implementation of the decision support tools are described in Section 6. Finally, Section 7 gives our conclusions.

2 Related Work

The moulding process is quite complex involving many variable process parameters like pressure, temperature and time settings. These process parameters have to be optimally set in order to improve part quality and maximize the production capacity of the injection moulding machine. Educated and experienced individuals are required to set up and optimize such a complex process. These individuals control the moulding process on a trial and error basis, which usually is time consuming. This method of controlling the moulding process relies heavily on operator intuition and a few "rules of thumb," which the operator develops over a period of time while working with different materials, pressures, temperatures and time settings.

Kumbakonam et al. [14] present an outline of ongoing research at the University of Louisiana at Lafayette involving the development of Intelligent Knowledge-Based Engineering Modules (IKEM) for the injection moulding process. IKEM has different modules, namely: parsing, mould design, cycle time and the manufacturing module, which are linked to each other. The manufacturing module is an intelligent manufacturing process tool called "The Optimizer." The optimizer captures non-deterministic knowledge in the injection moulding process from an expert in this field, and also uses deterministic knowledge available in the form of relations, look up tables, etc. The optimizer helps the manufacturer in the set up of the moulding machine by giving him the initial optimal process parameter values. These values could be later fine tuned for personal benefits by the operator using his intuition and guesswork.

Networking of people, machines, and services is rapidly becoming strategic for cost-effective solutions and to share the resources and confined knowledge among individuals and organizations. Also products and processes today demand a wide domain of expertise. Concurrent engineering addressees these issues and creates an unified environment for groups of people to share their resources, core competencies and develop products much faster and cheaper and gain competitive advantage. The framework proposed by Stroud et al. [28] consists of running high-end applications such as design, planning and analysis for the mould and die industries over the Internet. The designers should be able to communicate, share the knowledge and foresee any potential problems in downstream applications. The entire system is enriched by a knowledge engine which will provide advice on material selections, shrinkages, cavity layout, draft angles and cooling channels.

Mok et al. [17] develop a prototype of Internet-based intelligent design system for injection moulds. The architecture of the system consists of an interactive knowledge-based mould design system embedded in a Web based environment. A Java-enabled solution together with artificial intelligence techniques is employed to develop such a networked interactive CAD system. In this system, the computational module, the knowledge base module and the graphic module for generating mould features are integrated within an interactive CAD-based framework. The knowledge base of the system would be accessed by mould designers through interactive programs so that their own intelligence and experience could also be incorporated with the total mould design. The approach adopted both speeds up the design process and facilitates design standardization which in turn increases the speed of mould manufacture. Al-Ashaab et al. [1] introduce the SPEED (Supporting Plastic enginEEring Development) system designed to facilitate the sharing of injection-moulding information between interested parties via the Internet.

The global engineering environment has led to the distribution of product life cycle information and knowledge affecting the collaboration throughout product development. Although information technologies, such as the Internet, provide a partial solution to support such collaboration, there is still a need to support decision making by providing the right information and knowledge in the place, time and format required by the geographically distributed companies. In [26] Rodriguez & Al-Ashaab present the research conducted to develop a digitalised representation of product life cycle knowledge to support e-manufacturing of injection moulding products. Such representation is the kernel of a Knowledge driven Collaborative Product Development (KdCPD) system to support the effective collaboration in a global engineering environment.

In [32] Vasek & Stanek describe a utilization of the developed NAHOS information system as a tool management system used in injection molding to handle used tools (moulds). The information system gives to users the possibility to store all needed data as attributes of the moulds and to use these attributes for finding appropriate tool (mould).

Determining optimal process parameter settings critically influences productivity, quality, and cost of production in the plastic injection moulding industry. The research described in [2] presents an approach in a soft computing paradigm for the process parameter optimization of multiple-input multiple-output plastic injection moulding process. The proposed approach integrates Taguchi's parameter design method, back-propagation neural networks, genetic algorithms and engineering optimization concepts to optimize the process parameters. Chen et al. [3] integrate Taguchipsilas parameter design method, back-propagation neural networks (BPNN), and Davidon-Fletcher-Powell (DFP) method to optimize the process parameter settings of plastic injection moulding. The proposed approach can effectively assist engineers in determining optimal initial process parameter settings and achieving competitive advantages on product quality and costs. A self-organizing map plus a back-propagation neural network (SOM-BPNN) model is proposed in [4] for creating a dynamic quality predictor. This is an innovative neural network-based quality prediction system for a plastic injection moulding process.

The main objective of the study done in [6] is to determine an optimum and efficient design for conformal cooling/heating channels in the configuration of an injection moulding tool. Both simulation and experimental verification have been done by Saifullah et al. [27] with the cooling channels system. Comparative analysis has been done for an industrial part with conventional cooling channels using the Moldflow simulation software.

3 User Requirements Analysis

In the analysis of user requirements use-case diagrams are used to capture the platform requirements and understand what the system is supposed to do. Moreover, they can also be used to specify the behaviour of the system that is to be implemented. A use-case diagram presents a collection of use cases and actors and is typically used to specify or characterize the functionality and behaviour of a whole application system interacting with one or more external actors.

3.1 Defining TIPSS actor classes

TIPSS actors can be identified introducing two different high level classes:

(1) Technical - actor classes belonging to this upper-class are the ones based on roles and skills needed to provide/adapt data and set-up and deploy the software TIPSS platform as:

- *Field Device* - such as Mould Jini and similar tools, collects data from the production process;
- *Metadata provider* - arranges data gathered from the process, also adding information, enabling data source for its effective use within the TIPSS platform;
- *Application Developer* - uses data and metadata for testing activities and for the application configuration in the deployment phase;
- *Applications and Services* - the software components that use data and metadata and provide specific output as report, view or any other data needed to properly feed and launch other components;
- *Platform Administrator* - devoted to the management of workflows and any other

technical activity assuring the right working conditions to the TIPSS platform.

(2) Functional - actor classes belonging to this upper-class are the actual users of the functionalities provided with the TIPSS platform, basically they are:

- *Producers or Moulder* - produces end products. In the TIPSS scenarios, they are the users of the injection moulding systems equipped with smart tools and other data collection devices;
- *Mould Owner* - it is the customer that requires the mould and provides specifications to toolmaker. It directly uses the mould or provides it to different producers;
- *Toolmaker*- is the mould manufacturer develops moulds and dies;
- *Third party company* - uses TIPSS platform providing value-added activities committed from toolmakers or mould owners;
- *End Product Customer* - requires the mould owner for a product supply, specifying material, quality, delivery rate, etc. These market variables are among the main inputs for the mould design and development.

3.2 Identifying Use Cases

The top-level use-case diagram depicted in Figure 1 is just to visualize the industrial context software platform and the boundaries of the system's behaviour based on four classes of use cases. Since in such top level representation it is not needed the specification of the different user roles, the only conventional actor involved in this diagram is the generic user of the TIPSS platform.

- **Device management** - Mould Jini (MJ) main task is to track every event affecting the mould machine by recording process data collected through specific sensors.

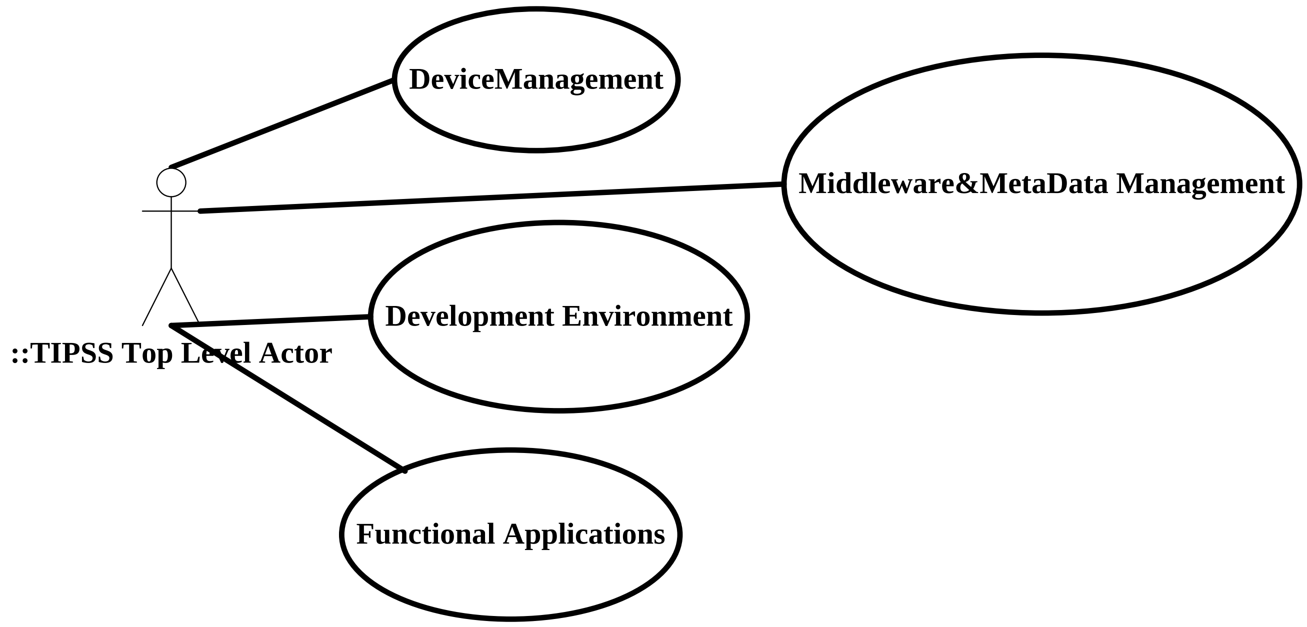

Fig. 1 Top-level use case diagram

- **Device management** - Mould Jini (MJ) main task is to track every event affecting the mould machine by recording process data collected through specific sensors. For the different purposes of the project, real time production data, as gathered from the Mould Jini must be made available within the TIPSS platform in order to properly feed the tool book application services. The general option is that the services can retrieve data from a database where they have been stored according to specific schemas or metadata. Another option is just to distribute process parameter values as real time data streams. In this case a fast data transfer protocol is required and, just in case, also concentrator devices for the aggregation of data coming from different moulds and MJs in the same company or production department.
- **Middleware and metadata management** – TIPSS project is based on the use of SOA technologies. In particular, the idea is to use an

Enterprise Service Bus (ESB) as a standard-based and flexible environment providing features for the implementation of the integrating architecture. These features, available as set of services exposed on the bus, mainly include data source mapping and access, event driven workflow management and application management. Data collected from the mould by MJ system or similar field devices and stored into local databases are raw process data. Metadata and specific formats needs to be provided in order to allow the set-up of data collection devices and make data accessible from TIPSS applications and services. Data interoperability services manage the communication between different and sometimes heterogeneous data sources. The common problem faced by these services is extracting information from a data source having a specific format to reverse it into another data source having a different format or provide it to an application in a format this application can manage.

- **Development environment** - The development of new algorithms or the setup of new business services may require the definition of new data schema and metadata as the application specific input/output. For instance, not only raw production data but also new specific metadata may be required concerning working conditions, injection machine type, company, etc. Any application module (algorithms or other services) available on the TIPSS platform then needs data specific format to be defined and provided as well as any supporting tool for data-management along its development cycle, as for prototyping, testing and deploying the software modules.
- **Functional applications** - Running algorithms, such as for process control and preventive maintenance requires data gathered from field devices to be properly formatted and sent as input by the application management services. Any time an application process is launched, a service can be triggered reading data from the data source (logbook databases or data streams) according to specific schema defined for each "calling" process, including process data and any metadata. The figure 2 is a general view that includes all the use cases and actors working within the functional application management scenarios.

TIPSS project is mainly focused on this last class of functional application management use cases. Within this class, the following are the most relevant specific use cases to be considered:

- **Tool Maker running mould qualification process** – this process aims at the optimization of control parameters as the value set that will be implemented during the first setup of the mould for the production phase. The process is carried out by expert engineers in the toolmaker's plant before delivering the mould and follows a strictly "tool specific" workflow that depends on the tool and machine features as well as on the injection materials used and other industrial variables. The final goal is to detect the correct working conditions for the mould and to produce the related reference values for the process parameters that could be controlled from the operators. The process parameters that can be monitored by means of current MJ are temperatures and pressures up to 23 zones of the mould, by means of specific thermocouple and analogical sensors (0-10V), as well as 16 indirect measures obtained for differential values of any of the 23 direct measures available.
- **Moulder working close to the mould machine** - In the first phase of production campaign, this process is devoted to control the mould process parameter values versus the setting points and ranges defined during the mould qualification phase (to keep the process within the given specifications) in order to verify the actual behavior and achieve all possible and needed optimizations. Expertise of the process engineers is provided in this phase with software tools supporting the decision activity required to realize acceptable vs. potentially dangerous conditions and, in this case, identify the actions to be taken. A "rule based" warning mechanism is implemented to warn if process parameters or any indicator based on such values (data interpretation tools) leave the range of values that have been set during the initial set up of the tool for operation. At this stage of development the warning contains the information that the tool is operating outside of the predetermined specifications. For the further research progress it is the goal of the consortium members to couple these warnings with an advise-functionality for measures that the operator has to take to prevent damage to the tool. An important function in this phase is the measurement and analysis of the change rate of process parameters. Another way to detect "out of specification operation" of the mould consists on analyzing the long term changes of process

parameters over the tools lifetime. These changes of process parameters are detected and analyzed by the rules in order to evaluate, whether maintenance or an interruption of the production process is necessary in order to prevent damage to the tool.

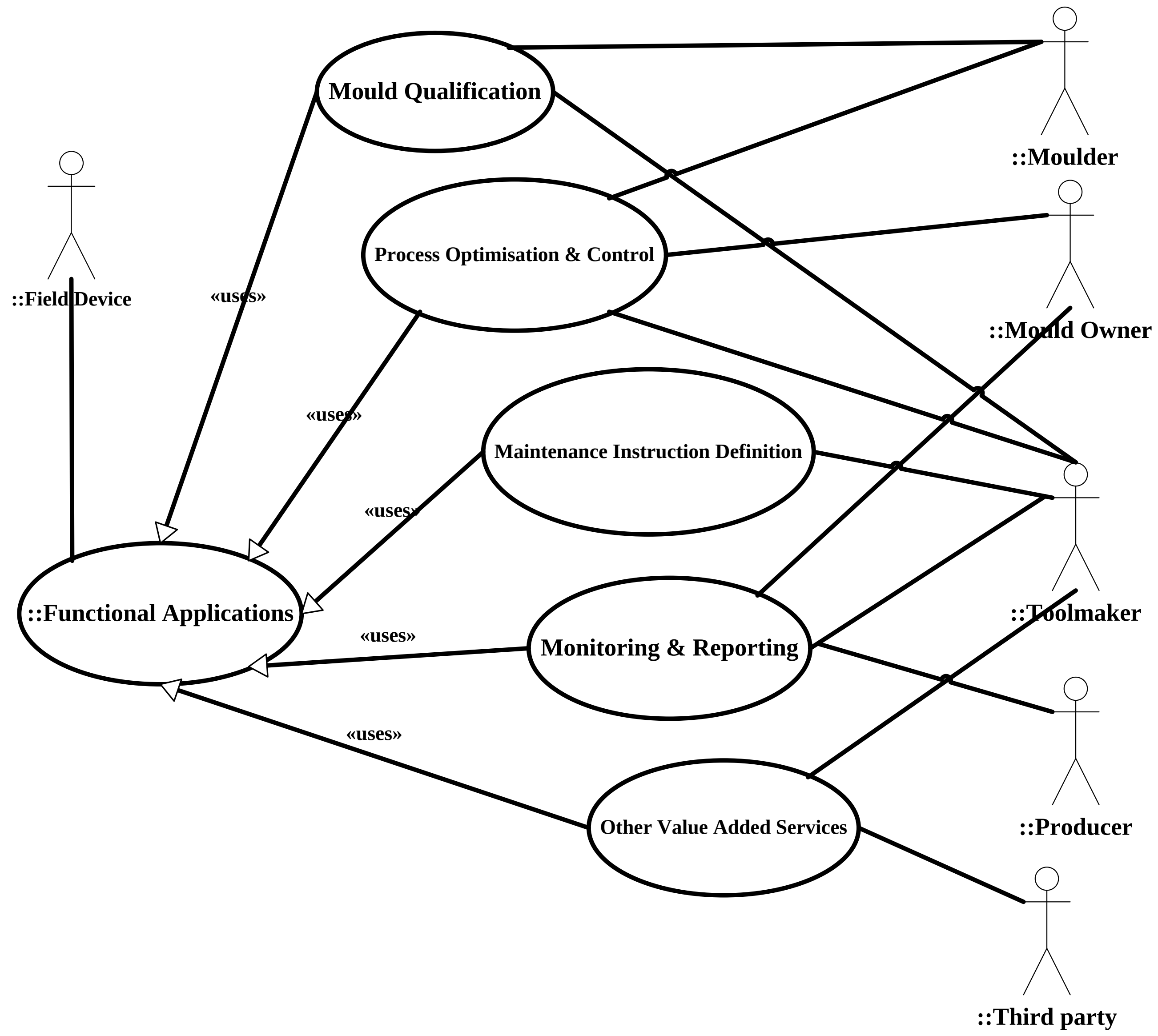

Fig. 2 Use case diagram of the functional application management scenario

- **Toolmaker collecting process data from a remote workstation** - After the set-up of the molding process at molder facility, during the actual production phase, the toolmaker could be interested on collecting process data from the mould delivered to a customer (producer) that show the way the tool is being used (monitoring and storing process parameter data). For this general situation mainly software tools for an improved reporting and tracking activities are required, as well as having a user friendly and flexible access to process data. The toolmakers additionally, by accessing process data can also use some "expert" models and algorithms predicting the behavior of the system. This could turn useful in case of services assigned to the toolmaker for the standard monitoring activities finalized to a preventive maintenance agreement. The logged process data also allows the toolmaker to perform different kind of analysis like data mining, benchmarking, simulations etc., aimed at evaluating the molders and improve their own process models.

- **Mould Owner collecting process data from a remote workstation** - This process is mainly based on the use of data collection and storage tools finalized to tracking and reporting for quality, security, reliability reasons related to the use of the mould put in place in different locations/plants. Also for this use case mainly software tools for an improved reporting and tracking activities are required, as well as having a user friendly and flexible access to process data.

4 Service Oriented Architecture Background

Service oriented architecture (SOA) is an approach to designing a system that allows for loose coupling, interoperability, and standards-based computing [5]. Service oriented architecture enabled us to provide the public with vital information. SOA refers to the design of new applications that involve the incorporation of "services" from existing systems [7]. SOA is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different domains. In general, a service consists of a set of named operations, each of which takes a number of parameters and returns a result [16, 25]. Converging organizations and their data to information technology with human interaction is a challenge. SOA is a key to solving this challenge [15, 19, 23, 29]. SOA is an architectural paradigm and discipline that is used to build infrastructures enabling those with needs (consumers) and those with capabilities (providers) to interact via services across disparate domains of technology and ownership. Services act as the core facilitator of electronic data interchanges yet require additional mechanisms in order to function [33]. In SOA, the system is divided into multiple sections, each of which assumes responsibility for a portion of the overall process. The three parties involved in SOA are service providers, service brokers, and service requesters [7, 16].

Service-oriented architectures deliver a number of recognized advantages, including more open and efficient access to key enterprise services, applications and information. But the very openness of an SOA also creates unique and significant security challenges for organizations [21]. A fundamental aspect of a successfully implemented SOA security is a well-defined, well-planned and well-implemented security model/strategy that is focused around the three basic principles of security (confidentiality, integrity and availability). Another important and often overlooked aspect of SOA security is well defined business agreements, service level agreements and security metrics between service providers and service consumers [22]. They provide a foundation for developing security strategy and governance. It also can be used as a measuring stick for quality assurance, business monitoring and SOA management.

Publish/subscribe systems are becoming increasingly popular in building large distributed information systems [30]. In such systems, subscribers specify their interests to the system using a set of subscriptions. Publishers submit new information into the system using a set of publications. Upon receiving a publication, the system searches for matching subscriptions and notifies the interested subscribers. Unlike the client/server model, the publish/subscribe model decouples time, space and flow between publishers and subscribers, which may lead to benefits such as reduced program complexity and resource consumption. Publish/subscribe systems have a number of interesting characteristics [12, 18, 31]. Firstly, producers do not need to address consumers. Instead, consumers simply specify the notifications they are interested in. This loosely coupled approach facilitates flexibility and extensibility because new consumers and producers can be added, moved, or removed easily. Secondly, communication is asynchronous, thereby removing the disadvantages and inflexibility of synchronous communication. Thirdly, producers and consumers do not need to be available at the same time. This means that a subscription causes notifications to be delivered even if producers join after the subscription was issued. Finally, publish/subscribe directly reflects the intrinsic behavior of information-driven applications because communication is initiated by producers of information.

5 TIPSS Software Platform Architecture

The architecture of TIPSS software platform is depicted in Figure 3. We can distinguish the following main logical and technical set of components according to the TIPSS three-tier architecture.

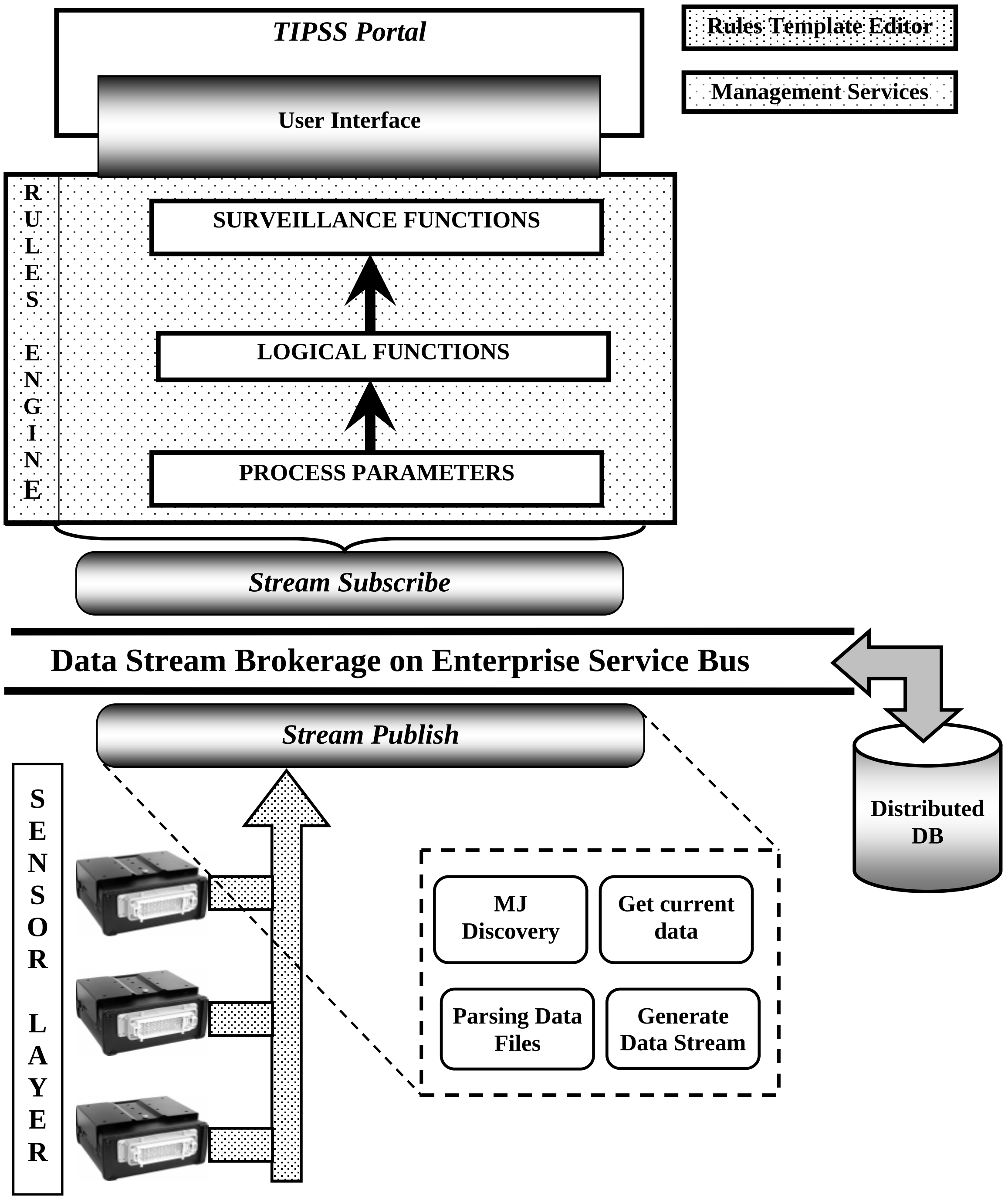

Fig. 3 The architecture of TIPSS software platform

The lower layer of the architecture represents the data integration aspects that include the collection process data from the field (Mould Jini or similar devices) and the data store and management facilities for local and remote access. Mould Jini [8] is the electronic log book of the injection mould. Referring to the equipment of the mould different data may be stored for nearly all sequences of injection and maintenance. Mould Jini supervises and alarms all temperatures. The actual values of each injection cycle are stored in a data logger to be used for a statistic analysis. This enables the user to avoid damages and loss of production by early detection. The nonresetable counter sums the total of the cycles. Independently of the cycle counter all measured values of the mould are supervised and set the alarms in question in case of deviation from the adjusted tolerances. The actually measured and stored values can be read out from Mould Jini via web browser. Mould Jini is configured via web interface.

According to the principles to implement the TIPSS software platform as a service based architecture, also the provision of data streams to the application layer is based on the set-up of specific services. The design approach used in Mould Jini is a web services based (SOAP) technology enabling for a more flexible and open access to the devices. The implementation is based on a SOAP service to get the real time measurements per web service. This feature provides a flexible way to access real time data gathered from the mould and an easy and flexible direct access to process data from any type of application for real time data processing.

Open source frameworks and SOA compliant technologies are used for this tier enabling for a flexible and efficient process data provision to different kind of players using the TIPSS platform. Although SOA could be considered the state-of-the-art for many industrial and business sectors, is still long to be fully adopted and deployed in the manufacturing environments since across the industry, SOA interoperability dramatically lowers costs between the enterprise, supply chain, and plant operations systems and other manufacturing/ production applications. The choice of SOA is strongly suitable for this kind of research activities to provide a collaborative environment for virtual organizations, as can be the systems involving toolmakers, customers and partners of support activities.

A publish/subscribe design approach is used into the TIPSS architecture to implement the access to the large scale distributed information dissemination applications. Publish/subscribe systems support loosely coupled asynchronous communication between information producers and consumers. Producers (publishers) inject messages into the system, which routes messages to consumers (subscribers) that register interest in certain messages using subscriptions. There is a data source (a publisher) that collects data and arrange them for a general provision, a broker that distributes the data on the bus, a subscriber that gets the data and put them within a XML file to which an application can access. The ActiveMQ is used as broker activating the publish/subscribe services. ActiveMQ is an open source message broker that supports Java Message Service (JMS). It is a standard Java API for creating, sending, receiving and reading messages. The communication between the broker and any application client is based on API (JMS interface). The logical and physical communication layer is represented from the Enterprise Service Bus component (ESB). ESB (ServiceMix/OpenESB) is a software component that lies between the business applications and enables communication among them. The brokerage system provides the process data gathered from the MJs to be “published” as data streams on the ESB. The brokerage system then enables any authorized application to “subscribe” the streams of specific process parameter available from any MJ within the domain.

The integration environment provides two kinds of MJ data: a database in which all MJs configuration history and data is stored. This database provides a web service based query interface to extract MJ data in a specific time period. The second kind of data is the MJ data streams provided as JMS topics. Each "application" to use a MJ data stream must subscribe to the topics of interest to acquire the data.

The upper layer of the architecture represents the application layer devoted to the end users, as toolmakers and any other potential business user of the TIPSS platform. User interfaces are accessible through the TIPSS portal environment as web applications to be used both in remote or local working conditions. The architectural details of the decision support tools (rules engine, rules template editor, management services) will be described in the next section.

The front end of the system is the user interface, which is provided either by the data interpretation tools, rules engine tool or a web browser. Further, the user interface is linked to underlying system information stored in various repositories via a web services layer.

6 Rule-based Decision Support Tools

Electronic, encoded knowledge is used effectively in a variety of highly successful industrial applications and to provide automated decision support [9, 13]. One of the most popular approaches to knowledge representation is to use rules. Such knowledge is encoded in actionable form, usually as *if…then* rules, where the *if* portion is a logical condition expression, and the *then* portion is an action – usually to recommend something and carried out via various means of notification (directly on screen for synchronous applications, or, for asynchronous applications, by email, text page, or by scheduling a notice for popup or display during a future login). Some rules may be encoded as tables in which conditions for firing are listed, for example, drug interactions. Those can also be recast in the form of *if…then* rules. Some of the benefits of the rules are that they are modular, each defining a relatively small and, at least in principle, independent piece of knowledge [10, 20]. New rules may be added and old ones deleted usually independently of other rules. The advantages and disadvantages of a rule-based approach are defined in [11]. The advantages are as follows: (1) easy to understand the knowledge `content'; (2) explanation for the reasoning is easily shown i.e. a list of which rules fired (and in which order) during reasoning; (3) maintenance (or modification) is easy, provided the rule base structured well; (4) uncertainty can be incorporated into the knowledge base. The following are main disadvantages (or limitations): (1) rule bases can be very large (thousands of rules). In this case the problem with rule sets is the speed. Matching a condition, selecting a rule and firing needs much more time than other methods and is so inherently inefficient; (2) rules may not reflect the actual decision making; (3) the control knowledge is frequently mixed up with the domain knowledge. So the context in which a rule applies is to be considered.

The most important component of a rule-based system is the rules engine. The rules engine interprets the rules in the light of the facts that are available to them and select which of the rules to fire and in which order. Rules engines are needed anywhere it is not desirable for the logic to be hard coded into the application and has to be amenable for modification and customization for different needs and situations [24]. Rules engines allow separation of the business logic from the application code and data, which reduces the software development and maintenance costs. The way in which the knowledge base is used is determined by the rules engine.

In the TIPSS decision support tolls we define the rules engine per the conventions introduced by Greenes et al. [9] as a software component of a rule-based system that assesses individual rules and determines their applicability in a specific case or situation. The general functionality of the rules engine tools is depicted in figure 4.

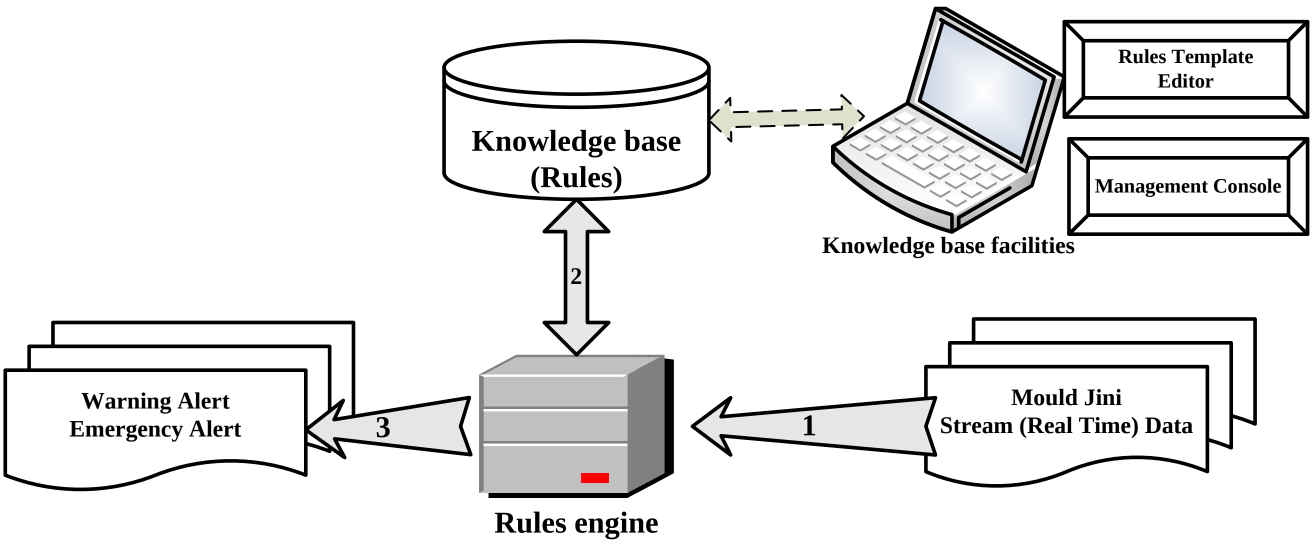

Fig. 4 General functionality of the Rules Engine Tools

Real time data (1) is reasoned upon by the rules engine using the rules encoded in the knowledge-base (2), in order to generate some form of output (3), such as warning or emergency alerts. The knowledge base facilities (Rules Template Editor + Management Console) provide the knowledge base content management. The rules engine actually acts upon specific incoming data via the application of rules embodied in the knowledge-base, in order to generate some type of output. The rules engine takes as its input one or more elements of data, applies a set of rules to that data, and generates some form of output.

The setup function of the rules engine is used by the service technician at the production site of the customer. There are some preconfigured functionalities that represent a standard setup for the injection tools. The first step of the setup is the assignment of the characteristic names, categories, etc. to the input signals of the mould-jini coming from the sensors of the injection tools. For this purpose a template is designed to support the service technician in the configuration and assignment of the parameters. The second step is the configuration of the rules consisting of the selection of input signals, the configuration of the logical functions and also the tool condition surveillance functions.

The basic logical operations are employed to real time data and to historical data. They convert the input of one or two input signals into an output signal, which is used for the process surveillance within the process data surveillance operations. They represent basic operations that are fitted to the surveillance requirements of a specific tool by the toolmaker prior to or during the tool setting. The basic logical operations are as follows:

- Sum (addition of two values);
- Difference (subtraction of two values);
- And/ Or function;
- Average;
- Trend analysis – The trend analysis function describes the change of a process parameter over a predefined time frame;
- Change rate –The change rate is determined as the change of the process parameter over a minimal range of time. It is the first derivative of the process parameter function.

For the surveillance and data interpretation of real time and historical process data the following two functions are realized: “Threshold Control” and “Peak Detection”. The process parameter surveillance mainly consists of the threshold control in order to monitor and evaluate whether process parameters, a difference between process parameters, their change rate is within the given specifications. The peak detection function has the main purpose to stabilise the surveillance of process parameters through filtering temporary parameter variations out of the predefined specifications. When a peak is detected within the process parameters, the rules engine classifies the peak as being either “uncritical” or “critical” which then leads to further actions. It is important to notice that the peak detection function is applicable to inputs that can be a single process parameter value, the difference between two process parameters, the trend or change rate of a process parameter.

The composite structure of the decision support tools architecture is depicted in Figure 5. The rules engine is the heart of the decision support tools. Its main objective is to handle events from the ActiveMQ topics, to collect data needed to execute a rule and to call the rule execution service passing collected data. Every rules engine has a part, called ‘Subscriber’. The ‘Subscriber’ is the actual topic subscriber that receives data (messages) from the message broker. Received messages must match a predefined and synchronized scheme (represented with ITopicEvent interface). Received data is stacked in the ‘Parameters stack’ and when all needed values for a rule are collected, rule service is called with that data, passed as arguments. A rule is made of a ‘rule template’ and a ‘rule instance’, i.e. every template must be assigned to a data source (rule template → data source = rule instance) in order to be active and to be executed.

The rule services are web service application. Its purpose is to maintain a catalog of the rules templates and to provide a method to execute a rule with specific data passed as parameter.

The management services glue together other parts of the system. They maintain catalogues of the data sources (topics/jinis), of the rules engines and the rule instances. Every rules engine is identifiable with the machine name it is running on. Management services provide interface to define which data sources should be handled by specific rules engine.

The analysis services are a web services application, providing entry point to the warehouse. It exposes interfaces for running a predefined analysis over the collected data. These interfaces are used in the rule activities.

The management console is the control centre of the decision support tools. It provides user interfaces for managing the catalogues of rule engines, data sources, rule templates and rule instances.

The rules template editor is a software graphical tool for creating a rule template. It has the functionalities needed to define the rule’s logic in a drag’n’drop manner. The rules template editor uses rule services to deploy a rule template.

A ‘rule template’ is a file, containing the rule logic and description of actions that must be taken when specified conditions are met. The ‘rule template’ contains description of rule parts in specific format and properties of the rule itself. A ‘rule part’ can be any of: conditions and conditional constructions, operators, activities etc. The purpose of the rules template editor is to make the target roles capable of creating description of rules in form of a rule template without strict binding with specific mould machine, factory, mould owner or any other conditions and/or environments. The rules template editor has rich, interactive user interface, enhanced usability and short learning curve

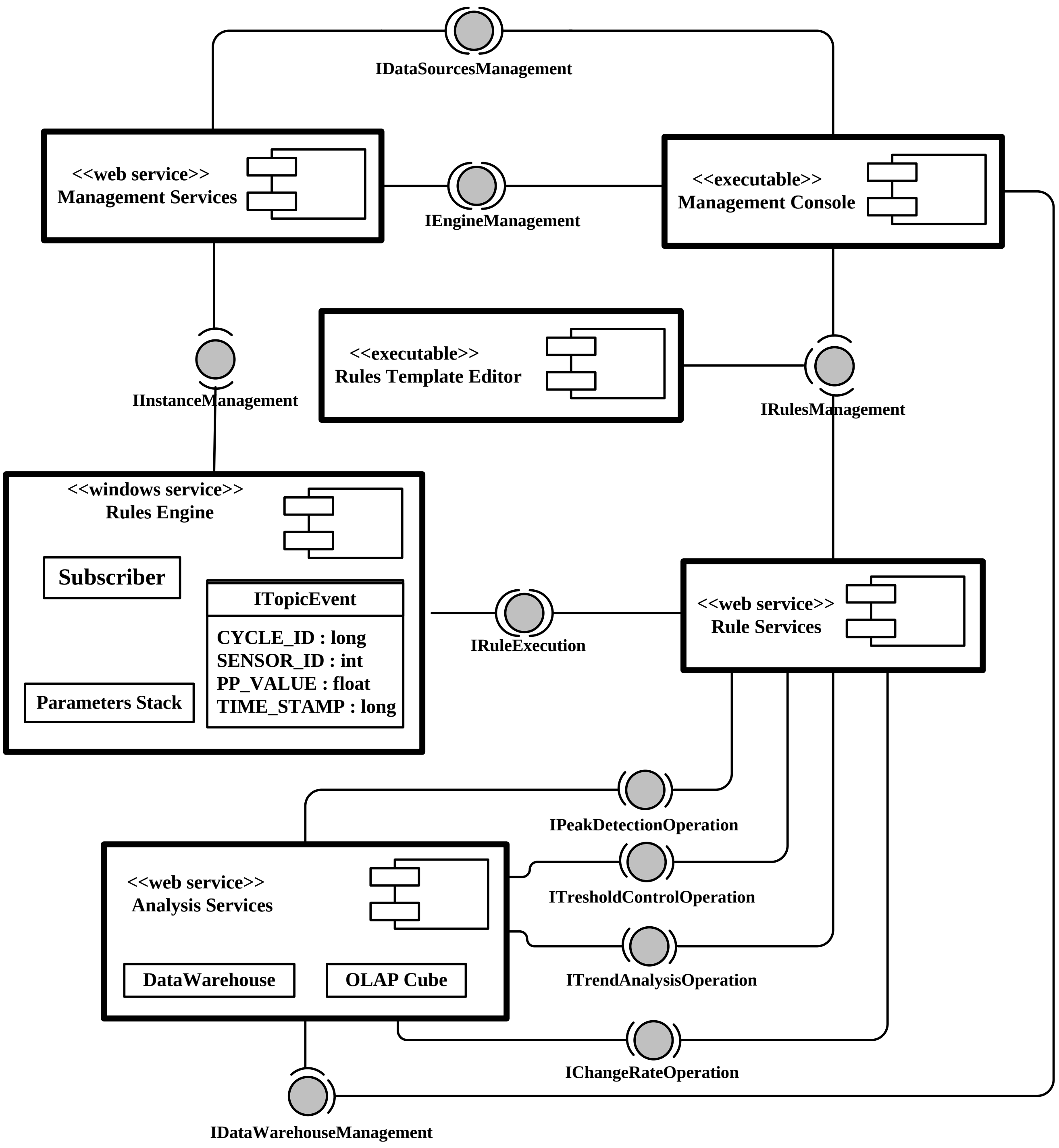

Fig. 5 Composite structure of the decision support tools architecture

The users of the rules template editor are toolmakers, mould owners and (eventually) third-party companies. If producers have experts with required knowledge, they can use the rules template editor for building specific for the mould process rules.

The main functionalities of the rules template editor are as follows:

- *Graphical representation of rules* – ability to build the rule templates in a diagram-like style, eliminating the need of any other technical skills, except the expert knowledge, required to build effective rules;

- *Known and approved user interface* – a user interface that is similar to the interfaces used in most of the modern graphical editors;
- *Ability to add extra rule parts on exploitation time* - rule template editor supports adding extra template parts after initial deployment by software expert or specially build installation tool, but not by the editor's end user;
- *Context sensitive configuration* – every part of the rules template can (and will) have different properties. The application supports properties definition of the part and provides the interface to operate these properties;
- *Loading and saving rule templates in file* – the editor has the ability to save created templates in a file with certain structure. This file format is used by all of the tools in the real time data processing suite;
- *Cut, copy and paste rule parts*;
- *Drag and drop, rule parts palette* – all available rule parts are listed in a panel. Rule parts are added to the rule design surface using drag and drop.

7 Conclusion
In the present paper we have proposed an approach to designing distributed knowledge-based software environment for injection mould industry. The platform accesses the data collected by the smart tools and the customers' production facilities. The architecture of the TIPSS platform is not only a SOA compliant approach, but finally an architecture that is based on loosely-coupled applications able to operate within an enterprise and inter-enterprises environment. SOA compliant technologies enable for a flexible and efficient process data provision to different kind of players using the TIPSS platform. The integration environment based on technologies like ESB and message broker supports a loosely integration of "TIPSS services". This makes possible to complete decouple the "TIPSS services" form issues related to other services location and implementation details.

The architecture of the TIPSS decision support tools is based on the rules knowledge representation. The proposed approach allows separation of the business logic from the application code and data, which reduces the software development and maintenance costs. The way in which the knowledge base is used is determined by the rules engine. The tools have a highly interactive and user-friendly interface. The rules template editor provides the domain experts with a simple and powerful tool to manipulate the rule database. The decision support tools are implemented in .NET 4.0 runtime environment and uses basic platform components, plug-in infrastructure, graphical user interface components (menus, buttons, tree views, event handling) from MS Visual Studio 2010. The implementation of the rules template editor and the rules engine are based on Windows Workflow Foundation. Workflow Foundation ships with a set of general purpose activities for defining workflows. This base activity library provides the ability to define control flows using familiar constructs such as if/else and while loops. It also includes a rules engine, support for communicating with other software using Windows Communication Foundation. For the development of analysis services the MS SQL Server Analysis and Reporting Services are used.

Acknowledgements:
This work is funded by the EU FP7-Small Collaborative Project under Agreement no. 214794.

References:

[1] A. Al-Ashaab, K. Rodriguez, A. Molina, M. Cardenas, J. Aca, M. Saeed, H. Abdalla, Internet-Based Collaborative Design for an Injection-moulding System, *Concurrent Engineering*, Vol. 11, No. 4, 2003, pp. 289-299.

[2] W. Chen, G. Fu, P. Tai, W. Deng, Process parameter optimization for MIMO plastic injection molding via soft computing, *Expert Systems with Applications: An International Journal*, Vol. 36, No. 2, 2009, pp. 1114-1122.

[3] W. Chen, M. Wang, G. Fu, C. Chen,. Optimization of plastic injection moulding process via Taguchi's parameter design method, BPNN, and DFP, *In: Proc. of 2008 International Conference on Machine Learning and Cybernetics*, Vol. 6, 2008, pp. 3315 – 3321.

[4] W. Chen, P. Tai, M. Wang, W. Deng, W., C. Chen, A neural network-based approach for dynamic quality prediction in a plastic injection moulding process, *Expert Systems with Applications: An International Journal*, 35(3), 2008, pp. 843-849.

[5] J. Davidson, M. Ragwar, A. Smith, G. Amoussou, S. Steinberg, J. Eckroth, A framework using service oriented architecture in a community information & referral

System,. *Journal of Computing Sciences in Colleges*, Vol. 24, No. 4, 2009, pp. 252-258.

[6] D. Dimla, M. Camilottob, F. Mianib, Design and optimisation of conformal cooling channels in injection moulding tools, *Journal of Materials Processing Technology*, Vol. 164-165, 2005, pp. 1294-1300.

[7] T. Erl, *Service-Oriented Architectures: Concepts, Technology and Design*, Prentice Hall, Indiana, USA, 2005.

[8] Feller Eng, Operator manual. *Feller Engineering GmbH*, Germany, 2008. [Online]: http://www.fellereng.de/

[9] R. Greenes, M. Sordo, D. Zaccagninia, M. Meyer, G. Kuperman, Design of a standards-based external rules engine for decision support in a variety of application contexts. *In: M. Fieschi, E. Coiera, Y. Li, eds. Proc. of the 11th World Congress on Medical Informatic MEDINFO*, 2004, pp. 611-615.

[10] Y. Guizhen, M. Kifer, Z. Chang, FLORA-2: A rule-based knowledge representation and inference infrastructure for the Semantic Web, *In: Lecture Notes in Computer Science*, Vol. 2888, 2003, pp. 671-688.

[11] C. Holsapple, A. Whinston, *Decision Support Systems: A Knowledge-based Approach*. West Publishing Company, 2000.

[12] F. He, L. Baresi, C. Ghezzi, P. Spoletini, Formal Analysis of Publish-Subscribe Systems by Probabilistic Timed Automata, *In: Lecture Notes in Computer Science*, Vol. 4574, 2007, pp. 247-262.

[13] J. Krogstie, G. Sindre, H. Jorgensen, Process models representing knowledge for action: a revised quality framework, *European Journal of Information Systems*, Vol. 15, 2006, pp. 91–102.

[14] A. Kumbakonam, T. Chambers, S. Dwivedi, Intelligent Manufacturing Process Tool For Plastic Injection Molding, *In: Proc. of ASEE Gulf- Southwest Annual Conference*, 2002. [Online]: http://www.aseegsw.org/Proceedings/VA5.pdf

[15] M. Lopez-Ramos, J. Leguay, V. Conan, Designing a novel SOA architecture for security and surveillance WSNs with COTS, *In: Proc. of the Workshop on Mobile Ad hoc and Sensor Systems for Global and Homeland Security (MASS-GHS)*, 2007. [Online]: http://jeremie.leguay.free.fr/lip6/files/lopez-mass-ghs.pdf

[16] C. McMurtry, M. Mercuri, N. Watling, M. Winkler,*Windows Communication Foundation*, Sams Publishing, USA, 2007.

[17] C. Mok, K. Chin, H. Lan, An Internet-based intelligent design system for injection moulds. *Robotics and Computer-Integrated Manufacturing*, Vol. 24, No. 1, 2006, pp. 1-15.

[18] G. Muehl, Large-scale content-based publish/subscribe systems. *Thesis (PhD)*, Department of Informatics, Technical University of Darmstadt, 2002.

[19] C. Pahl, Y. Zhu, Model-driven connector development for service-based information system architectures, *Journal of Software*, Vol. 4, No. 3, 2009, pp. 199-209.

[20] A. Paschke, M. Bichler, Rule-based knowledge representation for service level agreements. *In: Proc. of the 4th Conference on Multiagent System Technologies (MATES'06)*, Germany, 2006.

[21] G. Peterson, Service oriented security architecture, *Information Security Bulletin*, Vol. 10, 2005, pp. 325-330. [Online]: http://www.arctecgroup.net/ISB1009GP.pdf

[22] G. Peterson, 2008. Security in SOA. *SOA Magazine*, Issue XV, 2008. [Online]: http://soamag.com/I15/0208-2.pdf

[23] K. Rao, A. Pal, M. Patra, A service oriented architectural design for building intrusion detection systems, International Journal of Recent Trends in Engineering, Vol. 1, No. 2, 2009, pp. 11-14.

[24] A. Rajasekar, M. Wan, R. Moore, W. Schroeder, A prototype rule-based distributed data management system, *In: Proc. of the HPDC workshop on Next Generation Distributed Data Management*, Paris, France, 2006.

[25] O. Rishi, A. Sharma, A. Bhatnagar, A. Gupta, Service oriented architecture for business dynamics: an agent-based approach, *In: Proc. of the 6th International Conference on E-Governance ICEG*, 2008. [Online]: http://www.iceg.net/2008/books/2/4_19-28.pdf

[26] K. Rodriguez, A. Al-Ashaab, Knowledge web-based system to support e-manufacturing of injection moulded products, *International Journal of Manufacturing Technology and Management*, Vol. 10, No. 4, 2007, pp. 400-418

[27] A. Saifullah, S. Masood, I. Sbarski, New Cooling Channel Design for Injection Moulding, *In: Proc. of the World Congress on Engineering*, 2009, pp. 700-703. [Online]: http://www.iaeng.org/publication/WCE2009/WCE2009_pp700-703.pdf

[28] I. Stroud, S. Ahamed, J. Neelamkavil, M. Ostojic, Intelligent manufacturing and mold making, *In: Proc. of the International IMS Forum*, Lake Como, Italy, 2004, pp. 54-61.

[29] C. Sun, C. Jung, F. Kuo, W. Lee, M. Lien, K. Hung, Development of an Ontology-based Service Oriented Architecture Framework for National Geographic Information System in Taiwan, *In Proc. of 11th GSDI World Conference*, Netherlands, 2009. [Online]: http://www.gsdi.org/gsdiconf/gsdi11/papers/pdf/61.pdf

[30] D. Tam, R. Azimi, H. Jacobsen, Building Content-Based Publish/Subscribe Systems with Distributed Hash Tables, *Lecture Notes in Computer Science*, Vol. 2944, 2004, pp. 138-152.

[31] W. Terpstra, S. Behnel, L. Fiege, A. Zeidler, A. Buchmann, A peer-to-peer approach to content-based publish/subscribe, *In: Proc. of the 2nd international workshop on distributed event-based systems*, San Diego, California, 2003, pp. 1-8.

[32] L. Vasek, M. Stanek, Utilization of information system for injection molding tools, *Archives of Materials Science*, Vol. 28, No. 1-4, 2007, pp. 171-175.

[33] T. Vitvar, A. Mocan, M. Kerrigan, M. Zaremba, M. Moran, E. Cimpian, T. Haselwanter, D. Fensel, Semantically-enabled service oriented architecture: concepts, technology and application. *Service Oriented Computing and Applications*, Vol. 1, No. 2, 2007, pp. 129-154.