

# Kivonat

Szoftver rendszerek megvalósításának egyik tipikus módja az üzleti folyamat alapú tervezés, melynek előnye, hogy a rendszer működését az adott terület szakértői számára átláthatóbbá teszi. Ilyen rendszerekben felmerül a kérdés, hogyan lehet i) igazolni azt, hogy a specifikáció (folyamatmodell) megfelel a felhasználók, üzemeltető, hatóságok, stb. elvárásainak, ii) igazolni azt, hogy a rendszer működése megfelel a specifikációban foglaltaknak, iii) kiértékelni magának a rendszernek a működését mind informatikai, mind üzleti teljesítmény, valamint szolgáltatásbiztonság szempontjából. Emellett a folyamatok komplex rendszerekre építhetnek, melyek belső hibái mellett adathibák, ill. felhasználói hibák akadályozhatják a rendszer elvárt működését. A gyakori futtatások, vagy a hosszú idejű használat esetén nagy mennyiségű historikus adat termelődik. Ilyenkor már nehézkessé válik az adatok elemzése, ezért szükség lehet a folyamatok hatékony diagnosztikai támogatására, ami segíti a szakértők munkáját.

A probléma megoldására készítettünk egy olyan környezetet, ahol a szakértő a folyamat saját fogalomkészletét használva írhat fel szabályokat, amiket a korábban futtatott folyamatok lefutásából generált eseményeken kiértékelhet. Az elkészített szabályrendszer ellenőrizhető teljességi és helyességi szempontból. Például ha a szakértő megjelöli a modellben a kritikus részeket, akkor ellenőrizhető, hogy az üzleti elemző az összes fontos esetre vonatkozó szabályt felírta. Az összes szabály felírásán túl, az alkalmazott szabályrendszer konzisztenciájának ellenőrzése is fontos feladat, hiszen ellentmondásos szabályok használata helytelen, megtévesztő eredményt adhat. A szabályok szemantikai vizsgálata tehát egy olyan lépés, amely nélkülözhetetlen a helyes és valós diagnosztika elkészítéséhez.

A szabályrendszer továbbfejlesztésének egy módja lehet a szabályok tényleges lefutásának vizsgálata. Például ha egy riasztási jellegű szabály kiugróan sokszor fut le, az átgondolásra figyelmeztethet, hiszen ennek oka lehet, hogy hibás a feltétel, vagy érdemes felbontani a szabályt több alesetre. De az is elképzelhető, hogy ez a rendszer kiugróan rossz működéséről tanúskodik. A dolgozatban bemutatjuk, hogyan lehet ilyen vizsgálatokat végezni, feltáró adatanalízis segítségével.

Mivel a dolgozat ipari projektből indult ki, ezért fontos szerepe volt a sebességnek, valós környezetben való alkalmazhatóságnak is, így több tesztet is végeztünk.

Vizsgáltuk a szabálykiértékelés sebességét, egy komplexebb példán ellenőrizzük a teljesség és a helyesség vizsgálat működését, valamint ugyanezen a példán bemutatjuk a feltáró adatanalízis hasznosságát.

# Abstract

The implementation of software systems is often based on a design captured by business processes. This has the advantage that a domain expert can easily understand the operation of the system. In such systems, some important questions can be raised; i) how to prove that the specification meets the stakeholders' (users, operators, authorities, etc..) expectations, ii) how to prove that the operation of the system meets the specifications, iii) how to evaluate the system in terms of IT and business performance and dependability. In addition, activities in the processes can be based on functionalities provided by complex systems where internal faults and data errors can prohibit the desired functionality. Rule-based diagnosis can help to answer these questions. However, after many runs or long operational time, huge amount of data can be produced. In this case the analysis may become difficult therefore an effective tool is needed, which supports the experts' work.

For this reason, we created an environment where the business expert has the possibility to perform analysis by writing rules in a domain specific language, generated from the process model. These will be evaluated on events, generated by previously executed processes. The correctness and completeness of this rulebase can be analysed. For instance, if an expert notes some critical parts of a model, then it can be checked whether the business analytic has written rules for all important cases. In the design of diagnosis rulebases it is important to check the rules consistency since usage of controversial rules can lead to an incorrect, false statement. So the semantic analysis of the rules is essential to make correct diagnostics.

In the maintenance and development of such rulebases it's worth to investigate the behaviour of rules. For instance if a rule is executed extremely often, then it may indicate a design flaw, (e.g., the cause of this effect can be a bad condition formulation in the rule, or maybe it's worth to split the rule for multiple smaller cases) or it can be caused by a faulty component of the system. We present how to support this investigation by exploratory data analysis.

Our research was inspired by an industrial project, so it was important to create a software which meets practical performance and usability requirements.

We tested the usability of the method on a real-life case study, and using a complex example we checked the work of the examination of correctness and completeness. In this same example we present the usefulness of exploratory data analysis.