J Inno Sci Eng, 2021, 5(2):143-161
https://doi.org/10.38088/jise.848350

Research Article

An Action Management System Design and Case Study on Its Usage for Cyber Fraud Prevention and Risk Analysis

Abdulkadir Battal [1], Ruya Samli [1*]

[1*] *Department of Computer Engineering, Istanbul University-Cerrahpasa, Istanbul, Turkey.*

Cite this paper as:
Battal, A., Samli, R. (2021). *An Action Management System Design and Case Study on Its Usage for Cyber Fraud Prevention and Risk Analysis.* Journal of Innovative Science and Engineering. 5(2): 143-161

*Corresponding author: Ruya Samli
E-mail: ruyasamli@iuc.edu.tr

Received Date: 28/12/2020
Accepted Date: 23/05/2021

Abstract

Today, various types of devices are used to be connected to Internet at every moment of the life. This variety of devices present both benefits and problems to individuals and companies. One of the more important problems in cyber world is "fraud detection". It is known that, malicious uses in the cyber world are increasing rapidly, fraudsters who seek openness in systems cause material and moral damages to both individuals and companies. In the relevant literature, there are many studies on the detection and prevention of cyber frauds in many sectors such as finance, telecommunication and online shopping. This study aims to develop an action management system to detect cyber frauds and follow the actions of the cases with automatic workflows so as to protect the users according to their next actions. It also reveals the results of this action management system's usage in a case study of a telecommunication company in Turkey.

Keywords: Action management system, risk analysis, cyber fraud

1. Introduction

In today's digital world, information technologies are used to provide many personal and corporate transactions. While transactions over the Internet provide many advantages to both individuals and corporations such as speed, time-saving, labour-saving and convenience, the security of these transactions is becoming a serious issue. In the cyber world, for the safe completion of transactions; the users must ensure that the information is transmitted to the other party without any corruption (integrity), the information is not received by another person (confidentiality) and the individual/company that received the information is the one to whom the information is intended to be sent (authentication). Each of these topics gains importance day by day in cyber security world. The topic handled in this study is "confidentiality". Today, as the type, the amount and the importance of the information shared over the Internet increase, the possibility of the information being received by any unauthorized persons also increases. Cyber-attacks or/and cyber wars intrude from the systems of individuals, institutions and even countries and receive their information. As this situation causes high material and moral damages, large scale studies are being conducted for analyzing attack risks and detecting/preventing attacks. It is of great importance that a cyber-attack on an institution can be determined in advance and as a result, be prevented. However, it can be possible only with an accurate risk analysis. This study aims to develop an action management system to prevent cyber fraud which can be accepted as a form of cyber-attack and also, perform risk analysis. The action management system is a structure that ensures the correct and safe processing of actions that need to be taken automatically for workflows. If the actions in the workflows are handled correctly and safely, then cyber-crimes will decrease significantly and possible attacks will able to be prevented.

The rest of this study is organized as follows: In Section 2, a wide range of literature review is carried out. In Section 3, the technological structures of the application developed in this study are explained. In Section 4, the developed application is explained by examples, and also the results of a case study of this implementation in a telecommunication company of Turkey are given. The study is concluded in Section 5.

2. Related Works

2.1. Studies About Action Management Systems

Especially companies that provide services to a large number of customers use systems that take automatic actions in certain situations in order to prevent assigning all the work to the employees. By automating some of the actions, more work can be accomplished in a shorter time frame. Particularly, action systems that is related with "fraud and risk management" are becoming more common every day. These systems are generally configured with a marker and do not allow the action units to be used separately. The information of the action is uploaded to the system and the action is taken automatically after the process definitions. Information on how the processes are defined or which parts of the processes are available cannot be accessed. Furthermore, in systems of the prior art, structured and sequential actions cannot be provided, since the user's interaction with the system is kept to a minimum value. There are some patents on this subject in the literature. The patent [1] discloses an action method and apparatus for determining the dynamic flow in distributed systems. The dynamic flow detection apparatus enables dynamic determination of flow through the action chain in event processing performed in the distributed system. The fact that actions are general and independent enables event processing to take place or to be changed flexibly. The patent [2] discloses a method and system for setting up,

processing and implementing scenarios in event-based information systems. The present invention allows the actions to be reused in specified scenarios, to collect action and action chains throughout the scenario, and to minimize the operations in each defined scenario. The workflow management system mentioned in a patent given by [3] provides computer-aided, graphical tools for defining and managing complex processes in terms of workflow. The patent [4] discloses a workflow management system that automates the definition and application of a process that can be performed according to defined rules. The system is used to ensure that all individual activities are received in defined sequence, form and time. The system is based on three bases (coordination unit, organization unit and messaging unit). In [5], a selective action management system across a computer system was presented. In this action management system, actions can be selectively retrieved, fixed and reconfigured. In [6], the patent mentions an action management system that is responsible for managing and supporting the work of various tasks in production. The system uses process information defining the flow of many jobs in a task executed by at least one person and product information comprising the content of a product produced in each of the many jobs. It also includes a controller that controls the system to control the relationship between product information and most of the tasks, the input and output information, and the relationship that includes a product generated in each task immediately before each task, and a memory that stores and holds each of the work and the associated product information.

2.2. Studies About Cyber Fraud

Since investigating cyber fraud is a very popular and wide concept, many studies on this subject have been carried out in the related literature. Some of these studies have developed a new system for detecting cyber fraud in a private company or institution, some have investigated the reasons and some have researched on how to detect cyber fraud. In the study [7], it is aimed to use an anomaly detection algorithm performed using data mining for fraud detection in some transactions performed over the Internet in real-time. In this study, credit card fraud has been handled specifically between anomalies and cyber frauds. The aforementioned anomaly detection algorithm is based on Artificial Neural Networks (ANN) and is designed by using supervised learning. The study [8] attempted to identify cyber frauds on a global digital network and stated that cyber frauds are riskier than many other threats and must be identified and prevented surely. In [9], it is presented a framework designed to build Internet banking security based on a multi-layered ANN. This system, which can detect anomaly detection, intrusion detection and cyber fraud, has made significant gains by identifying these threats which are particularly harmful in financial terms. The study [10], which is conducted on 22 banks in total, identified deficits caused by lack of information and carelessness, identified cyber fraud transactions performed by using these deficits and took necessary measures. In [11], a cyber-fraud detection system using ANN was developed. In [12], cyber frauds in the banking sector were tried to be determined by the behavior of the customers and the transactions they performed. In this study, a combination of data mining, artificial intelligence and classification methods has been used and thus, frauds have been identified. The study [13] aimed to detect cyber frauds in online finance systems using blockchain technology. The study given by [14] designed a system that uses the Hidden Markov Model (HMM) to detect online frauds. In the study [15], cyber frauds about smart grid energy were determined by using ANN. This study presents a new application based on machine learning to determine energy consumption. The study [16] developed a system that uses Bidirectional Associative Memory (BAM) to detect cyber fraud in mobile phones. The main advantage of the system is the ability to detect not only cyber frauds, but also real-time frauds. In [17], cyber frauds were identified by using self-organizing maps.

There are also some survey studies about cyber fraud in the literature. The study [18] surveyed several techniques for detecting credit card frauds. In this study, some techniques such as ANN, artificial intelligence, Bayesian, data mining, k-nearest neighborhood, decision tree, fuzzy logic, Support Vector Machine (SVM), machine learning and genetic algorithm are examined. The study [19] examined the studies on the detection of cyber frauds in the field of e-commerce. The study [20] is one of the largest studies on cyber fraud detection which have focused on how to detect cyber fraud in many areas such as e-commerce systems, health systems, credit cards, and so on.

There are also several patent designs related to cyber fraud detection. The patents given by [21-23] have tried to detect frauds in health systems. The systems designed in these patents try to detect anomalies in health and analyze to distinguish frauds between these anomalies. The patent given by [24] designed a system for detecting frauds in interactive link analysis and the patent [25] developed a complete system of a device and software for detecting cyber frauds.

3. Material and Methods

Fraud detection systems consist of four main steps: data collection, detection, examination and action. In this study, an action management system that simulates all these steps was designed. In this system, the actions defined by the user throughout the smallest building blocks are recorded in the system and depending on the action building blocks; it is ensured that it is operated safely and sequentially according to the rules previously determined by the user.

3.1. Methods and Technologies

This section describes the methods and the technologies used for the designed action management system. These methods and technologies are as follows: Spring framework, Hibernate framework, Java messaging service, Quartz framework, Ehcache framework, microservice architecture and database.

Spring is an open source application development framework for Java and can be considered as “framework of frameworks” because it supports the use of frameworks such as Hibernate and Quartz. Hibernate is an object/relational mapping library designed for software developers. By providing the relationship with the database according to the object-oriented models, it simplifies the operations performed on the database and also strengthens the established structure. Java Messaging Service (JMS) is a synchronous or asynchronous programming interface that enables messaging between Application Programming Interface (API) software. In software projects, there is an increase in the dimension of the code when new functionality features are added. After a while, it becomes difficult to dominate the project because of this increase. In order to combat such problems in a monolithic project, abstractions and modules are created in the code as much as possible. Micro-service is the name given to this type of small and co-operating services. Quartz is an open source business planning system that can be used for small or large enterprise systems. Basically, the structure proceeds in two stages. Firstly, the active job must be determined. Secondly, the next job is determined. Figure 1 shows the Quartz framework structure.

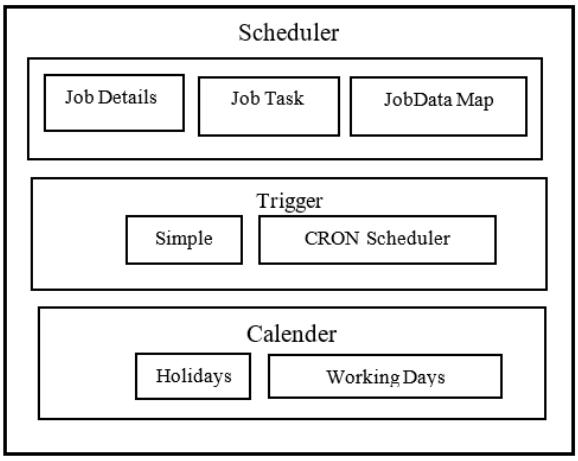

Figure 1. Quartz framework structure

The Ehcache framework is an open source caching mechanism used to improve system performance, reduce database load, and simplify scalability. Ehcache is a very powerful framework, and it has become one of the most widely used Java-based caching mechanisms for distributed caching and for the use of disk memory in cases of insufficient RAM. In Figure 2, Ehcache framework structure is shown.

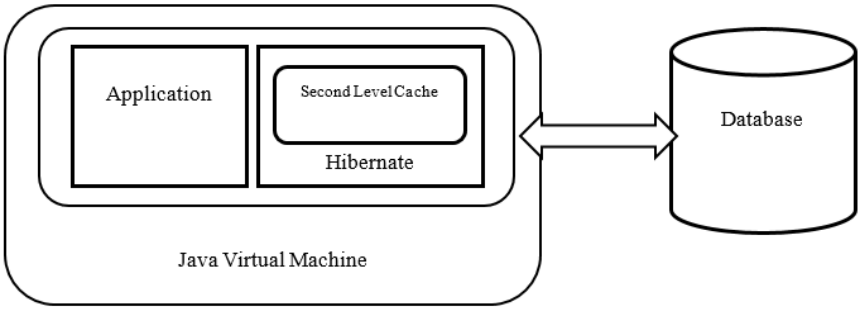

Figure 2. Ehcance framework structure

The application developed during this study has no database dependency. Thanks to the frameworks used, it can work on all of the preferred databases. In this example, the application runs on the Oracle database. Designed with information security and performance in mind, the data in this system can be divided into 4 groups according to the information security and performance. Figure 3 shows the configuration database model of the application.

- Action configuration data: Action data designed by users on the management screens.
- Profile data of the customer: Personal and profile data of the customer.
- Customer-transaction-event data: Customer-owned; motion and event data.
- Action data: The records of the actions taken to the customer.

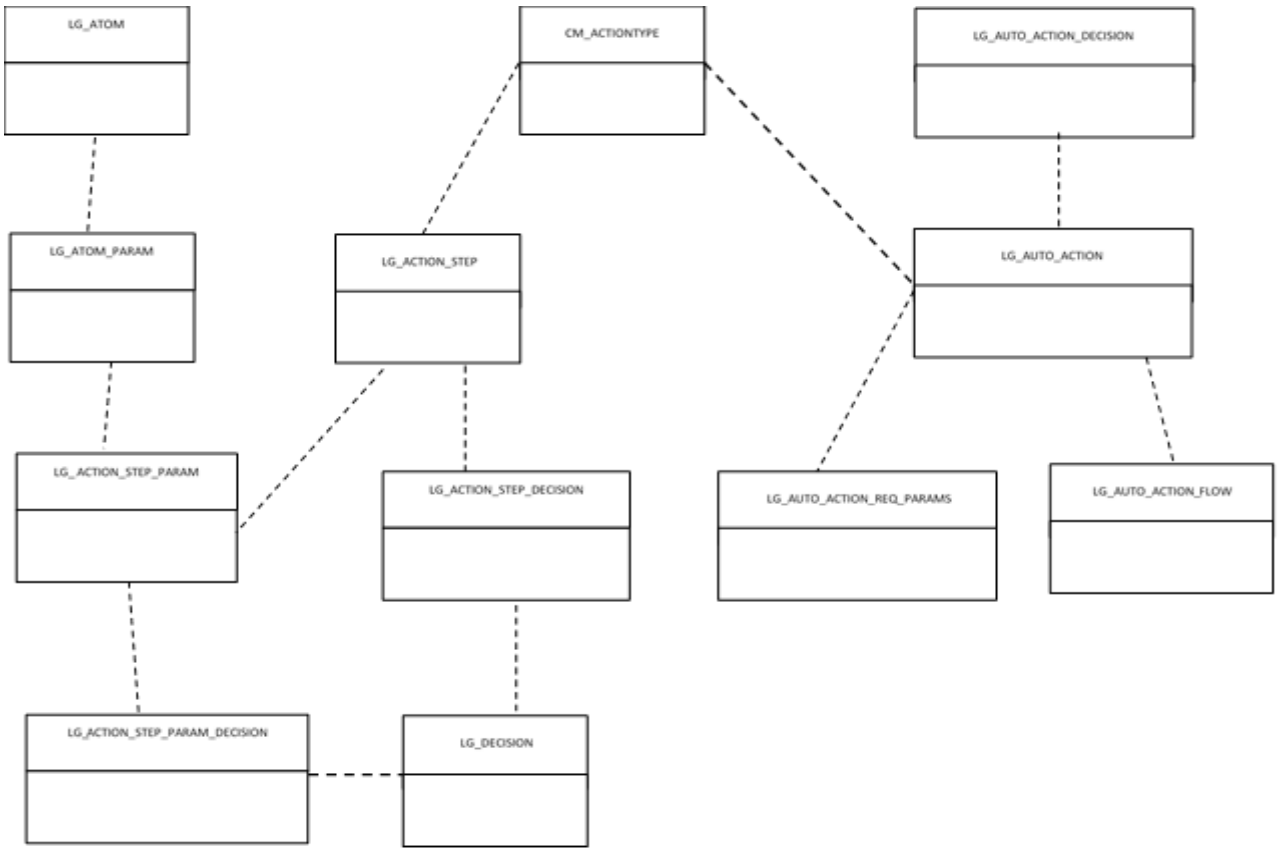

Figure 3. The configuration database model of the application

The tables in this database are described below.

LG_ATOM: It holds the information of atoms which are the basic building block of the action system. The fields in this table are ATOM_ID (NUMBER (10)), NAME (VARCHAR2 (100 Byte)), DESCRIPTION (VARCHAR2 (500 Byte)), CLASS_NAME (VARCHAR2 (500 Byte)).

CM_ACTIONTYPE: It holds the information about action types. The fields in this table are ACTION_CODE (NUMBER), ACTION_TYPE (VARCHAR2 (50 Char)), MAIN_XML_TEMPLATE (XMLTYPE), REF_PARAM_COUNT (NUMBER), ENTITY_PARAM_COUNT (NUMBER), CHILD_XML_TEMPLATE (XMLTYPE).

LG_ATOM_PARAM: It holds the parameters of atoms, which are the basic building blocks of the action system. The fields in this table are ATOM_PARAM_ID (NUMBER (10)), ATOM_ID (NUMBER (10)), PARAM_NAME (VARCHAR2 (200 Byte)), REQUIRED (NUMBER (1)).

LG_ACTION_STEP: It holds information about the conditions under which the flow will continue. The fields in this table are ACTION_STEP_ID (NUMBER (10)), CM_ACTION_TYPE_ID (NUMBER (10)), ATOM_ID (NUMBER (10)), NAME (VARCHAR2 (100 Byte)), DESCRIPTION (VARCHAR2 (500 Byte)), ORDER_NO (NUMBER (5)), EXIT_ON_ERROR (NUMBER (1)), IS_ROLLBACK_ACTIVE (NUMBER (1)).

LG_ACTION_STEP_PARAM: It holds the method of calculating the parameters for each step of the action. The fields in this table are ACTION_STEP_PARAM_ID (NUMBER (10)), ACTION_STEP_ID (NUMBER (10)), ACTION_PARAM_ID (NUMBER (10)), PARAM_SOURCE_ID (NUMBER (10)), ORDER_NO (NUMBER (10)).

LG_ACTION_STEP_DECISION: It holds the decision control conditions for each step of the action. The fields in this table are ID (NUMBER (10)), ACTION_STEP_ID (NUMBER (10)), DECISION_ID (NUMBER (10)), VALUE (VARCHAR2 (1000 Byte)), COMPARISON_TYPE (VARCHAR2 (100 Byte)), EXIT_FROM_ACTION (NUMBER (1)).

LG_ACTION_STEP_PARAM_DECISION: It holds the operating conditions of the parameters for each step of the action. The fields in this table are ID (NUMBER (10)), ACTION_STEP_PARAM_ID (NUMBER (10)), DECISION_ID (NUMBER (10)), VALUE (VARCHAR2 (1000 Byte)), COMPARISON_TYPE (VARCHAR2 (100 Byte)), EXIT_FROM_ACTION (NUMBER (1)).

LG_DECISION: It holds the decision structures of the action system. The fields in this table are DECISION_ID (NUMBER (10)), NAME (VARCHAR2 (500 Byte)), CLASS_NAME (VARCHAR2 (500 Byte)), GUI_SQL_TEXT (CLOB), PARAM_ID (NUMBER (10)), PARAM_SOURCE_ID (NUMBER (10)).

LG_AUTO_ACTION_DECISION: It holds the definitions of decisions for sequential action definitions of the action system. The fields in this table are ID (NUMBER (10)), AUTO_ACTION_ID (NUMBER (10)), PARAM_SOURCE_ID (NUMBER (10)), STATUS (NUMBER (1)), CREATE_DATE (DATE), ORDER_NO (NUMBER (4)), FAIL_AUTO_ACTION_ID (NUMBER (10)), FAIL_MISSION_RM_SRC_ID (NUMBER (10)).

LG_AUTO_ACTION: It holds the sequential action of the action system. The fields in this table are AUTO_ACTION_ID (NUMBER (10)), NAME (VARCHAR2 (100 Byte)), DESCRIPTION (VARCHAR2 (500 Byte)), STATUS (NUMBER (1)), CM_ACTION_TYPE_ID (NUMBER (10)), EVENT_CODE (NUMBER (10)), REASON_CODE (NUMBER (10)), DEFAULT_RUNDATE_IN_MINUTES (NUMBER (10)), RETRY_IN_MINUTES (NUMBER (10)), RETRY_MAX_COUNT (NUMBER (10)), CREATSE_DATE (DATE), DEFAULT_RUNDATE_RULES (VARCHAR2 (100 Byte)), GROUP_NAME (VARCHAR2 (40 Byte)), WAITING_ACTION (NUMBER (5)).

LG_AUTO_ACTION_REQ_PARAMS: It holds the parameter values for sequential action definitions of the action system. The fields in this table are ID (NUMBER (10)), AUTO_ACTION_ID (NUMBER (10)), REF_OR_ENTITY (VARCHAR2 (1 Byte)), NT_PARAM_CODE (NUMBER (10)), PARAM_SOURCE_CODE (NUMBER (10)), STATUS (NUMBER (1)), CREATE_DATE (DATE).

LG_AUTO_ACTION_FLOW: It holds the flow structure for sequential action definitions of the action system. The fields in this table are ID (NUMBER (10)), AUTO_ACTION_ID (NUMBER (10)), PARAM_SOURCE_CODE (NUMBER (10)), NEXT_AUTO_ACTION_ID (NUMBER (10)), ORDER_NO (NUMBER (10)), STATUS (NUMBER (1)), CREATE_DATE (DATE).

3.2. Proposed System

This section describes the design of the application performed in this study. For this purpose, the concepts of the atom, parameter, action, sequential action and decision-control building blocks are explained. Figure 4 shows the proposed system design.

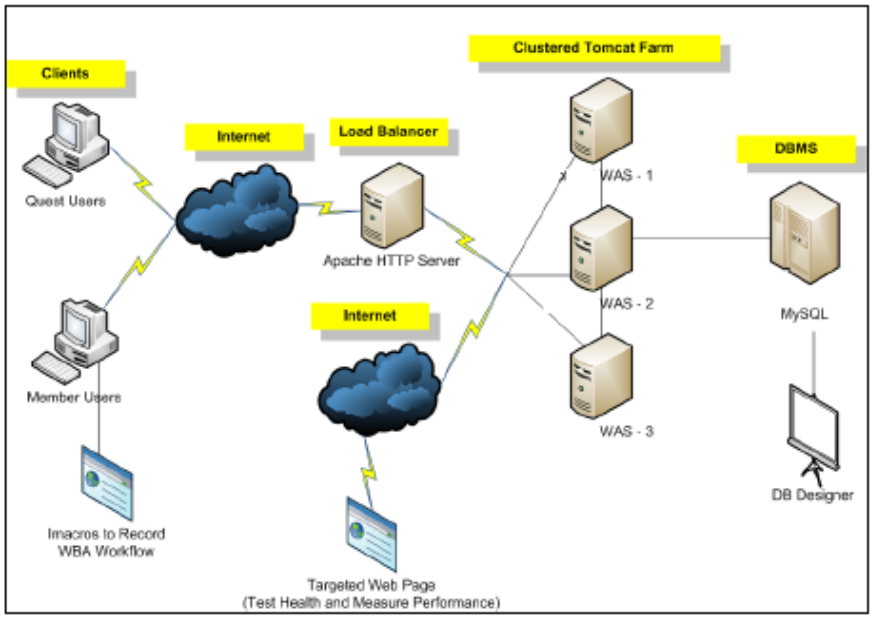

Figure 4. Proposed system design

3.2.1. Atom

Atom is the smallest building block of the designed action management system and is the name of each code snippet that provides the specified interface design. It is designed to perform a certain simple operation. Its task is to perform singular operations. In order to give the system a new simple process, it is necessary to develop it according to the rules and define it to the system. This situation allows the application to have an expandable design. When a new atom is added to the action system, it can be used as an action step to create an action definition. There is three basic information about atoms: performing basic operations, withdrawing the basic operation and keeping the information whether the basic operation is suitable for the withdrawal. Apart from these, there are various atomic processes specific to the application. Examples of atomic transactions are: money withdrawal, invoice opening, list addition, list removal, sequential action request generation.

3.2.2. Parameter

Atoms need some information while they are working. The parameter structure is the structure that allows this information to be automatically retrieved from external systems. This structure allows to read any parameter value with a specified design and provides flexibility during the action by reading the desired type of values from internal or external systems. Using these calculated values, the action is taken for users. The parameter structure of the action system is also designed to expand dynamically. Using the specified design, customized parameter reading services can be added to the action management system. For example, in one embodiment, current debt amount of the subscriber and data subscriber's data service status on-off information can be considered as parameters.

3.2.3. Action

Action is formed by combining one or more action steps. The action steps hold information about the sequence in which they will run, how to proceed if there is an error at which step, and whether there is a process that requires rollback. System administrators design their actions from the action configuration unit and make them requestable from the action collection unit.

3.2.4. Ordered Action

Ordered Action is the structure that enables the creation of workflows in the action management system. Actions are designed to be self-feeding and can trigger a different action. This triggering occurs through scheduled tasks. A request is left to the queue to run at any time. When the time comes, the conditions of the sequential action are controlled, and a new action is run. Sequential action consists of three foundations. These foundations can be expressed as:

- decision parameters for taking action,
- parameter values required for action
- another action as a scheduled task after the action is taken.

3.2.5. Decision – Control Building Blocks

The flexible decision-control building blocks provide the control mechanism of the action management system. With the improvements made according to the rules, new decision structures can be added to the action management system. It provides instant access to data from the internal system or external system to change the flow at the time of action and to select the parameters. The general control structures provide the decisions that affect the flow in the action management system by comparing the values with the specified values in the system after reading the values from the desired external systems. Decision-control building blocks are used in two different structures:

- **Parameter Control Structures**: They allow the user to select the parameters of the action step. They provide the parameters under which the system will be retrieved. For example, different conditions may be added to determine the invoice amount when opening a payment invoice. Different amounts of rules are provided to work according to these conditions.
- **Action Step Control Structures**: They allow the user to decide whether operations can be executed during action steps. For example, for a subscriber who does not want a Short Message Service (SMS), the SMS information step may not need to be run.

4. Experimental Study

In this study, an action management system has been designed for the prevention of cyber fraud and risk analysis by using Java programming language, and it was used in a telecommunication company in Turkey. This application, which is designed in accordance with the micro-service architecture and called Lego, consists of a combination of different units operating independently from one another. In this way, application interruptions are kept to a minimum and load distribution is ensured. In this embodiment, the units described below and described in the subsections are included. In Figure 5, these units were shown.

- Action Configuration Unit
- Action Collection Unit
- Action Management Unit
- Action Processing Unit

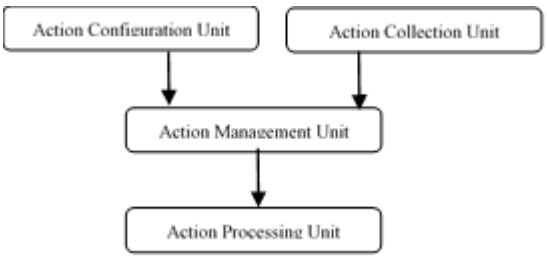

Figure 5. Action units

4.1. Action Configuration Unit

In the study, first of all, Lego Action Configuration Unit was developed using Java programming language. This unit enables the actions to be configured by authorized users from the Action Configuration Unit screens and to be defined from the atoms that are the smallest unit that make up the action. The action consists of at least one structured atom. The structured state of the atoms is called the "Action Step". The unit consisting of atom, input parameters, input parameter controls, atom controls is called action step.

An action can consist of one or more action steps. An action step refers to the smallest unit of work called the atom. For each action step, the required parameter definitions are configured. More than one source can be specified for a parameter. In this case, the priority order of the parameters is determined. In addition, control steps for reading a parameter value are defined. If it passes the control steps, the relevant parameter source is used. If it cannot pass the control step, the next parameter source is passed. Control steps are defined for the action step. Control steps are used to decide whether the action step will run or not. They can be defined in general if necessary, in case of not passing the control defined in this way, the action will be terminated. Each of the action steps is configured for transaction integrity. An action step keeps the information that in case of an error, the next step will be passed, the action will be completed at this step, or the previous steps will be undone. Table 1 shows an example of configuration screen.

Table 1. An example of configuration screen

Step	Order	Name	Exit CnError	Parameters
610	1	Reactivation	√	No records found.
620	2	Remove Blacklist	X	Order | Code Name | Req. | Source | NTParam | Default | Param Decisions (Value | Decision | Type) 1 | ASE_PARAM | true | Ase Param | - | 2 | No records found.
630	3	Remove FboList	X	Order | Code Name | Req. | Source | NTParam | Default | Param Decisions No records found.
211	4	Reactivation	X	Order | Code Name | Req. | Source | NTParam | Default | Param Decisions (Value | Decision | Type) 1 | YTS_FRAUD | false | Update YTS Fraud | - | 1 | No records found.
640	6	Auto action	X	Order | Code Name | Req. | Source | NTParam | Default | Param Decisions (Value | Decision | Type) 1 | AUTO_ACTION_DEF_ID | false | Update YTS Fraud | - | 1622 | No records found.
650	7	VAS	X	Order | Code Name | Req. | Source | NTParam | Default | Param Decisions (Value | Decision | Type) 1 | SDP_TEMP_BLOCK_FLAG | true | false | - | false | No records found.

As an example, an action consisting of six steps has been created. If the first action step receives an error, the action process will be completed here and will be deemed as incorrect. Since the "exitOnError" feature is not active in the other steps, it can be called optional steps. Again, three decision control structures were added for the first step and the "Exit" feature was defined as active for these steps. In this case, if any decision control is negative, the action process will be terminated and there will be no transition to other steps. In case of positive results from all three controls, the action step

will work. As can be seen on the second step, an atom parameter has been defined and its value is given a fixed value of two. No decision control structure has been added on the parameter.

4.2. Action Collection Unit

In this study, Action Collection Unit was designed secondly. This unit receives the information of the actions to be taken via Representational state transfer (REST) service, which is in communication with the external systems and saves it to the database system. In addition, it is responsible for carrying out the necessary authorization and security controls regarding the actions to be taken. It transmits to the Action Management Unit for simultaneous or asynchronous action processing on demand. Figure 6 shows examples of customer requests and responses in this unit.

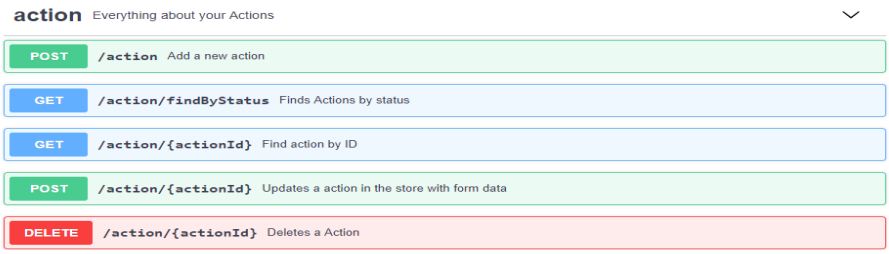

Figure 6. Request and response examples

As shown in Figure 6, the action collection service offers five different operations which are described below.

- **New Action Request**: It enables the action request from external systems to be recorded in the action management system. Authorization checks and time information of the relevant system are determined.
- **Finding Action Request by Status**: Action listing service is offered to external systems. It gives the action list according to date, action status.
- **Find Action with Reference Number**: Another action listing service offered to external systems. It gives the action list according to the reference number.
- **Action Request Update**: It performs the update of action requests from external systems.
- **Action Request Delete**: It performs action cancellation requests from external systems.

An example of action request information is given in Figure 7. As shown in the example, action request data consists of customer information and action information. The applications that make action request indicate which action to take with code. Here, the user is told which action to take and the parameters of the action are transmitted as a list from the key-value structure.

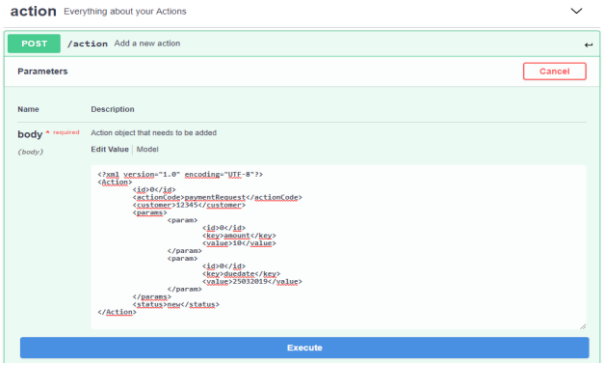

Figure 7. An example action request data

4.3. Action Management Unit

In this study, Action Management Unit was designed as the third. This unit ensures the correct and safe reception of the defined actions and communicates with the Action Collection Unit, the Action Configuration Unit, the Action Processing Unit and the database. The way this unit works consists of the following steps:

- It finds the desired action from the database with the action information collection unit.
- It transmits the small action particles recorded together with the action to the Action Processing Unit for sequential operation in order to fully perform the action.
- It is connected to the Action Processing Unit by taking action particles, conditions and parameters.
- It transmits the action steps to the Action Processing Unit according to the sequence at the specified time.
- The condition connected to the action particles is retrieved from the Action Processing Unit either successfully or unsuccessfully.
- It takes the action results of the action particles from the Action Processing Unit as successful or unsuccessful and records the last status of the action by updating the database.
- Based on the definition of the action particles and the predetermined conditions, it decides on the failure of the action performed by the Action Processing Unit to quit the action and no further processing. Figure 8 is a flow chart of the Action Management Unit.

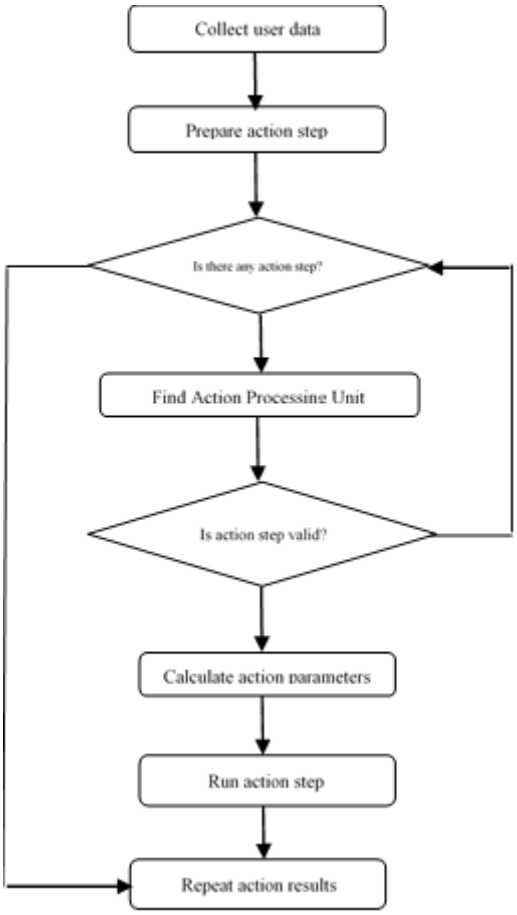

Figure 8. Action Management Unit. Flowchart

The Action Management Unit is responsible for process integrity management at all times. If a step is completed successfully, it completes the relevant records and transaction integrity of the action step; otherwise, it performs process integrity management according to the settings of the action step. At this point, the user decides what to do when he/she gets an error in the step. When an error is received, if there is an adjustment to skip the step, it records the corresponding step incorrectly, undoes the operations of the step and sends a message to the Action Processing Unit for the next step to run. If there is an adjustment to undo all action when it receives an error, it finds the previous running steps and performs the undo procedures for each of these steps. One of the most important parts of the action management system is the decision control structures. These structures, which determine how the action flow occurs, also decide which parameter calculation method to use. Table 2 shows the definitions of an exemplary action step decision control structure.

Table 2. Action step decision control definition

Value	Decision	Exit	Type
1	Reason	√	eq
240	Event	X	eq
1	CustomerType	X	eq
101	CustomerType	X	eq
30	Event	X	eq
OK	Control	√	eq
1	SubsciberType	X	eq

The functional relationships between the selected decision control service and the defined values are listed below.

- **equal to**: In order to accept the result as positive, the value defined in the action step must be equal to the value in the decision-control service.
- **greaterThan**: In order to accept the result as positive, the value defined in the action step must be greater than the value available in the relevant decision-control service.
- **smallerThan**: In order to accept the result as positive, the value defined in the action step must be smaller than the value available in the relevant decision-control service.
- **inList**: In order to accept the result as positive, the value defined in the action step must be equal to one of the values in the relevant decision-control service.

Figure 9 shows an example of a parameter calculation request information.

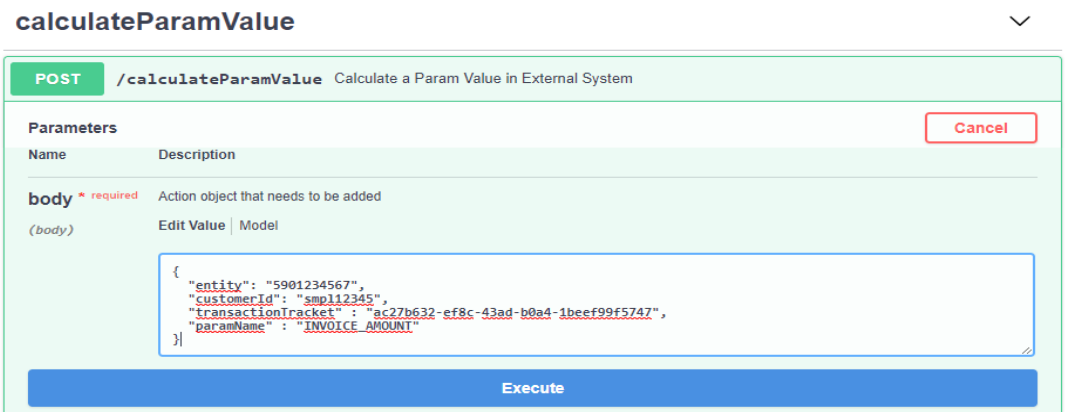

Figure 9. An example of a parameter calculation request information

While the action management system executes the action flow, it manages the flow according to different system variables. In this application, the action management system has gained the ability to manage the flow by reading values over different systems. Thus, faster integration of the action management system in all systems and an increase in development speed has been achieved. For this, a common parametric data reading service structure has been established. It integrates with external systems with a specified REST service interface. Here, responsibility is given to external systems. They are expected to offer services that support a common messaging type. It is ensured that the action management system takes the actions with the correct values by offering the REST services in accordance with the sample in Figure 9 for many different external systems. It is ensured that the latest status of user data in different systems is easily retrieved. Thus, actions are processed more reliably and verified with up-to-date data.

4.4. Action Processing Unit

In this study, Lego Action Processing Unit was developed using Java programming language. The Action Processing Unit in the action management system is in communication with the management unit. The Action Processing Unit manages the processing of the atoms that make up the action so that they can perform the action steps that come to it from the management unit.

The Action Processing Unit manages the calculation of the parameter values of the action step. Thus, it enables the action step to be processed according to the selected parameter values. Parameter definitions of an example action step are shown in Table 3.

Table 3. Action step parameter definition

Order	Code Name	Req.	Source	SQL	NTParam	Default	Param Decisions		
							Value	Decision	Type
1	ADVANCE_AMOUNT	False	70 AMOUNT_REQUEST		70-E		123	Event	eq
							45	Event	eq
							34	Event	eq
2	ADVANCE_AMOUNT	False	72 CALCUNBAMT		72-E		Value	Decision	Type
							49	Event	eq
1	ADVANCE_DUEDATE	False	17 DUEDATE	Service 36	17-E		Value	Decision	Type
							No records found.		
3	ADVANCE_AMOUNT	False	70 AMOUNT_REQUEST		70-E	14	Value	Decision	Type
							No records found.		

The Action Processing Unit will calculate the value of the parameters related to the definitions here. If there is a service parameter definition as a priority, parameter information will be calculated from this service. Then, if there is a variable reading definition for which action is requested, these values are prioritized. If these two last values are undefined, the fixed value definition will be considered. As seen in Teble 3, the invoice amount information has been repeated three times. In this case, the calculation of parameter values is processed by the Action Processing Unit in a way that the definition sequence number is taken into account. Again, as seen in Table 3, parameter decision controls are seen. There are three decision control definitions in the first row for the invoice amount parameter. In the second row, there is one decision control definition. If a positive result cannot be obtained from these two control sets, the invoice amount parameter in the third row will be used as an alternative fixed value.

The Action Processing Unit is also responsible for the registration, time and process integrity management of the atoms being operated. The transaction results are returned to the Action Management Unit as successful or unsuccessful. Depending on the definition of the action steps and the pre-determined conditions, if the result of the operation performed

by the Action Processing Unit fails, a message is notified to the management unit in order to exit the action and to revoke all transactions related to the particles that were previously run. Figure 10 shows the flowchart of the Action Processing Unit.

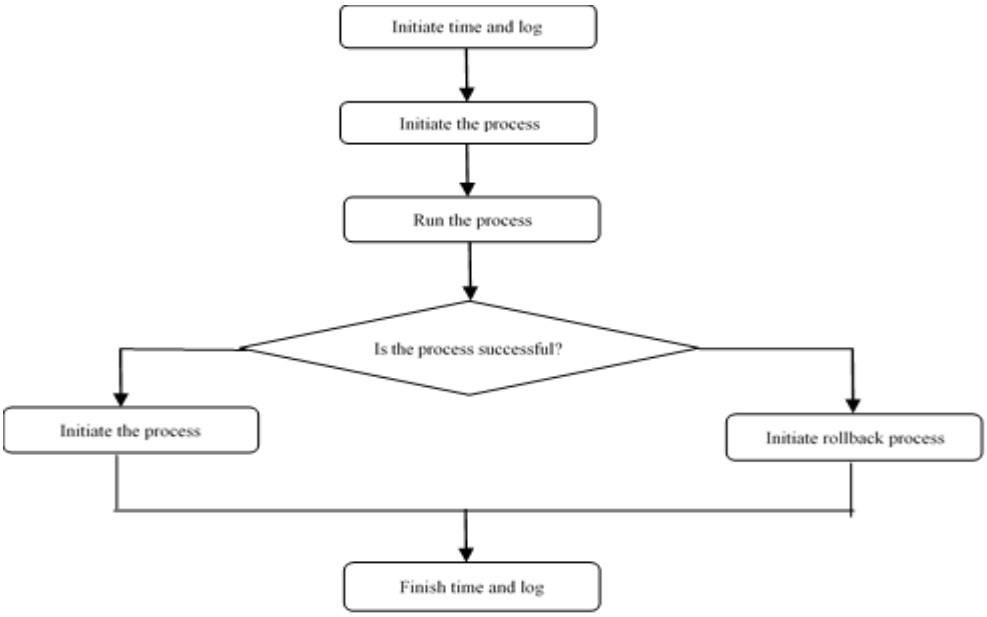

Figure 10. Action processing unit flowchart

Adding New Atom to the Action Processing Unit: Action management systems should ensure that action is taken on all systems with fast integrations in a structure with complex systems. In this application, new atoms that operate on different systems were developed for the action management system, enabling the action management system to manage risky situations in all systems. By connecting to external systems with REST services, which are mostly provided, software development of atoms specific to the system is provided with the messaging types specific to that system. However, it is aimed to ensure faster integration of the action management system in all systems and to increase the speed of development. For this situation, a common atom processing service structure has been established. It can be integrated into external systems with a specified REST service interface. Here, responsibility is given to external systems. They are expected to offer services that support a common messaging type. Thus, the Action Management Unit knows only the Uniform Resource Locator (URL) information of the relevant service, thanks to a common atom development. This service provides actions to be taken in external systems with the requests to be sent to the URL address. Figure 11 shows a sample of the request of the relevant service. Figure 12 shows Action Management System sample application flow.

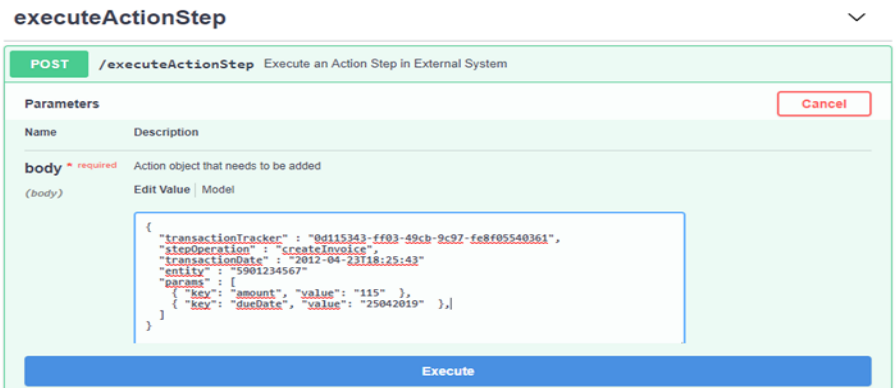

Figure 11. A sample of the request of the relevant service

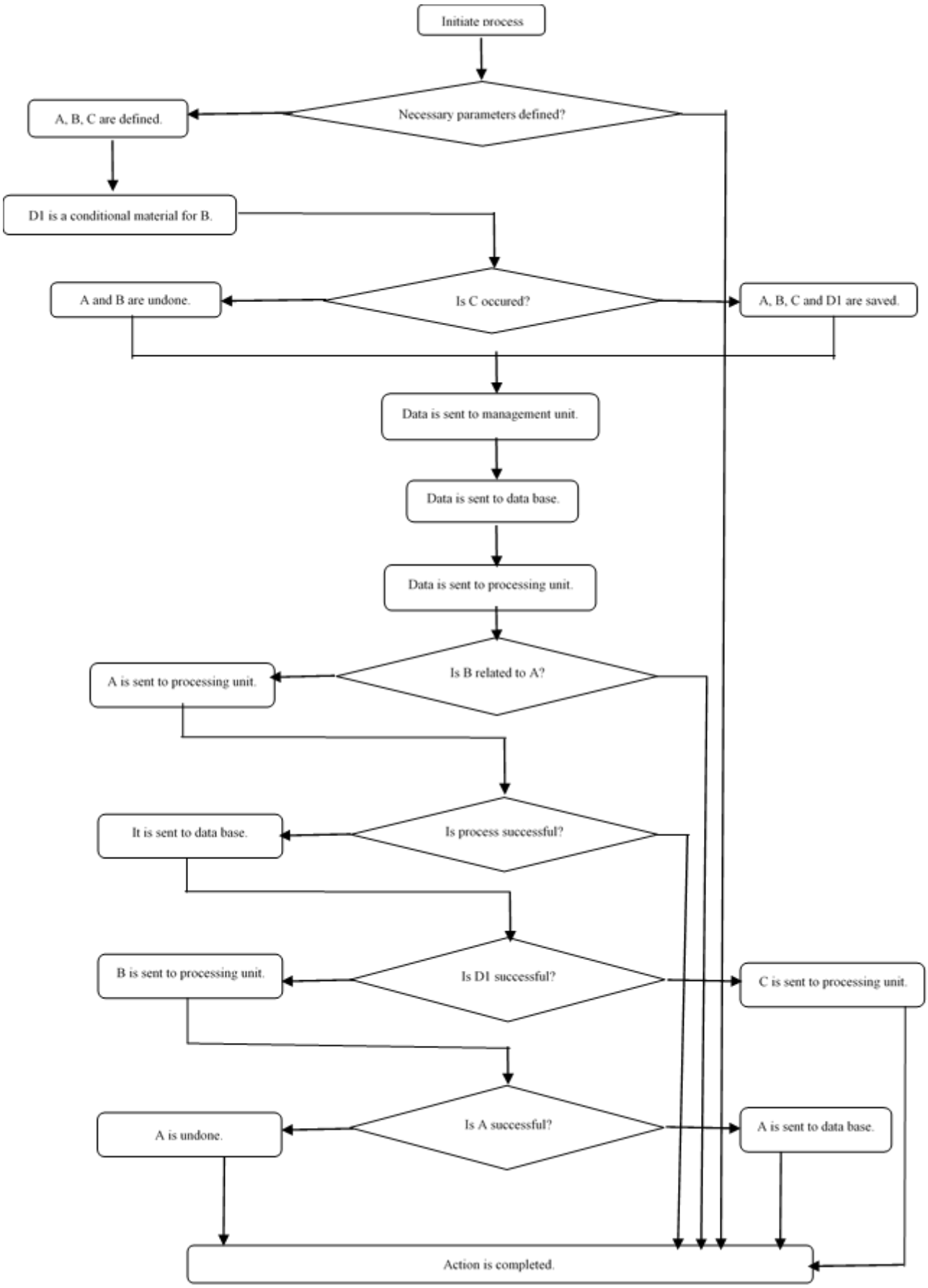

Figure 12. Action Management System sample application flow

4.5. Experimental Results

After the design of the action management system has been completed, case study on its usage for cyber fraud prevention and risk analysis is done with the data of a telecommunication company in Turkey as a case study. The proposed action system was used in 2014 for this case study. Action values obtained before and after the use of this action system are presented in Table 4 as monthly data for 2014 and 2015. After this date, any data cannot be obtained from the company. These data were used to be an example of the usage of proposed action management system. The explanations of the actions in the table are as follows:

- **Action A - Investigation**: The action of suspicious transactions belonging to the subscriber.
- **Action B - Service opening**: The action taken for a subscriber service via SMS, Internet and so on.
- **Action C - Call center**: The action for the information and additional actions made by reaching the subscriber through the call center.
- **Action D - Information**: The action for transactions made to the subscriber via SMS, e-mail, and so on.
- **Action E - Trip**: The action for activation of the search of the line again.
- **Action F - Shutdown**: The action for suspending the search of the line.

Table 4. Action numbers in 2014 and 2015

	ACTION TYPES						
YEAR-MONTH	**Action A**	**Action B**	**Action C**	**Action D**	**Action E**	**Action F**	**TOTAL**
2014-01	26K	24K	36K	76K	24K	31K	217K
2014-02	23K	36K	35K	75K	24K	26K	219K
2014-03	22K	31K	35K	96K	32K	33K	249K
2014-04	24K	24K	45K	96K	30K	53K	272K
2014-05	21K	21K	34K	81K	33K	44K	234K
2014-06	28K	23K	41K	96K	32K	42K	262K
2014-07	22K	29K	37K	84K	38K	50K	260K
2014-08	38K	23K	34K	89K	51K	71K	306K
2014-09	56K	40K	48K	178K	75K	113K	510K
2014-10	47K	38K	38K	201K	59K	56K	439K
2014-11	46K	33K	46K	270K	54K	66K	515K
2014-12	54K	38K	50K	262K	59K	88K	551K
2015-01	50K	33K	41K	230K	54K	80K	480K
2015-02	48K	34K	32K	218K	48K	69K	449K
2015-03	63K	35K	72K	202K	45K	61K	478K
2015-04	53K	41K	67K	178K	47K	65K	451K
2015-05	50K	28K	58K	192K	39K	50K	417K
2015-06	46K	28K	63K	169K	41K	54K	401K
2015-07	49K	30K	68K	219K	42K	63K	471K
2015-08	45K	29K	74K	207K	48K	82K	485K
2015-09	56K	24K	95K	209K	53K	72K	509K
2015-10	54K	26K	98K	207K	60K	87K	532K
2015-11	66K	28K	86K	193K	54K	73K	500K
2015-12	53K	31K	81K	213K	65K	84K	527K

The results show that, after using the proposed Action Management System, the number and the dimension of all actions (Action A, Action B, Action C, Action D, Action E and Action F) individually and the total number actions captured are increasing except for a few exceptional situations.

5. Conclusion

In this study, an action management system has been developed to ensure that the actions required to be taken automatically in order to protect users from risks in the cyber world are processed correctly and safely. This system allows actions to be defined independently and processed in a user-specific manner, and undertakes all process management tasks. It also provides fast asynchronous return as well as asynchronous operation options for long-term operations and allows the creation of a workflow, allowing sequential operations to be performed reliably in sequence. The action management system provides fast and accurate actions thanks to its flexible and modular configuration. This study is conducted in a telecommunications company in Turkey, it examines the action values prior to and after the use of the system, and the proposed action values are shown to increase with the help of an action management system. Thanks to the system, which does not wait for end-user approval, the following benefits have been observed because reliable actions are taken around specified rules:

- decision-making times are shortened,
- transaction waiting queues are shortened,
- the decision-making process is automated,
- resource utilization is optimized,
- operational costs are reduced,
- human resources are used correctly,
- productivity is increased,
- changes in the action process take place quickly,
- processes are adapted to changing cases quickly,
- human errors are mostly eliminated,
- cases are assigned to the right people when necessary.

References

[1] Fukuta, S.F.L. and Nishigaya, T.F.N. (2000). Method and apparatus for determining dynamic flow in a distributed system, EP1120710A2 numbered patent.

[2] Alshab, M.A., Bales, P.J., Covington, R.D., Theophilus, J.D. and Trotter, L.M. (2006). Method and system for building, processing, and maintaining scenarios in event-driven information systems, WO2007035452A1 numbered patent.

[3] Berg, W.C., McCallum, D.J. and Newman, R.W. (1995). Method and system for managing workflow, US5999911A numbered patent.

[4] Randell, J. (1996). Workflow real time intervention, US5826020A numbered patent.

[5] Shapiro, M., O'Brien, J.W., Matheson, C.E., Rodriguez, P.R. and Costa, M. (2003). System-wide selective action management, US7290002B2 numbered patent.

[6] Kawai, M., Rimoldi, A. and Bassi, G. (2001). Action management support system, US20030233162A1 numbered patent.

[7] Akhilomen, J. (2013). Data Mining Application for Cyber Credit-Card Fraud Detection System, Industrial Conference on Data Mining, Advances in Data Mining. Applications and Theoretical Aspects.

[8] Mirea, V., Blăjan, A. and Ionescu, L. (2011). Fraud, Corruption And Cyber Crime In A Global Digital Network, Economics, Management, and Financial Markets, 6(2): 373-380.

[9] Bignell, K.B. (2006). Authentication in an Internet Banking Environment; Towards Developing a Strategy for Fraud Detection, International Conference on Internet Surveillance and Protection.

[10] Dzomira, S. (2014). Electronic Fraud (Cyber Fraud) Risk In The Banking Industry, Zimbabwe, Risk Governance & Control: Financial Markets & Institutions, 4(2): 16-26.

[11] Arya, A.S., Ravi, V., Tejasviram, V., Sengupta, N. and Kasabov, N. (2016). Cyber fraud detection using evolving spiking neural network, International Conference on Industrial and Information Systems.

[12] Singh, P. and Singh, M. (2015). Fraud Detection by Monitoring Customer Behavior and Activities, International Journal of Computer Applications, 111(11): 23-32.

[13] Cai, Y. and Zhu. D. (2016). Fraud detections for online businesses: a perspective from blockchain technology, Financial Innovation, 2.

[14] Gupta, P. and Mundra, A. (2015). Online in-auction fraud detection using online hybrid model, International Conference on Computing, Communication & Automation.

[15] Ford, V., Siraj, A. and Eberle, W. (2014). Smart grid energy fraud detection using artificial neural networks, IEEE Symposium on Computational Intelligence Applications in Smart Grid.

[16] Krenker, A., Volk, M., Sedlar, U., Bešter, J. and Kos, A. (2009). Bidirectional Artificial Neural Networks for Mobile-Phone Fraud Detection, ETRI Journal, 31(1): 92-94.

[17] Olszewski, D. (2014). Fraud detection using self-organizing map visualizing the user profiles, Knowledge-Based Systems, 70: 324-334.

[18] Sethi, N., Gera, A. (2014). A Revived Survey of Various Credit Card Fraud Detection Techniques, International Journal of Computer Science and Mobile Computing, 3(4): 780-791.

[19] Rana, P.J. and Baria, J. (2015). A Survey on Fraud Detection Techniques in Ecommerce, International Journal of Computer Applications, 113(14): 5-7.

[20] Abdallah, A., Maarof, M.A. and Zainal, A. (2016). Fraud detection system: A survey, Journal of Network and Computer Applications, 68: 90-113.

[21] Arkel, J.H.V., Wagner, J.J., Schweyen, C.L., Mahone, S.M., Tada, D.D., Curtis, T.J. and Hagins, S. (2012). Predictive modeling processes for healthcare fraud detection, US20130006668A1 numbered patent.

[22] Arkel, J.H.V., Wagner, J.J., Schweyen, C.L., Mahone, S.M., Tada, D.D., Curtis, T.J. and Hagins, S. (2012). Near real-time healthcare fraud detection, US20130006655A1 numbered patent.

[23] Tyler, M., Basant, N., Robin, P. and Rahman, S. (2010). Healthcare insurance claim fraud detection using datasets derived from multiple insurers, US8214232B2 numbered patent.

[24] Crawford, S.L., Erickson, C., Miagkikh, V., Steele, M., Thorsen, M. and Tolmanov, S. (2008). Systems and methods for fraud detection via interactive link analysis, S20090044279A1 numbered patent.

[25] Turgeman, A., Kedem, O. and Rivner, U. (2015). Method, device, and system of generating fraud-alerts for cyber-attacks, US9552470B2 numbered patent.