Quality Improvement and Quantitative Modeling – Using Mashups for Human Error Prevention

Carlos Raniery P. dos Santos,
Lisandro Zambenedetti Granville
Institute of Informatics - UFRGS
Porto Alegre, RS, Brazil
Email: {crpsantos, granville}@inf.ufrgs.br

Larisa Shwartz, Nikos Anerousis
IBM T.J. Watson Research Center
Hawthorne, NY 10532, USA
Email: {lshwart, nikos}@us.ibm.com

David Loewenstern
WhitePages
New York, NY 10018, USA
Email: dloewenstern@whitepages.com

***Abstract*—Modern IT service provider organizations are under a continuous pressure to increase their competitiveness. Ways to reduce costs while improving performance – in terms of effectiveness, productivity, and quality – of services are a key focus area for companies in the IT industry. The existence of human operators in this industry, although required, may introduce defects in the process. In IT service provider organizations, preventing human errors from affecting the system is critical because of the strict requirements for quality. Our work in this paper is inspired by Six Sigma and is based on partial automation and process redesign to prevent human errors from occurring, or at least to reduce their frequency. In particular, we analyze the usage of mashups as an effective approach to cope with errors introduced by human operators while performing their daily activities in the context of IT Service Management (ITSM).**

I. Introduction

Modern IT service provider organizations are under a continuous pressure to increase their competitiveness. Ways to reduce costs while improving performance – in terms of effectiveness, productivity, and quality – of services are a key focus area for companies in the IT industry. However, despite all the solutions that have been proposed, modeling and optimizing human-centered processes remains a burdensome task. The human operator may be influenced by multiple factors and execute the process in a different way each time, thus introducing a significant variability in the final process outcome.

Automation is often used by the companies to obtain tight performance bounds [1]. However, that may not be feasible in some cases because of additional effort to deploy and maintain the automation infrastructure [2]. This is specially true when processes are constantly changing or are too complex to be automated. In such cases, the existence of human operators, although required, may introduce defects in the process. Previous researches have showed that human error is the largest contributor to reduced dependability in IT systems, and occur despite experience [3] [4] [5]. Even additional training and familiarity with systems cannot eliminate all the errors (*i.e.*, mistakes, slips, or lapses) made by the human operator. For example, a database administrator can accidentally delete a semicolon when executing consecutive commands, which may lead to a system outage or to a permanently damaged data.

In IT service provide organizations, preventing human errors from affecting the system – by avoidance or interception – is critical because of the strict requirements for quality. Errors in such environment introduce a considerable variance in quality, thus reflecting in severe penalties from Service Level Objective (SLO) violations. Variability is measured as the number of defective units at the output of the process, and is usually addressed by IT service quality engineers using techniques from the manufacturing domain. For example, SixSigma [6] and Lean [7] are two popular methodologies used in conjunction (*i.e.*, Lean Sigma [8]) to identify and remove the causes of defects and minimize the variability in both manufacturing and business processes.

In this paper we address the problem of reducing the occurrence of human errors in the Request Fulfillment process by using mashups technology. In the context of ITSM, mashups can be used to improve performance in human-centered processes through partial automation and process redesign. We propose a methodology inspired by SixSigma and based mashups to improve performance of ITSM processes by eliminating defects and reducing variability. In a previous work [9], we have showed that mashups are a feasible solution to tackle inefficiencies in ITSM processes, thus increasing the human operator productivity. Once more, we do not address exogenous elements, such as answering telephone calls. After all, we cannot change the human condition, but we can change the conditions under which humans work. In particular, we focus our work on error-prone activities performed by human operators that can be identified and quantified. The goal of our research is to analyze the usage of mashups as an effective approach to cope with errors introduced by human operators while performing their daily activities in the context of IT Service Management (ITSM).

The key contributions of our research work are:

- Introduce a comprehensive methodology to analyze error-prone activities in human-centered ITSM processes;
- Present a set of mashup patterns and interaction elements to cope with human errors;
- Provide a quantitative model to evaluate and predict the impact of employing such patterns in ITSM processes.

In order to reach these objectives, we focus on the Request

978-3-901882-50-0 ©2013 IFIP

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Downloaded on May 16,2026 at 01:17:39 UTC from IEEE Xplore. Restrictions apply.

Fulfillment process, which is one of the operational processes in IT management and presents a high risk of human error. The proposed methodology uses the Human Error Assessment and Reduction Technique (HEART), which is widely used in the field of Human Reliability Assessment (HRA) for the purposes of evaluating the probability of a human error occurring throughout the completion of a specific task. Event Tree Analysis (ETA) is used for identifying and evaluating the overall probability of human error in the sequence of events performed during the Request Fulfillment process.

The remainder of this paper is organized as follows. In Section II we present the technical background related to this paper. In Section III we identify error prone activities in the context of ITSM and propose a methodology to account for those errors in Section IV. In Section V we introduce a set of mashup patterns and interaction elements for coping with human errors in ITSM processes. In Section VI we demonstrate the application of the proposed methodology on the request fulfillment process and present results of our case study. The paper concludes in Section VII, with a brief summary of our findings and an outline of future work.

II. Background

The objective of this section is to review the technical background of our proposal, which includes more in depth discussions of human error assessment and the mashup technology.

A. Mashups

Over the past few years, a new set of Web 2.0 applications have gained interest from both industry and academia. These applications, termed as mashups, are Web applications created through the integration of external resources available on the Web (*e.g.*, RSS feeds, Web services, and online APIs) [10]. Essentially, the main objective of mashups is to graphically combine and integrate disparate assets (*e.g.*, presentation, data, and functionality) in new ways. Unlikely traditional composition technologies (*e.g.*, BPEL and WSCI), mashups are focused on the end-users, allowing them to create their own applications and encouraging cooperation and reuse.

Mashups can be created following the methodology proposed by Jhingran et. al [11]. In this methodology, three main steps are defined: ingestion, augmentation, and presentation. The ingestion is the process of gathering data from heterogeneous sources (*e.g.*, Web services, RSS feeds, online APIs). During this process wrappers are often used to access and standardize the interested data. In the augmentation step, the retrieved data is integrated and transformed in order to create more meaningful information with purposes different from the original ones. During the presentation step, the result of the augmentation process is displayed in a Web browser to the human user. The presentation can range from a simple text file containing the results of the composition, to a complex and highly interactive map display.

Besides mashups can be created through traditional programming techniques, specific systems (*i.e.*, mashup systems) can be employed by users with no programming expertise during the creation process. Such systems are Web-based development tools that enable end-users to create their own mashups; they are what an Integrated Development Environment (IDE) is to a programming language. Usually, mashup systems perform three main tasks: provide users with visual tools to define new mashups, store created mashups in mashup repositories or libraries, and execute mashups when requested. The three tasks are often performed from a single mashup system, but it is possible for a mashup to be created in one system, executed in another, and stored in a third to form a remote mashup repository.

In the context of ITSM, mashups can be used to improve performance in human-centered processes. We have ourselves [9] introduced a set of mashup patterns to address suboptimal execution of activities, providing a significantly improved orchestration of the process, thus improving the productivity. Further productivity, mashups can also be applied to decrease the number of errors introduced by human operators. The main concepts under the human error area and the technical background related are presented in the next subsection.

B. Human Errors

Human beings have the ability to execute multiple and complex tasks (mental and physical) at the same time. However, although skills and expertise level can vary among people, all humans reach their natural limits. When such limits are reached or exceeded, humans become susceptible to making errors.

Several works are found in the literature aiming to define ”human error”. However, up to now, there is no agreement on a unique definition for the term. In this paper, we assume the definition proposed by Reason [12], who considers that erroneous actions include all situations in which a planned sequence of physical or mental activities failed to obtain a result and whose errors cannot be attributed to interventions of external causes. Reason’s work is based on the Skill, Rule, Knowledge (SRK) based approach [13], which introduces a classification for errors common to all professions (*e.g.*, aviation, military, nuclear power, health care). These errors are of the following types:

1) **Knowledge-based errors** - Occur because of a deficit of knowledge. For example, a person may want to perform some plan or action, but the plan or action is implemented partially or incorrectly because of a lack of knowledge, thus not achieving the proper result;
2) **Rule-based errors** - Occur when a well-known rule is wrongly used, or a situation is misinterpreted. For example, when a person using a particular operating system is forced to use a different one, and then tries to perform the same actions even if the Graphical User Interface (GUI) is different;
3) **Skill-based errors** - Occur when actions are divergent to the original intentions. For example, when an user performs an automatic action, with little attention, and forgets to execute a specific task;

Reason [12] also introduced the Generic Error Modeling System (GEMS), which is an extension of the SRK approach. In that work, Reason observed that the type of errors can be

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Downloaded on May 16,2026 at 01:17:39 UTC from IEEE Xplore. Restrictions apply.

either involuntary or intentional. Involuntary actions (*i.e.*, slips and lapses) are those that deviate from planned intentions and, thus, do not reach their goals. Intentional actions (*i.e.*, mistakes and violations), on the other hand, are performed consciously but the desired result is not achieved. Figure 1 presents GEMS human error taxonomy.

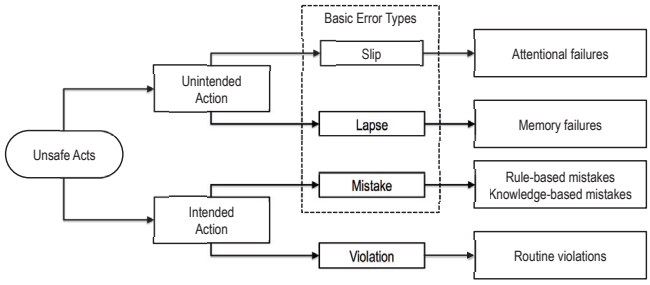

Fig. 1. Generic Error Modeling System (GEMS)

Previous investigations found that 61% of the human errors are skill-based, which means that humans are prone to slip and lapse with familiar tasks. When such tasks become harder, rule-based (28%) and knowledge-based (11%) become the most common failures made by humans. In the context of ITSM, human operators are well trained in specific areas of expertise and mostly of the work include routine tasks (*e.g.*, dispatching service requests, configure database server, restore password). In this paper, we focus on such skill-based errors, which can be tackled through proper instrumentation.

C. Quantitative Modeling for Human-Error Assessment

Diverse techniques to account for human errors have been developed in the last few decades, such as Technique for Human Error Rate Prediction (THERP) [14], Justification of Human Error Data Information (JHEDI), and Human Error Assessment and Reduction Technique (HEART) [15]. Although such techniques might appear to be outdated, enhanced methods were not developed in the field of Human Reliability Assessment (HRA) [16]. These techniques are still used in diverse high-risk professions such as nuclear power, healthcare, and aviation. Previous investigations [17] [18] [19] have shown that all these techniques present a very significant correlation between their predictions and the corresponding real measurements of human errors.

In this paper, we quantify human reliability using HEART, which is considered the most comprehensive method in the field of HRA. HEART method does not require the use of real human error measurements to make the predictions; instead, it specifies 9 Generic Task Types (GTTs) and provides the nominal Human Error Probability (HEP) for each one. HEART also specifies 38 Error Producing Conditions (EPCs), which are factors that can affect human performance making it less reliable (*e.g.*, distractions, overload). HEART methodology is based on the following four steps:

1) Identify all sub-tasks a system operator would be required to execute in order to complete an specific activity;
2) Identify the generic task type corresponding to the activity and its HEP;
3) Select the perceptible EPCs in the activity and that have a negative effect on the outcome;
4) Calculate the final HEP using the equation 1.

$$HEP[final] = HEP * \left\{ \prod_{i=1}^{n} \left[(EPC_i - 1)\, Ap_i + 1 \right] \right\} \quad (1)$$

In this equation 1, $HEP[final]$ is the final probability of human error for an specific activity, HEP is the nominal human unreliability accordingly to the task type, EPC_i is the ith error-promoting condition, and Ap_i is the proportion assessment factor (from 0 to 1) defined by the analyst. Because of space constraints, we suggest the interested reader to refer to the work of Williams [20] in order to observe HEP and EPC values in more details.

III. Human Errors in Service Management Processes

Continuous pressure on IT companies to increase their competitiveness forces human operators to reach or exceed their natural limits. Such constant effort, poorly designed systems, and lack of experience are examples why the human operators make errors during their daily activities [21]. In particular, we address in this paper error-prone activities, *i.e.*, portions of an IT Service Management (ITSM) process characterized by high risk of human error, that can be measured through instrumentation or observation, and can be improved through design and automation. Errors in ITSM processes can occur due to a wide variety of causes, for example:

- *Attention Span*: As the number and complexity of tasks increase, our ability to focus and maintain our attention decreases;
- *Memory*: Our short term (or working) memory is limited to the equivalent of retaining around seven distinct items at a time. This may not be sufficient for active processing of some complex tasks;
- *Situation Awareness*: A person's awareness (or not) and perception of different elements in their environment affect their information processing and actions;
- *Personal Resources*: the ability to process information appropriately diminishes as stress and fatigue increase.

In order to discover common error-prone activities in ITSM, we have analyzed tasks performed by a group of Subject Matter Experts (SMEs) in a very large IT Service Support and Delivery organization. Based on documents related to service incidents, it was possible to identify the non-exhaustive list of human errors presented below, presented in rough order from lowest to highest level. We illustrate these errors using two scenarios: a change management scenario in which a dispatcher assigns change requests to the appropriate technician, who is responsible for resolving them; and a continuity management scenario in which an analyst designs a disaster-recovery plan which will then be tested manually.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Downloaded on May 16,2026 at 01:17:39 UTC from IEEE Xplore. Restrictions apply.

- *Action Errors* are associated with actions performed by an operator on an object (*i.e.*, element of interest) and that change the state of the system. For example: the change request contains correct information, but the technician implements the change on the wrong server; the change request requires two technicians to operate in order (first reinstall the operating system, then install the data base) but the actions are performed out of order; the business recovery plan is complete but the tester fails to implement some part of the plan, causing the test to fail.
- *Retrieval Errors* are failures to retrieve correct information, either from memory or from a visual display, to be used in a further step. For example: the technician retrieves and acts upon a change request that had been assigned to a different technician; a business continuity analyst does not notice a list of required static IP addresses in the client configuration and so fails to include them in the business recovery plan.
- *Checking Errors* are errors that occur when the operator fails to check some information. It can be the combination of action and retrieval errors. For example: the technician marks the wrong change request as having been resolved.
- *Decision Errors* relate to the reasoning capabilities or training of the human operator and occur when the operator has to make an explicit choice between multiple alternatives. For example, a dispatcher assigns a change request to a technician who does not have the appropriate skill level for the task; an analyst suggests an unsuitable hardware substitution in a recovery plan.
- *Communication Errors* occur when the operator fails to pass information to another person, either in person or through a system (*e.g.*, instant messenger, e-mail). For example: a dispatcher fails to inform a technician of a change request that had been assigned to him or her;

Where it is feasible, companies can reduce human errors through complete automation of the process. However, even where this is not possible, it is very often the case that the process may be partially automated, using software to alert the operator to potential errors or provide support to reduce task complexity [22] [23]. The next section presents a methodology for locating error-prone components of ITSM processes and applying partial automation in the form of mashups to reduce the potential for human error.

IV. Methodology for Quality Assessment

We combine Six Sigma with various techniques to improve the performance of ITSM processes by eliminating defects and reducing variation. The HEART technique, for example, was extensively validated by experiment in other fields and represent one of the most widely used technique for Human Reliability Assessment (HRA) [17] [18] [19]. Event Tree analysis, in turn, is mainly used in the field of safety engineering and reliability engineering and can be used combined HEART to determine the probability of failure in a sequence of events. Finally, mashups represent an effective approach for eliminating root cause problems that degrade the quality in ITSM processes. Our methodology is based in the following stages:

1) The analyst works with a Subject Matter Expert (SME) to determine the process workflow and root causes of problems;
2) Using HEART technique the analyst determines the Human Error Probability (HEP) of each activity in the workflow;
3) Event Tree analysis is used to evaluate the performance of overall workflow;
4) The analyst identifies the most appropriate Mashup Patterns and interaction elements to solve problems;
5) The analyst evaluates the process once more to ensure improvements become embedded to the process.

Based on the obtained process workflow (stage 1) and on the individual HEP values for each activity (stage 2), the analyst builds an Event Tree (stage 3) to identify and evaluate the sequence of events in a failure scenario. Once the most error-prone activities are identified, the analyst selects the most appropriate mashup patterns to improve the process (stage 4). Finally, the analyst run through the methodology again (stage 5), from stage 2 to stage 4, in order to ensure the improvements become embedded, and to detect new improvement areas in the process.

Besides its simplicity and easy of application, the HEART technique presents several problems. For example, the Error Producing Conditions (EPCs) are not independent of each other; moreover, the use of the method is extremely subjective and relies heavily on the experience of the analyst; and finally, from a qualitative point of view, the EPCs are a useful list of factors to guide quality analysts, but the numerical values (*i.e.*, Assessed Proportion of Affect) are subjective and imprecise. In order to take into consideration such weaknesses, we have analyzed the values of 2024 EPCs chosen by Subject Matter Experts (SMEs) in 8 common ITSM activities.

TABLE I. Assessed Proportion of Affect

Class	Average	σ
Low	0.1833	0.086157
Medium	0.5032	0.106578
High	0.8316	0.077288

A clustering algorithm, called k-means, was employed to group these EPCs' values in clusters. k-means is a partitioning technique that groups a dataset into k clusters. The criterion used to assess the number of clusters is the average silhouette of the data. Three significant clusters – *i.e.*, low (L), medium (M), and high (H) – have been identified. For each cluster we have measured the EPC value average, which will be used as input to expression 1. In Table I, we show these averaged results and the respective standard deviation σ. We employed "linguistic variable" to represent each cluster because it is very useful when situations are too complex of ill-defined to reasonably described in conventional quantitative expressions.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Downloaded on May 16,2026 at 01:17:39 UTC from IEEE Xplore. Restrictions apply.

V. Copying Human-Errors in ITSM Processes Through Mashups

As previously discussed, although required, the existence of human operators in ITSM processes can introduce a significant variability in the final process outcome. These operators execute multiple and complex tasks, usually employing different systems, and them becoming susceptible to making errors. In our previous work [9], we have introduced a set of mashup patterns for eliminating inefficiencies in ITSM processes, thus improving the human operator's productivity. Such inefficiencies are caused by insufficient design of the process itself, or defects in the tools being used. We argue that these mashups patterns can be used to cope with human errors too. In special interest of this paper, we analyse involuntary actions (*i.e.*, slips and lapses) performed by human operators and that can be tackled through instrumentation and redesign. The list bellow summarizes the mashup patterns we have introduced before.

- **Alerter pattern**: periodically monitors a system of interest on behalf of the user and, based on previously established conditions, sends notifications only when events of interest take place;
- **Importer pattern**: abstracts the different methods used to access the external data so that data consistency maintenance becomes transparent to the user;
- **Transformer pattern**: enable the processing of certain types of data and thus both materializing the compatibility between systems and satisfying the requirements of the IT process;
- **Displayer pattern**: enforces the concept of a "single pane of glass", allowing the combination of data from multiple sources and present the results of this combination in a Web page.

In order to prevent human errors from occurring, or at least to reduce their frequency, we have created a new type of interaction elements, called **Error Prevention** modules, to be used by mashup developers. These interaction elements are used to hide technical details from the mashup developer [24]. We define two types of this **Error Prevention** module: *Buffer* module and *Forcing* module. The *Buffer* module allows developers to specify for how long time a specific action should be delayed before being executed, thus providing a recovery window during which an erroneous action can be cancelled. According to Reason [12], roughly 78% of human errors can be detected immediately after they are executed. The *Forcing* module, on the other hand, allows developers to introduce confirmation points in the process workflow in order to force the human operator to consciously accept them before proceeding with the mashup execution.

In Figure 2, we provide an example illustrating how the **Error Prevention** modules are used in the creation of a mashup application. In Step (1) two adaptation modules are used to retrieve information from external data sources, which is merged In Step (2). The result of this composition is sent to a third system using another adaptation module (5). *Buffer* and *Forcing* modules are used (Step (3) and Step (4)) to first, force the human operator to confirm if the mashup should continue its execution, and second to wait for an amount of time before sending the merged data to the external system.

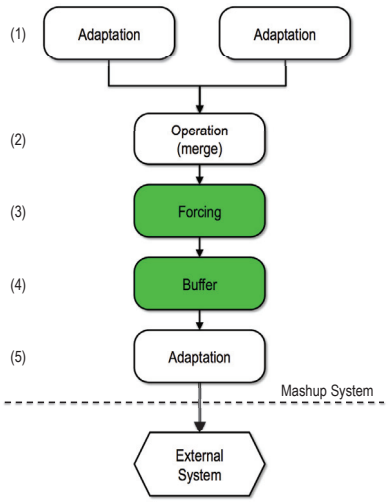

Fig. 2. Usage Example of Error Prevention Modules

VI. Case Study and Evaluation

We concentrate on the Request Fulfillment process, one of the operational processes in IT Management, responsible for carrying out service requests, and that interfaces with Service Desk, Incident Management, and Change Management. ITIL v3 describes it as "management of customer or user requests that are not generated as an incident from an unexpected service delay or disruption" [25]. This process includes two main activities: support the requests (*i.e.*, tickets) made by the customers, and solve those requests. The first activity is performed by system administrators (SAs) responsible for solving specific requests; the second activity is executed by dispatchers, who are human operators responsible for monitoring new requests, dispatching requests to the right SAs, and monitoring compliance with Service Level Agreements (SLAs).

A. Dispatch Activity

In this paper we focus our analysis on dispatch in Request Fulfillment, an activity centered on humans with knowledge of standard fulfillment procedures. Once the customer creates a new request (*i.e.*, incident, problem or change request) in Service Desk, a ticket is routed to a dispatcher who is responsible for analyzing the ticket and determining the appropriate SA to solve it. Usually, each dispatcher is responsible for teams of system administrators, each with a specialized technical background. Once the dispatcher receives the ticket, he analyses if it was misrouted or not. To accomplish this task, dispatchers determine if his team is able to resolve it (using his knowledge of his team's schedule, workload and expertise). If his team is not able to solve the ticket, he forwards it to another dispatcher who is responsible for a different team of SAs.

Although fully automated solutions offer many advantages, they may not be feasible for the dispatch activity. Dispatch is

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Downloaded on May 16,2026 at 01:17:39 UTC from IEEE Xplore. Restrictions apply.

constantly changing and is too complex to be automated. For example, in some cases a dispatcher may want to train a new administrator, and so may intentionally assign a request to a less skilled SA than is available. Although required, the use of human dispatchers in service delivery centers may introduce defects in the process, thus reflecting in severe penalties from Service Level Objective (SLO) violations. In this context, mashups are a new technology that simplifies and makes more predictable the creation and execution of dispatching systems, focusing on the activities performed by each dispatcher, thus decreasing the possibility of errors.

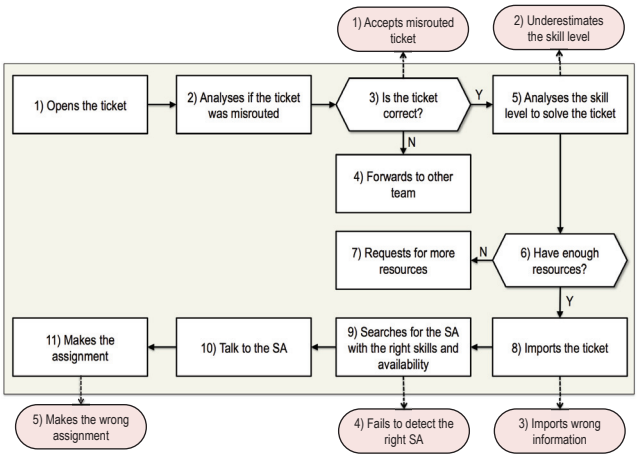

Fig. 3. Workflow of activities performed by dispatchers in Service Desk. Ovals represent points of failure.

In order to detect problems and perform a root cause analysis in dispatch process, we have shadowed dispatchers in their daily activities. Based on interviews, was possible to determine the workflow of activities presented in Figure 3. Once a new request is received from a customer, the dispatcher determines if his team has the proper expertise to solve it, *i.e.*, he analyses if the ticket was misrouted. If the ticket was not misrouted, the dispatcher analyses the required skill level to solve it. Next, the dispatcher imports data from the customer's ticketing system. This activity is usually performed manually because data usually need to undergo some processing (*e.g.*, anonymising confidential information). Once the ticket is imported, the dispatcher uses SA availability, workload and skills to determine the proper SA to solve it.

In order to accomplish his tasks, dispatchers interact with multiple tools (*e.g.*, multiple ticketing systems, calendar-based systems, spreadsheets). Furthermore, dispatchers may also be overloaded with customer requests, which can vary from just a few dozen to hundreds in a busy day. Due this complex nature and continuous pressure, dispatchers are vulnerable to making errors. Based on the interviews and on documents related to service incidents, it was possible to identify the most common failures in dispatch, presented in Figure 3. These failures may occur due the combination of the error types we have depicted in section III. The dispatcher may, for example, accept a misrouted ticket. This can happen due to a decision error, when he fails to decide if the ticket was misrouted; or due an action error, when he mistakenly clicks on the accept button instead of rejecting it. The skill level to solve the ticket can also be underestimated, characterized as a decision error. Retrieval errors may also be introduced during the importing step. Another example of a decision error occurs when the dispatcher fails to detect the right SA to solve the ticket. Finally, even if the dispatcher knows the proper SA, he may fail to make the assignment because selects the wrong SA in the ticketing system (*i.e.*, an action error).

B. Baseline Performance

In order to discover the human error probability (*i.e.*, $HEP[final]$) of each task performed by dispatchers, we have classified them as type G. According to HEART, the generic task type G involves "completely familiar, well designed, highly practised, routine task occurring several times per hour, performed at the highest possible standards by highly motivated, highly trained and experienced person, totally aware of the implications of failure, with time to correct potential error, but without the benefit of significant job aids". This generic task type has a nominal human error probability of 0.0004. For each task, we also have selected the EPCs that present a negative effect on the outcome and them calculated the $HEP[final]$ according to equation 1. We illustrate in Table II the HEART calculations for task 3 (*i.e.*, import wrong information about the ticket).

TABLE II. HEART CALCULATIONS FOR TASK 3

EPC	HEART effect	Assessed proportion of effect	Assessed effect
Suppressing information	x 9	High	7.48
Channel overload	x 6	High	4.99
Little checking	x 3	Medium	1.51

Based on these values, the total assessed EPC effect is $(7.48 * 4.99 * 1.51) = 56.36$ and the assessed human error probability is $(0.0004) * 56.36 = 0.022$. Therefore the probability of occurring a human error in Task 3 is 2.2%.

TABLE III. ORIGINAL FAILURE PROBABILITIES

i	Task	$HEP[final_i]$
1	3	0.002391857
2	5	0.019094864
3	8	0.022549899
4	9	0.019094864
5	11	0.002391857

By following the HEART methodology for the remaining tasks, we were able to get the failure probabilities presented in Table III. We do not present tasks 4 and 7 because in our scenario, the tickets were not routed to the wrong dispatcher and we also determined that he always had enough resources (*i.e.*, System Administrators) to solve them. Task 10 is not shown because it was not observed during our evaluations. With these values it was possible to employ the Fault Tree Analysis (FTA) method in order to estimate the overall workflow failure rate. This failure rate was obtained using the equation 2, presented bellow.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Downloaded on May 16,2026 at 01:17:39 UTC from IEEE Xplore. Restrictions apply.

$$P[failure] = 1 - \prod_{i=1}^{n}(1 - HEP[final_i]) \qquad (2)$$

Let $P[failure]$ be the probability of failure in a sequence of events and $HEP[final_i]$ be the final probability of human error for a specific task, which is obtained from equation 1. By taking the equation 2, we obtain $P[failure] = 0.064015658$, which means that there is a chance of 6.5% for the dispatcher to make at least one error during the execution of the dispatch process.

C. Applying Mashup Patterns and Error Prevention Modules to Dispatch

All the mashups patterns proposed in our previous work and the error prevention modules introduced in this paper can be used to create a mashup-based solution for the above-mentioned dispatching scenario. By using these patterns and modules, we aim to eliminate, or at least reduce, the probability of dispatchers making errors during their daily activities. Figure 4 shows how the mashup patterns and proposed error prevention modules relate with each specific task in the dispatching scenario.

The displayer pattern is used to show in a single screen all the useful information for the dispatcher (*e.g.*, workload, skills, ticket description). This pattern aids to reduce the occurrence of retrieval errors, either from memory or from a visual display. The combined use of both importer and transformer patterns is useful to avoid action and retrieval errors. Even with the introduction of these patterns, the dispatcher may still make some error. In order to reduce this probability even more, the two error prevention modules introduced in this paper are used after the assignment task (11). If the dispatcher mistakenly selects the wrong SA to assign the ticket, the system will require a confirmation to execute the assignment task. Even if the dispatcher confirms that, he still has a few seconds to cancel the operation before it is executed.

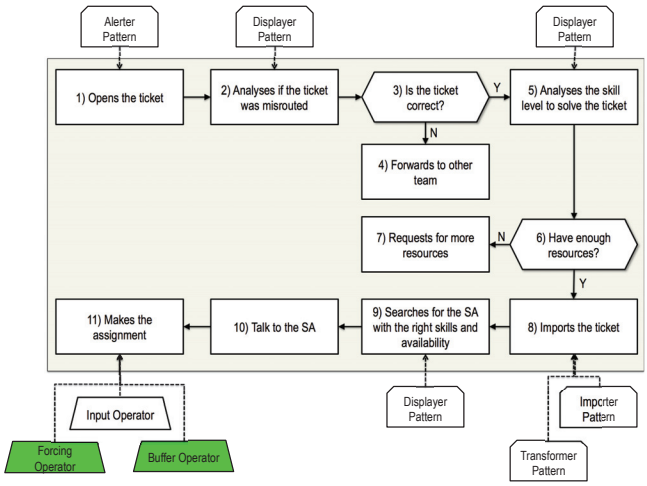

Fig. 4. Relating mashup patterns and operators with dispatching tasks

D. Predictions on Mashups Usage

Analyzing the tasks of the dispatching scenario and using the methodology described in section IV we estimated the human error probabilities presented in Table IV. Each line of this table represents a specific task of the dispatching process. Task 8 was completely automated by employing mashup patterns, thus the human error probability is 0%.

TABLE IV. FAILURE

i	Task	$HEP[final_i]$
1	3	0.00091324
2	5	0.00445674
3	8	0.00000000
4	9	0.00476030
5	11	0.00091324

From equation 2, we obtain $P[failure] = 0.011004691$, which means that the use of mashups can reduce to 1.1% the chance of dispatchers to make at least one error during the execution of the dispatch process. The obtained result show the significant reduction of 83.07% in the probability of human error, which mostly was due to the automation of step 8 by using the importer pattern. The usage of a displayer pattern contributed to reducing the chance of failure in tasks 5 and 9. In both tasks, the dispatcher no longer needs to look up for any information in a different system, thus eliminating all the possible retrieval errors.

VII. Conclusions and Future Work

In this paper, we extended our previous research on applying mashups to improve the performance of ITSM processes by eliminating defects and reducing variability. We have introduced a methodology inspired on Six Sigma and based on partial automation and process redesign to tackle error-prone activities in human-centered ITSM processes. HEART and Event Tree Analysis were combined to evaluate and predict the impact of employing mashups in ITSM processes. We have focused our analysis on dispatch in Request Fulfillment, an activity centered on humans with knowledge of standard fulfillment procedures. Based on interviews with a group of SMEs in a very large IT Service Support and Delivery organization was possible to determine the workflow of activities performed and the most common failures in dispatch.

The new interaction elements, *Buffer* and *Forcing*, in conjunction with a set of mashup patterns enable mashup developers to create new applications that ultimately prevent human errors from occurring, or at least reduce their frequency. Two ways to prevent human errors were observed: avoidance and interception. Error avoidance (*i.e.*, keeping people from making errors) was possible through the use of mashup patterns and the *Forcing* module. Error interception (*i.e.*, stopping the errors from reaching the system) was possible through the *Buffer* module. Altough was not observed the reduction of Knowledge-based errors (*i.e.*, decision errors), the presented patterns and interaction elements avoided the occurrence of Rule-based and Skill-based errors (*i.e.*, action and retrieval errors), which represent 89% of human errors according to previous investigations. We also demonstrated the significant

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Downloaded on May 16,2026 at 01:17:39 UTC from IEEE Xplore. Restrictions apply.

reduction of 83.07% in the probability of dispatchers making errors during their daily activities.

As future research, we aim to expand our exploration of performance in business processes by evaluating the correlation of productivity and quality. We also plan to extend the analysis of the ITSM process beyond productivity and quality, considering other aspects such as effectiveness. Finally, we will explore the influence of these three aspects in ITSM process costs.

REFERENCES

[1] A. Keller, J. L. Hellerstein, J. L. Wolf, K. L. Wu, and V. Krishnan, in *Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP*, vol. 1, 2004, pp. 395–408.

[2] A. B. Brown and J. L. Hellerstein, "Reducing the cost of it operations: is automation always the answer?" in *Proceedings of the 10th conference on Hot Topics in Operating Systems - Volume 10*, ser. HOTOS'05. Berkeley, CA, USA: USENIX Association, 2005, pp. 12–12.

[3] T. Benson, S. Sahu, A. Akella, and A. Shaikh, "A first look at problems in the cloud," in *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, ser. HotCloud'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 15–15.

[4] J. Gray, "Why do computers stop and what can be done about it?" in *Symposium on Reliability in Distributed Software and Database Systems*, 1986, pp. 3–12.

[5] D. Oppenheimer, A. Ganapathi, and D. A. Patterson, "Why do internet services fail, and what can be done about it?" in *Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems - Volume 4*, ser. USITS'03. Berkeley, CA, USA: USENIX Association, 2003, pp. 1–1.

[6] P. Pande, R. Neuman, and R. Cavanagh, *The Six Sigma Way: How GE, Motorola, and Other Top Companies are Honing Their Performance*. McGraw-Hill, 2000.

[7] J. Krafcik, "Triumph of the lean production system," *Sloan Management Review*, vol. 30, no. 1, pp. 44–52, 1988.

[8] T. T. Allen, *Introduction to engineering statistics and lean sigma: statistical quality control and design of experiments and systems*. London: Springer, 2010.

[9] C. dos Santos, L. Granville, W. Cheng, D. Loewenstern, L. Shwartz, and N. Anerousis, "Performance management and quantitative modeling of it service processes using mashup patterns," in *Network and Service Management (CNSM), 2011 7th International Conference on*, oct. 2011.

[10] D. Merril, "Mashups: The new breed of web app - an introduction to mashups." 2003, http://www.ibm.com/developerworks/web/library/x-mashups.html.

[11] A. Jhingran, "Enterprise information mashups: integrating information, simply," in *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 2006, pp. 3–4.

[12] J. Reason, *Human Error*. Cambridge [England] ; New York : Cambridge University Press, 1990. xv, 302 p., 1990.

[13] J. Rasmussen, "Skills, rules, and knowledge: signals, signs, and symbols, and other distinctions in human performance models," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 13, no. 3, pp. 257–266, 1983.

[14] A. Swain and H. Gutterman, "Handbook of human reliability analysis with emphasis on nuclear power plant applications," Sandia Laboratories, Albuquerque, New Mexico, Tech. Rep. NUREG/CR-1278, 1980.

[15] J. Williams, "Heart - a proposed method for achieving high reliability," in *Symposium on the Achievement of Reliability in Operating Plant, Safety and Reliability Society*, 1985, pp. 87–109.

[16] R. Whittingham, *The Blame Machine: Why Human Error Causes Accidents*. Taylor & Francis, 2012.

[17] B. Kirwan, "The validation of three human reliability quantification techniques therp, heart and jhedi: Part 1 -ttechnique descriptions and validation issues," *Applied Ergonomics*, vol. 27, no. 6, pp. 359 – 373, 1996.

[18] B. Kirwan, R. Kennedy, S. Taylor-Adams, and B. Lambert, "The validation of three human reliability quantification techniques therp, heart and jhedi: Part 2 - results of validation exercise," *Applied Ergonomics*, vol. 28, no. 1, pp. 17 – 25, 1997.

[19] B. Kirwan, "The validation of three human reliability quantification techniques therp, heart and jhedi: Part 3 - practical aspects of the usage of the techniques," *Applied Ergonomics*, vol. 28, no. 1, pp. 27 – 39, 1997.

[20] J. Williams, "Validation of human reliability assessment techniques," *Reliability Engineering*, vol. 11, no. 3, pp. 149 – 162, 1985.

[21] A. B. Brown, "Oops! coping with human error in it systems," *Queue*, vol. 2, no. 8, pp. 34–41, Nov. 2004.

[22] V. R. Madduri, M. Gupta, P. De, and V. Anand, "Towards mitigating human errors in it change management process," in *ICSOC*, 2010, pp. 657–662.

[23] L. Shwartz, D. Rosu, D. Loewenstern, M. J. Buco, S. Guo, R. Lavrado, M. Gupta, P. De, V. R. Madduri, and J. K. Singh, "Quality of it service delivery - analysis and framework for human error prevention," in *SOCA*, 2010, pp. 1–8.

[24] C. dos Santos, R. Bezerra, J. Ceron, L. Granville, and L. Rockenbach Tarouco, "On using mashups for composing network management applications," *Communications Magazine, IEEE*, vol. 48, no. 12, pp. 112 –122, december 2010.

[25] OGC, "Information technology infrastructure library v3 (itil v3)," Office of Government Commerce, May 2008. [Online]. Available: http://www.itil-officialsite.com/

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Downloaded on May 16,2026 at 01:17:39 UTC from IEEE Xplore. Restrictions apply.