Automatic Resource Assignment for Issue Resolution

Steven James Ross
Christopher Farrar

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation
https://www.tdcommons.org/dpubs_series/1479

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.
Automatic resource assignment for issue resolution

ABSTRACT

Issue tracking systems are widely issued for managing the reporting and addressing of various issues in a software system. In many cases, determining the specific software components connected to the issue and the appropriate persons to resolve the issue is not straightforward and requires a series of assignments until the right components and persons are assigned to understand and resolve the issue. The techniques of this disclosure employ a machine learning model trained on existing labeled data from an issue tracking system to automate the process of assigning appropriate components to issues and routing them to personnel most suitable for handling them. Additionally, the model allocates a priority for each issue and reroutes issues in case the initial allocation fails to resolve the issue within reasonable time.

KEYWORDS

- Issue tracking
- Bug tracking
- Resource allocation
- Automated issue routing
- Bug reports

BACKGROUND

In software development, issue tracking systems are widely issued for managing the reporting and addressing of various issues, such as bugs, desired features, customer-reported problems, etc. In many cases, determining the specific software component(s) connected to the issue and the appropriate person(s) to resolve the issue is not straightforward and may require a series of assignments until the right components and persons are assigned. Such manual issue
routing flow that often requires multiple tries prior to reaching the correct assignment requires substantial time and effort on the part of the person filing the issue and the persons involved in examining and routing, thus wasting resources.

DESCRIPTION

The techniques of this disclosure operate within any system that tracks issues, such as a bug tracking system for software (and/or hardware). The system accepts manual or automated issue reports that include various details pertinent to the issue, such as a text description of the problem with relevant links, logs, and other automatically generated data associated with the issue along with corresponding timestamps and system information. The system may solicit suggestions, from the reporting entity, regarding possible components that are likely to be connected to the issue and individuals that are likely to be suited to resolve the issue and may provide a list of possibilities for these suggestions the user can choose from, based on already-entered information. The system further allows the addition of manual or automated comments or changes after an issue has been filed.

Once an issue is filed, the report is passed to a machine learning model trained on existing extensive labeled data within the issue tracking system. Such data includes information such as filing time, resolution time, routing, components affected, persons involved, etc. Data is used for training with specific consent for such use, e.g., from individuals or entities that submitted the information. Based on the training data, the training process involves taking into account various factors, such as the final components selected when an issue was resolved, the amount of time taken to resolve the issue, the persons who contributed to resolving the issue, the final assignees when the issues was resolved, etc. The training can also additionally consider the features associated with presently unresolved issues.
Upon receipt of a new issue report, the trained machine learning model analyzes the report to detect whether the associated feature is likely working as intended. In such cases, the entity that reported the issue is directed to appropriate documentation that might help resolve the matter. Next, the model considers whether the issue is a likely duplicate of a previously reported issue. If the issue is considered a duplicate with a level of confidence higher than a specified threshold value, it is automatically marked as a duplicate. If the level of confidence is lower than the threshold, the reporting entity is presented with a list of potential duplicates from previous reported issues and offered the ability to indicate whether the currently reported issue matches any in the list.

The model analyzes any issues not filtered as duplicates to determine the appropriate components connected to the issue and, subsequently, the person or entity tasked with resolving the issue. When determining the most appropriate assignee, the model additionally takes into account various permitted contextual factors, such as on-call status for the components in question, vacations and leaves, past experience resolving similar issues, current time of day and working hours, schedules and meeting, current assignment workload, etc. Information on these factors is obtained, with prior user consent, from various appropriate enterprise systems, such as calendars for schedules, human resource management for leave and vacation information, etc.

The model may additionally assign the reported issue an indication of priority or severity. For instance, an issue that involves a single machine (e.g., a server computer) is marked as low priority while one that might affect large numbers of machines, e.g., millions of machines, is considered critical. After the initial assignment of components, persons, and priority is made, the model continues to track the progress of the issue. Issues that remain unresolved beyond a specified amount of time are examined again to determine if any changes need to be made to the
initially assigned components, persons, or priority, additionally taking into account any actions taken after the initial assignment.

Fig. 1: Issue report processing for automated component and personnel assignment

Fig. 1 shows the flow of operation of the trained machine learning model (102) in processing an issue report (100) received as input. First, it is determined if the underlying feature works as intended. If so, the reporting entity is directed to pertinent documentation (104). If the determination is that the underlying feature doesn’t work as intended, it is determined whether the issue is a likely duplicate of an existing issue. In case the issue is considered a duplicate with
confidence higher than a specified threshold value, it is automatically marked as duplicate in the issue tracking system (106).

For cases in which the confidence does not meet the specified threshold, the reporting entity is presented with a list of likely duplicates (108) and allowed to choose whether the current issue matches any in the list. Any issue not filtered as unproblematic or duplicate proceeds down the pipeline to be assigned appropriate components, persons, and priorities determined by the trained machine learning model. When assigning personnel, the model considers various contextual factors, such as schedules, workloads, past experience, etc., obtained from appropriate enterprise systems (110). After the initial assignment the issue is examined periodically to check whether it has been resolved. In case the issue remains unresolved beyond a specific period, the assignment process is repeated to determine if any changes are necessary to the originally assigned components, personnel, or priority.

The proposed system can be further enhanced with additional features for issue management. For instance, the system can be set up to page or otherwise appropriate individuals when critical issues are reported. The system can also be augmented by incorporating various appropriate constraints, such as limits on the number of issues assigned to a person, control on the depth of the issue queue, etc. In addition to component assignment, the model training may take into account code changes and logs needed to resolve issues, and suggest files, related (including closed) bugs, or code snippets likely to correspond to the reported issue during the operational phase. If the model detects inadequate or potentially inaccurate information in the issue report at any point during the operational phase, the reporting entity is prompted for clarification or additional information.
The threshold value of confidence for automatic flagging of duplicate issues and the time value of the period after which existing issues are checked for resolution can be specified in advance, e.g., as a setting of the issue tracking system, by the reporting entity, or the developers. Alternatively, the values may be determined dynamically based on the data. Further, the appropriate value may vary based on the issue, the system, the components, the personnel, the priority, etc.

With little to no modification, the techniques described can also be applied to domains other than software, such as hardware systems or customer support. The automated issue assignment and routing enabled by the described techniques have the potential to save substantial human time and effort, speed up issue resolution, and improve product quality and reliability.

CONCLUSION

The techniques of this disclosure employ a machine learning model trained on existing labeled data in an issue tracking system to automate the process of assigning appropriate components to issues and routing the issues to the most suited personnel. Additionally, the model allocates a priority for each issue and reroutes issues in case the initial allocation fails to resolve the issue within reasonable time. The automated issue assignment and routing provided by the described techniques have the potential to save substantial human time and effort, speed up issue resolution, and improve product quality and reliability.