2017 IEEE 24th International Conference on Web Services

SaaS for Automated Job Performance Appraisals using Service Technologies and Big Data Analytics

I-Ling Yen, Farokh Bastani, Yongtao Huang
Computer Science Department
University of Texas at Dallas
{ilyen, bastani, hxy131530}@utdallas.edu

Yuqun Zhang, Xin Yao
Department of Computer Science and Engineering
Southern University of Science and Technology
{zhangyq, xiny}@sustc.edu.cn

Abstract. In this paper, we present a new SaaS (software as a service) design for employee job performance appraisals, SaaS-JPA. We use IoT and computer systems to collect data related to the daily works of employees. A semantic model is developed to guide the data collection process, facilitate data interpretation and interoperation, and enable big data analysis to make job performance appraisal decisions. We also propose two new performance assessment models: The similarity-based relative performance model and the revenue-based performance model. These performance models are enabled by the service technologies and big data analytics. Finally, we discuss the design of SaaS-JPA.

Keywords: Job performance appraisal, semantic model for performance appraisal, similarity-based relative performance, revenue-based performance, software as a service (SaaS).

1 Introduction

Job performance appraisal (JPA) is the process for evaluating the effectiveness of the employees at their jobs and it is a critical step in every organization of any type of business and of any size. The quality of many human resources related decisions, such as promotions, merit raises, terminations, compensation, training and mentoring needs, improvement in organizational supports, etc., depends heavily on the performance appraisals. In modern society, many jobs require specific expertise and in many organizations and enterprises, the human resources are considered valuable assets. Correspondingly, the role of performance appraisals has growing importance.

Though job performance appraisal is highly critical, it has long been viewed as a difficult and distasteful task, disliked by both managers and employees. There are many different methodologies to guide the job performance appraisal process [1]. These methods consider the aspects regarding how to select the evaluation criteria, who should perform the evaluation, and how to represent the assessment outcomes. In recent years, the 360 degree appraisal and 720 degree appraisal methods [2,3] have become the most popular performance appraisal framework. There are also many software systems for managing the performance appraisal processes and ease the appraisal tasks, including IBM Kenexa, SuccessFactors, Oracle Taleo, BambooHR, Cornerstone Performance, Reviewsanp, Trakstar, Halogen, etc. These systems provide tools to support the workflows for job performance appraisals. They generally allow the company to tailor the review forms and templates, let managers schedule employee reviews and record the outcomes and feedbacks, help managers and employees set the job objectives and help document the progress of the employees towards the objectives, etc. Most of these systems also support automated appraisal workflow execution, including routing the related documents and emailing the appraisal activities according to the schedules defined in the workflows. However, the major functionalities of these software systems are simply to provide the interfaces for easy appraisal workflow and template definitions, support appraisal document management and retrieval, and automate appraisal workflow execution. In other words, these systems can only help with the mundane chores in the appraisal process, while the challenging tasks, such as assessment criteria determination and evaluation decision making, are still to be done manually.

In fact, current computing technologies can be applied to offer additional help for job performance appraisals.

(1) Nowadays, a lot of office workflows are through computer systems and records are available in company databases. Employees are used to interact via emails and other online means. For office workers, these data can be mined to obtain useful data regarding tasks performed by them and potential feedbacks and reviews for these tasks. Even phone conversations and daily interactions outside the computing world may potentially be captured by IoT devices and analyzed to obtain work related data. Text and other mining techniques can be used to collect the information.

(2) For non-office works, in most of the situations, they leave records in computer systems or mobile devices. For example, in a car repair store, this includes the nature of each repair, the time it took, etc. In a moving company, all the records about loading, driving, unloading, etc., and the time these actions take are documented. In a hospital, the nurses may have to give shots, measure blood pressures, change IVs, schedule medicine intakes, etc., and these works also have records in the computer systems or mobile devices. In fact, these data are already collected and used for work related assessment. With additional software mining, more comprehensive information can be collected for performance appraisals.

(3) For various design works, such as hardware, software, mechanical, architectural, and so on, the works are mostly done on the computer systems. Also, management systems are generally used to allow the designers to check in/out of

978-1-5386-0752-7/17 $31.00 © 2017 IEEE
DOI 10.1109/ICWS.2017.47

the partial works to support collaboration. Hence, even if the employees use different computers to perform the work, the periodical check in/out records can still be useful and can be mined to infer the workplace performance. Some metrics should be designed to assess such professional activities from the incremental work records.

(4) Big data analytics can be applied to infer performance metrics from works that are similar in full or in parts. This can be considered for all the cases above.

Since privacy may be a concern, information collection should be under the knowledge of the employees and tools should be designed and provided to the employees to allow them to effectively filter out the information that they are not willing to share.

In this paper, we consider the design of a Software as a Service (SaaS) system, SaaS-JPA, to assist with the job performance appraisals. The assistance technologies to be considered are those that may help with the appraisal decision making process, not just information management. Our contributions in this work include the following:

(a) We analyze the information required for assisting with appraisal decision making as discussed in the four points above and build sematic models to capture the information. The semantic models can also facilitate the appraisal decision reasoning services and enable the big data analysis services for appraisals.

(b) We develop the mechanism of using mining techniques to discover the tasks performed by an employee and obtain data regarding how well the task is performed. We also propose to apply big data analysis to derive the relative performance of an employee, including relative performance based on similar tasks and relative performance based on similar jobs.

(c) Employee performance can also be judged based on her/his contribution to the company's revenue/profit. We develop a revenue-based performance appraisal approach. In this approach, the major company services and their workflows and corresponding profits are identified. A scheme to distribute the profit from an overall service to individual tasks comprising the workflow of the service is proposed. From the per task profit, we can derive the employee's total contribution to the company's profit, and then the revenue-based performance for the employee can be derived. The final performance appraisal should be the integration of the similarity-based relative performance, the revenue-based performance, and the performance assessed using existing methods.

In the next section, we review the literature in performance appraisal. Section 3 defines the semantic models that are useful for the job performance appraisal process. In Section 4, we propose automated evaluation methods based on the semantic models. The overall design of the SaaS-JPA, which incorporates the semantic models, the automated evaluation methods, as well as repository management components, is discussed in Section 5. Section 6 states the conclusion of this paper and identifies some future research directions.

2 Literature Review

There are many different job performance appraisal methods proposed and used in the history [1]. Generally, a certain list of "traits" are identified and evaluation is done accordingly. For each trait, various evaluation methods have been considered traditionally. Some use the rating scale which defines a numerical range for the evaluator to rate the employee. Since the meaning of the numerical rating scales may be unclear to the evaluators, linguistic scales can be defined to allow evaluators to give ratings more intuitively [4]. Instead of rating individuals independently, the ranking method lets the evaluator rank a specific group of employees for each specific trait. Similar to ranking, the paired comparison method enables the evaluator to compare each pair of employees in a group and the final evaluation is the summary of such comparisons. This approach can incur high overhead due to the large number of comparisons required. In the essay method, the evaluator describes her/his impression of the employee's performance, strengths and weaknesses in terms of each trait. In this method, each trait would be more coarse-grained. Though this method can be more informative, it also has the potential of creating confusion due to fuzzy wording and requires additional methods to summarize the essays to derive a decision.

Besides the methods that are based on the evaluators' impressions, there are also some based on factual events or specifically specified behaviors. In the critical incident method [5], the performance of the employee in terms of a certain trait is evaluated based on certain critical behaviors that are implicative for the trait. For example, this technique was recommended as one criteria for evaluating pilots based on the number of errors incurred in reading and interpreting aircraft instruments. In the behaviorally anchored rating scale (BARS) [6], some concrete examples to illustratively define various performance levels are given. The evaluator serves as an observer and rates the employee by matching her/his actual behaviors against the examples of the corresponding ratings. These methods can be more subjective and avoid potential biases. However, it may be difficult to define the factual events or example behaviors for some evaluation traits.

In most traditional performance appraisal methods, the supervisors are the major evaluators. In the 360 degree appraisal (or integral evaluation) [2], evaluators are selected from different categories, including supervisors, co-workers, and subordinators of the employee. It is also recommended to include individuals external to the company who have interactions with the employee, such as the customers, the suppliers, and so on. This integral view can help improve the reliability of the evaluation by reducing the potential of bias and can take into account the effect of the employee's general behaviors on her/his colleagues and other individuals with whom she/he interacts. The 720 degree appraisal is double of the 360 degree appraisal because it performs the appraisal at least twice [3]. The 360 degree appraisal framework does not contain the concept of feedbacks and improvements. In the

720 degree appraisal method, the supervisor should discuss with the employee about her/his performance, feedbacks, objectives, suggested improvements to accomplish the objectives, etc. After a period of time, the supervisor meets the employee again to evaluate her/his improvements and/or progress toward the objectives set in the previous meeting. Both 360 degree and 720 degree appraisals are popular methods in modern organizations and enterprises.

Major steps in the performance appraisal processes are generally performed manually, such as the assessment decision process by the evaluators and the integration of various assessment outcomes to the final result. Software tools are available to assist with the bookkeeping and managerial tasks in the appraisal process. The only works that apply computing technologies in performance appraisals are the use of fuzzy logic. Generally, performance appraisal decisions are fuzzy memberships. Correspondingly, fuzzy logic techniques have been used to integrate multiple fuzzy evaluations to a single one. In [7], a defuzzification method is proposed to hierarchically integrate the fuzzy evaluation decisions for many traits into one appraisal result. In [8], a fuzzy model with multiple granularities is proposed. It allows the evaluators to use different number of scoring levels and different linguistic descriptors to provide evaluation decisions. Fuzzy logic techniques are used to integrate these evaluation results into a single appraisal decision.

3 Semantic Modeling

As the first step toward job performance appraisals, we analyze the basis for appraisals and develop the semantic models that facilitate the appraisal tasks. First, we consider the basic unit for appraisal, the employee, and build the employee model as shown in Figure 1.

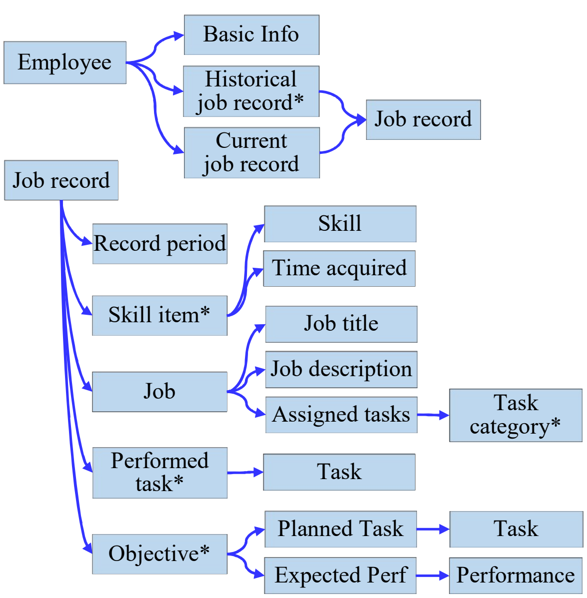

Figure 1. Employee model.

An employee has basic information such as her/his name, employee ID, date of birth, salary, education, and experience. The job related information for the employee is captured in the job records, including historical and current ones. Each job record maintains the skills the employee has at the time, the job she/he holds, and tasks performed in this period (or so far). For job specification, job title gives the position of the employee, the division chain (e.g. engineering college, computer science department), and the company. Each job has some expected tasks to be performed and these are not the concrete tasks, but task categories. For example, a secretary may be expected to perform the travel arrangement task category, and she/he may have performed the concrete travel arrangement tasks many times. Also, the actual tasks performed by an employee may not be in the assigned categories of her/his job. The Objective class captures the expectations set for the job for the specific employee after a certain performance review. Each objective is set for a planned goal (expressed as a task category) and the expected performance.

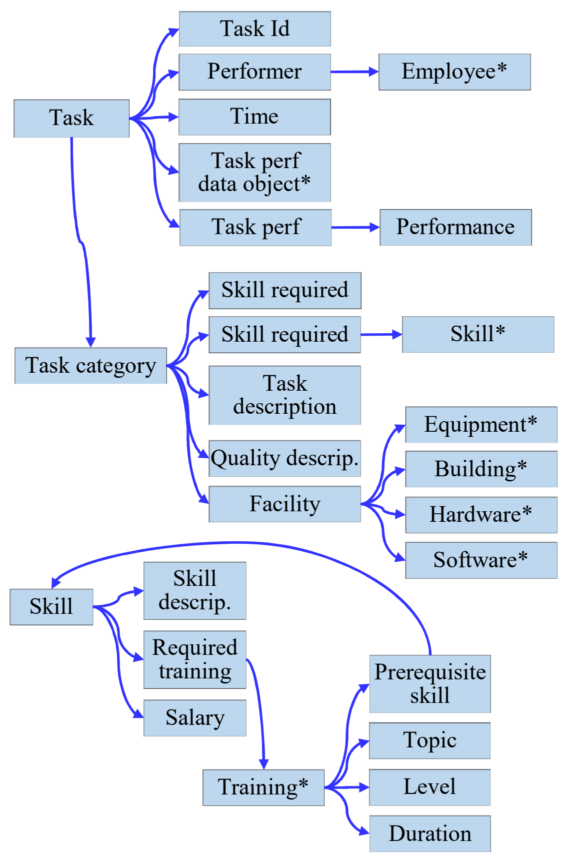

Figure 2. Low-level task model.

The task model referenced in the employee model is shown in Figure 2. This is the low-level task model and will be expanded for workflow model later. It specifies the detailed task information. A task has an ID, the time spent for it, a link to an employee who performed the task. Some data objects that may be used to assess the performance of the task are captured in the Task performance data object (TPDO) class of the task model. Based on the TPDOs, we can assess per task performance, which is given in the Task performance class.

A task belongs to a Task category and some task related facts can be specified at the category level instead of being repeated for each task in the category. The task categories form a class hierarchy. Each task category class has task description and task quality description classes, each can be a set of keywords or sentences defining the concepts of the tasks in the category and the corresponding quality. When performing a task, some facilities may be needed, such as some equipment, some computing platforms, some software tools, and so on. The task may be performed in some specific buildings. These pieces of information are captured in the Facility class. Most of the tasks have the required sets of skills. In order to obtain a skill, some trainings through various means, such as courses, training sessions, self-study, and so on, may be required. Before a training can be effective, some pre-requisite skills may be needed. The salary class in the skill model provides the average salary (and standard deviation) for people with the specific skill. This is used later in the model for computing the contributions toward a revenue.

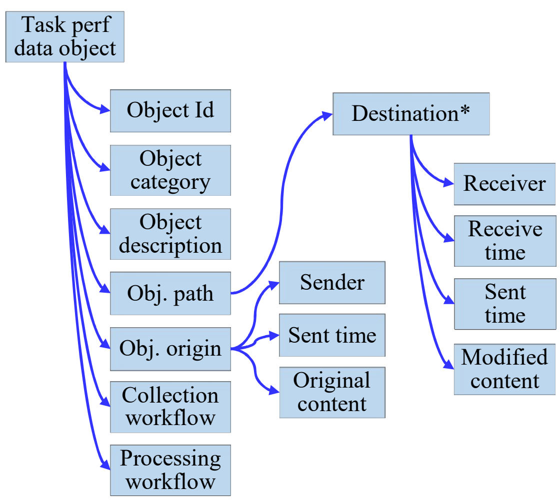

Figure 3. Performance data object model.

The Task performance data object, referenced in the Task model, captures the data objects used and routed during the execution of a certain task. The semantic model for TPDO is shown in Figure 3. From the TPDO, we may be able to reversely identify what task is performed and determine how well the task is performed. The object category may be document, database, media, etc. For different object categories, different data processing services will be needed to mine the task related information. During the execution of the task, the object may be routed among some employees (destinations) following a certain procedure and some modifications may be made to the object and the routing information is captured in the sender-receiver chain. The TPDOs are collected during task execution and will serve as the basis for automated performance assessment.

The goal for developing the semantic models is to facilitate performance appraisal. Thus, we also need to define the semantics for the performance model (as shown in Figure 4) to enable the mapping from tasks and TPDOs to performance appraisal results. We classify job performance into four major areas, namely, quantity (how much work has been performed), quality (how well the work is performed), reliability (whether the employee is reliable in dealing with tasks), and cooperation. The quality metric considers the quality of the task outcome and the employee's approach in dealing with the task. Whether the employ can correctly and quickly understand the task, deal with the task systematically, solve problems when they occur, and be innovative and take adaptive solutions when necessary. The Reliability of an employee is justified by whether she/he can correctly make decisions, complete tasks on time, does not give up when problem occurs, have the same attitude and behaviors throughout, and follow safety rules in the workplace. The cooperation area is judged by the behaviors of the employee when working in a group. The performance model can be adapted for different organizations and for different job characteristics. For each performance trait, some concepts (keywords) can be defined to guide the judgement for the trait.

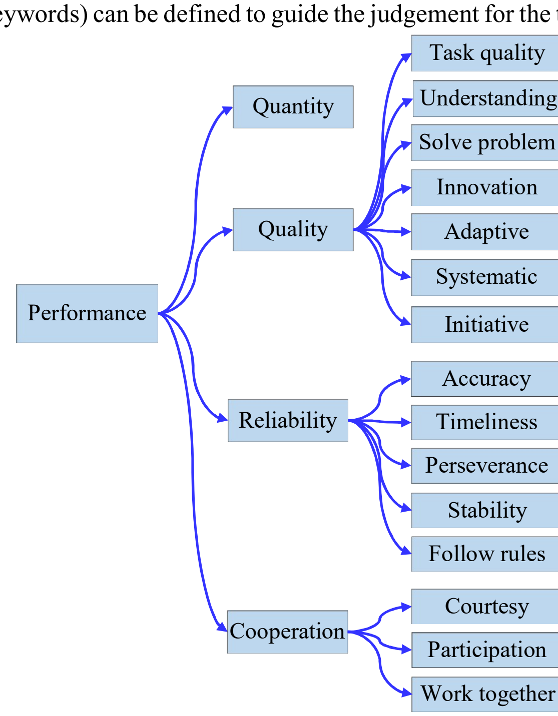

Figure 4. Performance appraisal model.

4 Automated Performance Appraisal

Job performance appraisal (JPA) is considered difficult and distasteful by managers and employees. Though JPA software tools can help manage the performance related data and appraisal reports over the time periods, the difficult decisions are still manual based. In this section, we consider automating the appraisal decisions to a certain extent. We attempt to map task performance data objects (TPDOs) collected for individual tasks as well as employee data defined in employee model to assessment scores for some traits defined in the Performance model. Though there may be a variety of TPDOs, text, code, media, etc., we only consider two categories, namely, text based and code based (software), with a focus on text based TPDOs. Also, we develop two novel automated performance appraisal approaches. In Subsection 4.1, we discuss the similarity-based relative performance appraisal approach. Subsection 4.2 presents the revenue-based performance appraisal.

4.1 Similarity-Based Relative Perf**ormance Appraisal

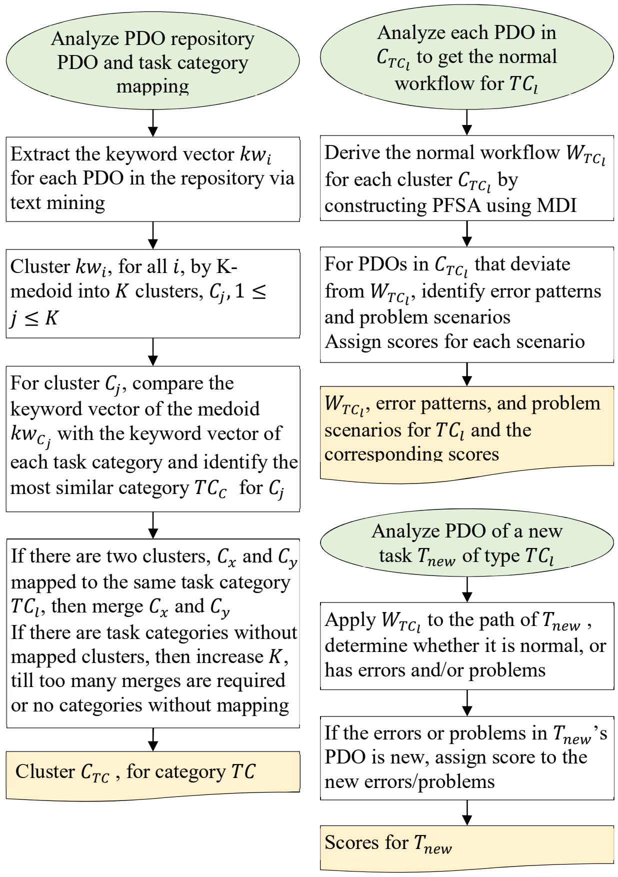

Figure 5. Per-task performance appraisal via text-based PDOs.

Figure 5 illustrates the basic mechanism for processing TPDOs and deriving similarity-based relative performance for each task performed by an employee. The assessment process references the individual TPDO for the task as well as the similar TPDOs in a repository. First, we analyze the pool of TPDOs in the repository and derive the model for identifying the task category of the TPDO. The flowchart for the algorithm is given in the left of Figure 5. We use text mining libraries to extract the keyword vector (with counts) of each TPDO. Note that there may be multiple documents in one TPDO and the keyword vector is the summary of all these documents. Then, we cluster the keyword vectors of all the TPDOs in the repository and associate the clusters with the tasks based on the similarity between the keyword vector of the cluster and that of the task. When analyzing a new task of an employee, we simply extract the keyword vector from TPDO of the task and determine its task category.

After clustering, we consider automated performance appraisal based on the characteristics of the TPDO of the task relative to the TPDOs in the cluster for the same task category. Some performance traits defined in the performance model are considered. In order to properly evaluate the per task performance, we need to also know whether the task processing has encountered any problems or whether the performer has made any mistakes. This can be analyzed based on two methods, quality/mistake/problem keywords analysis and analyzing deviations from normal processing procedures.

Normal workflow derivation and deviation analysis. Generally, the tasks in the same task category should follow a common workflow, which may have some branches for different but normal situations. From the information captured in a TPDO, we can determine how the documents are passed around for the task via the sender-receiver chain. We attempt to discover the normal workflow from a pool of TPDOs for the same task category. The workflow discovery problem is similar to the problem of constructing a probabilistic finite state automata (PFSA) from the execution traces of the automata [9]. PFSA construction techniques have been used in the literature to reconstruct the control flow structure of a program from its execution traces and detect incorrect executions [10].

A probabilistic deterministic finite-state automata (PDFA) is a tuple $\langle \Sigma, Q, \delta, q_0, P, F \rangle$ where Σ is a finite alphabet, Q is a finite set of states, $\delta: Q \times \Sigma \rightarrow Q$ is a transition function, q_0 is the initial state, $P: Q \times \Sigma \rightarrow [0,1]$ is the transition probability, $F: Q \rightarrow [0,1]$ is the final state probability, P and F are functions that satisfy the following constraints:

$$P(q,a) = 0, if\ \delta(q,a)\ is\ undefined, \forall a \in \Sigma$$

$$\sum_{a \in \Sigma} P(q,a) + F(q) = 1, \forall q \in Q$$

A PDFA $\mathcal{A}$ models a probability distribution over Σ^* and $\sum_{x \in \Sigma^*} Pr_{\mathcal{A}}(x) = 1$. The probability $Pr_{\mathcal{A}}(x)$ of generating a string $x = x_1 x_2 \dots x_n$ (where x is an execution trace) from PDFA $\mathcal{A}$ can be computed as follows:

$$Pr_{\mathcal{A}}(x) = \begin{cases} F(q^{n+1}) \prod_{i=1}^{n} P(q^i, x_i) \\ where\ q^1 = q_0\ and\ \delta(q^i, x_i) = q^{i+1}, \\ i = 1,2,\dots,n \\ 0, \quad otherwise \end{cases}$$

A commonly used PFSA construction approach is the Minimal Divergence Inference (MDI) algorithm [9], which generates PFSA that trades off minimal divergence from training sample distribution and PFSA size.

We use MDI for workflow discovery from all the TPDOs of each task category as illustrated in the top right flowchart in Figure 5. The bottom right flowchart of Figure 5 shows how to determine whether the execution of a given task T_{new}, with task category TC_l, is normal or has mistakes or problems based on the analysis of the TPDO of the task, $T_{new}.TPDO$. We first apply the automata W_{TC_l} to $T_{new}.TPDO$ and determine the normality of T_{new}. If its execution is normal, the scoring for T_{new} can be derived from the score given to the workflow as well as from the feedbacks identified from $T_{new}.TPDO$. Also, the time taken in each step in $T_{new}.TPDO$ can be compared to other TPDOs that are similar to T_{new}'s to derive the timeliness appraisal. If the execution of T_{new} is abnormal, then additional analysis is needed to figure out whether there are errors and problems in T_{new}.

Keyword based analysis to identify errors/problems. Generally, an abnormal execution would leave traces in the data objects associated with the task (in the TPDO). We can analyze $T_{new}.TPDO$ and identify keywords and sentences that indicate the potential errors and problems. The analysis can reference some predefined word ontologies that are related to common errors and problems. We can also compare $T_{new}.TPDO$ with similar TPDOs in the repository to identify the exceptional keywords and subsequently categorize them in terms of errors and problems. The mistakes in the task execution will impact the accuracy in performance appraisal. In case there are problems during the task execution, then the specific solutions taken can be mined and analyzed to determine the performance of the employee in terms of problem solving, innovation, and persistence in solving a problem encountered during a task.

Processing code-based TPDOs. Similar to text-based TPDOs, we can use big data analytics to analyze code-based TPDOs for the appraisal of software developers. In this case, the performance data can be collected related to new code developed and code modified. The metrics to be considered include code size, code complexity, level of difficulties, test cases, bugs, etc. All these metrics should be relative by comparing similar code-based TPDOs and the similarities can be measured based on similarity in requirement specifications, similarity in code structure, etc.

4.2 Revenue-Based Performance Appraisal

Revenue and profit are very important factors in commercial organizations. Thus, it is not new to analyze the contribution of an employee towards the revenue/profit of the company and use it to evaluate the job performance of the employee. Of course, such performance appraisal can be partial and should be used jointly with other performance appraisal metrics. In this section, we develop a service model (as shown in Figure 6) to help automate the revenue-based performance appraisal.

Various businesses in a company can be modeled as services, which could be production services, customer oriented services, etc. These services are specified by the business workflows. The service and workflow models in Figure 6 mostly follow the existing service models such as BPMN, but with some terminology substitutions. However, we differentiate the relation between the Task category and the Workflow from the relation between the Support services and the Task category. A Task instance is completely dedicated in a workflow instance while a Support service instance can serve multiple tasks at the same time. Thus, their contribution derivations will be somewhat different. For example, consider the workflow for providing a customized design for a customer. The tasks ranging from preparing the proposal for bidding to presenting the final design to the customer are all completely parts of the workflow. But the task of preparing a company-wide party to build good relations with all the customers is only a support task which contributes to many potential projects.

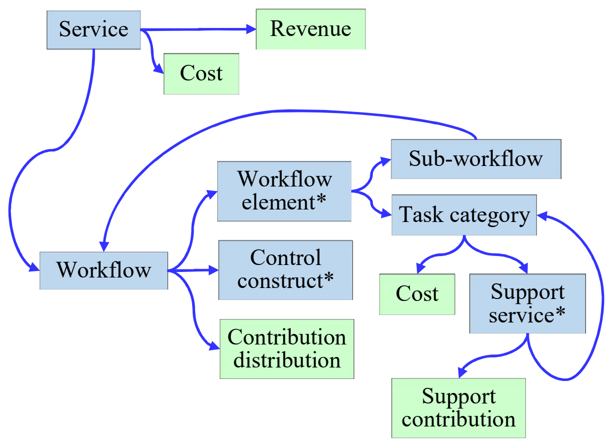

Figure 6. Service and workflow-level task models with revenue-based contribution concept.

To derive the contribution of the individual tasks in the workflow toward the overall service, we need to consider the man-hour spent on the task and the value of the contribution. Generally, the higher the degree of expertise (skill) required for the task, the higher is the contribution of the task toward the overall service. Also, the degree of expertise can be reflected by the salary. However, the contribution distribution should not be solely based on the specific service instance and the corresponding task instances, but should also consider similar scenarios. Thus, we mine the repositories to obtain average data for contribution derivation and adjust it by the data in the specific service instance.

Consider a service s with a workflow that consists of task categories $TC_1, TC_2, \ldots, TC_n$. Let the skill set required by TC_i be $TC_i.skill$. Let $\{E_{i,1}, E_{i,2}, \ldots, E_{i,m_i}\}$ denote the set of employees who actually performed tasks in category TC_i, where $E_{i,j}$ performed $T_{i,j}$ with time $T_{i,j}.time$ and $E_{i,j}.job$ and $E_{i,j}.salary$ are the job and salary of $E_{i,j}$. We mine the employee repository to obtain the set of employees $\{E^R_{i,1}, E^R_{i,2}, \ldots\}$, where for all j, $E^R_{i,j}.job$ has the set of required skills $E^i_j.job.skill$ that has sufficiently high similarity as $TC_i.skill$. We also mine the task repository to obtain the set of tasks $\{T^R_{i,1}, T^R_{i,2}, \ldots\}$, where $T^R_{i,j}$, for all j, has category TC_i. Correspondingly, we compute the contribution $ctrb(\ , T_{i,j})$ of task $T_{i,j}$ performed by employee $E_{i,j}$ toward service s as

$$ctrb(s, T_{i,j}) = \delta * E_{i,j}.salary * T_{i,j}.time$$
$$+\delta' * \operatorname*{avg}_{z}(E_{i,z}.salary * T_{i,z}.time)$$
$$+\delta'' * \operatorname*{avg}_{x}(E^R_{i,x}.salary) * \operatorname*{avg}_{y}(T^R_{i,y}.time$$

Here, $\delta, \delta', \delta''$ are parameters to adjust the weights for the actual average and individual data and the reference data from the repository. The specific revenue-based performance for task $T_{i,j}$ can then be expressed as

$$rate\left(\frac{ctrb(\ , T_{i,j})}{\sum_{x=1}^{n}\sum_{y=1}^{m_i} ctrb(\ , T_{x,y})} * s.revenue\right)$$

$rate$ is a function to convert the revenue data to fuzzy performance ratings for related traits.

The contribution of each task category can also be defined manually when defining the workflow. The contribution of a support task can be derived in a similar way as above, but it should consider the total revenues from multiple services the task has contributed to.

Note that both similarity-based relative performance and revenue-based performance are per task assessments. To derive the performance of an employee, we need to integrate the performance ratings obtained from different models as well as the performance rating of all the tasks the employee has performed. Fuzzy integration methods discussed in [7,8] can be applied to this integration.

5 SaaS-JPA Design

Based on the semantic models and the automated job performance appraisal schemes, we design a new SaaS platform for job performance appraisal. The architecture of the SaaS-JPA is shown in Figure 7. In the figure, the brown lines are for management, green lines indicate foreign data links, and blue lines are data flows.

Data collection manager. SaaS-JPA needs to host the data collection mechanism to collect task performance data objects (TPDOs) and this activity is managed by the Data Collection Manager (DCM). DCM supports TPDO data source specifications and maintains a tool suite to interact with various servers to retrieve the logging data. The tenant can configure the SaaS via data source specifications to collect TPDOs from desired sources. Since data collection is external to SaaS, DCM cannot be fully centralized and needs to deploy some services to the remote site and integrate them with internal SaaS services to achieve data collection. Each tenant may have its own workplace environment and some data collection services may have to be offered by the tenant and integrated with DCM. Security is an important issue in such situations.

Employee and Task data management. The subsystems for Employee Information Management (EIM) and for Task Information Management (TIM) have similar structures. They maintain the corresponding semantic model, support storage, regular and filtered retrievals of the collected data. They also support various mining functions to achieve the specified mining goals on the collected data. The storage, retrieval, and mining are based on the semantic data models. The TIM maintains the Task and TPDO models and the Task category and Skill class hierarchies. The EIM maintains the Employee and Job record models. Note that TIM and EIM cannot be completely independent. There are mining tasks that involve both Task data and Employee data. We put an additional interface in SaaS-JPA to support the cross repository accesses and analyses.

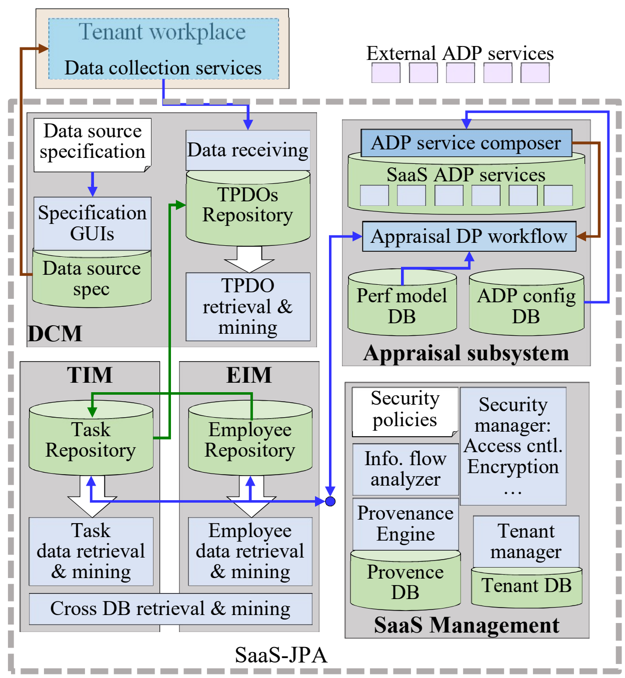

Figure 7. SaaS-JPA architecture.

The Appraisal subsystem. The appraisal subsystem hosts the performance models, the appraisal data processing (ADP) services, and the service selection and composition functions. The performance models include the SaaS-JPA model and customer defined models. The ADP services include the microservices implementing the two automated appraisal approaches discussed in Section 4, some customer ADP services, the supporting services for systematic manual appraisals, and the ADP services for performance data integration. The manual ADP supporting services generate appraisal forms and summarize performance related data based on the performance models and some appraisal templates and route them to designated people to facilitate their manual assessments. The services for appraisal data integration are based on fuzzy logic. They retrieve the task based evaluations as well as employee based evaluations by different appraisers for a spectrum of traits defined in the desired performance model and summarize them using tenant specific integration schemes. The ADP service composer composes various ADP services according to the specific tenant specifications for appraisal computation and integration. The composed ADP services take data from TIM and EIM repositories, perform ADP computation, and send appraisal results for tasks and employees back to TIM and EIM repositories, respectively.

Issues for repositories. SaaS-JPA, as shown in Figure 7, hosts several data and process repositories. They not only support the storage and retrieval of the historical and current data, but also support data mining. Some existing NoSQL databases, such as Cassandra, Radis, and so on, may be suitable for hosting these repositories. However, neither NoSQL nor relational databases have sufficient semantic support. The RDF [11] databases may be able to provide strong semantic support, but they are very inefficient, especially for big data. Thus, we add semantic knowledge to NoSQL database externally to allow the proper semantic based retrieval and inference. Also, since the quantity of TPDO data will keep on growing as time goes by, some compression or retention policies can be applied to the historical data to avoid the overflow problem.

SaaS management. In all SaaS, it is necessary to manage the tenants and their users and control the access of related data. Many PaaS, such as Azure, Heroku, Cloud Foundry, and so on, support these management activities and provide basic access control mechanisms. However, the data hosted on SaaS-JPA are sensitive to privacy while they are desirable for sharing to improve the effectiveness in big data analytics. Thus, we further offer integrated data provenance, access control and information flow control mechanisms to provide better protection to the private data of the tenants [12]. Additional security mechanisms should also be applied to the data collection processes and data collection channels. Physical system security and integrity protection should also be incorporated.

6 Conclusion

In this paper, we have designed a SaaS system for job performance appraisal. We focus on automated appraisal schemes and corresponding semantic models that are required to support the appraisal automation. We have developed two novel automated performance appraisal algorithms and designed the corresponding semantic support to enable the automation.

The future research spans several directions. First, we plan to collect data to test our automated performance appraisal schemes and design metrics to measure their effectiveness. From the evaluation results, we can improve or enhance these algorithms. We will also work with commercial organizations and compare their existing performance appraisal processes with our automated schemes. Furthermore, we will investigate additional schemes to further automate the performance appraisal processes.

7 References

[1] Gary Dessler, Human Resource Management, 15th Edition, Pearson, 2017.

[2] K. Sanwong, “The development of a 360-degree performance appraisal system: A university case study,” International Journal of Management, vol. 25, 2008, pp. 16-22.

[3] T. Anupama, M. Binu, T. Dulababu “The need of 720 degree performance appraisal in the new economy companies,” International Journal of Multidisciplinary Research, Vol.1 No. 4, August 2011.

[4] R. de Andres, M. Espinilla, L. Martinez, “An extended hierarchical linguistic model for managing integral evaluation,” International Journal of Computational Intelligence Systems, Vol. 3 No. 4, 2010, pp. 486 – 500.

[5] John C. Flanagan, “The critical incident technique,” Psychological Bulletin, Vol. 51, No. 4, July, 1954.

[6] F.J. Landy and J.L. Barness, “Scaling behavioral anchors,” Applied Psychological Measurement, Vol. 3, No. 2, April 1979, pp. 193-200.

[7] A. Shaout and J. Trivedi, “Performance appraisal system using a multistage fuzzy architecture,” International Journal of Computer and Information Technology, May 2013.

[8] R. de Andrés, J.L. García-Lapresta, L. Martínez, “A multi-granular linguistic model for management decision-making in performance appraisal.” Soft Computing, Vol. 14, No. 1, 2010, pp. 21-34.

[9] F. Thollard, P. Dupont, C. Higuera, “Probabilistic DFA inference using Kullback-Leibler divergence and minimality,” ICML 2000, pp. 975-982.

[10] W. Zhang, F. Bastani, I-L. Yen, K. Hulin, F.B. Bastani, L. Khan, “Real-time anomaly detection in streams of execution traces,” Symposium on High Assurance Systems Engineering, October 2012, pp. 32-39.

[11] W3C, “Resource description framework (RDF),” https://www.w3.org/RDF/.

[12] W. She, W. Zhu, I-L. Yen, F.B. Bastani, B. Thuraisingham, “Role-based integrated access control and data provenance for SOA based net-centric systems,” IEEE TSC, Vol. 9, No. 6, 2016, pp. 940-953.