How to Enable Multiple Skill Learning in a SLA Constrained Service System?

Sumit Kalra[1], Shivali Agarwal[2], and Gargi Dasgupta[2]

[1] Indian Institute of Technology, Kanpur, India
sumitk@cse.iitk.ac.in
[2] IBM Research, Bengaluru, India
{shivaaga,gaargidasgupta}@in.ibm.com

Abstract. In a knowledge based service system like IT services, the requirements of skills to service customer requests keep changing with time. The service workers are expected to learn the required skills very quickly and become productive. Due to high attrition rate and demand, service workers are given basic class room training and then rest of the training is carried out on-job. When a service worker learns multiple skills simultaneously, learning slows down due to factors like forgetting and interference. At the same time, the organization needs to meet service level agreements (SLA). We have developed a model for on-job training which extends the business process for IT service delivery. The key idea is to model learning, forgetting and interference in service time estimation to get realistic service times. Accurate estimation of service time taken by a service worker to resolve the service tickets helps in resource allocation and planning decisions for achieving the desired objectives of upskilling and SLA success. The simulation of execution of the augmented business process provides insights into what kind of planning and dispatch policies should be practiced for achieving the desired goals of multi-skill learning and SLA success.

1 Introduction

A *Service System (SS)* is an organization composed of (a) the human resources who perform work, and (b) the processes that drive service interactions so that the outcomes meet customer expectations [22]. Typically, a *service worker (SW)* represents a unit of human resource and a *service request (SR)* represents a unit of service work that (s)he is assigned. Hence, management of the SWs in service provider organizations is crucial. Over the past years, business and education groups have issued a series of reports indicating that due to rapid technological changes and increasing global competition, the skill demands of work are continually rising. Economists studying the changing workplace skill demands, have found that technological change is "skill-biased" thereby increasing the demand for people who have multiple skills. Many businesses are asking employees to assume multiple roles and because of this shift, hiring has become difficult in countries in spite of steady unemployment rates.

X. Franch et al. (Eds.): ICSOC 2014, LNCS 8831, pp. 260–274, 2014.

This need for multi-faceted workers entails not only retaining the right skills, but also transforming the skills of the workers as dictated by the changing business requirements. For example, in the IT services domain, it may so happen that due to a transformation in the customer's environment, a provider has to quickly upskill his team. The current team of 10 people who only had expertise in the Solaris operating system needs to be transformed to a team where both the operating systems of Windows and Solaris need to be supported. While one option for the provider is to replace some of the Solaris personnel with new hires having Windows skills, a better option is to impart new skills to existing SWs such that they collectively meet the target skill requirements.

There are several approaches for imparting new skills: (a) class room training, where SWs dedicate training time for a certain duration and incur costs, (b) shadowing, where SWs observe the work of skilled SWs and learn, or (c) on-job training, where SWs pick up skills while actually doing the work. The nature of work in services involves substantial interactions not only with the customer but also with colleagues. Also, carrying out a task is far more difficult than simply knowing how to carry out a task. Hence, on-job training following minimal classroom training is the approach commonly adopted by service providers. As of today, very little understanding exists on how the on-job training should be carried out. For example, how does the skill of a SW evolve when one or multiple new learnings are imparted ? Does this evolution of target skills change when (s)he already has some existing skills ? How do multiple learnings interfere with each other ? Can parallel learnings also reinforce ? How should the on-job training be planned and carried out such that impact to customer service in terms of service level agreement (SLA) is minimized ?

We have addressed the problem of incorporating on-job training in IT business process in this paper. This internalizes many of the questions raised above for on-job training. Our main contributions are:

1) We have developed an on-job training model based on the Dreyfus model of skill acquisition [10], the Learn-Forget-Curve-Model(LFCM)[15] and theory of interference in learning [19]. This model can be used to create a standalone training process or embedded into existing business processes. **The main components of the model are service time estimation model, skill distribution policies and finally the dispatch heuristics**.
2) The on-job training model has been woven into the IT incident management(ITIM) business process as a case study.
3) We have carried out an evaluation of the proposed model using discrete event simulation.

The evaluation focuses on understanding (i) the role of interference and skill multiplicity while imparting training for multiple skills simultaneously and (ii) how do dispatch(work assignment) policies influence learning. The rest of the paper is organized as follows. Section 2 describes the learning model based on service times during on-job training. Section 3 explains the skill distribution and dispatch heuristics components. Section 4 explains how the on-job training components get integrated into a business process. The evaluation of the training model as a part of business process is presented in section 4.1 and 4.2. Related work is discussed in section 5 and we conclude in section 6.

2 Learning Curve and Skill Progression Model

On-job training seems to be an effective way to bridge the gaps between the new and existing skills. In this scenario, a service worker gets to work on tasks which require the specific new skills (s)he is expected to be upskilled on and improvement in service times is the main observable measure to quantify learning. While initially the tasks will take longer to complete, as (s)he works on them the service time to complete tasks become smaller. In specific, authors in [15] have shown that the reduction in service time with experience follows the power-law [1]. However if there exists breaks between the new-skill tasks assigned to a worker, forgetting may happen. Also if multiple new skills are being learnt by a worker, learning interference may creep in among the multiple skills. Both forgetting and interference slow down the learning process and affect the service time. Keeping this in mind, the service time model has been designed drawing upon the existing work on learning and skill acquisition namely, LFCM and Dreyfus model respectively. We briefly explain the factors that play a role in the service time estimation below.

Learning Effect on Service Time: During on-job training, when people initially take-on new skill work, service times are longer. Assuming the difference between skills could be mapped to a gap function, we state that larger the gap between the skills, longer becomes the service times. This is modeled as gap learning factor or *glf*.

Forgetting Effect on Service Time: Time gaps between task executions [15]cause forgetting, which in turn has the effect of longer service times. Forgetting is proportional to the time gap [13].

Interference Effect on Service Time: When a service worker works on multiple new skills within the same span of time, the learning of these new skills interfere with each other. This interference results in lower recall accuracy of other skills [19] and hence in longer service times.

Skill Level Gap Effect on Service Time: Dreyfus model [10] of skill acquisition models the progression levels as Novice, Advanced beginner, Competent, Proficient and Expert. The interpretation of each level has been provided in terms of qualitative translation of each level to the task performance. This model is very appropriate for on-job training. The service times are least at the expert level and highest at the beginner's level. The time taken by a SW at any level to complete an SR is stochastic and is shown [1] to follow a lognormal distribution for a single skill.

[1] While this is true for manufacturing, the same principle can be applied to any industry where there is rhythmic and repeatable work, for example, IT service management.

We now present a service time model that takes into account the above factors. This represents the skill progression model of a worker as multiple new learnings are imparted to her.

Let T_s be the service time required by a SW for a SR with particular skill requirement while working for n^{th} , $n > 1$, time on the same skill where the SW is working on the skill after a time gap. T_{BS} is the base service time which denotes the time taken by service worker when working on the skill for the first time. T_{BS} is defined for each SW skill level. Let *dist* be the gap between required skill level of the SR and the current skill level possessed by SW. If the latter is higher or equal, *dist* is 0. The base service time is computed as $T_{BS}(1 + \log(1 + dist))$. Equations 1, 2 and 3 show the learning model while factoring in the time gap [13] only. The *timeGap* is the time spent on resolving SRs with other skills and the $timeUsed$ is the time spent on resolving SRs with relevant skill. The learning factor (lf) is a constant [15] which depends the learning pace of the SW. The gap learning factor (glf) incorporates the lf and γ, $0 \leq \gamma < 1$, which is function of *timeGap* and $timeUsed$.

$$\gamma = \frac{\log(1 + timeGap/timeUsed)}{\log n} \quad (1)$$

$$glf = lf * (1 - \gamma) \quad (2)$$

$$T_s = T_{BS} * n^{-glf} \quad (3)$$

There has been sufficient evidence in the literature to indicate that interference also causes forgetting. To include the interference in this model, we assumed that the effect of interference is equivalent to stretched time gap. To include the interference in this model, we used the results from [7] that show that the effect of interference is equivalent to stretched time gap and modify the Equation 1 as Equation 4.

$$\gamma = \frac{\log(1 + \frac{(timeGap + interferenceMeter)}{timeUsed})}{\log n} \quad (4)$$

$InterferenceMeter$ keeps the track of number of times the SW has worked on other interfering skills since last worked on the current skill. Each increment denotes a unit of time. This meter is reset to zero every time when SW works on the skill. If a SW works on the interfered skill less often, then the effect of interfering skills are more and vice-versa. However, as n increases, the impact of forgetting and interference reduces.

The quantitative model for skill progression corresponding to the Dreyfus qualitative model is obtained using time and motion studies. These studies provide a threshold on quantum of work to be done for being eligible to move to next skill level. We assume in this work that the SWs are provided basic classroom training for skills that they have never worked on before to make on-job training feasible. We shall now describe how to carry out on-job training.

3 How Is On-job Training Performed?

A SS has an existing set of skills according to the current requirements of clients. The SS periodically updates the target set of skills required based on changes in existing clients' requirements and IT infrastructure of new clients. The target skills may have only a partial overlap with the existing skills in the system. On-job training is used for transforming the existing skill profiles to the target skill profiles. We assume, w.l.o.g., that the total workforce remains same and the transition does not entail hiring. Given this setup, the training problem is solved in two phases. The first phase is the skill distribution phase that determines the skills for each SW on which she will be trained. This is a one time decision process based on heuristics as described in 3.1. The second phase is the dispatch of an incoming request to a SW as and when it arrives. This phase involves a continuous decision making process so that SS can achieve the training targets and ensure SLA success. The first phase fixes the target skills for each SW and this information is used for the dispatch of SRs upon arrival in the second phase. The two phases are explained next.

3.1 Skill Distribution

The decision of skill distribution is a preliminary step in carrying out the training. Table 1 shows the input to the distribution problem. It states the current skills possessed by three SWs with id SW_1, SW_2 and SW_3. They have 2 skills each. Last two columns in the table show the new skills required and the required number of SWs respectively. Here we can see that the skill id 5 is already present in the current skill profile while others are not. The skill distribution is carried out by adopting one of the following strategies i) *Balance skill load*, that is, balance the number of new skills to be learned per worker, ii) *Balance interference*, that is, balance the number of new skills to be learned in terms of interference. We explain both the strategies with the example input of table 1 . Let us assume that the pair of skills $\{5, 7\}$ is highly interfering and other pairs are not interfering. The output based on strategy (i), where target skills are distributed in such a manner that each SW gets a chance to learn equal number of new skill, is shown in second column of Table 2. The output provides the distribution of target skills per SW. Here, each SW requires to learn two new skills. However, the output looks different if we distribute using strategy (ii) where target skills are distributed by minimizing the interference among skills to be learned by a SW. Third column in the Table 2 shows one such possible distribution where no SW receives any interfering skill pair. We can observe that in strategy (i), the SW_3 has received skill 7 which is interfering with his existing skill 5, whereas in strategy (ii), SW_3 only gets one new skill to learn as he already knows skill 5 which is not interfering with existing skills.

We would like to add that there can be other heuristics as well for doing skill distribution. For the purpose of this work, we assume that the skill distribution is done using one of the strategies. We do not delve deep into the details of the algorithm as they are straightforward.

Table 1. SWs Current Profile and Target Requirements

Current Profiles		Target Requirements	
SW Id	Skill Ids	Skill Id	Requirement
SW_1	1,2	5	1
SW_2	2,3	6	3
SW_3	4,5	7	2

Table 2. Skill Distribution Strategies

	Balance New Skills	Balance Interfering Skills
SW Id	Skill Ids	Skill Ids
SW1	5,6	6,7
SW2	6, 7	6,7
SW3	6,7	5,6

3.2 Dispatching

The task of carrying out on-job training is equivalent to multiple sequential invocations of the task of assigning incoming SR to appropriate SW such that the SLA target is met and up-skilling of all the service worker is maximized. We have designed two different heuristics(policies) for selection of SW for an incoming SR which along with the naive policy of SLA Priority can handle different types of SS's goals. Simulations can be run for a real life SS to learn which is the best dispatch policy for the fore-casted demand. We describe the different policies next but before that, we briefly explain the interpretation of SLA in terms of timestamps.

SLA is specified by the customers for each incoming SR in terms of expected date and time of resolution. This is modeled in SS using timestamps according to the Equation 5.

$$SLA_{remain} = SR_{SLATime} - SR_{compTime} \tag{5}$$

$SR_{SLATime}$ denotes the timestamp by which the SR should be completed in order to meet SLA, $SR_{compTime}$ denotes the timestamp when the SR got completed and SLA_{remain} denotes the time remaining to meet the specified SLA.The positive value of SLA_{remain} indicates an SLA success otherwise SLA miss.The $SR_{compTime}$ is dependent on the expected service time of the SW who is working on it.

Skill-Level Priority Policy: Skill-Level Priority policy aims to maximize the SLA success, based on the observation that service worker w having matching skill level with least load, denoted by *minLoad*, is able to quickly complete the assigned SR, hence maximizing the SLA success. Each SR in the SW's queue contributes to load proportional to the skill level gap between incoming SR and the SW. A threshold on the queue load determines if a SW is overloaded.

Algorithm 1 formally describes the Skill-Level Priority policy for assigning a service request SR to assign appropriate service worker w among the pool of available service workers $SWList$. Initially Skill-Level Priority policy checks for all the service workers which have same skill level as required by the SR, are not overloaded and can meet SLA based on expected service time. Among them, it finds the least loaded service worker w. The load due to pending SRs in the queue is denoted by $SRPendingQueueLoad$. If all the service workers with equal skill level as required by SR are overloaded or not available, then the policy looks for the service workers having the same skill required by the SR with one level lower and higher which are not overloaded and so on. As we have finite skill levels, the algorithm terminates. If it does not find anyone, then the least loaded SW is chosen. Amongst the shortlisted SWs, it then computes γ to find who has the maximum learning potential.

```
Input: SR, SWList
Output: SW_id
id = φ
minLoad = 150
diff = 0
while id = φ AND diff < 4 do
    for each w_i ϵ SWList do
        if abs(SR_SkillLevel − w_i.SkillLevel) = diff AND w_i.overload = false
        then
            if minLoad > w_i.SRPendingQueueLoad then
                id = w_i.id
                minLoad = w_i.SRPendingQueueLoad
            end
        end
    end
    if id == φ then
        diff = diff + 1
    end
    else
        break
    end
end
return id
```

Algorithm 1. Skill-Level Priority Policy Outline

Learning Priority Policy: Learning Priority policy gives more chances to the service workers with lower skill levels in order to assign them more service requests and increase their experience and learning. This policy looks at all the service workers which can complete the SR and meet SLA by calculating the expected service time and check if it is less than the remaining SLA time. Since this policy always prefers the service worker with maximum worst case service time $maxWorstServiceTime$ who can complete the SR within SLA, there are higher chances of increasing the skill level of the service worker at the cost of increasing the probability of missing SLA.

Algorithm 2 formally describes the Learning Priority policy for assigning a service request SR to assign appropriate service worker w among the pool of available service workers $SWList$. Initially Learning Priority policys checks for all the service workers which have expected service time less than remaining service time and not overloaded. Among them, it finds the least loaded service worker w with highest value of worst case service time $maxWorstServiceTime$.

Input: $SR, SWList$
Output: SW_{id}
$id = \phi$
$minLoad = 150$
$maxWorstServiceTime = 0$
for *each* $w_i \epsilon$ *SWList* **do**
  if $w_i.expectedServiceTime < SR.SLA_{remain}$ *AND* $w_i.overload = false$ **then**
    if $minLoad > w_i.SRPendingQueueLoad$ *AND* $maxWorstServiceTime < w_i.worstServiceTime$ **then**
      $id = w_i.id$
      $minLoad = w_i.SRPendingQueueLoad$
      $maxWorstServiceTime = w_i.worstServiceTime$
    end
  end
end
return id

Algorithm 2. Learning Priority Policy Outline

SLA Priority Policy: The work dispatch based on SLA priority policy mimics on ground reality of existing service systems. It basically dispatches the SR to the first available SW who has skills to carry out the work. There is no other consideration like skill level, learning progress etc. Note that SLA Priority policy does not compute expected service time according to the learning curve model of section 2 while choosing the SW as is done by Skill-Level or Learning Priority policy thus differing from them in a crucial way.

4 Case Study - IT Incident Management

ITIM process is one of the main candidates for on-job training use case in SS, hence, we chose it for the case study. The IT incident management process extended with tasks for on-job training is illustrated in Fig. 1. We have built upon the ITIM process studied in [18] which is revisited briefly as follows. A problem or issue faced by a business user is reported to a help desk. The help desk personnel opens an incident ticket in a ticketing tool and records the description of the issue. Then the incident is assigned to a specific work group based on the problem described by the user. The incident, once assigned to a work group, is picked up by an available resource within the work group who then updates the assignment information indicating the ownership of the incident. The incident enters the resolution stage. The resource further analyzes the problem in the ticket, communicates to the business user for more input on the problem, and resolves the problem. Once an incident is resolved, the resource restores the functionality of the system as required by the business user. The business user validates and confirms the service provided by the resource. Once confirmed by the business user, the incident is closed.

The extension to the existing ITIM process for training is primarily in the dispatch task. The incident, once assigned to a group, is assessed by a dispatch engine for the skills that it requires and the load on the SWs that have been identified to work on those skills. Subsequently, the expected service time for the shortlisted SWs is computed according to the model in section 2. Then, after considering the SLA requirements, the engine selects an incident owner following one of the proposed policies and the incident is dispatched. Once the incident is resolved, the parameters that track the learning of the SWs are updated and so is the SLA measurement. The process to initiate the closure of the incident is also initiated in parallel.

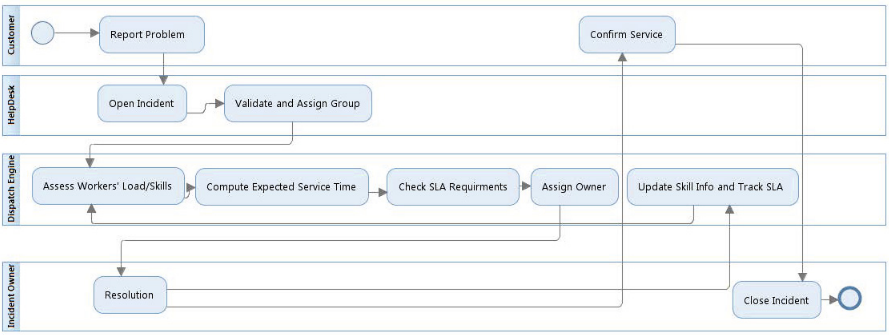

Fig. 1. IT Incident Management Process Extended with On-Job Training Tasks

4.1 Simulation Framework Overview of Enhanced ITIM Process

A discrete event simulation [1] of the ITIM process augmented with training has been used to gain insights into the proposed training model. A service request(SR) arrives in the system and is redirected to a service worker (SW) who resolves it. A valid set of states for a SR is *inqueue* (default), *pending*, *inservice*, *rework* or *completed* and similarly, set of valid state of service worker is defined as {Available, NotAvailable}. For each SW, we also maintain information such as existing skills, new skills being learned, working hours shift availability, overload status. The expected service time of a SR for each SW is computed using the learning curve described in section 2. SLA requirements for all customers are assumed to be: *as long as a provider completes* 95% *of all SRs received every month within specified hours*, the quality of service is deemed adequate. We also assume that each SR requires single skill like unix, windows, db2 etc. The main components are described below.

Global Queue: A global queue is maintained which accepts all the incoming SRs with different priority and different skill requirement. For every SR, we maintain the information such as priority, SLA deadline, skill and corresponding level requirement and status as *inqueue* (default), *pending*, *inservice*, *rework* or *completed.* This global queue serves as the input to the dispatching module.

Dispatch Module: The dispatching module accepts SRs from global queue one by one and uses list of all the SWs in order to search for the most suitable SW for the current SR based on a policy described in section 3. A policy remains in force for the period of simulation (say, a month). After identifying the SW, SR is sent to its queue and the status of the SR is changed from *inqueue* to *pending.* The *SRPendingQueueLoad* is updated as described in equation 6.

Service Workers' Queue: A service worker's queue can have SRs of skill levels different than his current level. The load value in such a situation is normalized by having more complex SRs contribute more to the load than the lower level ones. We assume the normal load of a SR for SW is equal to 20 and the Equation 6 is used to calculate the load due to different levels.

$$weight = 20 + (SR_{skill-level} - SW_{skill-level}) \times 5 \tag{6}$$

Let *curload* of a SW denote the load due to pending SRs in the queue. We calculate whether a SW is overloaded or not as follows:

$$overloaded = \begin{cases} yes & \text{if } curload \geq 100 \\ no & \text{otherwise} \end{cases}$$

Once SR is assigned to a SW, it remains in *pending* state in the queue of the SW till all the SRs which arrived before it are resolved. When SW works on the SR, the status is updated to *inservice.* To introduce some failure cases in the simulation model, where a SW fails to resolve the SR as per requirements, every SR with 0.01% prbability sent for rework. If the SR being sent for rework, it is status is updated to *rework* and placed on the global queue along with recently arrived SRs. Otherwise, the status is updated to *completed.*

Learning Parameters and Interference Meter: For simulation purpose, we set the learning factor lf to 0.1 and *timeGap* is captured in unit of work hours. We start an interferenceMeter for each skill for every SW with value 0. Whenever a SW works on a particular skill, the value of interferenceMeter of skills being interfered by current skill is incremented. It is reset to 0 for a skill for SW when he works on that skill.

Statistical data collection and Skill Progression: The framework continuously collects data such as skill level progression rate, SLA success rate. Skill level is upgraded after sufficient experience for the skill. In our simulation model, we assume that after working on 500 SRs of the same skill, SWs skill level is incremented by 1 with minimum value 1 and maximum value 4. These skill level from 1 to 4 represents the proficiency of SW as Basic, Developed, Advanced and Expert respectively. We have adopted a simplified model of skill levels as proposed in [10].

Workload Generation: Diverse workloads for the simulation are generated as described below. Given the average inter-arrival time, assuming Poisson distribution, the workload is generated for a specified number of weeks. For each arrival, there are associated parameters of priority, skill and skill-level required to resolve the SR. Here we assume that a SR is assigned to only one SW and that SWs can have multiple SRs with a limit upto 5 SRs with skill gap 0, pending in its queue at any given instance of time during simulation. For a particular skill, the number of current and required SWs is specified as illustrated in table 1. Dispatch simulation is performed on different combinations of skill distributions arising out of the two strategies. Under uniform skill load distribution, each SW gets equal number of skills where as in left (right) skewed skill load distribution, most of the SWs get less (more) number of skills. Analogously, the workload consists of left skewed and right skewed distribution of interference load.

Note: It is possible that one skill pair is more interfering than the other pair. Interference can also be unidirectional. However, there is no quantitative model for this yet in the literature. In the absence of any quantitative model for interference load, we consider only presence or absence of interference between two skills and assume symmetric interference. The value for Interference load is 10 if two skills are not interfering as oppose to value 90, which denotes that the pair of skills are interfering. The interference load is computed as a summation of pairwise interference when more than two skills are being learned.

4.2 Simulation Experiments and Results

The training process is simulated under different workloads and policies to understand the tradeoffs that exist in adopting such a process while delivering services. The key insights obtained from the experiments are listed below.

Observation 1 (On Skill Load vs. Interference): Skill load and interference are equally strong deterrents in learning. We carried out experiments with a target skill profile having high number of skills to be learned uniformly such that the skills do not interfere (Table 3 Scheme S-2) and compared the learning time with a target profile where the skills to be learned do not exceed *two* but these skills interfere with each other (Table 3 Scheme S-3). In both scenarios, we find the similar pattern of skill level progression. In Scheme S-1, we kept both the skill load and interference load to low. The entries in the table show the percentage of SWs at the levels L1 to L4 at starting of weeks 1, 10, 20 and 40. These numbers are of Learning First Policy. However, the observation holds in the other two policies also.

Observation 2 (On Learning Pace): Learning Priority policy aids in uniform learning while Skill-Level Priority policy aids in greedy learning. Fig. 2

Table 3. Effect of Skill Load/Interferene Load on skill progression

	S-1				S-2				S-3			
	L1	L2	L3	L4	L1	L2	L3	L4	L1	L2	L3	L4
1	100	0	0	0	100	0	0	0	100	0	0	0
10	5	80	15	0	100	0	0	0	100	0	0	0
20	0	35	45	20	58	42	0	0	67	33	0	0
40	0	0	25	75	38	55	7	0	36	51	13	0

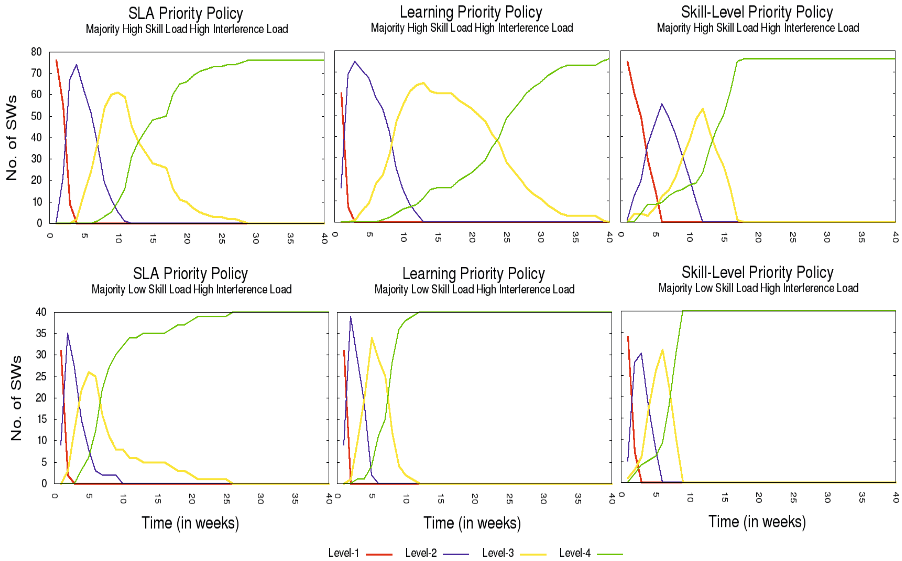

Fig. 2. Skill progression: High/Low skill load and high/low interference load

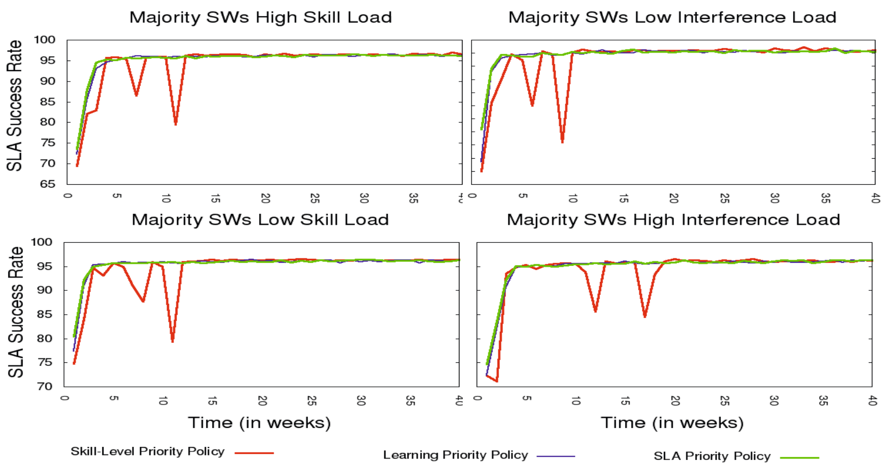

Fig. 3. SLA Success Rate Comparison

demonstrates this for two types of workloads. It can be seen in Learning Priority policy that at any given time more than 90% SWs are distributed at two consecutive levels but this is not so for the other policies. This is indicative of uniform learning where the level gap between SWs is not too much at any given point in time. Skill-Level Priority, however, follows non-uniform learning and SWs reach the highest level faster compared to the other policies. SLA Priority is neither uniform nor greedy. The simulations were run for all combinations of workload and the same trends were observed. We have presented results only for two distributions for sake of brevity.

Observation 3 (On SLA success): An interesting insight that we got was that if the dispatch policy tries to prioritize on exact match of skill levels as in Skill Level Priority, then sharp dips in SLA success rate are likely as shown in Fig. 3. This happens due to longer SR pending queue.

Applying the Insights: The insights obtained above from simulation runs can be applied in practice by SSs to achieve desired behavior. We summarize some of the important practical considerations that emerged from the experiments: i) SS should adopt Learning Priority policy if uniform learning is more desirable, ii) If the goal is to promote a competitive environment, Skill-Level Priority policy is most advisable provided SLAs are relaxed, iii) For efficient learning, the new skills to be learned per worker should be minimized; and an attempt should be made to minimize the interfering skills to be learnt per worker.

5 Related Work

In this section, we situate our work within prior research on team and organizational learning theories, resource planning, human skill evolution and learning.

One of the most recent works that studies multi-skill requirements in service delivery is [8]. This work studies the problem of optimal skills to train people on while in this paper we have studied how to train people on multiple skills. Learning has also been looked at in the context of human resource planning [4], [3], where there is a need to forecast the future skill mix and levels required, as well as in context of dynamic environments like call centers[12], where both and learning and turnover are captured to solve the long and medium term staffing problem.

There has been a significant body of work focused on teams and their learnings. About two decades back researchers[25,11] studied the effects of organizational structure (i.e. hierarchy, team etc.) on metrics like problem solving, cost, competition and drive for innovation and also the effect [6] of learning and turnover on different structures. At the same time, collaborations and communication with teams have also seen a comprehensive body of research. Carley's [5] theory of group stability postulates a relationship between individual's current knowledge and her behavior. She also found that a group's interaction increases as commonality across knowledge dimensions increases. Very recently [17] presented the notion of synergy in human teams or *how well* they work together.

In context of skill evolution, Dibbern et. al [9] captures the dependencies of expertise, task complexity, support information and learning tasks on learning effectiveness during Knowledge Transfer. Imparting knowledge with on-the-job training has also been another popular method for imparting skills. Work in labor economic theory [2] has attempted to assess how much on-the-job training is needed for a specific worker, based on his current expertise and learning ability.

In the domain of learning, authors [16] talks about accelerating learning of agents via human feedback. It is also shown that [24] optimizing skills in isolation does not necessarily benefit their combined operation. According to authors, how much an individual learns when challenged, depends on the skill level of the performer and the task complexity. Apart from the learning and forgetting models ([15,14,20,21]) presented in Section 1, recent work [23] presents interesting results on how memory consolidation and forgetting processes regulate the memory capacity, and can mutually improve the effectiveness of learning.

6 Conclusions

We conclude that distribution and dispatch policies play a crucial role in balancing SLA success and upskilling when performing on-job training. The presence of interference slows down the learning rate and so does the number of skills to be learned. As part of future work, we plan to formalize interference model and study semantic facilitation during training. We also plan to study the training method in context of other business processes where on-job training is practiced.

References

1. Banerjee, D., Dasgupta, G., Desai, N.: Simulation-based evaluation of dispatching policies in service systems. In: Winter Simulation Conference (2011)
2. Barron, J., Black, D.A., Loewenstein, M.A.: Job matching and on-the-job training. Journal of Labor Economics 7(1), 1–19 (1989)
3. Bordoloi, S.: A control rule for recruitment planning in engineering consultancy. Journal of Productivity Analysis 26(2), 147–163 (2006)
4. Bordoloi, S.K., Matsuo, H.: Human resource planning in knowledge-intensive operations: A model for learning with stochastic turnover. European Journal of Operational Research 130(1), 169–189 (2001)
5. Carley, K.: A Theory of Group Stability. American Sociological Review 56, 331–354 (1991)
6. Carley, K.M.: Organizational learning and personnel turnover. Organization Science 3, 20–46 (1992)
7. Das, A., Stuerzlinger, W.: Unified modeling of proactive interference and memorization effort: A new mathematical perspective within act-r theory. In: Proceedings of the Annual Meeting of the Cognitive Science Society, COGSCI 2013, pp. 358–363 (2013)

8. Dasgupta, G.B., Sindhgatta, R., Agarwal, S.: Behavioral analysis of service delivery models. In: Proceedings of the Service-Oriented Computing - 11th International Conference, ICSOC 2013, Berlin, Germany, December 2-5, pp. 652–666 (2013)
9. Dibbern, J., Krancher, O.: Individual knowledge transfer in the transition phase of outsourced softwaremaintenance projects. In: ISB-IBM Service Science Workshop (2012)
10. Dreyfus, S.E., Dreyfus, H.L.: A Five-Stage Model of the Mental Activities Involved in Directed Skill Acquisition. Tech. rep. (February 1980), http://stinet.dtic.mil/cgi-bin/GetTRDoc?AD=ADA084551&Location=U2&doc=GetTRDoc.pdf
11. Jablin, F.M., Putnam, L.L., Roberts, K.H., Porter, L.W. (eds.): Handbook of Organizational Communication: An Interdisciplinary Perspective. Sage (1986)
12. Gans, N., Zhou, Y.P.: Managing learning and turnover in employee staffing. Oper. Res. 50(6) (2002)
13. Jaber, M.Y., Bonney, M.: Production breaks and the learning curve: The forgetting phenomenon. Applied Mathematical Modelling 20(2), 162–169 (1996), http://www.sciencedirect.com/science/article/pii/0307904X9500157F
14. Jaber, M.Y., Kher, H.V., Davis, D.J.: Countering forgetting through training and deployment. International Journal of Production Economics 85, 33–46 (2003)
15. Jaber, M.Y., Sikstrom, S.: A numerical comparison of three potential learning and forgetting models. International Journal of Production Economics 92(3) (2004)
16. Knox, W.B., Stone, P.: Reinforcement learning from simultaneous human and mdp reward. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, vol. 1 (2012)
17. Liemhetcharat, S., Veloso, M.: Modeling and learning synergy for team formation with heterogeneous agents. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, vol. 1, pp. 365–374 (2012)
18. Liu, R., Agarwal, S., Sindhgatta, R.R., Lee, J.: Accelerating collaboration in task assignment using a socially enhanced resource model. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 251–258. Springer, Heidelberg (2013)
19. Jan Mensink, G., Raaijmakers, J.G.W.: A model for interference and forgetting. Psychological Review, 434–455 (1988)
20. Nembhard, D.A., Uzumeri, M.V.: Experiential learning and forgetting for manual and cognitive tasks. International Journal of Industrial Ergonomics 25, 315–326 (2000)
21. Sikstrom, S., Jaber, M.Y.: The power integration diffusion (pid) model for production breaks. Journal of Experimental Psychology 8, 118–126 (2002)
22. Spohrer, J., Maglio, P., Bailey, J., Gruhl, D.: Steps toward a science of service systems. Computer 40(1), 71–77 (2007)
23. Subagdja, B., Wang, W., Tan, A.H., Tan, Y.S., Teow, L.N.: Memory formation, consolidation, and forgetting in learning agents. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, vol. 2 (2012)
24. Urieli, D., MacAlpine, P., Kalyanakrishnan, S., Bentor, Y., Stone, P.: On optimizing interdependent skills: A case study in simulated 3d humanoid robot soccer. In: Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011) (May 2011)
25. Williamson, O.E.: The economics of organization: The transaction cost approach. American Journal of Sociology (1981)