DRIVING USER ADOPTION USING ENTERPRISE GRAPH CLUSTERING

Darko Zlatic
Keith Griffin
Johan Van Puymbrouck
DRIVING USER ADOPTION USING ENTERPRISE GRAPH CLUSTERING

AUTHORS:
Darko Zlatic
Keith Griffin
Johan Van Puymbrouck

ABSTRACT

Techniques are described herein to analyze collaboration behavior based on collaboration data inputs using graph clustering. This may enable generation of deployment models resulting in high collaboration application adoption.

DETAILED DESCRIPTION

Driving adoption of new team based messaging platforms can be challenging in enterprise environments. Deployments can have questionable chances of success for several reasons. First, in today’s enterprise, regardless of internal tools standardization, a wide range of implicit and explicit freedoms are given to users to choose tools they will use, as well as to choose between passive resistance, formal compliance or full commitment to the tools the enterprise provides to them. Second, a change in behavior of employees may be required from previous tools. Third, given that people follow behavior they see from watching others, it is critical to demonstrate good usage practice in the immediate enterprise social network early in deployment. Fourth, vendors require ways to measure and drive adoption rather than hope for organic growth.

In particular, the following three goals are important to the success of such deployment. The first goal is to maximize the chance for each employee in the enterprise to become a committed user of the team collaboration tool. The second goal is to generate best practice examples and hints to other users at the beginning of the new deployment. The third goal is to manage and report the team adoption process for an organization.

The techniques described herein solve these three problems for deployed applications. Briefly, deploying team collaboration applications in a controlled group based rollout results in better ongoing user adoption.

In a first step of the method, based on existing enterprise data for a given enterprise organization (e.g., conferencing logs, email, phone records, etc.), the weighted undirected
graph may be generated by showing each employee as a node, and having a link between two nodes representing the frequency of the communication between two employees. Frequency is just an example of the possible function.

In a second step of this method, graph clusterization techniques may be applied on the weighted undirected enterprise social graph created in the first step. As illustrated in Figure 1 below, the graph may be split into sub-graphs of nodes/employees who are engaged in a higher frequency of communication among each other than with the rest of the graph/organization. The communication capacity of the sub-graph may also be calculated as a model measure of the internal frequency of communication. The relative influence between two neighboring sub-graphs may also be calculated based on their communication capacities and number of interconnections. This enables the creation of a new aggregated graph with nodes representing sub-graphs of the original graph as deployment groups as the nodes, and links representing cumulative influence between two deployment groups. Besides calculated communication capacity for node data, a placeholder may be used for measured intra-node team adoption parameters.

![Figure 1](https://www.tdcommons.org/dpubs_series/1518)

In a third step of this method, deployment groups within the aggregated graph generated in the second example step may be selected with sufficient communication capacity to engage in the first round of deployments. These may be referred to as “activated groups.” As this third step progresses, intra-node team adoption may be measured for activated groups. New eligible groups may be selected once communication capacities rise above a threshold, after neighboring groups, with significant influence, show required adoption. This way a “controlled fusion” reaction may be created to make an optimal adoption flow for the organization.
In a fourth step of this method, as a possible extension, data used for activation decisions may be captured and used with supervised machine learning, with actual adoption data as labels, for making better decision in future deployment rounds, assuming that the model has been pre-trained with simulated adoption for the type and size of the organization.

Figure 2 below illustrates an example of possible implementation architecture.

The enterprise social graph is a software component that receives data from different sources and structures them in the form of a weighted graph based on preset graph generation parameters. Data sources are connected to the enterprise social graph via plug-ins. The deployment planner is a software component that makes an intelligent deployment plan based on the current snapshot of the enterprise social graph. Each deployment planner includes a deployment schedule for teams over certain clusters in the enterprise social graph, and also provides a prediction of a deployment dynamic that can be improved over time with machine learning extensions. It is a versioned document object with comparison methods for easy research and adjustments over the dynamic enterprise graph data. The adoption monitor collects specific team usage data and produces reports about adherence with deployment plan predictions and feeds the deployment data back to the deployment planner for use with the machine learning component.

This solution is cloud/on-premise agnostic, allowing any combination of cloud/on-premise/hybrid deployment. The collaboration software plug-in and adoption monitor are only two components required to interface with vendor collaboration cloud. The output of the system may be static or dynamic. Static output may simply produce a deployment plan. In the case of a dynamic system it is possible to automate the deployment, e.g., the system decides based on the output from the deployment planner component how to proceed with
deployment to maximize success. This may be semi-automated if an administrator wants to maintain control of the system.

Although the method described herein is focused on new client deployments and adoption of team collaboration tools as opposed to incremental feature deployments, these techniques may also be used for individual feature deployments. One goal could be to identify users who are mostly likely to communicate and influence successful adoption in team environments using graph clustering.

The graph may be built on existing communication patterns in the customer environment and even though this may be before the team platform is deployed, data may be ingested from the existing platform (e.g., communication records). This is typically a long manual process today.

One way could be to inform the enterprise administrator for which user groups to deploy a platform in order to achieve successful adoption. However, the system may perform an automated roll-out or, alternatively, may be a human-in-the-loop system that recommends a deployment plan that the enterprise administrator approves and/or adjusts before deployment.

In summary, techniques are described herein to analyze collaboration behavior based on collaboration data inputs using graph clustering. This may enable generation of deployment models resulting in high collaboration application adoption.