Available online at www.sciencedirect.com

ScienceDirect

Manufacturing Letters 10 (2016) 1–5

www.elsevier.com/locate/mfglet

Research Letters

A knowledge-based social networking app for collaborative problem-solving in manufacturing

D. Mourtzis *, M. Doukas, N. Milas

Laboratory for Manufacturing Systems and Automation, University of Patras, Greece

Received 6 November 2015; accepted 1 August 2016
Available online 4 August 2016

Abstract

Employees within companies act as "social machines" that process data and information, and produce valuable, yet tacit knowledge. This paper considers a smart factory framework inspired from social networking, in which employees collaborate to tackle problems related to product and production. A knowledge-based mobile app, based on the Root Cause Analysis process, formalises problem-solving instructions and expert opinions described in natural language via the Vector Space Model method. The utilised cloud infrastructure facilitates knowledge sharing and provides a business-oriented social network for employees. Both method and developed app are validated in a case study obtained from the automotive die-construction domain.
© 2016 Society of Manufacturing Engineers (SME). Published by Elsevier Ltd. All rights reserved.

Keywords: Manufacturing system; Knowledge management; Social manufacturing

1. Introduction and state of the art

Modern enterprises function on the basis of collaboration among stakeholders across the product/service value-adding chain [1,2]. Current collaboration practices, however need modernisation [3]. Meanwhile, employees are perceived as "social machines" [4], which process data/information, and produce valuable, yet implicit knowledge. Constituting this knowledge reusable, entails major advantages for problem-solving [5] among other domains [6,7]. The adoption of social media in manufacturing can empower corporate collaboration and provides a knowledge capturing mechanism [4].

Furthermore, Root Cause Analysis (RCA), a formal problem-solving method widely used in industrial practice [8] can be facilitated by Cloud Manufacturing. Moreover, mobile devices, which are acknowledged as IT mega-trends [9,10] can act as information brokers and enable interactivity and constant awareness on core enterprise functions [11].

2. Method

Towards improving problem-solving, this work proposes a method mimicking social networking in which employees collaborate to deal with problems emerging during an engineering project's lifecycle. Within the social manufacturing eco-system, the RCA method is used to support formal problem-solving, identification of causes, and formulation of effective solution steps (Fig. 1).

A key step in problem identification is problem statement [12], yet, current practice treats it from a heuristics approach. In this study this step is enhanced by exploiting previous knowledge. During the submission of a new problem to the social platform, natural language descriptions are used by engineers, which are processed by the Vector Space Model (VSM) method, to extract from the knowledge base similar problems with reported solutions. If a

* Corresponding author.
E-mail address: mourtzis@lms.mech.upatras.gr (D. Mourtzis).

http://dx.doi.org/10.1016/j.mfglet.2016.08.001
2213-8463/© 2016 Society of Manufacturing Engineers (SME). Published by Elsevier Ltd. All rights reserved.

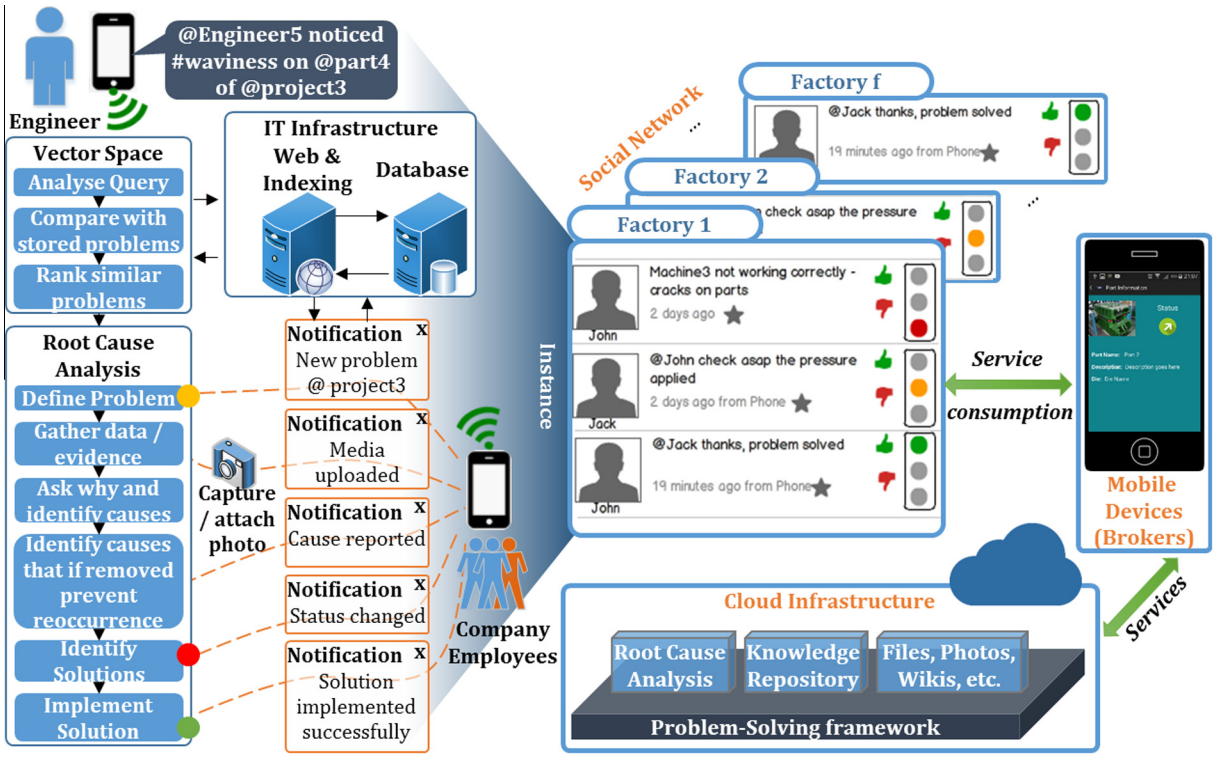

Fig. 1. Framework components and the problem submission process.

problem has been encountered previously, its solution is retrieved from the repository and is adapted to the new situation. If not, the problem description is posted to the business social network, in search for a solution. Problems already resolved are marked with a "green" traffic light. Problems still pending but with known solutions are marked with "orange". Serious problems that do not have a known solution or could cause major disruptions to a project are marked with "red".

Problem descriptions are restricted to 150 characters long for the sake of conciseness and avoidance of story-telling. Keywords, which characterise a problem, are signified using the '#' symbol and are automatically assigned with a higher similarity weight factor [13]. The recipient individual or group is signified using the '@' symbol. Engineers belonging to the same team are automatically subscribed to notifications. For instance, when a "red" problem arises, the project manager and team members are notified through the developed mobile app to submit possible solutions and causes. Each participant can vote submitted solutions/causes either positively (thumbs up) or negatively (thumbs down) to promote best practices. Finally, the problem, root causes, and solutions are queued up for approval by an administrator and are afterwards stored and indexed into the knowledge base, to become reusable in future cases.

Stored problems, in the context of Natural Language Processing (NLP) referred to as documents, represent formalised explicit problem-solving knowledge. An indexing method parses these documents to tokenise words and associate them with a unique document identifier [14]. After all documents are parsed, the inverted file is sorted out by terms.

A new problem description submitted by an engineer comprises essentially a query for this accumulated knowledge (Fig. 2). Considering that word-by-word comparisons are required for such short queries, VSM is selected as the most suitable NLP method [14] and implemented by the Elasticsearch engine [15]. In VSM, sets of documents and queries are represented as vectors with dimensionality equal to the number of dictionary terms. The bag-of-words assumption is considered, since the order of words in short descriptions does not affect the retrieval accuracy. Let us assume a query q, an already stored document d in a database of D documents, and t a term from the vocabulary T. Vectors d and q are formed to represent the total weights for each term in documents and queries respectively. The weighting method applied is an enhanced version of *tf-idf* (term frequency-inverted document frequency), which uses custom weights to distinguish keywords from less important terms. The *tf* method gives extra weight to terms used repeatedly in a document and the *idf* method attenuates the effect of common terms that occur in multiple documents, by counting the number of documents that contain a term t (df_t). The result is a composite weight comprised of individual weights for terms dt, $t \in [1,\ldots T]$, calculated by (1). In case a term is hash-tagged, i.e. is a keyword, then its weight is multiplied with the factor 1.5 for improving its relative significance. The cosine similarity between the query vector q and the document vectors d_i

Download English Version:

https://daneshyari.com/en/article/1697925

Download Persian Version:

https://daneshyari.com/article/1697925

Daneshyari.com