



HAL
open science

InnoCrowd, An AI Based Optimization of a Crowdsourced Product Development

Indra Kusumah, Clotilde Rohleder, Camille Salinesi

► **To cite this version:**

Indra Kusumah, Clotilde Rohleder, Camille Salinesi. InnoCrowd, An AI Based Optimization of a Crowdsourced Product Development. 18th IFIP International Conference on Product Lifecycle Management (PLM), Jul 2021, Curitiba, Brazil. pp.267-278, 10.1007/978-3-030-94335-6_19. hal-04186152

HAL Id: hal-04186152

<https://inria.hal.science/hal-04186152v1>

Submitted on 23 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

InnoCrowd, an AI based optimization of a crowdsourced product development

Indra Kusumah¹[0000-0003-4750-8998], Clotilde Rohleder¹[0000-0002-5638-5531], Camille Salinesi²[0000-0002-1957-0519]

¹ University of Applied Science HTWG Constanz, Germany

² CRI Paris 1 Sorbonne University, Paris, France

Abstract. The development of a new product can be accelerated by using an approach called crowdsourcing. The engineers compete and try their best to provide the related solution based on the given product requirement submitted in the online crowdsourcing platform. The one who has submitted the best solution get a financial reward. This approach is proven to be three time faster than the conventional one. However, the crowdsourcing process is usually not transparent to a new user. The risk for the execution of a new project for developing a new product is not easy to be calculated [1,2]. We developed a method InnoCrowd to handle this problem and the new user could use during the planning of a new product development project. This system uses AI concepts to generate a knowledgebase representing histories of successful product development projects. The system uses the knowledge to determine qualitative and quantitative risks of a new project. This paper describes the new method, the InnoCrowd design, and results of a validation experiment based on data from a current crowdsourcing platform. Finally, we compare InnoCrowd to related methods and systems in terms of design and benefits.

Keywords: New Product Development, Artificial Intelligence, Knowledge Management

1 Introduction

Howe has defined Crowdsourcing by 2006 [3]. Nowadays, crowdsourcing has gained popularity in several application areas because of significant cost and time savings [4]. Crowdsourcing covers quite a large range of application scopes, including simple tasks such as tagging a picture, medium tasks like programming a simple app, and even complex tasks such as designing and producing a vehicle [5]. Our work addresses primarily bigger projects, like the latter, a typical one for Local Motors. A traditional automobile manufacturer employs less than 5000 engineers to design a new car, but Local Motors depends on the support of more than 30.000 engineers (the crowd) for this purpose, as illustrated in Fig.1. Engagement of more people can make the design process up to three times faster and the manufacturing cost more than 10 times cheaper. Also, taking advantage of a greater variety of expertise and backgrounds tends to improve the innovation and quality of the product.

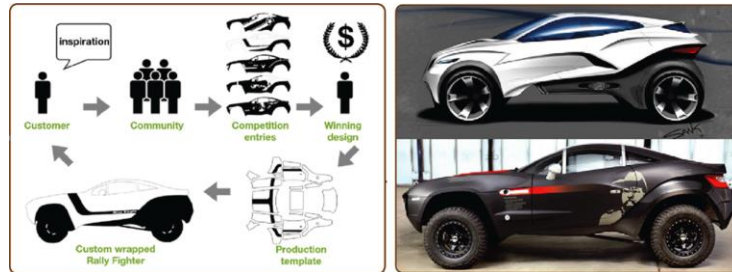


Fig. 1. Crowdsourced vehicle development in Local Motors [5]

Amazon Mechanical Turk, one of the best-known crowdsourcing platforms, offers micro tasks to the crowd, for example tagging images, and translating fragments of text. Top Coder offers low complex software tasks that together respond to user requirements. Similarly, Cad Crowd divides up new product design into smaller jobs. We used data from Cad Crowd experience to train our machine learning algorithm and for testing.

2 Problem in industrial business use cases

The difficulty of deciding whether to participate in a crowdsourcing competition increases with task complexity. Informal, purely manual procedures could be satisfactory only for simple projects. Determining the risk associated with complex projects requires more formal methods and tools, as reported in [1,2,6].

The crowdsourcing literature does not reveal any dominant theories or models [7]. Researchers have employed a wide range of techniques, including case studies, proofs-of-concept, design studies, usability studies, tool developments, and experiments. Findings tend to be ad-hoc and are sometimes conflicted. For instance, researchers differ on whether increasing payments tends to increase crowdsourcing output quality [8]. Crowdsourcing is an emerging research area with scattered knowledge.

In the following chapter we show the state of the art. The fourth chapter describes our development of methods for making decisions easier by making project risk more transparent. The fifth chapter presents results of the experiments. The sixth chapter discusses related works and enhancements to the state of the art through our contribution. The last section summarizes our current work.

3 State of the Art

Determining requirements for a system to assist in assessing the appropriateness of responding to a crowdsourced offer can be built around the goals for the system [7], using design patterns like reflection [8] or logical reasoning approaches such as constraint programming [9]. Both present limitations, such as rigidity and poorness of

expression in the representation of a wide spectrum of requirements, and in the representation of evolving systems [10]. Several authors have tackled the problem from a requirements and knowledge reuse perspective [11], however, ad-hoc and manual practices persist [12]. Crowdsourcing has proved efficient for building a knowledge-base. The author in [13] built an enterprise ontology of business process crowdsourcing. It includes the main business processes, data entities, data attributes, and their hierarchy relationships, which were structured into a lightweight ontology. He made it more robust by adding decision-making relationships and business rules. Still, it lacks quantitative and precise effort estimation. Time and project cost are not explicitly estimated.

If we analyze deeper the existing works, there are two aspects of crowdsourcing that addressed, namely task management and crowd worker management. [14,15,16] developed algorithm for optimizing the work of the crowd. [14] introduced an analytics-based decision support methodology to guide decision making of crowd workers. The results imply that such kind of dynamic decision support for crowd workers is critical towards achieving an increased submission rate and reduced failure rate due to no or poor submissions. [15] developed a Bayesian-generative model to exploit “who knows what” for the workers in the crowdsourcing environment. The model provides a principle and natural framework for capturing the latent skills of workers. [16] formulated three steps for the crowd management. Step one includes important considerations that shape how a crowd is to be constructed. Step two outlines the capabilities firms need to develop to acquire and assimilate resources (e.g., knowledge, labor, funds) from the crowd. Step three outlines key decision areas that executives need to address to effectively engage crowds.

[17, 18, 19] addressed task management in their work. [17] developed a recommendation system that support to consider different types of crowdsourcing platforms, workers, and tasks in the evaluation. It has proven that even simple recommendation systems lead to improvements for most platform users. [18] used game-theoretic models to develop design principles for crowdsourcing contests and answer the questions: what types of tasks should be crowdsourced? Under what circumstances? He found out that when a single task is to be completed, crowdsourcing can lead to higher quality outcomes than directed assignment if the pool of players is diverse, but can lead to suboptimal outcomes when workers have similar abilities. [19] developed a domain knowledge to accurately model a worker’s quality. He examined using knowledge base, e.g., Wikipedia and Freebase, to detect the domains of tasks and workers. He designed an Online Task Assignment algorithm, which judiciously and efficiently assigns tasks to appropriate workers.

Sample of works described above give an illustration on the limitation of the existing solution which concentrate only in two aspects, namely task and crowd management. We see a clear gap here because other aspects are still not covered in their work sufficiently. How about the financial aspect, the project cost. How about the timely effort, the project timeline. Minimally these two aspects are not addressed in their work.

We developed InnoCrowd to provide the necessary tools for the systematic reuse of requirements from project histories. It calculates risks while minimizing the neces-

sary resources. The decision-making process is based on a domain ontology, which describes the system components and the metrics that guide their selection, according to user requirements. This framework also manages the high level requirements and dependencies to guide implementations.

4 Research Methodology

We use “Design Science” as our research methodology. We collect system requirements, define acceptance criteria, consider past knowledge to the related research projects in order to produce research contribution and not routine design.

As mentioned in the second chapter we take real productive data from the crowdsourcing platform CAD Crowd. Typical crowdsourcing task in product development area is given in text as requirement information and also images which represent rough visualization of the required product. In relation to this we have to consider both types of input data, namely text and image. If we consider only text as input data then the crowdsourcing project classification result might be not precise enough. Since we will use artificial intelligent for the classification task then it is very necessary to get as much as possible data necessary for the training and for the testing. Therefore the first requirement in our research methodology is how to fetch all related text and image data from the crowdsourcing platform effectively. The data fetching process needs to be semi automated so that it can be processed by only a single click. The task can be fulfilled by choosing and using an optimum web scrapper.

The second requirement is how to prepare the data before it is consumed by the tool of artificial intelligent. Not all the data gained from the fetching process is relevant for the classification task. If the irrelevant data is kept then it may lead to imprecise data classification. Here we can use text and image editor tools.

The third requirement is to get the optimum algorithm for the classification of text and image. In context of text processing there are several algorithms of Natural Language Processing (NLP). And for image processing there are also many alternatives from machine learning algorithm. This is to be explored in the experiment which will be reported in a specific chapter in this paper.

As the next step is the definition of the acceptance criteria. The most important acceptance criteria is the level of accuracy for the data classification result. We will try to reach the level of accuracy 90%. More than that we will consider time, cost, and quality as parameter for benchmarking with other relevant methods developed by scholars.

A comprehensive literature study is conducted for knowing state of the art and the gap to be fulfilled by our solution. It is shown by mentioning the past knowledge to each related sub aspect in this work.

5 Contribution: InnoCrowd Method

We developed a novel method for the generation of a knowledge database relevant to a specific crowdsourcing provider. Figure 2 shows an overview of InnoCrowd includ-

ing system set up and system usage. The block "System set up" summarizes how we build the Decision Support System (DSS). From definition point of view a DSS represents a point of view on the role of the computer in the management decision making process [14]. Decision support implies the use of computers to:

- Assist managers in their decision processes in semi structured tasks.
- Support, rather than replace, managerial judgement.
- Improve the effectiveness of decision making

It starts with the collection of data from the crowdsourcing provider, which consists of text files (product requirement from customer), images, project histories, and digital models (2D and 3D files of the product to be designed). The second step uses NLP and Deep Learning Neural Network (DLNN) to extract information. The third step embeds product classification in the Neural Network (after NN training and test). The last step establishes relationships between each product sub class and the project specific information relevant for a decision: product specification, related time and cost, and a description of the working team from the crowdsourcing provider's project history.

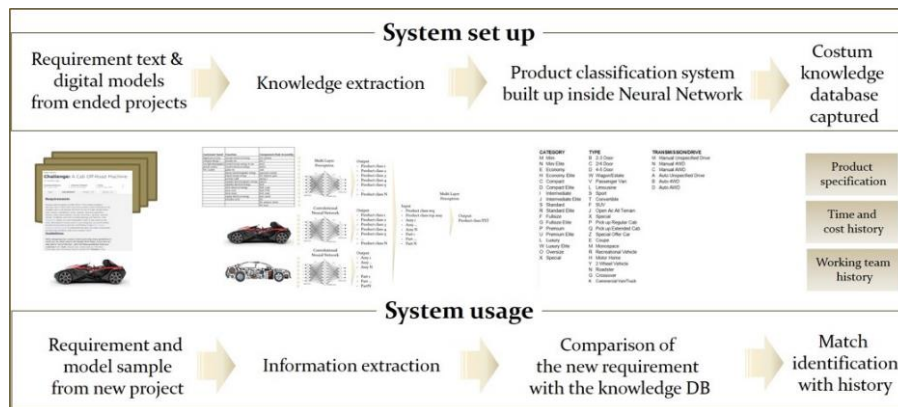


Fig. 2. InnoCrowd overview, system set up and system usage

The block "System usage" shows the way we use the DSS to decide on a new project. Requirements and model samples are input to the system. The second step is the extraction of relevant information from the given dataset (input in the previous step). In the third step, the DLNN classifies the new project by comparing it with others in the knowledgebase. The final step provides project-specific information relevant to a decision: product specification, related time and cost, and working team requirements. The user can then decide whether to apply for the new project.

We use DLNN for the automated recognition of the product design. For each step there is a special neural network trained for recognition. A deep learning neural network is known as the best analyzer / classifier of 2D pictures. It can process the data faster than a 3D model classifier [14].

The requirements are collected as free form text from the source system (crowdsourcing provider platform). Each relevant part of the text will be mapped to

the related domain or application or adaptation layer. Then the Neural Network classifies the proposed product.

6 Experiment

The dataset for the experiment was extracted from The crowdsourcing platform CAD Crowd. More than 900 crowdsourcing projects are listed in the platform. More than 800 projects were finished and therefore could be used as input for our knowledge database in InnoCrowd. The dataset consists of text and images. We extracted the data with a web scraper. At the beginning we processed the text and image data separately in the data mining tool “Orange.” Figure 3 shows the text processing procedure.

The data has to be preprocessed before being used in the classification system. The texts are grouped into several segments. We have found out that some segments are not relevant for the classification task. Therefore, we cut this from the dataset. We give specific group naming to some segment so that it gives more semantic meaning which is relevant for the classification task. Among others the group name “Dimension”, “IT specification and “General specification”. To the image data we also delete some images which are not necessary for the classification task. We concentrate on image analysis from the same perspective. It gives more accuracy when the image classification process is performed.

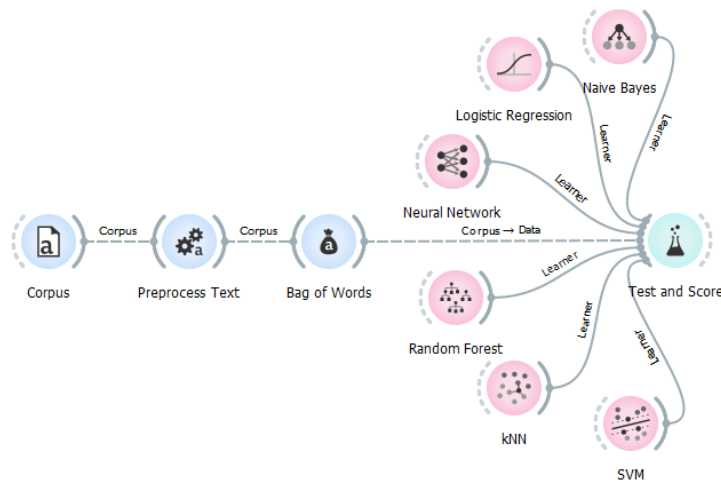


Fig. 3. Process chain for the text analysis

The process is initiated in the first step called “Corpus” which takes the text file in a specific data format. Therefore, it is necessary to convert the text file extracted from the web into this data format. The second step “Preprocess Text” is the step for handling parts of text that are not relevant for the data mining. For an example, commas, dots, low level words such as “and”, etc. The third step “Bag of Words” is the step where the words in each dataset are counted and then converted into numbers in order

to make it calculatable and comparable during the data classification in the next step. The fourth step is the testing and scoring process by using several algorithms. As seen in the picture, six algorithms are evaluated in order to get the best classification result. These algorithms are: Neural Network, Logistic Regression, Naïve Bayes, Random Forest, kNN and Support Vector Machine.

This is the first experiment for comparing the performance of each algorithm. In the figure 4 (left side) we can see the result, the performance comparison between those algorithms. The most important result is given as the parameter called AUC which represents the grade of classification accuracy from each algorithm. Two best performance are achieved by Logistic Regression (Accuracy: 0,657) and Neural Network (Accuracy = 0,653). Therefore, we limit the next experiments by using only these two algorithms.

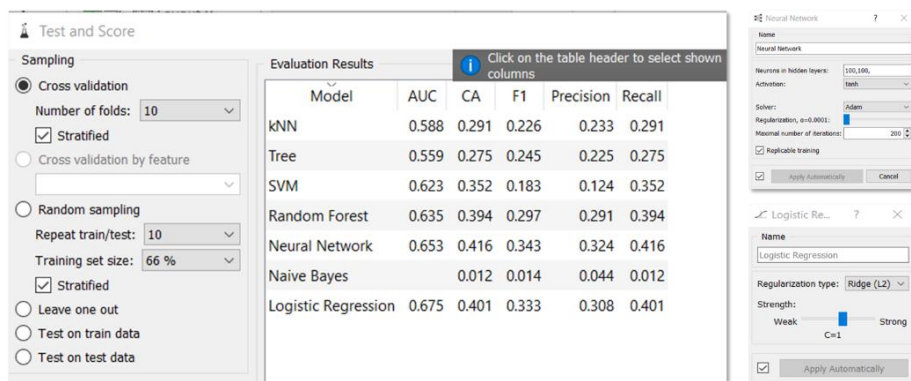


Fig. 4. Result of the text classification with several algorithms

On the right side of Figure 4, we can see parameters of Neural Network and Logistic Regression that can be set to get more accurate results.

We carried out further experiments by manipulating each parameter. Roughly these are parameters that modified during the experiments:

- KNN layer:
 - number of layer: 3, 4 and 5
 - neuron number in layer: 50, 100, 150, 200
- Regularization: 0,01 0,02 0,05 0,1
- Max. nr of iterations: 100, 200, 300, 400

Manipulating the parameters during experiments revealed the best configuration of each algorithm.

These are the best parameters for the neural network:

- Neuron in hidden layer: 100,100
- Activation: tanh
- Solver: Adam
- Regularization: 0,01

- Max. nr of iterations: 200

Final best accuracy (AUC): 0,91

For Logistic Regression these are the parameters:

- Regularization type: Ridge
- Strength: C=1

Final best accuracy (AUC): 0,90

The neural network had the best accuracy, so we adopted it.

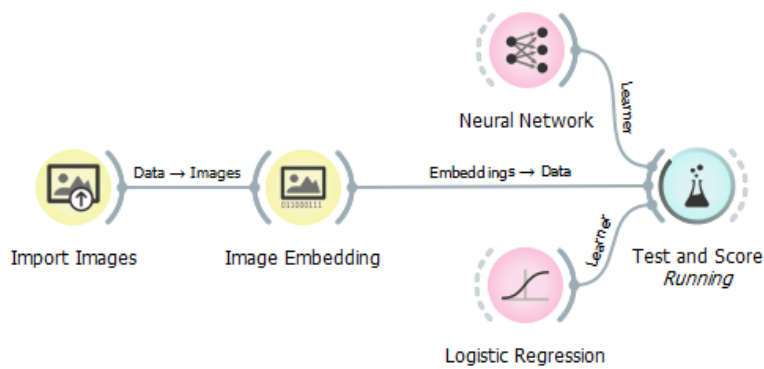


Fig. 5. Process chain for the image analysis

The second experiment was the processing of image files. Figure 5 shows the process chain. It starts with the module for importing the image. The second module embeds images and converts them into numbers according to the pixels inside the image. The third module evaluates each algorithm. This time we compared Neural Network and Logistic Regression. The neural network performed the best image classification, between 95% and 98% (fig. 6%).



Fig. 6. Result of the image classification with deep learning Neural Network

The third experiment is the integration of the text and the image processing (Figure 7). The process chain is integrated so that the system will give the final classification result. We used only the neural network, the best performer in previous experiments.

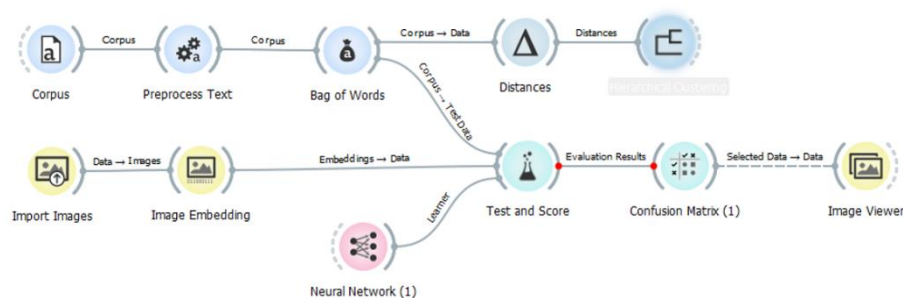


Fig. 7. Process chain for the integrated text and image analysis

We carried out further experiments by manipulating each parameter. Roughly these are parameters that modified during the experiments:

- KNN layer:
 - number of layer: 3, 4 and 5
 - neuron number in layer: 50, 60, 70, 80 90 100
- Regularization: 0,01 0,015 0,02 0,025
- Max. nr of iterations: 200, 230, 250, 270, 300, 330, 350

This experiment found the best result with the following parameters::

Neural network these are the parameters:

- Neuron in hidden layer: 80,90
- Activation: tanh
- Solver: Adam
- Regularization: 0,015
- Max. nr of iterations: 330

Final best accuracy (AUC): 0,90

This 90 % accuracy is sufficient to significantly supporting decisions on product specification / requirement, project time and cost, and the working / crowd team.

7 Comparison with related work

As described in the chapter “State of the art” there are related works which can be seen as a decision support system for crowdsourcing. But they are focused on limited aspects, namely task management and crowd management. They don’t take into account other aspects such as project cost and project timeline. By considering this limitation we compare our solution with the solution given by [13] because it covers the

same aspects as InnoCrowd does. Figure 8 below shows the conceptual model of the solution developed by [13].

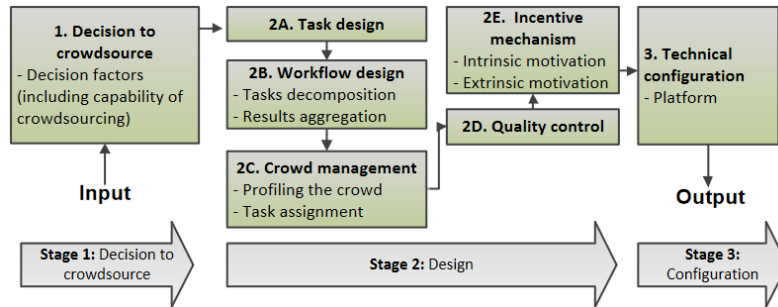


Fig. 8. Conceptual model developed in [13]

And for a comparison in the high level we show InnoCrowd’s conceptual model in the figure 9 below. The same aspect is that both concepts is that they are to be used as tool for a decision support in crowdsourcing project. In practice both solutions deal with the aspects task / product specification, crowd management, cost management, as well crowd management. Solution of [13] takes quality control into account. Meanwhile InnoCrowd seals with time management.

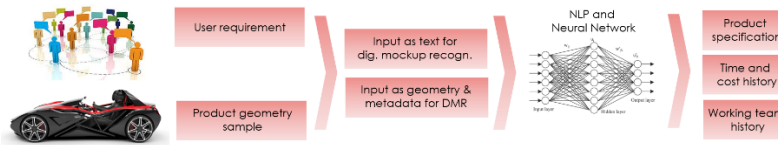


Fig. 9. InnoCrowd’s conceptual model

We found out that the biggest difference is related with the usage of the solution. In the solution of [13] everything is to be done step by step manually. Meanwhile InnoCrowd works automatically by giving input in form of initial product specification and initial possible product form as image file. The relevant output for decision is given automatically by the algorithm of InnoCrowd. This gives more advantage in term of easier usage and time necessary for making the decision.

Furthermore, we compare the both solution on three main aspects, namely time, cost and quality. The following is our estimation. InnoCrowd get 6 times “very good” and 1 time “sufficient”. Meanwhile solution of [13] get 2 times “good” and 5 times “acceptable”. The comparison of average InnoCrowd vs Solution of [13] is 3,6 vs 2,3. It means InnoCrowd performs better than the solution of [13].

DSS tool	InnoCrowd	Score	Solution [13]	Score
Time				
Time to market	Only digital input necessary	++++	Need consulting	+++
Lead time	Automated evaluation process	++++	Through the questionair	++
Cost				
Prediction of project cost	Based on facts from DB	++++	Qualitative prediction	++
Tool investition cost	Need customizing on classes	+	Customizing only in high level	+++
Quality				
Quality of crowd workers	Very detailed record	++++	Qualitative evaluation	++
Quality of project mngmt	Transparant process	++++	Qualitative evaluation	++
Quality of input vs output	Enable improvement	++++	Only in high level	++
++++ = very good +++ = good ++ = acceptable + = sufficient				

Fig. 10. Solution comparison

8 Conclusion and future work

Methods for handling complex product is developed and embedded in the method for handling innovation creation process. As the whole it builds InnoCrowd. We verify the solution concept by using dataset from crowdsourcing platform CAD Crowd. The verification result on product requirement analysis (as free text file) gives a good classification result, namely in the accuracy level between 90% and 91%. Meanwhile the verification result on digital model analysis gives even a higher accuracy, between 95% and 98%. The overall performance when the text and image analysis are integrated is 90%. This result gives a clear view on the usability of InnoCrowd.

The comparison of InnoCrowd with other solution gives a clear result, that InnoCrowd performs better. The performance of InnoCrowd is “good”, meanwhile the other solution is between “acceptable” and “good”.

In relation with the usage of InnoCrowd as a decision support system then it is technically done by providing the user the information related to the previous project matched with the planned project. The user will have access to the information of the previous project cost, the time needed to finish the project and also the crowd workers who had work for similar task. These information give a clear transparency on the project risk, so that InnoCrowd can be used as an answer to the management when they want to crowdsource a product development project.

So far, we have developed the basic functionality of InnoCrowd. What we still have to develop in the future is the user usability so that the process chain starting from data fetching, data preprocessing, data classification and result presentation can be done in an integrated environment. And we still need to optimize the processing time as well as the accuracy of the algorithm.

References

1. Buhse, W., Reppesgard, L., Lessmann, U.: Der Case Local Motors: Co-Creation and Collaboration in der Automotive-Industrie. Forschungsstelle für Customer Insight, St. Gallen (2011).
2. Chatterjee, S., Khandekar, P., Kumar, B.: Reimagining enterprise innovation through crowdsourcing. White Paper TCS (2014).
3. Howe, J.: Crowdsourcing: Why the power of the crowd is driving the future of business. Currency, Washington (2008).
4. Eagle, N.: txteagle: Mobile Crowdsourcing. pp. 447–456. Springer, Berlin, Heidelberg (2009).
5. Local Motors Homepage. <https://localmotors.com/>. last accessed 2019/11/21.
6. Schenk, E., Guittard, C.: Crowdsourcing: What can be Outsourced to the Crowd, and Why? In: Workshop on Open Source Innovation, Strasbourg (2009).
7. Allen, R., Fowler, F.: The pocket Oxford dictionary of current English. Oxford University Press (1976).
8. Achtaich, A., Souissi, N., Mazo, R., Roudies, O., Salinesi, C.: A DSPL Design Framework for SASs: A Smart Building Example. EAI Endorsed Trans. Smart Cities, vol. 2, (2018).
9. Pohl, K., Böckle, G., Van Der Linden, F.: Software Product Line Engineering. Foundations, Principles, and Techniques, vol. 49, no. 12. Springer, Berlin (2005).
10. Whittle, J., Sawyer, P., Bencomo, N., Cheng, B., Bruel, J.: RELAX: a language to address uncertainty in self-adaptive systems requirement. Requirement Engineering, vol. 15, no. 2, pp. 177–196, Zurich (2010).
11. Mongiello, M., Boggia, G., Di Sciascio, E.: ReIOS: Reflective Architecting in the Internet of Objects. In: Proceeding of 4th Int. Conf. Model. Eng. Softw. Dev., pp. 384–389 (2016).
12. Sawyer, P., Mazo, R., Diaz, D., Salinesi, C., Hughes, D.: Constraint Programming as a Means to Manage Configurations in Self-Adaptive Systems. Spec. Issue IEEE Comput. J. Dynamic Softw. Prod. Lines. vol. 45, pp. 56–63 (2012).
13. Thuan, N., Antunes, P., Johnstone, D., Xuan, H., Hoang, N.: Building an Enterprise Ontology of Business Process Crowdsourcing: A Design Science Approach. In: PACIS 2015 Proceedings, vol 112 (2015).
14. Yang, Y., Karim, M., Saremi, R., Ruhe, G.: Who Should Take This Task? – Dynamic Decision Support for Crowd Workers. In: Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 1–10 (2016).
15. Zhao, Z., Wei, f., Zhou, M., Chen, W., Ng, W.: Crowd-Selection Query Processing in Crowdsourcing Databases: A Task-Driven Approach. In: EDBT Computer Science (2015).
16. Prpic, J., Shukla, P., Kietzmann, J., McCarthy, I.: How to Work a Crowd: Developing Crowd Capital through Crowdsourcing. In: Business Horizons, pp 77-85 (2015).
17. Borchert, K., Hirth, M., Schnitzer, S., Rensing, C.: Impact of Task Recommendation Systems in Crowdsourcing Platforms. In: Proceedings of the First International Workshop on Crowdsourcing and Data Mining (2012).
18. Renade, G. Varshney, L.: To Crowdsourc or not to Crowdsourc? In: HCOMP@AAAI (2012).
19. Zheng, Y., Li, G., Cheng, R.: DOCS: DomainAware Crowdsourcing System. In: Proceedings of the VLDB Endowment, pp 361-372 (2016).