COMPUTER-AIDED DESIGN & APPLICATIONS, 2017
VOL. 14, NO. 5, 610–621
http://dx.doi.org/10.1080/16864360.2016.1273578

Methods for determining the optimal number of simultaneous contributors for multi-user CAD parts

Brett Stone, John L. Salmon, Ammon I. Hepworth, Edward Red, Matthew Killian, Alex La, Ariana Pedersen and Talmage Jones

Brigham Young University, USA

ABSTRACT
The development of multi-user CAD (MUCAD) tools has opened up exciting new opportunities and applications. The capability for multiple users to simultaneously model and design a CAD part has far-reaching potential. However, many basic questions remains unanswered, such as how many users should work together on a given part. This research proposes and develops a set of methods to determine the optimal number of users for a given part within a MUCAD environment, based on the characteristics of the part itself. Two candidate models are evaluated with a set of 60 experiments with design teams composed of different numbers of users. The models show modest correlations with the test data while more-refined models are explored to improve predictive power. On the other hand, highly significant correlations between the ability to predict completion time and multi-user team size were identified in the experimental data. Observations regarding the speed and quality of MUCAD teams are also made with future areas of research suggested.

KEYWORDS
Multi-user CAD; collaborative design; concurrent engineering

1. Introduction

Multi-user CAD (MUCAD) enables multiple users to simultaneously contribute to a CAD part or assembly. All users' contributions appear in real time, providing a collaborative environment where all users visualize changes to a CAD model as they occur. This type of environment allows for design tasks in CAD parts, which have been traditionally limited to a serial workflow, to be performed in parallel. Various research efforts related to collaborative CAD have been conducted since the late 1990's [2, 3, 5–8, 10, 12, 14, 16–22, 24, 28–31, 33, 36, 39].

Previous research shows that multiple users simultaneously working on a CAD part in parallel can significantly decrease the time it takes to complete the part [15, 16]. As more users are added, the time to complete the part tends to decrease. However, there is a point at which adding more users no longer decreases the time to completion, and in many cases it increases the time [16]. Therefore, there is an optimal point at which either increasing or reducing the number of users increases the design time. This point, which is specific for each CAD part, is what we call the optimal number of simultaneous contributors.

Although previous research suggests that an optimal number of simultaneous contributors for a specific CAD part exists, no one has attempted to determine the factors that influence this number. Furthermore, no one has yet determined any method to adequately predict this optimal number. In this paper, we present factors related to the part itself that appear to influence the optimal number of simultaneous contributors in a CAD part. We also present two methods to determine or predict this value. These methods use a taxonomy and a dependency tree structure to classify the part and estimate the optimal number of users. We then present experimental results to determine empirically which of the two methods most accurately predicts the optimal number of multi-user team members.

To support this prediction process, a taxonomy is developed that classifies parts according to the number of features and structure of branching dependencies. Once the classification is complete, the results are used to develop two predictive models of the optimal number of users for a given CAD part. These models are then validated through experiments to show the accuracy of each model.

2. Background

2.1. Taxonomy

A taxonomy is a structured way of grouping or distinguishing a large and diverse set of specimens, which is

CONTACT Brett Stone brett_stone@me.com

useful in many fields such as biology [9], astrophysics [35], or even systems engineering [13]. For example, biological taxonomy, with its designations of kingdom, phylum, class, order, family, genus, and species, allows us to classify living things in a neatly structured fashion. Todd *et al.* provide a similar method of classification for manufacturing processes, beginning with whether a process is shaping or non-shaping, and progressing all the way to specific processes such as Ion Beam Cutting and Swaging [37]. These taxonomies serve significant practical purposes beyond simply organizing objects. It is easy to see that much of biological research would be impossible without a standardized way of understanding how different species are related. Similarly, an organized way of thinking about manufacturing processes allows designers and manufacturers to systematically consider alternatives for making planned products a reality.

In order to identify the optimal number of multi-user teammates for a given part, a structured method of classification must be established. Just as living creatures and manufacturing methods can be classified and organized using a taxonomy, models of physical parts that are created in CAD can also be organized using a similar scheme. Our proposed taxonomic method is presented in Fig. 1. Starting at the top with "All Parts," the first level of distinction includes determining whether the part has a single feature or multiple features. In this research, a feature is defined as any of the geometry-creating methods in a modern CAD tool such as Siemens NX or Dassault CATIA. Examples include "Extrude" in NX or "Pad" in CATIA, "hole," "pattern," or "loft" features. Sketches, by themselves, are not considered features in this method. These features and their dependencies are defining characteristics of modern feature-based CAD systems [32].

If a part only has a single feature, it is considered unsuitable for MUCAD. This is because the feature is the atomic unit, meaning only one user can edit a feature at a time [17]. If, at some future period, a MUCAD system alters that paradigm and adds capability for MU sketching, this taxonomy would change (see "Sketch Domain" on the far left of Fig. 1). The other option at this classification level is for a part to have multiple features.

Level two of the taxonomy requires identification of whether the part has linear or branching dependencies. Dependencies occur when one feature in a part depends on another feature in some way. For example, a hole may depend on a surface or a solid on which it is based. If multiple features depend on a single parent feature, these children features are said to branch. An example of a part with purely linear dependencies is shown in Fig. 2. In contrast, Fig. 3 and Fig. 4 respectively show a piston head and an automotive fluid reservoir with their feature dependency trees. The automotive fluid reservoir tree demonstrates a relatively high number of branching dependencies.

A visual representation of a part's dependencies often bears resemblance to the structure of a tree. How complex the tree structure of any given part is, from mostly linear to complex and bushy, is the third and final level of our taxonomy.

Using this method, which is, as far as we are aware, unique in the field, we have classified a sample of more than 100 parts. To ensure a minimal breadth of part variety, we selected parts from among nine different manufacturing methods, such as blow molding, sheet

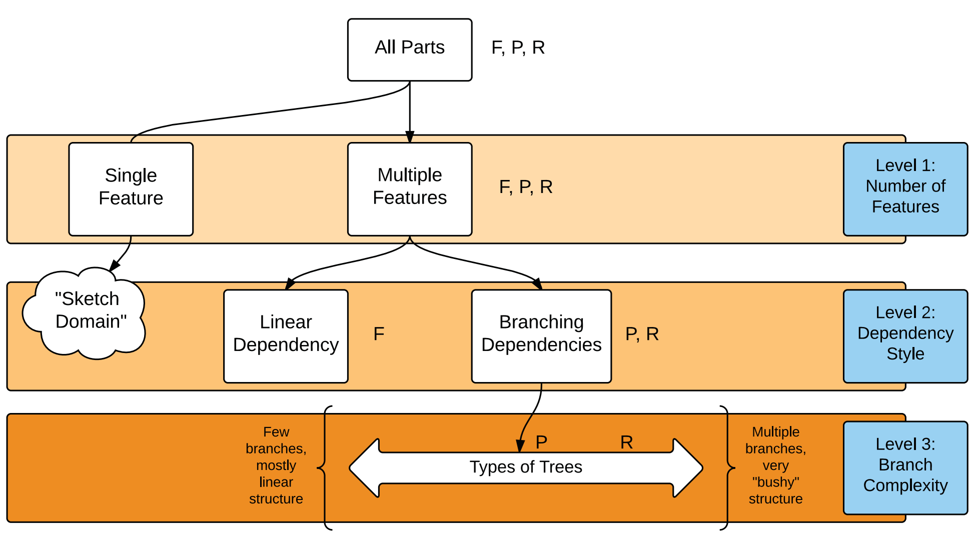

Figure 1. Proposed taxonomy for classifying parts for multi-user CAD. Note the "F" for the fan blade set, "P" for the piston head, and "R" for the automotive fluid reservoir in the subsequent examples.

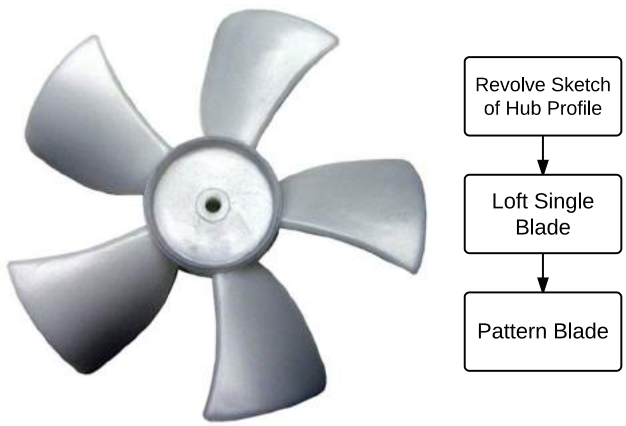

Figure 2. Model of a fan blade set and its feature dependency tree, which is completely linear.

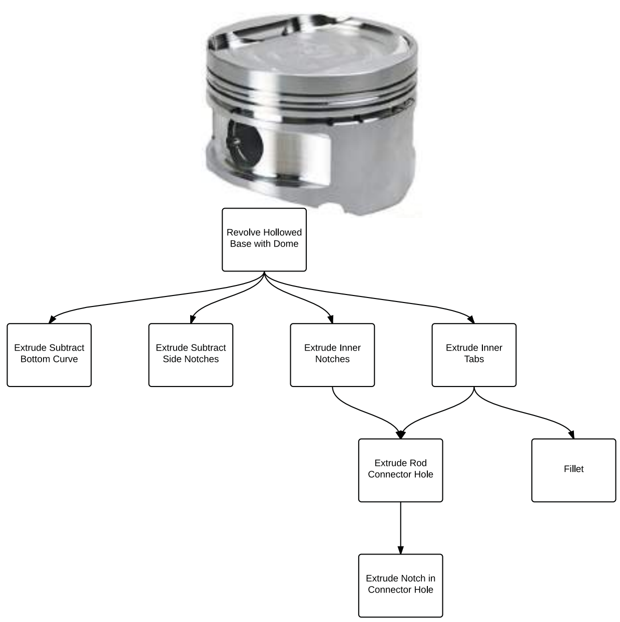

Figure 3. Model of a piston head and its branching feature dependency tree.

metal manufacturing, forging, and 3D printing. Next, a researcher classified each part using the taxonomy (aided by standardized classification forms) and created the feature dependency tree for each applicable part. A second and sometimes third researcher verified the classification and tree structure to confirm the part's taxonomic definition. This collective set of classifications could then be used to develop predictive models to estimate the optimal number of users for a given CAD part.

Currently, CAD part classification is a manual process requiring researchers to think extensively about how they would model a part and then check each other's proposed structure. In the future, an automated tool that leverages machine learning could be developed to automate this process. However, such a tool would be required to handle the ambiguity of multiple options for how to model a given piece of geometry.

2.2. Predictive models

To accurately predict the optimal number of users for a given CAD part, we proposed a set of models, an overall methodology, and hypotheses. Since this research was the first step in filling an apparent gap in MUCAD implementation, we have endeavored to follow a classic pattern of increasing fidelity from simple to more complex models of prediction. This is not unlike various methods for aircraft design and aerodynamics where lower-order models are initially applied to obtain first-order approximations, followed by more accurate and sophisticated methods [1].

For example, during conceptual design, an aircraft designer may simply apply Bernoulli's incompressible flow equations to extract simple estimates of drag polars from a point design. During preliminary and detailed design, one may invoke Euler's and Navier-Stokes equations, which can include compressibility and viscosity, respectively, resulting in more accurate predictions for the aircraft's various aerodynamic performance characteristics. Finally, aircraft models are tested in wind tunnels validating the models' predictive capability for a particular geometry.

In the context of multi-user CAD, we proposed that the lowest-order model to predict the optimal number of multi-users working concurrently in a CAD part is a simple function of the number of features within that part. Selecting a part from the sample of parts classified using the taxonomy previously described, the number of features in a part is quickly calculated and the optimal number of users can be extracted from a linear regression model. Under this model, we hypothesized that for parts with few features (i.e., less than 10) no significant benefits will be obtained from more than one concurrent user. Therefore, a single user would be optimal. The additional overhead of multi-user environments and the necessary communication requirements may outweigh the benefits with so few features in a part. However, we hypothesized that with 10 or more features, the potential for multiple users working simultaneously in the same CAD part will become increasingly attractive. When these parts are modeled by multiple users, the team can experience reduced modeling time, reduced or accelerated error checking, and enabling of earlier efforts by analysts and subject-matter experts down-stream.

It is important to note that while there are many ways of defining an "optimal" number of users, we

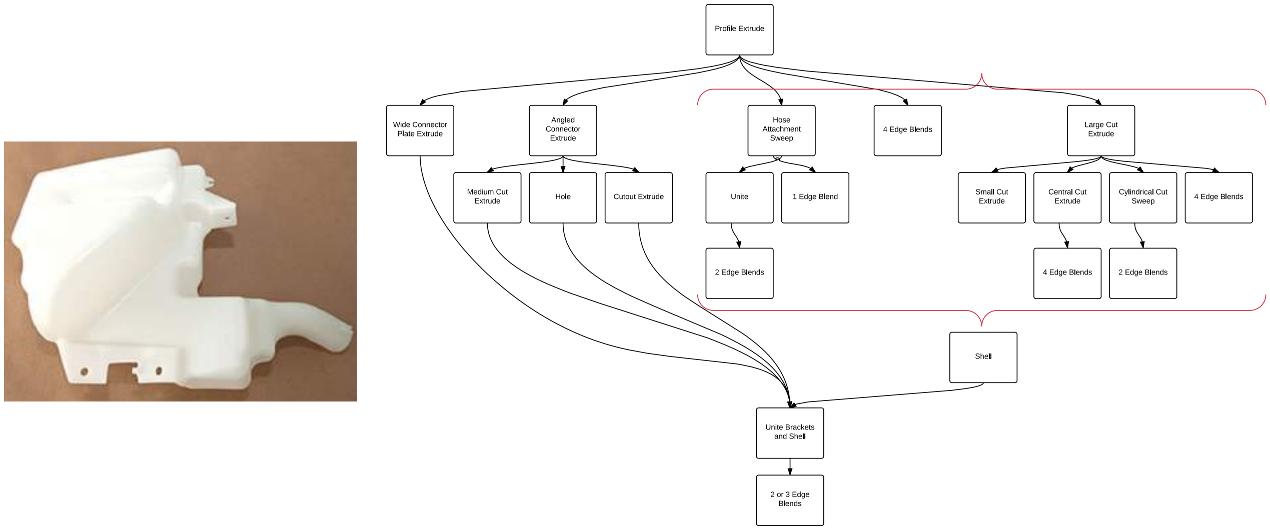

Figure 4. An automotive fluid reservoir and its feature dependency tree.

have limited our research to consider the optimal number to be that which minimizes the "calendar time" or "actual time" from beginning to completing the model (as opposed to total man-hours). Given Brooks' and other's research regarding increasing overhead costs with increasing numbers of teammates, we do not consider man-hours in this investigation [4].

A more sophisticated second-order model would take into consideration not just the number of features but the features' locations and orientations with respect to the feature dependency tree. We hypothesized that a tree with little to no branching, even with many features, will not allow multiple users to concurrently model a part. On the other hand, a part with significant branching suggests potential for many simultaneous users. This model uses the feature dependency trees generated during the taxonomic classification to count the number of features within a particular tier or level of each tree's hierarchy. Then, a weighted sum across all branches and levels is performed to predict an optimal number of multiple users. We hypothesized that this model would more accurately predict the optimal number of users for a given part than the first-order model.

A third and more complex model would make fewer assumptions about the feature dependency tree and would consider the time and complexity associated with modeling each feature with an evaluation of the interfaces between them. Additional factors could be included in this model that would drive the optimal number of users, including ideas from graph theory such as connectivity, path lengths, and cycles [38]. Since this third type of model requires information beyond what was gathered in the taxonomic classification of the part sample described, it forms the thrust of future research efforts whereas this paper will address the first two models described. Finally, efforts to validate these models were performed through 60 design tests with teams of different numbers of users.

Brooks addresses attributes of teams of various sizes and task types [4]. For teams working on tasks that require communication, Brooks argues that adding more members to the team does not shorten the time to task completion in a linear fashion [4]. Instead, he shows that each time a new teammate is added, the marginal improvement decreases. For tasks with more complex interrelationships, such as the software development projects he studied, a point comes at which adding team members begins to negatively affect the time to completion. Hepworth *et al.* demonstrated similar results in a MUCAD environment [15].

3. Methods

The first and second models were investigated empirically by measuring the time required to model 13 "small" parts (20 or fewer features) and two "larger" parts (more than 20 features). Each part was modeled with one, two, three, and four multi-user team members. Users were never allowed to model the same part twice to control for learning and reduce the bias in observed quality and modeling time. Because of the number of models that had to be created, 26 volunteers from the Brigham Young University (BYU) CAD Lab and other student-volunteers with significant NX CAD experience modeled the parts. Students were mostly undergraduate mechanical engineering majors.

In order to calibrate and compensate for the large variety of modeling skill levels, each user took a modeling speed test. This test, completed individually by each volunteer, required the examinee to model a basic part. Trained proctors verified satisfactory completion of the part and recorded the amount of time taken. Equation (3.1) shows how a correction factor is calculated to normalize the individual skill level for all participants,

$$R_c = \frac{t_{avg.}}{t_{user}} \tag{3.1}$$

where R_c is the correction factor, $t_{avg.}$ represents the average of all of the examinees' speed tests, and t_{user} represents the time for each individual's speed test.

Another potentially confounding factor that we attempted to mitigate is the beta status of the NXConnect MU software. Software bugs did occasionally cause individuals to spend time waiting or restarting the program. To compensate for this, video recording of each user's screen was examined after each model was completed and the time a user spent waiting due to bugs was subtracted from his or her total modeling time to produce the active modeling time for each user. Each user's active modeling time was then added to the other members of his or her team and averaged to produce the corrected calendar time for each modeling effort described in Eqn. (3.2).

$$T_c = \frac{R_{c,min} \sum_1^k (t_m - t_{bugs})}{k} \tag{3.2}$$

T_C is the corrected calendar time for each model, k represents the total number of users on the team, t_m is the raw modeling time for each user, and t_{bugs} represents the time a given user spent waiting because of software bugs.

Steiner, Page, and Moynihan state that the performance of teams whose members are highly interdependent (those performing "conjunctive" tasks) depends most on the team's weakest member, or the team member with the lowest rating in the relevant skill [23, 26, 34]. In the case of the MUCAD teams in this study, R_C was used to indicate team member skill. In other words, the lowest R_C, or the $R_{C,min}$, was applied to weight each team's T_C. This assumption was supported based on the observations of MUCAD teams, which demand high levels of interdependence: they must agree on how to orient the part, decide who will model which sections, and depend on each other's sketches and features to create their own.

4. Results

Results of the part-modeling experiments can be seen in Tab. 1, arranged in order from smallest number of features per part to the highest. Some parts varied significantly from the expected overall trends, but many matched well.

Comparing the T_C of the 13 small parts to the number of users per team, one can observe similar results to those found by Hepworth *et al.* and Brooks [4, 15]. Fig. 5 shows the time to complete each part compared with the number of users on each team, as well as a line connecting the mean time in each category with 95% confidence intervals.

Given the data's non-normality and potential for inequality of variances, a non-parametric, Wilcoxon each pair comparison was used to compare the means of each group. Mean values were 1 User: 27.1 minutes, 2 Users: 20.5 minutes, 3 Users: 15.3 minutes, and 4 Users: 13.5 minutes. The difference between the 4-User teams and the 1-User teams was statistically significant ($p = 0.04$). The next closest difference to statistical significance was the difference between the 3-User and the 1-User teams ($p = 0.06$).

The optimal number of MUCAD teammates was determined for each part by identifying the point at

Table 1. Time completion results of the part modeling experiments.

Part Name	Total # of Features	Avg. # of Features/ row	*Tc* 1-User (min)	*Tc* 2-User (min)	*Tc* 3-User (min)	*Tc* 4-User (min)
Sintered Part	3	1.5	9.89	8.29	6.58	9.03
Cup	4	1	1.82	3.54	11.12	5.80
Ball Valve	4	1.33	2.20	6.07	3.43	2.51
3D Printed Hinge	7	1.75	8.83	16.91	11.95	7.93
Tablet Mount Arm	7	2.33	34.76	18.02	13.36	8.40
Chocolate Container	9	2.25	27.49	39.78	12.07	12.34
Mining Machinery	10	1.43	28.91	16.93	13.75	17.91
QuadCopter Arm	10	2.5	35.71	37.44	20.17	12.94
Fan Housing	13	6.5	27.17	22.16	12.91	13.56
Kitchen Sink	15	3	64.59	12.97	25.27	19.44
Car Door Panel	17	2.83	39.59	32.17	20.87	18.47
Gear Pump Housing	17	4.25	40.53	35.96	26.95	23.98
Pump Casing	19	3.16	30.38	16.55	21.37	22.71
Airplane Rib*	32	10.67	18.59	28.62	26.01	24.03
Tray*	59	5.9	25.08	27.03	25.28	31.93

*Included as case studies

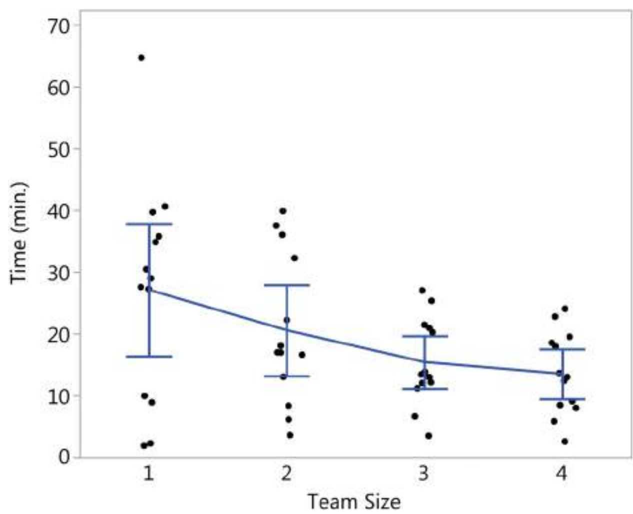

Figure 5. The average amount of time to complete a model decreases as the number of teammates increases; improvement or reduction in time begins to level off by four teammates.

which adding more users no longer saved time, or, in the case that the classic Brooks pattern was not displayed, the number of users correlated with the shortest time to completion. A first-order linear regressed model of the optimal number of teammates was determined from the number of features per part and the average number of features per row within the part's feature dependency tree. These curves are shown in black on both the left and right hand side of Fig. 6. The linear relationships do demonstrate a positive correlation, as expected, but both are quite weak statistically with a small R^2 value of just 0.065 when the model is based on the number of features, while the model for the average number of features per row was only slightly better at 0.076.

However, since the true model would be constrained to have "1" as the optimal number of users when the total number of features equals one, and the model should asymptotically approach a maximum number of users for practical reasons (i.e., the overhead of integrating a large number of modelers overpowers the benefits), various non-linear models were considered and applied to the data set. A similar argument is made for the second type of model using the average number of features per row. One such approximation, based on the Michaelis–Menten equation [25], offers a better model to regress the experimental data and provide a prediction for parts with numbers of features up to 20. The Michaelis–Menten models, shown with the red lines in Fig. 6, offer 2.37 and 2.72 times more predictive power with R^2 values of 0.206 and 0.153, respectively. Not only do these models offer a more accurate prediction for the optimal number of users, but they are also characterized by a more feasible non-linear trajectory consistent with literature on team or group size and performance [11, 27].

Another way of looking at the ability of the proposed models' predictive power is to consider time to completion vs. feature count (or average number of features per row) by size of team. The results of this analysis are shown in Fig. 7.

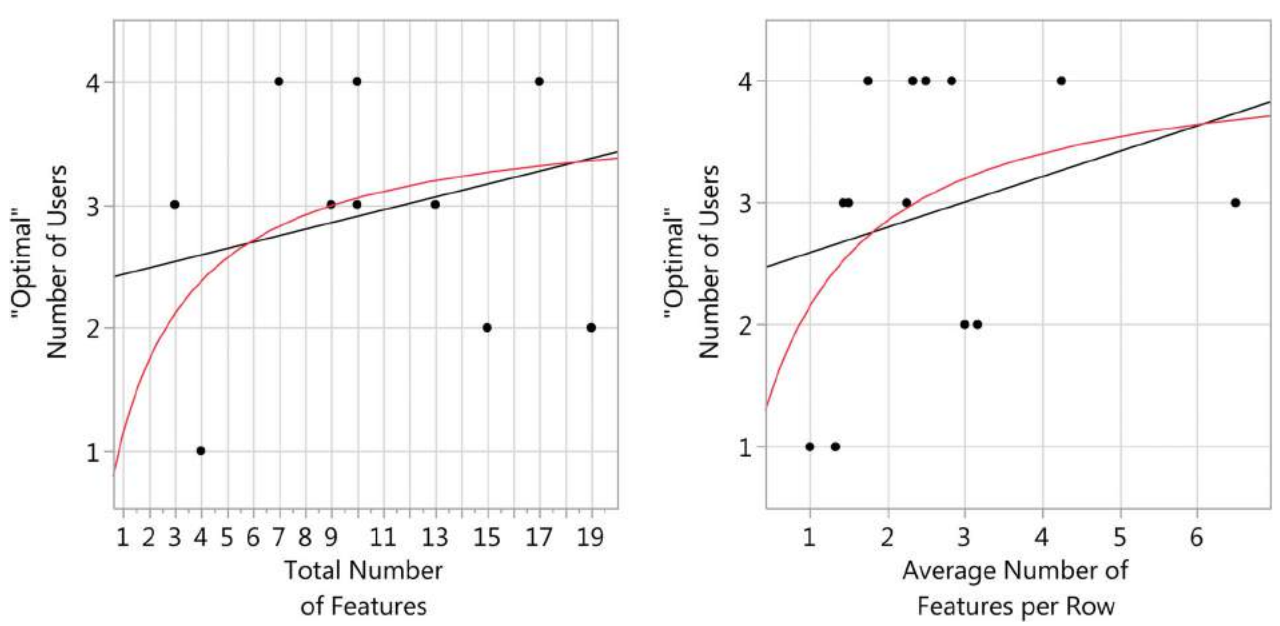

Figure 6. The optimal number of teammates by (a) the total number of features and (b) by the average number of features per row (first-order linear regression (red), Michaelis–Menten model (black)).

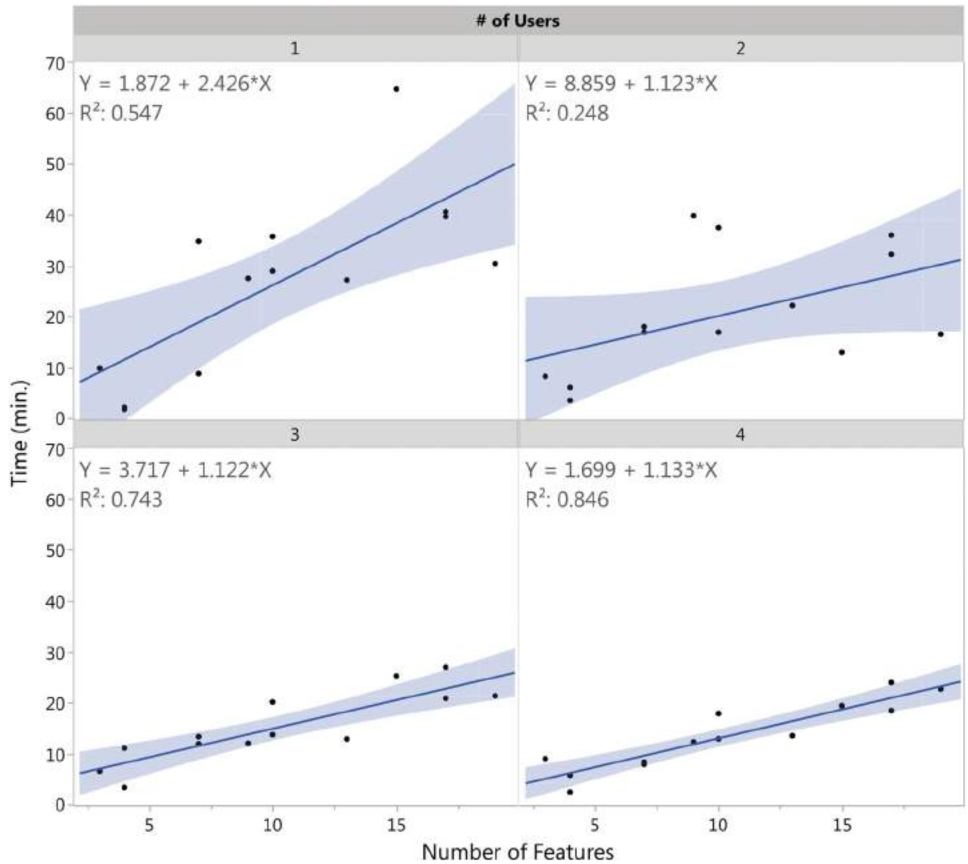

Figure 7. Time to complete each part vs. the number of features by the size of each team with 95% confidence intervals, linear regression equations, and R^2 values.

As demonstrated in Fig. 7, the number of features shows a positive correlation with time to completion. These correlations were statistically significant, with p-values less than 0.05 in all cases except for the 2-user teams ($p = 0.08$). It is also interesting to note the increase in R^2 values as the size of the team increases. Statistical results for comparing completion time with average number of features per row yielded similar but weaker results, with p-values ranging between 0.08 and 0.16.

5. Discussion

Results of our analysis show that the proposed models using the number of features and the average number of features per row do correlate with the optimal number of users, although weakly. It is likely that more repetitions of the same parts, and by larger sizes of teams (i.e., greater than four), will be necessary to fully validate these models statistically. Furthermore, the parts used were all primarily simple with respect to the total number of part features (i.e., less than 20). Team behavior and performance may be different with more complicated parts and offer more stable effects.

However, the theory that MU teams may allow more accurate prediction of time to completion for a model of a given size was observed and found to be statistically significant in most cases. This finding matches our observations in other studies and experiences. One explanation for this phenomenon may be that teammates tend to complement each other's skill sets so that when one user is less knowledgeable or skilled, other users can provide the needed ability or will naturally compensate out of necessity. For example, clear instances were observed where MU teammates learned from each other's modeling techniques during the experiments. The following sections described some of the findings from these observations.

5.1. Trunking

One trend we observed during the experiments was the types of strategies teams employed to try to deal with "trunks." Following the idea that a part's dependency branching is similar to the structure of a tree, the first element of the tree structure as we imagined it (see Fig. 3 and Fig. 4 for examples) was a single feature created by one user and gives the rest of the MU team the context it needs to model other features. In theory, while many different features of a part could be chosen as the trunk, the classification process identified features which seemed to be the most likely chosen as the trunk. We assumed that most teams, for the sake of avoiding confusion about how the part was oriented, would choose to follow a "single trunk" strategy. The obvious drawback to this strategy is that the rest of the team must wait while one person creates the trunk.

Test volunteers were not informed or instructed how to organize their modeling efforts, and we observed that most teams did follow a single-trunk strategy anyway. However, we also observed several enterprising teams attempt to improve on the single-trunk method for MUCAD modeling. Some teams would attempt to "shrink" the trunk of a part by having one user complete a very simple version of a sketch of the trunk feature. The user who sketched this initial feature would then quickly exit the sketch to allow the other team members to view it. In many cases, the sketch was not completely constrained or even dimensioned correctly, but it sufficiently communicated the general size, shape, and orientation of the feature well enough for the other team members to begin creating their features. Often, the initial user who modeled the trunk would return to refine it later on.

One example of this strategy can be seen in Fig. 8, where, after discussing their strategy, one user created a very rough, incomplete sketch. He then exited the sketch so it would be committed to the server and his teammate could see it. Then, he reentered the sketch to refine it while his teammate began working on other portions of the model.

Other teams attempted to "multi-trunk" their parts. After attempting to explain the general orientation of the part's features to each other, two or more team members would simultaneously sketch and create their features. The risk taken by teams that attempted to multi-trunk, of course, was that once completed, their features would sometimes not properly relate to each other. Sometimes this took little effort to correct, while other times it meant completely redoing features, which increased confusion among teammates. In most cases, multi-trunking required much more effective coordination before initiating modeling activities. Future research should investigate this tactic, its potential, and its implications.

5.2. Quality difference for simple parts

It was initially predicted that small, simple parts, or those we classified with 10 or fewer features, would see little benefit from being modeled by a MU team. However, we observed that in some cases, while MU teams completed their simple parts slower than single-user teams, they also greatly increased the quality of the part.

The Cup is one example of when larger teams took longer to model the part than the single-user team. Despite efforts to control for quality, MU teams often insisted on including a higher level of detail in their part as shown in Fig. 9. From the beginning of their modeling efforts, some MU teams seemed to have a sense of obligation to involve as many of their users as much of the time as possible. This led to teams altering their modeling strategy to make more, simpler features and/or consider strategies such as subtractive modeling to allow more users to contribute to the model simultaneously. A ratio of features added per minute of modeling time shows that even though the two- and three-user teams were much slower than the single-user team, the features-to-minute ratios of two of the MU teams were higher than the single-user team.

5.3. Case studies

Although many parts were expected to be suitable for multi-user teams and the experiments confirmed our predictions, a few parts were surprisingly not conducive to MUCAD. Those that met our expectations included the Car Door Panel, Fan Housing, Gear Pump Housing,

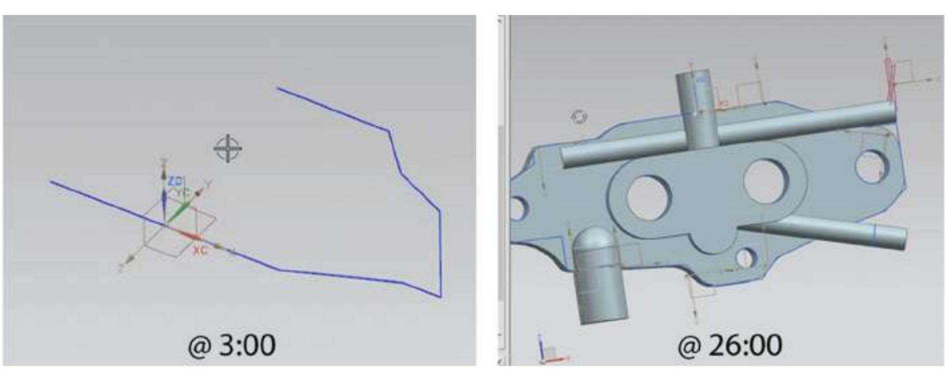

Figure 8. Example of (a) rough-trunking an initial sketch, and (b) the more fully developed model.

Figure 9. The level of detail included in the model of the cup generally increased with the number of users.

Mining Machinery, Pump Casing, QuadCopter Arm, and Tablet Mount Arm. The Tray and Airplane Rib, two parts with the highest number of features (more than twice the average number of features of all the others), were less appropriate for MUCAD based on the results of the experiments. Completion times for each team size for each of these nine aforementioned parts are shown in Fig. 10.

Several parts, such as the Fan Housing and Pump Casing, appear to demonstrate Brooks style curves. Others, such as the Car Door Panel and Tablet Mount Arm, could also potentially be Brooks curves, but with their optimal points at a higher number of team members than tested. The Tray and Airplane Rib do not match these trends. In fact, the Tray's completion time remains relatively flat for team sizes of one to three users, and finally increases with four users. This is opposite of our initial predictions that the Tray would be very suitable for MU modeling considering its large and widely branching tree structure.

After reviewing the video and audio recordings of the Tray's teams, we discovered a large difference in way the single user modeled a few important portions

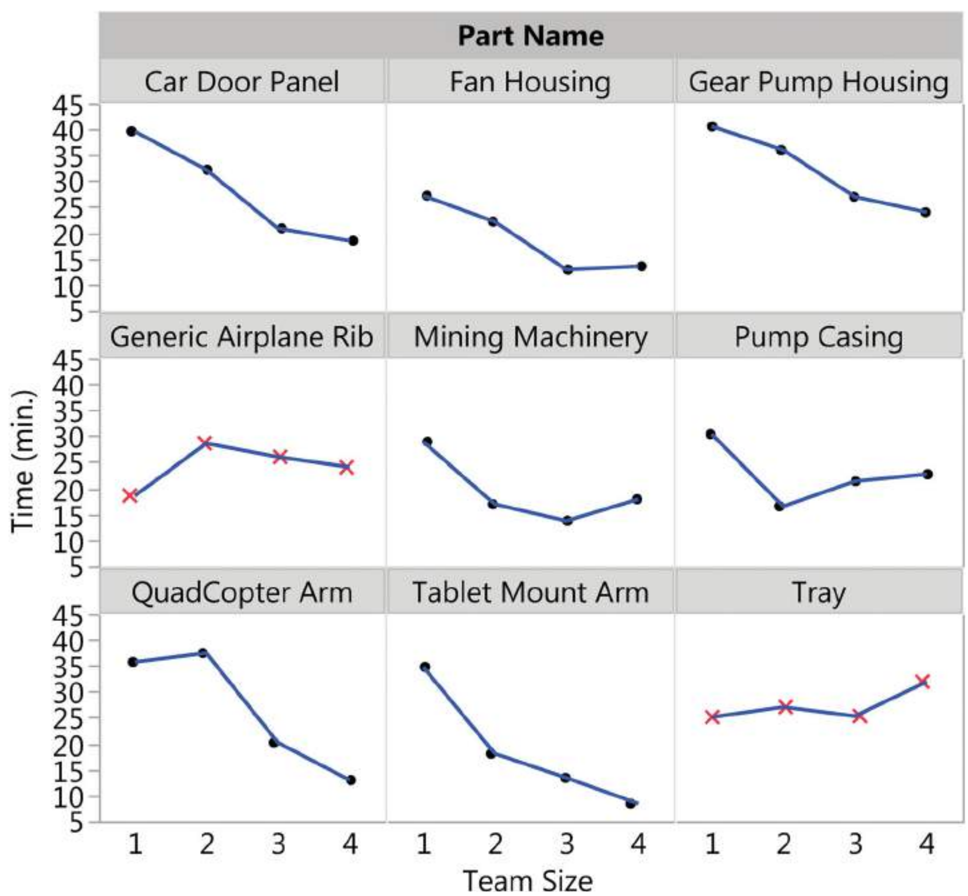

Figure 10. Completion times for each team size for the nine parts considered.

of the part compared to the MU teams. For example, to add angular draft to the multiple negative extrudes in the part, members of the three-person team specified the amount of draft as part of each extrude feature. Meanwhile, the single user quickly created many simple extrudes, and then, using the draft feature, returned and selected multiple extrusions to which he applied draft globally. This technique served the single user especially well, perhaps unknowingly, on one particular portion of the Tray, which was considered more complex. On the three-person team, the contributor who worked on the same portion, despite having the second fastest speed-test time, struggled significantly. In the end, he spent more than double the time to finish the section as the single user. Members of the four-person team experienced similar challenges. We suspect that the style of this single-user may be rare and that additional repetitions would reveal the Tray to be a strong candidate for MUCAD teams as originally predicted.

The observations of the teams modeling the Airplane Rib part, also predicted to be suitable for MUCAD, revealed some interesting insights. The single-user team was able to use a spline and model a satisfactory airfoil shape in roughly four minutes. For contrast, the four-member team decided to have each team member attempt to sketch an airfoil and then choose the best among the designs. After that effort failed, one team member went onto the internet, found a set of coordinates for a NACA airfoil, downloaded it, and created points for a spline. This entire process took approximately 12 minutes and significantly delayed the team's completion time.

6. Conclusions

By classifying a sample of parts using a taxonomic scheme we developed, we were able to test two proposed models for predicting the optimal number of multi-user team members for modeling a given part. The empirical data through testing strengthen the idea that an optimal number of members exists for MUCAD teams, and that the optimal number of users can be predicted with varying accuracy by different kinds of models. We also find strong evidence to support the theory that increasing the size of a team from a single user to larger teams can increase the accuracy when predicting the time for completion.

6.1. Limitations

Although 60 different team size-part type combinations were performed, the required constraint that no volunteer repeat a part limited the amount of collected data. An enlarged dataset with additional repetitions could significantly enhance this research and the validation of the proposed models. Furthermore, only teams of one to four users in size were tested with relatively simple parts composed of few features. Both of these could be expanded in future experiments. It should also be noted that the type of work best represented by these experiments would be the work done to create the CAD models after most design decisions had been made. Design tasks such as determining the best shape of a design for optimal functionality were considered to be outside the scope of this research.

6.2. Future work

Since this research considers the first step in closing an apparent gap in MUCAD understanding, a number of topics have been identified for future research. Experiments testing a wider range in part complexity and team size would help validate the models presented in this research. More advanced models, including graph theoretic principles, should be developed and evaluated for improved accuracy. Future testing should also include additional methods for measuring the individual user's CAD skills and assessing the impact of combining users with different or similar skill levels and their effect on the multi-user teams. Explorations into whether a correlation exists between the type of manufacturing process used to create a physical part and the model's "MU-friendliness" would likewise be beneficial. Finally, development and testing of an automated tool for part classification would be a major but useful undertaking.

Acknowledgements

We wish to thank the National Science Foundation Center for e-Design site at Brigham Young University and the Boeing Company for their support of this research [grant number 1267247]. We also wish to thank Joanna Hugo, Landon Wright, Scott Egbert, Eric Bowman, and David French for their contributions.

ORCID

John L. Salmon http://orcid.org/0000-0002-8073-3655
Ammon I. Hepworth http://orcid.org/0000-0001-7651-1059
Edward Red http://orcid.org/0000-0002-9321-6913

References

[1] Alexandrov N. M.; Lewis R. M.; Gumbert C. R.; Green L. L.; Newman P. A. Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models, J. Aircr., 38(6), 2001, 1093–1101. http://dx.doi.org/10.2514/2.2877.

[2] Bidarra R.; van den Berg E., Bronsvoort W. F. A Collaborative Feature Modeling System, J. Comput. Inf. Sci. Eng., 2(3), 2002, 192. http://dx.doi.org/10.1115/1.1521435.

[3] Briggs J. C.; Hepworth A. I.; Stone B. R.; Coburn J. Q.; Jensen C. G.; Red E.: Integrated, Synchronous Multi-user Design and Analysis, J. Comput. Inf. Sci. Eng., 15(3), 2015.

[4] Brooks F.: The mythical man-month, Vol. 1995. Reading, MA: Addison-Wesley, 1975.

[5] Bu J.; Jiang B.; Chen C.: Maintaining semantic consistency in real-time collaborative graphics editing systems, IJCSNS, 6(4), 2006, 57.

[6] Cera C. D.; Braude I.; Comer I.; Kim T.; Han J.; Regli W. C.: Hierarchical Role-Based Viewing for Secure Collaborative CAD, in ASME International Design Engineering Technical Conferences & The Computer and Information in Engineering Conference, 965–974, 2003.

[7] Chan S.; Wong M.; Ng V.: Collaborative solid modeling on the WWW, Proc. 1999 ACM Symp. Appl. Comput., 1999, 598–602. http://dx.doi.org/10.1145/298151.298487.

[8] Chen S.-J.; Lin L.: Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering, Eng. Manag. IEEE Trans. Eng. Manag., 51(2), 2004, 111–124. http://dx.doi.org/10.1109/TEM.2004.826011.

[9] Crowson R. A.: Classification and biology. 1972.

[10] Dietrich U.; Lukas U.; Morche I.: Cooperative modeling with TOBACO, in Proceedings of the TeamCAD97 Workshop on Collaborative Design, 1997, 115–122.

[11] Dyer W. G.; Dyer J. H.: Team Building, 5th ed. San Francisco: Jossey-Bass, 2013.

[12] French D. J.; Stone B.; Nysetvold T. T.; Hepworth A.; Red W. E.: DETC2014-35279 Collaborative Design Principles From Minecraft With Applications, 2014.

[13] Gideon J. M.; Dagli C. H.; Miller A.: Taxonomy of Systems-of-Systems, Inst. Electr. Electron. Eng. IEEE, January 2005, 356–363.

[14] Hepworth A.: Scalable Integration of Commercial File Types in Multi-User CAD, Comput. Aided. Des. Appl., 11(4), 2014, 459–467. http://dx.doi.org/10.1080/16864360.2014.881190

[15] Hepworth A.; DeFigueiredo B.; Shumway D.; Fronk N.; Jensen C. G.: Semantic Conflict Reduction Through Automated Feature Reservation in Multi-User Computer-Aided Design, 2014 Int. Conf. Collab. Technol. Syst., 2014, 56–63. http://dx.doi.org/10.1109/CTS.2014.6867542.

[16] Hepworth A.; Halterman K.; Stone B.; Yarn J.; Jensen C. G.: An Integrated Task Management System to Reduce Semantic Conflicts in Multi-user CAD, Concurr. Eng. Res. Appl., March, 2015, 1–12. http://dx.doi.org/10.1177/1063293X15573595.

[17] Hepworth A. I.; Tew K.; Nysetvold T.; Bennett M.; Jensen C. G.: Automated Conflict Avoidance in Multi-user CAD, Comput. Aided. Des. Appl., 11(2), 2014, 141–152. http://dx.doi.org/10.1080/16864360.2014.846070.

[18] Jing S. X.; He F. Z.; Han S. H.; Cai X. T.; Liu H. J.: A method for topological entity correspondence in a replicated collaborative CAD system, Comput. Ind., 60(7), 2009, 467–475. http://dx.doi.org/10.1016/j.compind.2009.02.005.

[19] Kao Y.-C.; Lin G. C. I.: Development of a collaborative CAD/CAM system, Robot. Comput. Integr. Manuf., 14(1), 1998, 55–68. http://dx.doi.org/10.1016/S0736-5845(97)00014-8.

[20] Li W. D.; Fuh J. Y. H.; Wong Y. S.: An Internet-enabled integrated system for co-design and concurrent engineering, Comput. Ind., 55(1), 2004, 87–103. http://dx.doi.org/10.1016/j.compind.2003.10.010.

[21] Mishra P.; Varshney A.; Kaufman A.: Collabcad: A toolkit for integrated synchronous and asynchronous sharing of cad applications, Proceedings TeamCAD: GVU/NIST workshop on collaborative design, 6, 1997, 131–137.

[22] Moncur R. A.; Jensen C. G.; Teng C. C.; Red E.: Data consistency and conflict avoidance in a multi-user CAx environment, Comput. Aided. Des. Appl., 10(5), 2013, 727–744. http://dx.doi.org/10.3722/cadaps.2013.727-744.

[23] Moynihan L. M.; Peterson R. S.: A contingent configuration approach to understanding the role of personality in organizational groups, Research in Organizational Behavior, 23, 2001, 327–378. http://dx.doi.org/10.1016/S0191-3085(01)23008-1.

[24] Nam T. J.; Wright D. K.: CollIDE: A shared 3D workspace for CAD, in Proceedings of Conference on Network Entities, 1998.

[25] Nelson D. L.; Lehninger A. L.; Cox M.M.: Lehninger Principles of Biochemistry. W. H. Freeman, 2008.

[26] Page E.: The Rate of Team Performance Change Over Time, Georgia Institute of Technology, 2004.

[27] Parker G. M.:Cross-Functional Teams: Working with Allies, enemies, and other strangers, 2nd ed. San Francisco: Jossey-Bass, 2003.

[28] Qiang L.; Zhang Y. F.; Nee A. Y. C.: A distributive and collaborative concurrent product design system through the WWW/Internet, Int. J. Adv. Manuf. Technol., 17, 2001, 315–322. http://dx.doi.org/10.1007/s001700170165.

[29] Ramani K.; Agrawal A.; Babu M.; Hoffmann C.: CADDAC: Multi-Client Collaborative Shape Design System with Server-based Geometry Kernel, J. Comput. Inf. Sci. Eng., 3(5), 2003, 170. http://dx.doi.org/10.1115/1.1582882.

[30] Red E.; French D.; Jensen G.; Walker S. S.; Madsen P.: Emerging Design Methods and Tools in Collaborative Product Development, J. Comput. Inf. Sci. Eng., 13(3), 2013. http://dx.doi.org/10.1115/1.4023917.

[31] Red E.; Jensen G.; French D.; Weerakoon P.: Multi-user architectures for computer-aided engineering collaboration, in Proceedings of the 2011 17th International Conference on Concurrent Enterprising (ICE 2011), 2011, 1–10.

[32] Shah J. J.; Mantyla M.: Parametric and Feature Based CAD/CAM: Concepts, Techniques, and Applications. New York, New York, USA: John Wiley & Sons, Inc., 1995.

[33] Shen L.; Hao Y.; Li M.; Zhao W.; Zheng J.: A Synchronous Collaborative Environment for Engineering Design Education Collaborative Discussging, 2007, 298–303.

[34] Steiner I. D.:Group Process and Productivity, 8th ed. New York: Academic Press, 1972.

[35] Steinmetz M.; Navarro J. F.: The Hierarchical Origin of Galaxy Morphologies, 2002, 1–9. http://dx.doi.org/10.1016/S1384-1076(02)00102-1

[36] Stork A.; Lukas U.; Schultz R.: Enhancing a Commercial 3D CAD System by CSCW Functionality for Enabling Co-operative Modelling via WAN, in Design Engineering

Technical Conferences, Computers in Engineering Conference, 1998.
[37] Todd R.H.; Allen D.K.; Alting L.: Fundamental Principles of Manufacturing Processes, 1st ed. New York, New York, USA: Industrial Press Inc., 1994.
[38] West D. B.: Introduction to Graph Theory, Volume 2. Upper Saddle River: Prentice Hall, 2001.
[39] Xu Y.: A Flexible Context Architecture for a Multi-User GUI, Comput. Aided. Des. Appl., 8(4), 2011, 479–497. http://dx.doi.org/10.3722/cadaps.2011.479-497.