Characterizing Relationships for System Dynamics Models Supported by Exploratory Data Analysis

A Conceptualizing Approach about the Meeting Diversity in Student Software Projects

Fabian Kortum, Jil Klünder, Kurt Schneider
Software Engineering Group
Leibniz Universität Hannover
30167 Hannover, Germany
{fabian.kortum, jil.kluender, kurt.schneider}@inf.uni-hannover.de

Abstract— **Estimating dynamic components in projects involves understanding human factors which are substantial in software development. Communication and collaboration in teams consist of social-driven characteristics with influences on the continuous delivery of software. Efficiently estimated meetings become increasingly important due to budget calculations and shortened release cycles. Experiences of project managers combined with retrospectives on historical data records support a better understanding of team dynamics. But interpreting complex effects is not always trivial, in particular without further analyzes. In several studies, information relationships are investigated through linear correlation measures. Additional analyses for higher correlations are often neglected due to the advanced functional characterization. This leads to statistical gaps with significances for explored data relationships and their functional interpretation. In this paper, we present a systematic identification and visualization of team communication effects and diversities for field study records of 34 student software projects. We combine methodologies from system dynamics with exploratory data analysis to extract and emphasize significant effects. These insights help to sensitize for advanced investigations about the statistical measures of correlation and to interpret sophisticated structures. Furthermore, it reinforces potentials for a team's communication performances and enables an enhanced understanding about how student teams meet and communicate.**

Keywords— Team dynamics; exploratory data analysis; data visualization; student software projects

I. INTRODUCTION

Human factors have become increasingly important for software engineering disciplines and processes. Especially the relationships of single factors are often difficult to understand or too complex to be directly interpreted [10]. It remains to the experience and knowledge of project leaders to estimate future projects inter alia according to team structures, capacities, and budget capabilities. The longer a project manager collaborates with a particular team, the more harmonized and predictable becomes the team's typical behavior. Knowledge transfer and exchanges are often achieved by self-reflections and retrospectives of all involved project members at the end of the project [16]. The common use of system records during the development process allows subsequent statistical analysis on available project data, team performances and comparable attributes that can be monitored. This analysis leads to the identification and visualization of important compounds. Illustrating sophisticated information structures and the interpretation of these can be realized in various ways [14]. But interpreting data relationships is often linked with risks if relationships are insufficiently analyzed. This occurs when data dependencies become statistically explained without further proof for multiple kinds of functional characteristics, e.g. determining a data pair's non-linearity. This can cause gaps in interpretation and inadequate significance measures on explored data relationships. However, another problem is the number of records: The more records, the more combinations and types of connections exist. Anyhow, each of these relationships requires further examination for complex functional relationship characteristics [7]. This takes an enormous amount of extra time and effort when analyzing data manually and is almost unrealistic for large data sets.

In this study, we explore some important effects on student teams' communication and meeting behavior. The underlying data set originates from a previous field study focusing on group effects and the communication behavior in 34 student software projects. The data consists of weekly data reports from each team leading to more than 15.000 database entries. This includes communication paths, intensities and network structures, social manner, used media channels, mood and team spirits, meeting quantity, duration and participation from 165 student participants [6]. Our study concerns on meeting diversity effects about the following seven categories of system components shown in Fig. 1.

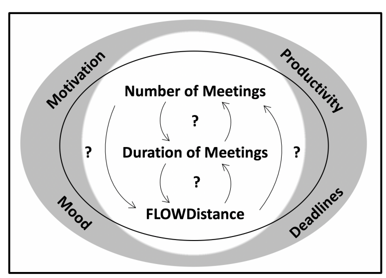

Figure 1. Report components with relevance for team's meeting diversity

DOI reference number: 10.18293/SEKE2017-143

Based on previous studies [6, 8] and a subset of open questions about team communication effects, this approach focuses on the verification of three assumptions with strong relevance to meeting manners during student software projects. Insights about the following questions can help us to optimize the educational concept of student software projects, also identifying stereotypes about students operating manner.

RQ1 is derived from expectations on student's meeting estimation when quality gates occur. In the best case, the meeting times remain at an almost constant level that would represent balanced working process without firefighter situation.

> *RQ1:* *Does the duration or quantity of team meetings increase for project weeks that are terminated to have quality gates or deadlines?*

Insights from RQ2 present two indicators for the duration of team meetings. For early communication diagnoses and tendency estimators, it is important to understand whether the team's perceived productivity of communications ends up with longer or shorter meeting durations. The same applies for team member's subjectively reported motivation during each project week. A decreasing level of team's motivation can be probably associated with shrinking meeting hours. Thus, it can be one potential indicator for an ongoing motivation condition in teams.

> *RQ2:* *Does the communication productiveness or team member's motivation have an effect on the duration of team meetings?*

In RQ3, we want to verify whether a decentralized meeting for instance through video chats or other digital channel have positive, negative or even no effects on the motivation of a team. As a matter of fact, face to face communication is an approved contact form with maximum information flow [17]. Compared with other communication channels, face to face communication presents additional perspectives for the motivation in teams.

> *RQ3:* *FLOW distance is defined as a metric to describe a team's decentralized communication and meetings [10]. Does a decentralized meeting also affect the team's perceived motivation?*

In previous approaches on early communication diagnoses for tendency forecasts, we revealed the importance of characterizing data relationships, that should be made with most maximal aware for accurate estimators [8]. For the verification of RQ1, RQ2, and RQ3, we apply exploratory data analysis to characterize and interpret key components that are later used to describe the meeting dynamics. So far unknown data relationships are analysed through mutual information consideration. At the same time, such techniques resolve a more efficient interpretability of data relationships. These insights can help to sensitize on investigations about the statistical measure of correlation and clarifies the structures.

We also present a simplified process based on techniques for conceptualizing a system dynamics model [1, 12]. The conventional conceptualization of this kind of model is often realized supported by statistical methods, e.g. analysis of regressions, distributions, correlations and data significances. At this moment, semi-automatic statistical methods for identifying linear, non-linear and other functional relationships are rarely taken into account. We introduce the maximal information-based non-parametric exploration (MINE) method which can detect and classify up to 27 different types of functional data relationships [7]. In addition to the verification of RQ1, RQ2, and RQ3, it is also our aim to apply a conceptualizing process for building a system dynamics model with the support of exploratory data analysis on student team records. This allows us to visualize the basic mechanisms of effects within a system. Although, this paper does not include neither the formulation nor the simulation of the system dynamic model. Such models are mostly used to perform experiments and simulations to discover further, not yet identified effects in software projects. However, these subsequent processes will be introduced in later publications and are not part of this paper. Although more research questions are conceivable, we focus on those mentioned above. We will present solutions for visualizing the results, represent the identified communication and meeting effects, give context information about relationships and serve the methodical conception of dynamically influencing components within a system.

II. Origin of Team Communication Records

Software projects at Leibniz Universität Hannover take an important role for students in the fifth semester of their undergraduate computer science studies. These projects fulfill real world's customer requirements, time pressure and self-managing organizations within each developer team. Self-chosen student project leader and quality associates navigate the team through each phase of a waterfall-oriented development process. In an interdisciplinary cooperation with psychologists, Schneider et al. [6] studies both the dynamics of communication behavior and information flow from student software developers since a few years. Several student software projects have been monitored to grasp data about social driven team dynamics and to establish early diagnoses and tendency forecasts for communication diversity [8]. The following list shows conditions about teams and projects from the previously taken field study [6]:

a) 34 student software projects with waterfall-oriented development process and durations of 15 weeks
b) The students were academia undergraduates with at least five semesters in major of computer science.
c) Team sizes: 31 of 34 teams consisted of five people
d) Projects were comparable in complexity, effort of time and fulfilled real end-user requirements.
e) Self-organization: The teams had to manage tasks, project scheduling, social conflicts and problems.
f) Three quality gates were integrated during the project phase to ensure the quality of product.
g) At project's end, all teams had to fulfil a customer acceptance test, individually formed based on each projects requirements compliance.
h) Students have basic knowledge in software engineering disciplines, programming skills in Java, and mixed experiences in working as a group.

III. Related Work

Our research is based on related work with manifested methods in the field of system dynamics modeling as well as the applied statistical information analysis.

Decades ago, Forrester [1] pioneered the well-known system dynamics model, which is a methodology for holistic analysis and model based simulation of complex and dynamic systems. The quantitative modeling represents a strategy to identify and investigate forces within systems which have led to a problem in the past or are of particular importance. The author manifested qualitative models through flow diagrams to simulate system behavior that enables a deeper understanding. Stocks, rates, and auxiliary variables are used to describe system interconnections and show how the effects of action lead to the behavior of systems that are partly non-linear and contra-intuitive [1, 5]. The processes used in this study are based on Forrester's [ref] conceptualization processes to achieve quantitative knowledge about dynamics.

Also, especially for the principles of dynamic events in software project management, Abdel-Hamid et al. [3] introduced various case studies with large software projects providing relevant metrics. They applied data records and experiences to build system models with different stages of complexity. The authors present a detailed overview of plenty dynamic modeling examples like the dependence of productivity on the motivation of development teams and even an entire software development process chain.

Madachy et al. [4] describe the early stages of modeling communication and team issues, including Brooks law. The authors applied qualitative model simulations to understand different process dynamics through regular boundary expressions under a range of parameter settings.

Houghton et al. [2] focused on the data inclusion and consideration for system dynamics modeling procedures. The authors describe investigations about the possibilities for expanding the conventional system dynamics methodology by conceptualizing and formalizing data collections using statistical methods.

IV. Methodology

The methodology of this approach describes the investigative processing of students' communication and meeting diversity in software projects through the novel exploratory data analyzing technique MINE [7]. First, we present MINE's operational advances and power for the identification and characterization even for complex data relationships and structures. We continue with the conceptualization stage for building a system dynamics model that can be later used for visualizing and abstracting mechanism effects within the target system. Parallelly, we apply conventional data visualization techniques that are commonly used to explore linear data relationships like line charts and cross-correlation plots to demonstrate the difficulties and gaps when data dependencies only become expressed as linear relationships. Using the exploratory relationship identification through MINE, we also present force-based network diagram that graphically highlights the affecting and affected team communication components within the system.

A. Data analyses for maximal information coefficients (MIC)

MIC [7] is a novel measure of data dependence that captures linear, non-linear and more complex functional associations between a pair of variables. The algorithm identifies the maximal available mutual information through consideration of relationships types and their functional properties characteristics. This enables the identification and estimation even of a complex association between a pair of data compounds, where conventional statistical correlation measures would solely consider the linearity of relationship. Fig. 2 summarizes MIC's power for the identification of relationships compared to other correlations methods that take common place on data analysis.

Relationship Type	MIC	Pearson	Spearman	Mutual Information *(KDE)*	Mutual Information *(Kraskov)*	CorGC *(Principal Curve-Based)*	Maximal Correlation
Random	0.18	-0.02	-0.02	0.01	0.03	0.19	0.01
Linear	1.00	1.00	1.00	5.03	3.89	1.00	1.00
Cubic	1.00	0.61	0.69	3.09	3.12	0.98	1.00
Exponential	1.00	0.70	1.00	2.09	3.62	0.94	1.00
Sinusoidal *(Fourier frequency)*	1.00	-0.09	-0.09	0.01	-0.11	0.36	0.64
Categorical	1.00	0.53	0.49	2.22	1.65	1.00	1.00
Periodic/Linear	1.00	0.33	0.31	0.69	0.45	0.49	0.91
Parabolic	1.00	-0.01	-0.01	3.33	3.15	1.00	1.00
Sinusoidal *(non-Fourier frequency)*	1.00	0.00	0.00	0.01	0.20	0.40	0.80
Sinusoidal *(varying frequency)*	1.00	-0.11	-0.11	0.02	0.06	0.38	0.76

Figure 2. Power for identifying relationships types using MIC [7]

MIC represents a relationship coefficient score considering several further statistical correlation measures. This includes, for example, the Spearmen correlation coefficient, Kraskov et al.'s [9] mutual information estimation, maximal correlation estimation using ACE [15] and the principle curve-based CorGC dependency measures [11] for the identification of interdependencies for up to 27 different functional relationship types. Therefore, the algorithm can automatically determine whether a data relationship is interpretable as linear or more complex functional dependence. Estimating tendency models, these further differentiation results have a significant meaning for the outcome's accuracy of forecasting models. In particular, the algorithm determines the existence of a relationship between two data variables in regression problems and non-linear information analysis. Its analysis is grounded on curve-fitting techniques and resembles an automated function regression with maximal mutual information consideration.

B. Conceptualization of a system dynamics model

The commonly applied building process for system dynamics models is divided into four main stages. Each of these steps consists of sub-steps as shown in Fig. 3. This paper only covers the first stage about the conceptualization of system dynamics models and provide practical advice for completing each sub-step. Due to the scope and limitations of this approach, the three remaining stages in the model building process will take part in continuous publishing. Our aim about a conceptual model is to reach an enhanced understanding of the way how student teams communicate and manage their meetings. We also want to simplify the identification and characterization of effects without loss of quality and gaps due to solely linearity measures.

BUILDING A SYSTEM DYNAMICS MODEL

Stage 1	**Conceptualization (focus of this approach)**
	(1) Define the purpose of the model
	(2) Define the model boundary and identify key variables
	(3) Describe the behavior with reference plots of key variables
	(4) Diagram the basic mechanism of the system
Stage 2	**Formulation**
	(1) Convert Feedback diagrams to functional equations
	(2) Estimate operating ranges through parameter values
Stage 3	**Testing**
	(1) Simulate the model and test the dynamic hypothesis
	(2) Test the models' assumptions
	(3) Test model behavior and sensitivity to perturbations
Stage 4	**Implementation**
	(1) Test the models' response to different policies
	(2) Translate study insights to an accessible form

Figure 3. Forrester's stages for building a system dynamics model [12]

Stage 1.1 Define the purpose of this dynamic system modelling

In the first stage, conceptualizing a system dynamics model helps to understand a model's boundary and operating limitation, influencing factors and effects between components. This encloses the meeting and communication behavior of teams, and also their organizational structures during a waterfall process-driven software project. A model about team communication diversity could be rather used to estimate the tendency course of team dynamics for future projects. The identified relationship characteristics, stereotype, and effects express the model's operating mechanism. Simulation runs and experiments with varying communication constellation additionally can resolve knowledge and information about dynamic behavior over time. Furthermore, explicit situational cases or exploratory project planning and knowledge infusions are of higher interests. Therefore, internal structures and interactions during different phases of the development process are more interpretable and transparent through key components with known relationship effects. The conceptual model in this paper shall be used to abstract and visualize the mechanism of teams meeting diversity according to RQ1, RQ2, and RQ3. Findings of coherent meeting effects become resolved through the exploratory data analyses in MINE.

Stage 1.2 Define the system boundary and identify key variables

The guide about how to build a system dynamics model by Forrester [1] continues with the determination of key variables and components of the target system. All elements within a data set become manually reviewed and marked due to their assumed relevance for the system boundary. The marked elements represent a closed systems boundary within which the behavior, e.g. meeting diversity, is analyzed. The primary components list must be reasonable, aggregated and if possible directionally labeled. For example, the *MoodPositive* element from the initial components boundary list in Fig. 4 represents an aggregated component for ten psychological characteristics like happy or satisfied to express a team's mood. These features are part of the team reports and were documented weekly by each member of a team.

Initial Components Boundary List		
Motivation	Holiday Break	Total N° of Commits
Communication Productiveness	Used Communication Channel	Commitment Rate (Member)
Phase of Project	Change Lines of Code	Team Size
Mood Positive	N° of Meeting Attendants	Missing Member (e.g. Sick)
Mood Negative	Progressed Project Time	Quality Gates
Duration of Meetings	Project Life Span	Requirements Compliance
N° of Meetings	Communication Intensity	FLOWDistance

Dynamic components with feedback mechanism in the system

Endogenous Components Boundary	
Motivation (S)	N° of Meetings (S)
Com. Productiveness (S)	N° of Meeting Attendants (F)
Mood Positive (S)	FLOWDistance (S)
Mood Negative (S)	Communication Intensity (S)
Duration of Meetings (S)	Communication Channel (F)

Components without direct affects by the system

Exogenous Components Boundary	
Phase of Project	Holiday Break
Missing Member (e.g. Sick)	Progressed Project Time
Quality Gates	Project Life Span
Team Size	

Figure 4. System boundary with endo- and exogenous component separation

The review of all data variables, especially the subjectively selected system boundary components requires a subsequent categorization. All components in the initial list need to be classified and separated into endogenous or exogenous system elements. However, Fig. 4 only presents an orientation guideline as summarized overview of possible system components and does not necessarily bind the model's frame. After further examination, the initial boundary list also contains system components that seem to be unnecessary for the current research questions. The components with assumed relevance for the ongoing investigations about the three research questions should be classified as an endogenous or exogenous component to set borderlines for this approaches system boundary. Exploratory data analysis like MINE [7] also helps to identify key elements of a sophisticated system as well as analyzing the maximal available mutual information between associations. The key factor findings in MINE plotted as a weighted network diagram are shown in Fig. 5. Beside the *ProgressedTimeOfProject*, the components *FLOW Distance*, *ProjectPhase, NumberOfTeam-Meetings* and *NumberOfAttendedMeetings* seem to be of central relevance for the system.

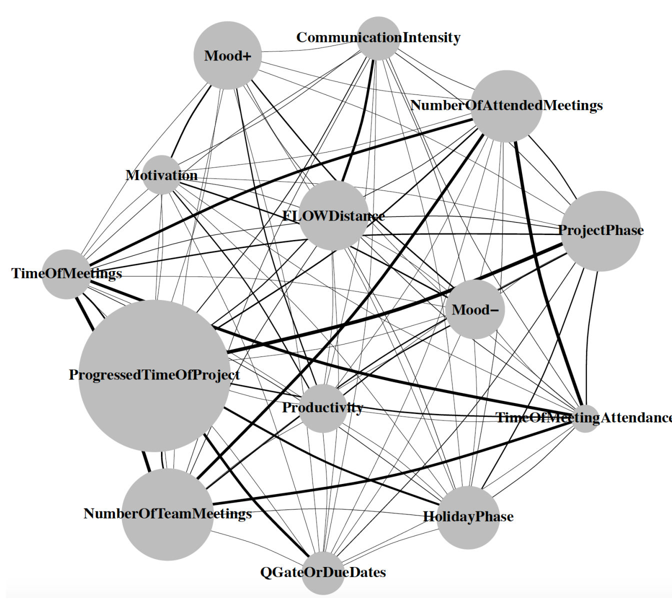

Figure 5. MIC-weighted network diagram on team meeting components

This network plot enables an overview of all investigated component's relationship with statistically identified maximal information coefficient strengths. The diagram nodes present varying sizes according to each measured ongoing MIC strength.

Nodes or component relationships are intersected through edges that vary in their intensity to express the relationship strengths for identified functional properties through MINE. The graphics are realized through a minimalistic Java tool reading data records from CSV-files, performing exploratory data analysis through integration of the MINE application and plotting the identified relationship as an MIC-weighted network diagram. For plotting the graphic, the tool uses an embeddable R-library named igraph [14]. This library also allows extended network visualizations as shown in Fig. 8. It can derive force-based network diagrams with positive and negative notated arrows to describe an effects polarity.

Stage 1.3 Describe the key variables behavior as reference plot

The traditional conceptualization process for system dynamics models by Forrester [5, 12] continues with plotting the assumed components of interest. Through cognitive reviews and interpretations, the difficulty remains in identifying distinct phenomena, patterns or other relationship types between the components. The line chart in Fig. 6 shows meeting characteristics of the teams and their dynamic changes during a project.

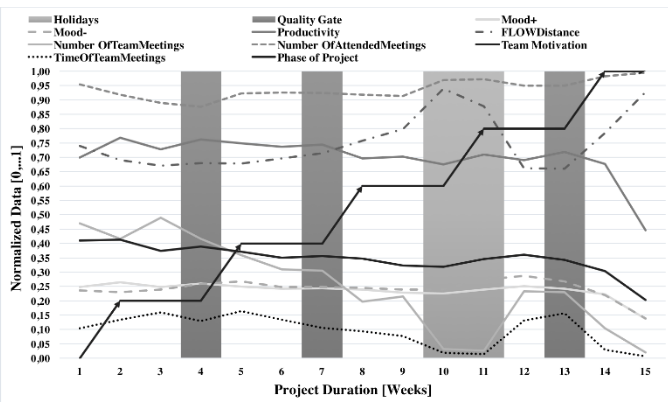

Figure 6. Reference chart for comparisons of team meeting dependencies

The more component's and their changes over time became plot and compared with other elements, the more difficult it is to make qualitative interpretations for data relationships and types. Cross-correlation plots in R-statistics help to measure two component's linear relationship statistically as in Fig 7. The applied visualizations foster a general understanding for meeting diversities and data variances over time. However, it represents an insufficient method, especially for detailed identifications of component's relationships in larger or complex systems.

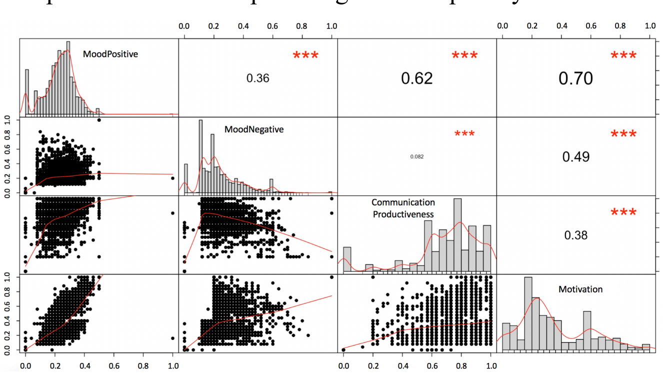

Figure 7. Linear data analysis through cross correlation matrix in R-statistics

As an example, MINE identifies relationship strengths for the components *MoodNegative* and *CommunicationProductiveness* in Fig. 7 differently compared to the linearity measures from the cross correlations in R. In this qualitative comparison, MINE measures a stronger correlation coefficient with strong significance based on the additionally considered functional relationship properties. The example demonstrates the statistical outcome with identification gaps in case of sole linearity analyzes without further relationship characterization. Both correlation coefficient measures and their particular statistical significance with Fisher's exact tests [13] are shown in Tab 1.

Table 1. Identification of relationship strength: MIC vs. Pearson correlation

Component X	Component Y	MIC	sig. *p*	Pearson Cor.	sig. *p*
Mood-	Productiveness	0.30527	0.00780	0.08199	0.0004

For the verification of RQ1, RQ2, and RQ3, we consider all team records for the exploratory data analysis in MINE. This step enables to identify each possible relationship within the data components, shown in the network diagram in Fig. 5. As the subsequent step, we limit the considered boundary of relationships from the entire data structure to only relevant subjects for the RQs. This centralizes the focus and simplifies the interpretability for particular component's effects. In particular, it enables to compute only the sub-network diagram that describes a component's direct interplay as an affected or affecting unit. The Java tool that was established to analyze the team data records includes a feature to select components for managing the focus of relationship analyses in MINE. Therefore, resulting network diagrams can shrink and expand their structure, depending on the selected items and research focus. According to RQ1, RQ2, and RQ3, we chose all components that were identified through MINE to be affected by elements or affect others. On the affected side, we mark the features that were of interest to the research questions. This includes five components: *DurationOfTeamMeetings, FLOW-Distance, NumberOfTeam-Meetings, Motivation*, and *Productivity* (communication productiveness). The force-based sub-network diagram in Fig. 8 shows all affected components with interest for the RQs, as well as all the elements that are affecting these.

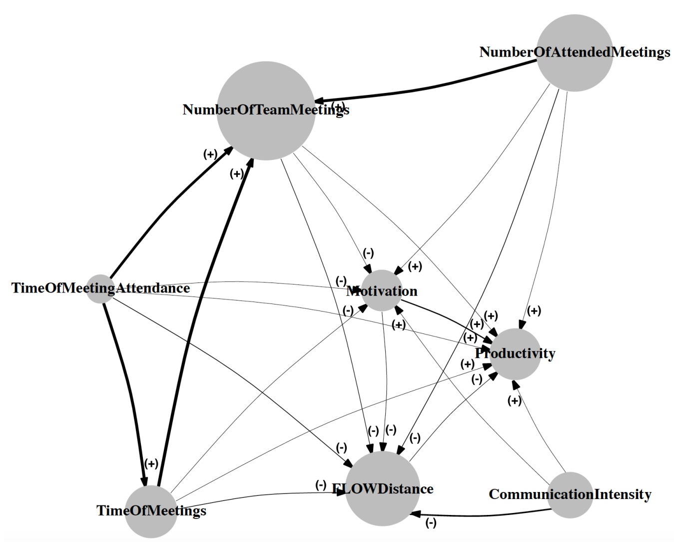

Figure 8. Relationships with directed polarities between meeting components

This diagram is a visualized result of the exploratory relationship identifications through MINE and our Java tool which performs the embedded visualization steps. In Tab. 2, all findings through MINE are listed numerically including their references to the conventional correlation measures on a relationship's linearity.

Table 2. Relationship identification and MIC-scoring through MINE

Effecting Component X (identified with MINE)	Affected Component Y (selected scope)	MIC (strength)	MIC-pearson² (nonlinearity)	MEV (functionality)	Linear regression (pearson)
TimeOfMeetings	**NumberOfTeamMeetings**	**0,88203**	**0,37842423**	**0,88203**	**0,7096519**
TimeOfMeetingAttendance	**TimeOfMeetings**	**0,81826**	**-0,040376246**	**0,81826**	**0,92662627**
NumberOfAttendedMeetings	**NumberOfTeamMeetings**	**0,79197**	**-0,077647746**	**0,76115**	**0,93253297**
TimeOfMeetingAttendance	**NumberOfTeamMeetings**	**0,73744**	**0,28851897**	**0,73744**	**0,6700157**
CommunicationIntensity	**FLOWDistance**	**0,68798**	**-0,05492699**	**0,68668**	**-0,86192054**
Motivation	**Productivity**	**0,30437**	**0,16183046**	**0,30437**	**0,37754408**
TimeOfMeetingAttendance	**FLOWDistance**	**0,19455**	**0,054150358**	**0,19455**	**-0,37469938**
TimeOfMeetings	**FLOWDistance**	**0,19312**	**0,054751784**	**0,19312**	**-0,3719788**
NumberOfAttendedMeetings	**FLOWDistance**	**0,18831**	**0,03741516**	**0,18831**	**-0,38845184**
NumberOfTeamMeetings	**FLOWDistance**	**0,1804**	**0,024076387**	**0,1804**	**-0,3953778**
CommunicationIntensity	**Motivation**	**0,14371**	**0,075237975**	**0,14371**	**0,2616716**
Motivation	**FLOWDistance**	**0,1404**	**0,097615935**	**0,1404**	**-0,20684311**
CommunicationIntensity	**Productivity**	**0,13312**	**0,015243918**	**0,13312**	**0,34333086**
FLOWDistance	**Productivity**	**0,1305**	**0,05606755**	**0,1305**	**-0,27282313**
TimeOfMeetings	**Motivation**	**0,08316**	**0,07907622**	**0,08316**	**-0,06390443**
NumberOfTeamMeetings	**Motivation**	**0,06959**	**0,06958476**	**0,06959**	**-0,002290251**
TimeOfMeetingAttendance	**Motivation**	**0,06919**	**0,06621243**	**0,06891**	**-0,05456712**
NumberOfAttendedMeetings	**Motivation**	**0,06334**	**0,06332517**	**0,06334**	**0,003851846**
TimeOfMeetings	**Productivity**	**0,05778**	**0,03728396**	**0,05778**	**0,14316438**
NumberOfTeamMeetings	**Productivity**	**0,05536**	**0,03248602**	**0,05536**	**0,15124145**
TimeOfMeetingAttendance	**Productivity**	**0,05209**	**0,03416415**	**0,05209**	**0,13388745**
NumberOfAttendedMeetings	**Productivity**	**0,04873**	**0,03124734**	**0,04873**	**0,13222201**

The used Java tool enables to mark whether an effect has negative or positive influences as directed polarity label for each identified relationship. Each polarity is detected through the trendline course of the identified functional relationship between a pair of components. The formerly visualized relationships in Fig. 8 with the directed polarities combined with the statistical measures in Tab. 2 allow us to interpret and describe the behavior of each component and therefore to verify the RQs.

According to RQ1, we want to prove whether quality gates are a reliable indicator for an increasing number or duration of team meetings. The methodology of this study allows to verify the RQs in two ways: Through statistical measures and also as cognitive interpretation for visualized relationship networks. The secondary is less accurate but has strong benefits for simple relationship representations. The affected component *Number-OfMeetings* in Fig. 8 reveals three incoming positive influences from the affecting components *NumberOfAttendedMeetings*, *TimeOfMeetings*, and *TimeOfMeetingAttendance*. There is no evidence in the empirical records confirming the dependency between an increasing number of team meetings and occurring quality gates. This is also correct for the duration of team meetings. The component *TimeOfMeetings* is only affected by the time of students' typical attendance in a meeting. Therefore, we cannot confirm that *QualityGates* influence the quantity and duration of student team meetings.

RQ2 is about whether the communication productivity or motivation have an influence on the duration of team meetings in the student projects. We show in Fig. 8, that the component *TimeOfMeetings* is affected by the component *TimeOfMeeting-Attendance*. It also affects four others differently i.e. positively the *FLOWDistance*, negatively the *Motivation*, positively *Productivity* and positively the *NumberOfTeamMeetings*. All these components have endogenous characteristics within the system. Means they all represent affected elements, but also can affect other components at the same time. Vice versa, the diagram indirectly obtains that *TimeOfMeetings* will be reduced with an increasing *Motivation*. An increasing communication *Productivity* leads to a positive effect on the *TimeOfMeetings*. Therefore, the higher the perceived communication productivity of team members, the more time they will spend in meetings.

RQ3 concerns potential effects on a team's perceived motivation in the case of distributed communication structures which can be expressed by a high *FLOW-Distance*. Fig. 8 that this component is negatively affected by the *Motivation*. Both components are endogenous units as well. Therefore, they also consist of feedback loop mechanisms. For instance, this means that an increasing *FLOWDistance* will decrease the team's perceived *Motivation* within a project and vice versa. The graph also shows that a decentralized communication has plenty effects on e.g. quantity, duration, participation in meetings with relevance for teamwork.

Stage 1.4 Diagram a system's basic mechanism

The dependencies in our system with specific component scopes can be expressed through the notated polarity interactions shown in Fig. 8. So far, these diagrams do not cover or describe feedback loops for components' interplay. Forrester [5] applies quantitative modeling through stocks and flows diagrams as in Fig. 9 to investigate interactive effects on endogenous components. The stocks represent components controlled through auxiliaries. Each auxiliaries function describes a single stock's level change over time by associated effects that were identified to influence a stock. In fact, auxiliaries regulate effects through functional equations, which is part of the second stage to formulate a system dynamics models. However, this paper limits its focus on the conceptualization of a system dynamics model. We manually build the stock and flow diagram in Fig. 9 to quantitatively visualize the dynamic mechanism regarding the RQs about students' meeting diversity in software projects. The model is realized with the educational license for the modeling and simulation software Anylogic.

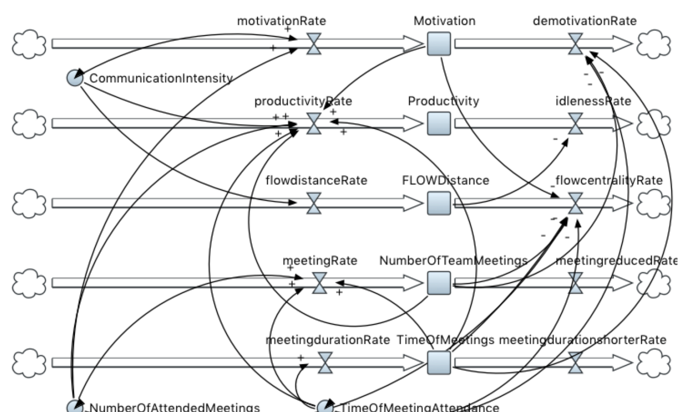

Figure 9. Stock and flow diagram for a meeting diversity system

This quantitative model abstracts the component's interplay through Forrester's notation REF. Formalizing the so far only conceptual auxiliaries will resolve a qualitative model that also enables simulations. However, this requires manually effort by analysts working on the *Formulation* stages listed in Fig. 3. Formulating relationships also requires detailed knowledge about the type of connection. It is an incremental building process that helps project analysts to detect further heuristics and gather information for all project associates.

V. Interpretation and Validity of Results

The objective of this research was to combine the system dynamics terminology with exploratory analysis on information provided by student reports during software projects. The MINE methodology and visualization techniques, e.g. the conceptual modeling and force-based networks are applicable for further system conceptualization and data structure characterization [7].

A. *Interpretability of statistical results*

Before we started with our investigations, we approved the quality of data records. Therefore, we performed univariate variance analysis for all data records through ANOVA to ensure that the data is normally distributed. For the comparison of different data types, all analyses are done with normalized values. The applied RQs are defined due to assumptions with relevance on student teams' meeting behavior in university software projects. All relationships listed in Tab. 2 are verified through Fisher's exact tests with p-values < 0.05, i.e. with significances. The explored insights help to enhance framework conditions and educational concepts for future student software projects. Supplementary, the identified meeting dependencies in Fig. 8 characterizes that a centralized communication structure has strong importance for the motivation and positive atmosphere in teams. Beside newly gathered insights, some trivial relationships could be noted as well like that a team's motivation has positive effects on the communication productivity in meetings. Additional analyses on other teams and software projects probably discover more aspects of dynamic team behavior and show human factors in a more understandable way.

B. *Threats to validity*

This approach underlies several threats to validity. All results are derived from the student records in a previously taken field study [11]. Experience records of other teams with different framework conditions may lead to other results. The authors subjectively set the initial system's component boundary listed in Fig. 4 with assumed relevance for meeting factors. Different researcher's experiences may result in other selections. Due to the nature of system conceptualization, our meeting model might be incomplete and missing exogenous influences. The reliability and quality of insights about the meeting diversity in teams are only statistically validated and resolutely remain solely applicable for other student software projects with similar frameworks. Consequently, the results should not be over-generalized for regular developer teams in software projects.

VI. Conclusion

We introduced the conceptualization steps about modeling system dynamics [12] in combination with the MINE terminology [7] that establishes an advanced identification of functional relationships in sophisticated data structures. The objective of this study is to resolve a better understanding about dynamic dependencies on meeting diversity in student software projects. Human behaviors are not always trivial to comprehend. Therefore, we applied field study records from 34 student software projects to an exploratory analysis and visualization proceeding, which verified three research questions about meeting behavior with assumed relevance for educational perspectives on project scheduling. Beside the system dynamics mechanism, we derived force-based network diagrams providing a simplified visualization for meeting dependency structures that are also understandable not especially for analysts. We could prove that the centrality of meetings takes an important role in the communication productiveness and motivation of student developer teams.

The exploratory terminology helps us to characterize the meeting dependencies through functional property analyses on data relationships which are verified as an enhanced identification to complement gaps that are not covered by solely linear correlation measures [7]. Data collections from completed software projects are valuable goods for future decision and planning improvements. It is desirable to simplify the interpretability of structural dependencies information. This study helps researchers and educational staff to understand better and interpret student's meeting diversities, also how to statistically analyze and visually diagram effects in teams.

Acknowledgments

This work was funded by the German Research Foundation under grant number 263807701 (TeamFLOW, 2015-2017).

References

[1] J. W. Forrester, *World dynamics*. Wright-Allen Press. 1971.

[2] J. P. Houghton, M. D. Siegel, A. Wirsch, A. Moulton, S. E. Madnick, & D. K. Goldsmith, A Survey of Methods for Data Inclusion in System Dynamics Models. MIT. Engineering Systems Division. 2014.

[3] T. Abdel-Hamid, & S. E. Madnick, *Software project dynamics: an integrated approach*. Prentice-Hall, Inc. 1991.

[4] R. J. Madachy, *Software process dynamics*. John Wiley & Sons. 2007.

[5] J. W. Forrester, System dynamics, systems thinking, and soft OR. *System dynamics review*, *10*(2-3), 245-256. 1994.

[6] K. Schneider, O. Liskin, H. Paulsen, & S. Kauffeld, Media, mood, and meetings: related to project success? *ACM Transactions on Computing Education (TOCE)*, *15*(4), 21. 2015.

[7] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, ... & P. C. Sabeti, Detecting novel associations in large data sets. *Science*, *334*(6062), 1518-1524. 2011.

[8] F. Kortum, & J. Klünder, Early Diagnostics on Team Communication: Experience-Based Forecasts on Student Software Projects. *Quality of Information and Communications Technology (Quatic)*. IEEE. 2016.

[9] A. Kraskov, H. Stögbauer, & P. Grassberger, Estimating mutual information. *Physical review E*, *69*(6), 066138. 2004.

[10] J. Klünder, K. Schneider, F. Kortum, J. Straube, L. Handke, & S. Kauffeld, Communication in teams-an expression of social conflicts. *International Conference on Human-Centered Software Engineering* (pp. 111-129). Springer International Publishing. 2016.

[11] P. Delicado, Another look at principal curves and surfaces. *Journal of Multivariate Analysis*, *77*(1), 84-116. 2001.

[12] J. W. Forrester, & S. Albin, Building a System Dynamics Model Part 1: Conceptualization. Massachusetts Institute of Technology. 1997

[13] R. A. Fisher, Tests of significance in harmonic analysis. Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, 125(796), 54-59. 1929.

[14] K. Ognyanova, & P. Monge, A multitheoretical, multilevel, multidimensional network model of the media system: Production, content, and audiences. *Annals of the International Communication Association*, *37*(1), 67-9. 2013.

[15] L. Breiman, & J. H. Friedman, Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association*, *80*(391), 580-598. 1985.

[16] A.Birk, T. Dingsoyr, & , T. Stalhane, Postmortem: Never leave a project without it. *IEEE software*, *19*(3), 43. 2002.

[17] S. W. Ambler, Agile modeling. 2002