15TH MIDDLE EASTERN SIMULATION & MODELLING MULTICONFERENCE

MESM'2015

5TH GAMEON'ARABIA

CONFERENCE

EDITED BY

Marwan Al-Akaidi

MARCH 2-4, 2015

Manama, Bahrain

A Publication of

EUROSIS-ETI

Printed in Ghent, Belgium

15TH MIDDLE EASTERN SIMULATION & MODELLING MULTICONFERENCE

5TH GAMEON-ARABIA CONFERENCE

March 2-4, 2015

Organised by

ETI

Sponsored by

EUROSIS

AOU

Ghent University

IEEE UKRI - SPC

University of Skovde

JPCS

APSA
Complexity and Public Policy Group

Policy Studies Organization

Hosted by

AOU – Arab Open University

Manama, Bahrain

EXECUTIVE EDITOR

PHILIPPE GERIL
(BELGIUM)

EDITORS

MESM'2015-GAMEON-ARABIA'2015

General Conference Chair

Prof. Dr Marwan Al-Akaidi
Vice-Rector Academic Affairs
AOU, Kuwait

Local Chairs

Dr. Khaldoon Al-Roomi, AOU, Bahrain
Dr. Roustem Narimanovich Miftahof, Arabian Gulf University, Bahrain

INTERNATIONAL PROGRAMME COMMITTEE

Modelling Methodology and Modelling Tools
Helge Hagenauer, Universitaet Salzburg, Salzburg, Austria
Brian Hollocks, Bournemouth University, Bournemouth, United Kingdom
Bjoern Johansson, Chalmers University of Technology, Goeteborg, Sweden
Yan Luo, NIST, Gaithersburg, USA
Dmitry Zaitsev, Odessa National Telecommunications Academy, Odessa, Ukraine

Analysis Methodology
Jairo R. Montoya-Torres, ENS des Mines de Saint-Etienne, Gardanne, France

Simulation of Networks and Communication
Eugen Dedu, FEMTO-ST, Montbéliard, France
Carlos Palau, UPV, Valencia, Spain

Antenna Design and Measurement
Hassan Salti, Arab Open University, Kuwait Branch, Al Ardia, Kuwait

Complex Systems Simulation
Marta Pla Castells, Universidad de Valencia, Valencia, Spain
Krzysztof Fujarewicz, Silesian University of Technology, Gliwice, Poland
Jan Studzinski, Systems Research Institute of Polish Academy of Science, Warsaw, Poland

Simulation of Signal and Image Processing
Renate Sitte, Griffith University, Gold Coast Mail Centre, Australia
Joao Manuel R. S. Tavares, University of Porto, Porto, Portugal

Energy Systems Simulation
Musleh B. AL-Otaibi, Alahmadi, Kuwait
Janos Sebestyen Janosy, KFKI, Budapest, Hungary
Patrick Waeger, EMPA, St. Gallen, Switzerland

Simulation in the Chemical and Petrochemical Industry
Ammar Al Khani, Espoo, Finland
Diganta Bhusan Das, Loughborough University, Loughborough, United Kingdom
Eugenio Ferreira, Universidade do Minho, Braga Portugal
Mohammad Al-Nawafleh, Tafila Applied University College, Al-Balqa' Applied University, Tafila, Jordan
Cornel Resteanu, Research Institute of Informatics, Bucharest, Romania
M R Riazi, Kuwait University, Safat, Kuwait

Multimedia and Virtual Reality Systems
Christos Bouras, University of Patras, Rion, Patras, Greece
Peter Kvasnica, Alexander Dubcek University of Trencin, Trencin, Slovak Republic
Zhisheng Huang, Vrije University Amsterdam, The Netherlands
Marco Roccetti, University of Bologna, Bologna, Italy

Decision Processing in Management
Hadhoum Boukachour, Le Havre University, Le Havre, France
Naoufel Cheikhrouhou, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland
Joel Colloc, Le Havre University, Le Havre, France
Jeanne Schreurs, Universiteit Hasselt, Diepenbeek, Belgium
Christine Sybord, Lyon 2 Lumiere University, Lyon, France
Behrouz Zarei, Sharif University of Technology, Tehran, Iran

Modelling and Simulation in Biomedical Systems
Manuel Alfonseca, Escuela Politecnica Superior, Universidad Autonoma de Madrid, Madrid, Spain
Igor Kvasnica, Regional Department for Environmental Issues of Trencin, Trencin, Slovak Republic
Tomislav Pribanic, University of Zagreb, Zagreb, Croatia

Modelling and Simulation for Industrial Applications
Luis Felipe Acebes Arconada, Universidad de Valladolid, Valladolid, Spain
Pascal Berruet, Universite Bretagne Sud, Lorient cedex, France
A. Chatzinikolaou, Bosch Rexroth SA, Athens, Greece
Olaf Diegel, Massey University, Albany Campus, North Shore Mail Centre, Auckland, New Zealand
Sergej Kalaschnikow, VA TECH ELIN EBG Elektronik GmbH & Co, Vienna, Austria
Markus Koch, C-LAB, Paderborn, Germany
Habtom Mebrahtu, Anglia Polytechnica University, Chelmsford, United Kingdom
Pascal Meyer, Forschungszentrum Karlsruhe, Karlsruhe, Germany
Lars Moench, TU Ilmenau, Ilmenau, Germany
Carlos E. Pereira, Universidad Federal do Rio Grande Do Sul, Porto Alegre, Brazil
Caterina Rizzi, Universita degli Studi di Bergama, Dalmine, Italy
Amir Shirkhodaie, Tennessee State University, Nashville, USA
Hans Veeke, TU Delft, Delft, The Netherlands
Ramiro Velazquez, Universite Paris 6 - CNRS, Fontenay aux Roses, France
Joao de Silva Wellington, UFPE, Recife, Brazil

INTERNATIONAL PROGRAMME COMMITTEE

Simulation in Logistics, Traffic, Transport and Harbour Simulation
Isabel Garcia Gutierrez, Universidad Carlos III de Madrid, Leganes, Madrid, Spain
Daniel Krajzewicz, GermanAerospace Centre, Berlin, Germany
Herve Manier, Universite de Technologie de Belfort-Montbeliard (UTBM), Belfort Cedex, France

Web based Simulation
Jorge Marx Gomez, Otto-von-Guericke-Universitaet Magdeburg, Magdeburg, Germany
Alfonso Urquia, UNED, Madrid, Spain

Simulation in Information Processing
Khalil Drira, LAAS-CNRS, Toulouse Cedex, France
Jean-Claude Hochon, GFI Consulting, Toulouse, France
Peter Lawrence, Deakin University, Bentleigh, Australia
Azeddien Sllame, Tripoli University, Libya
Jim Torresen, University of Oslo, Oslo, Norway

Simulation in Fuzzy Systems, Neural Networks and Genetic Algorithms
Track Chair: Kifaya Qaddoum, Prince Sultan University, Riyadh, Saudi Arabia
Chrissanthi Angeli, Technological Education Institute of Piraeus, Athens, Greece
Marie-Ange Manier, Universite de Technologie de Belfort-Montbeliard (UTBM), Belfort Cedex, France
Viorel Nicolau, University of Galati, Galati, Romania

Simulation in the Built Environment
Bernadette O'Regan, University of Limerick, Limerick, Ireland
Agustin Yague, Universidad Politicnica de Madrid, Madrid, Spain

Simulation in Archaeology
Jean Yves Blaise, UMR CNRS/MCC 694 MAP, Marseille, France
Anders Hast, University of Gavle, Gavle, Sweden
Wolfgang Kreutzer, University of Canterbury, Christchurch, New Zealand
Marcos A Rodrigues, Sheffield Hallam University, Sheffield , United Kingdom

Simulation Projects in the Arab World
Saleh Al-Jufout, Tafila Technical University, Tafila, Jordan
Mohamed-Aly O. Louly, University of Technology, Troyes, France

GAMEON-ARABIA
Eva Hudlicka, Psychometrix Associates, Blacksburg, VA, USA
Pieter Jorissen, Karel de Grote Hogeschool, Hoboken, Belgium
Ian M Marshall, Coventry University, Coventry, United Kingdom
Joseph M. Saur, GTRI, Joint Systems Integration Center, USA

MESM'2015

GAMEON-ARABIA'2015

All articles published in these Proceedings have been peer reviewed

EUROSIS-ETI Publications are ISI-Thomson, IET, SCOPUS and Elsevier Engineering Village referenced

A CIP Catalogue record for this book is available from the Royal Library of Belgium under nr.12620

For permission to publish a complete paper write EUROSIS, c/o Philippe Geril, ETI Bvba, Executive Director, Greenbridge Science Park, Ghent University-Ostend Campus, Wetenschapspark 1, Plassendale 1, B-8400 Ostend Belgium.

EUROSIS is a Division of ETI Bvba, The European Technology Institute, Torhoutsesteenweg 162, Box 4, B-8400 Ostend, Belgium

Printed in Belgium by Reproduct NV, Ghent, Belgium
Cover Design by Grafisch Bedrijf Lammaing, Ostend, Belgium

EUROSIS-ETI Publication
ISBN: 978-90-77381-87-8
ISN: 978-90-77381-87-8

Preface

Dear conference delegate,

I have the pleasure to welcome you to the combined 15th Middle Eastern Modeling & Simulation Multiconference (MESM2015) and 5th annual Pan-Arabic - GAMEON-ARABIA'2015 organized by EUROSIS and hosted and sponsored by the Arab Open University (Bahrain Branch) in Manama, Bahrain. The MESM'2015-GAMEON-ARABIA'2015 is co-sponsored by the IEEE – UKRI SPC, Ghent University, The University of Skövde, JPCS, APSA Complexity and Public Policy Group, and The Policy Studies Organization.

Next to the programme, featuring the refereed and selected papers in the fields of Data Simulation, Networks and Logistics Simulation, and Gaming the joint event also features a specialized track on Biomedical Simulation, a Biomedical Simulation poster session and a workshop entitled" Introduction to Computational Biology" organized by the Arabian Gulf University, Kingdom of Bahrain - under the Patronage of His Excellency Dr. Majid Bin Ali Al-Noomi- Minister of Higher Education, Bahrain and represented on his behalf by Professor Riyad Y. Hamza, General Secretary of the high education council.

As General Conference Chair of both events, I would like to express my thanks to Professor Moudi Al Hmoud, the Arab Open University Rector, for giving me the time to organize this conference and thanks also to the committee members for reviewing the papers and to our local chairs Dr. Khaldoon Al-Roomi, AOU, Bahrain and Dr. Roustem Narimanovich Miftahof, Arabian Gulf University, Bahrain in organizing this event at our Bahrain Branch.

Thanks to my colleague Philippe Geril, executive director of EUROSIS office for supporting the event and for his time. Last but not least thanks to all authors without whom the conference would not be a successful conference.

Professor Dr Marwan Al-Akaidi
General Conference Chair
EUROSIS – M. East Chair
Vice-Rector Academic Affairs
Arab Open University
HQ- Kuwait

CONTENTS

SCIENTIFIC PROGRAMME

DATA SIMULATION

Ghana Classroom Interactions Evaluated Through Networks and Simulation

Liz Johnson

Complex Systems Institute
University of North Carolina at Charlotte
8813 Covey Rise Court
Charlotte, NC 28226 United States
Ljohnson1@carolina.rr.com
1704/293-1482

Key words: Ghana education, classroom climate, network analysis, interactions, ABM

Abstract

Ghana faces daunting challenges in education with a national focus to improve achievement despite high drop out rates and lack of resources. Identifying the characteristics of various classroom network structures and corresponding processes can serve as the means to determine the ideal relationships and balance between classroom environment, educational opportunity, and achievement. The classroom environment survey CES was administered to six Ghanaian classrooms, as well as observations on climate conducted in 4 of the classrooms. A network analysis approach and assessments of classroom climate provided the means to map and determine conditions and processes. This research presents a comprehensive methodological approach of qualitative, quantitative, network analysis, and agent-based model. The model provided the means to unravel classroom interactions and processes into fundamental relational properties of learning environments, which include instructional, emotional, and behavioral interactions. The classroom network structures supported a positive environment and high number of instructional and emotionally supportive interactions in all the classrooms. Overall, the research contributes to the knowledge about the structure, interdependencies, and complexity of interaction processes in classrooms that impact achievement.

1. Introduction

Improving educational outputs is a critical way to raise a country's productivity and thus standard of living (Glewwe & Jacoby, 2003). Over twenty nine million primary school aged children that do not attend school reside in sub-Saharan Africa (Federal Ministry for Economic Cooperation and Development, 2015). Despite a national focus on education countries like Ghana suffer student dropout rates as high as 15% at the primary level and 35% at the junior high-level (Akyeampong & Ananga 2014). Irregular school attendees primarily come from the lowest income households and students often stay out of school because of economic strains. Adding to the challenge is many Ghanaian families simply cannot afford the cost of fees for their children to attend public schools. Many irregular school attendees withdraw transiently and end up dropping out permanently. Research on the causes of student drop out in Ghana has been attributed to poor quality in teaching methods (Akyeampong & Ananga, 2014, para. 1; Kadingdi, 2006). Also according to Apaganda, the Northern Regional Director of Education, the quantity of students in Ghana classrooms range from 80 to 120 causing poor performance on year-end achievement tests (GhanaWeb, 2014).

Hadžikadić (2014) warned if you do not understand enough about a problem to solve it, intervening is meddling. Not only do we *not* know enough about student engagement and achievement but also we know little about what really goes on in classrooms, in relation to interactions and processes impact on achievement. Westaby (2012) asserted, "goal achievement and performance results from network interactions" (p. 205). So classroom interactions have variable effects on student performance, achievement, and dropping out (p. 56). Mapping classrooms with teachers and students represented as nodes with varied levels of volition, then connect them by links representing interactional causal sequences, represents unique patterns of interdependencies in the social space classrooms occupy (Carolan, 2014; Westaby, 2012). Interactions can be defined as “mutual or reciprocal action or influence” (Merriam Webster Online, n.d.), as well as “the quality, state, or process of two or more things acting on each other (Biology Online, 2014). Salmon (1984) defined processes as entities that display a consistent organization of structure over time (p. 144).

If you do not know why a classroom is the way it is a approach offers a unique strategy to understand what causes a classroom to be the way it is (Bellinger, 2014; Carolan, 2014) and also how to better understand *how* classroom processes relate to interactions and causality at the micro and macro levels of analysis. Determining an accurate architecture facilitates increased understanding of the interdependent actors and environment because teachers and students “both shape and are shaped by the context in which they interact” (Carolan, 2014, p. 10). A networked relational approach has the “potential to reveal new patterns of behavior not captured through traditional means” and underlying determinants of high performance and low performance (Westaby, 2012, p. 208).

2. Research context

Whereby interactions in classrooms have been extensively researched, there is still limited knowledge about the structure, interdependencies, and complexity of interaction processes in classrooms. Qualitative, quantitative, social network analysis, and agent-based modeling provides a means to further examine *teacher-student* interactions, *student-student* interactions, *teacher-whole classroom* interactions, and also linkages to essential instructional, emotional, and behavioral processes. A multi-method approach can provide specific insight into the characteristics of network dynamics, potential causality, and can allow focus on vital patterns of interactions that create desired outcomes. Consequently, identifying the characteristics of various classroom network structures and corresponding processes can guide educators into in determining the optimal relationships between classroom environment, educational opportunity, and achievement.

Furthermore Hadžikadić argued (2014), you cannot understand inherent complexity unless you map out and understand the network behind the system. Networks can be defined as interactions between parts with social configuration and with distinct and discernable structures (Scott, 2009). A school classroom is a complex adaptive system; and thus contextual depending on the environment and subsumed by “the nature of the ever-changing interactions among the constituent parts” (Obrien, Hadžikadić, & Khouja, p. 11). A classroom is a complex intricate wiring of networks that define the interactions between students and teachers, which directly impact behavior. A network analysis and ABM are the means to characterize a complex adaptive system like a classroom (Hadžikadić, 2014).

A classroom, examined as a whole interdependent system of relations, has properties distinguishable from those of individual students and teachers, which further determine the behaviors of the interrelated parts (Scott, 2009). Consequently, not only does the field of social forces determine aggregate classroom behavior but also it is the perceived environment that really matters to the group behavior and molds outcomes (Lewin, 1936; Scott, 2009, p. 11). For example, the social reality in a classroom is co-constructed by teachers and students based on their perceptions and experiences of the contexts in which they take action (Scott, 2009, p. 11). Consequently, it is critical to assess classroom environment, which is assessed in this research.

3. Research problem

A major national focus in Ghana is “improving the quality of learning and teaching,” as well as reducing student dropout rate in order to improve achievement scores (To Be Worldwide, 2011, p. 6). According to Mashburn et al. (2008) improving educational outcomes should include investigating the nature and effectiveness of *teacher-student* interactions (CASTL, 2013). Developmentally, middle school and high school students report interactions with teachers as “frequently” unsatisfying, unmotivating, and lacking as a supportive relationship (CASTL, 2013, p. 1). Student dropout is considered at crisis level in Ghana, potentially a resulting from unengaged students (Akyeampong & Ananga, 2014). Increased understanding of classroom climates and interactions are needed to improve classroom-level learning environments to reduce adolescent withdrawal from educational pursuits and improve achievement (CASTL, 2013; Rolland, 2012). This research examines the dynamics of change at the classroom level to determine agent processes, as well as collective influences that impact academic achievement and outcomes within adolescent learning environments.

4. Research questions

1. Given classroom climate, what are the interrelated network structures of agents and processes in classrooms and outcomes from simulation?

2. What is the frequency, duration, and depth of classroom interactions and what impact do they have on achievement?

5. Methodology

Qualitative, quantitative, and network analysis approaches were employed as the means to provide data for an agent-based model. With this approach the researcher was able to observe the dynamics of agents, the collective, and the interrelating environment in a more comprehensive way than using a single approach. Alternate complexity methodologies could have been used but network analysis and an ABM simulation was suited for capturing relationship connections and interaction processes from heterogeneous agents of students and teachers (Ghorbani, et al., 2014; Johnson, 2015a; Johnson, 2015b).

A multiphase design was implemented in order to explore interactions through a series of sequentially aligned interations from qualitative, quantitative, and complexity methodologies. The introduction of each additional methodology built upon previous results, as to transform static qualitative and quantitative data into first, a dynamic network analysis of relationships, and next into an agent-based model simulation of networked relationships and parameterization of qualitative and quantitative data (Johnson, 2015a).

5.1. Qualitative.
A survey of classroom environment and observations were conducted. The Classroom Environment Scale (CES), a 90-question survey, provided an assessment of standardized classroom experiences from students and teachers. The tool assessed the effects of interactions and characteristics of the classroom environment (Trickett & Moos, 2002). CES has been used extensively internationally to determine classroom climate (Trickett & Moos, 2002). CES captures aggregate scores on classroom characteristics and how a student or teacher personally experiences their classroom and their relationships (Trickett & Moos, 2002). CES is based on the perspective a classroom is a dynamic system that includes the following domains:

Relationship Dimensions	
1. Involvement	the extent to which students are attentive and interested in class activities, participate in discussions, and do additional work on their own
2. Affiliation	the friendship students feel for each other, as expressed by getting to know each other, helping each other work with homework, and enjoying working together
3. Teacher Support	the help and friendship the teacher shows toward students; how much the teacher talks openly with students, trusts them, and is interested in their ideas
Personal Growth/Goal Orientation Dimensions	
4. Task Orientation	the emphasis on completing planned activities and staying on the subject matter
5. Competition	how much students compete with each other for grades and recognition and how hard it is to achieve good grades
System Maintenance and Change Dimensions	
6. Order and Organization	the emphasis on students behaving in an orderly and polite manner and on the organization of assignments and activities
7. Rule Clarity	the emphasis on establishing and following a clear set of rules and on students knowing what the consequences will be if they do not follow them; the extent to which the teacher is consistent in dealing with students who break rules
8. Teacher Control	how strict the teacher is in enforcing the rules, the severity of punishment for rule infractions, and how much students get into trouble in the class
9. Innovation	how much students contribute to planning classroom activities, and the extent to which the teacher uses new techniques and encourages creative thinking

Figure 1. CES Subscales and Descriptions (Trickett & Moos, 2002)

Additionally, the Classroom Assessment Scoring Systems (CLASS) was used, as a basis for the Observation Mapping Checklist of classroom interactions is the second source of data collection (see Appendix 1). The CLASS instrument was adapted into an Observation Mapping Checklist in order to capture single interactions based on objective behavioral markers. A trained CLASS observer scored the Observation Mapping Checklist in classrooms during the half hour sessions. Additionally, the observations sessions were videotaped via IPad as to verify scoring of the Observation Mapping Checklist.

5.2. Quantitative.
Preliminary regression analyses were run, though all student achievement scores were not fully available at the time of this report. Ghanaian year-end achievement scores of 2014 served as the dependent variable. The independent variables included student gender, years at school, months at school, student age, a model index of all CES variables, and each CES variables separate.

5.3. Complexity methodologies.
A network analysis in Gephi was run based on CES and classroom observations. An agent-based model in Netlogo was developed based off the qualitative, quantitative, and network data.

6. Sample

The research used non-probability sampling from six junior high and senior high classes. North Ghana rural schools were selected on the basis that the known and unknown characteristics of the sample would best represent the population of rural junior high and high school classes (O'Sullivan, Rassel, &

Berner, 2008). The sample included three junior high and three high schools with a total of two hundred and fourteen students and six teachers participating. There were five mathematics classes that participated and one science class.

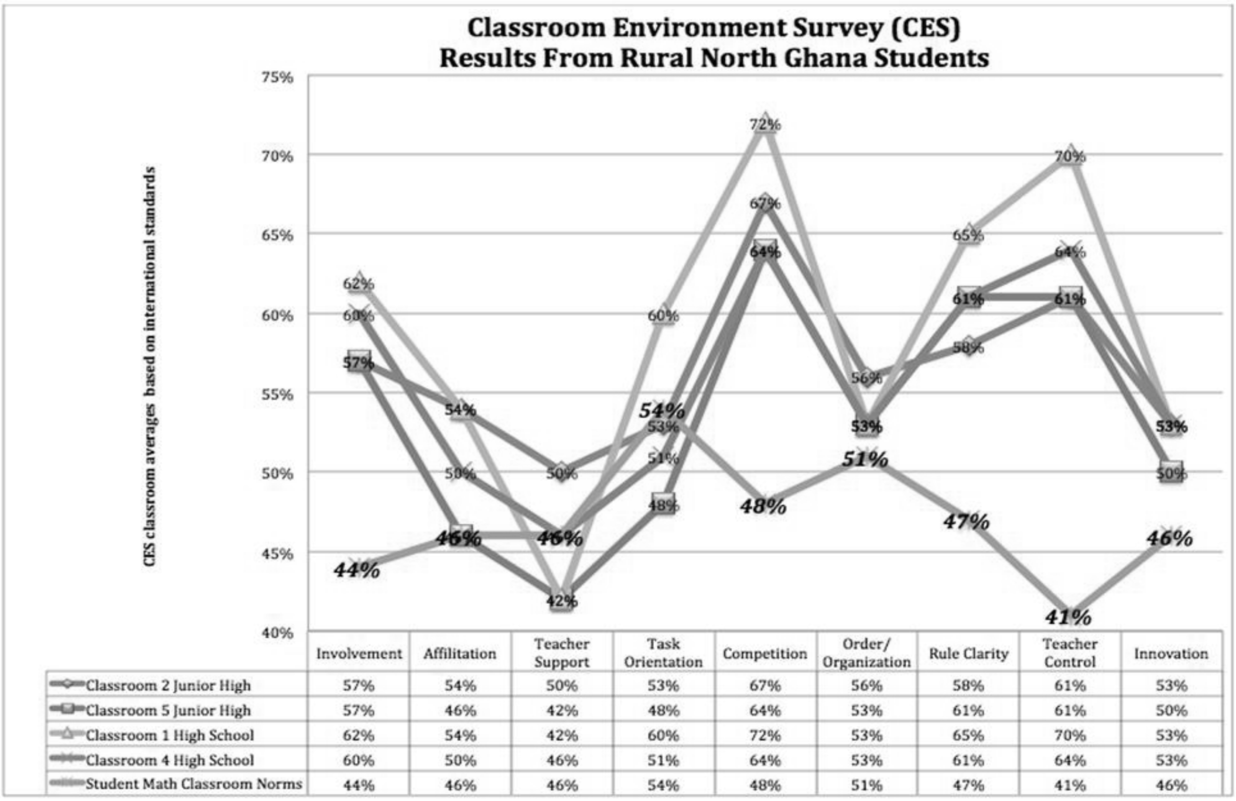
Ghanaian students and teachers were administered the CES classroom environment survey the second week in July of 2014. This was less than a week before year-end achievement tests were administered. The Ghanaian teachers were not given advanced notice of survey administration and observations. All six teachers willingly agreed to participate. The administration of the CES survey took place before the CLASS classroom observations. Ghanaian students were given instructions in both English and their native language. Student participants in the CES survey took longer than the standard time of fifteen minutes (Trickett & Moos, 2002) with some students taking up to twice as long. This can be attributed to the fact English is a second language of a majority of students and the CES survey format posed challenges in translating. Observation sessions were conducted in four mathematics classrooms for a half hour (Johnson, 2015a).

7. Results

7.1. Classroom Analysis

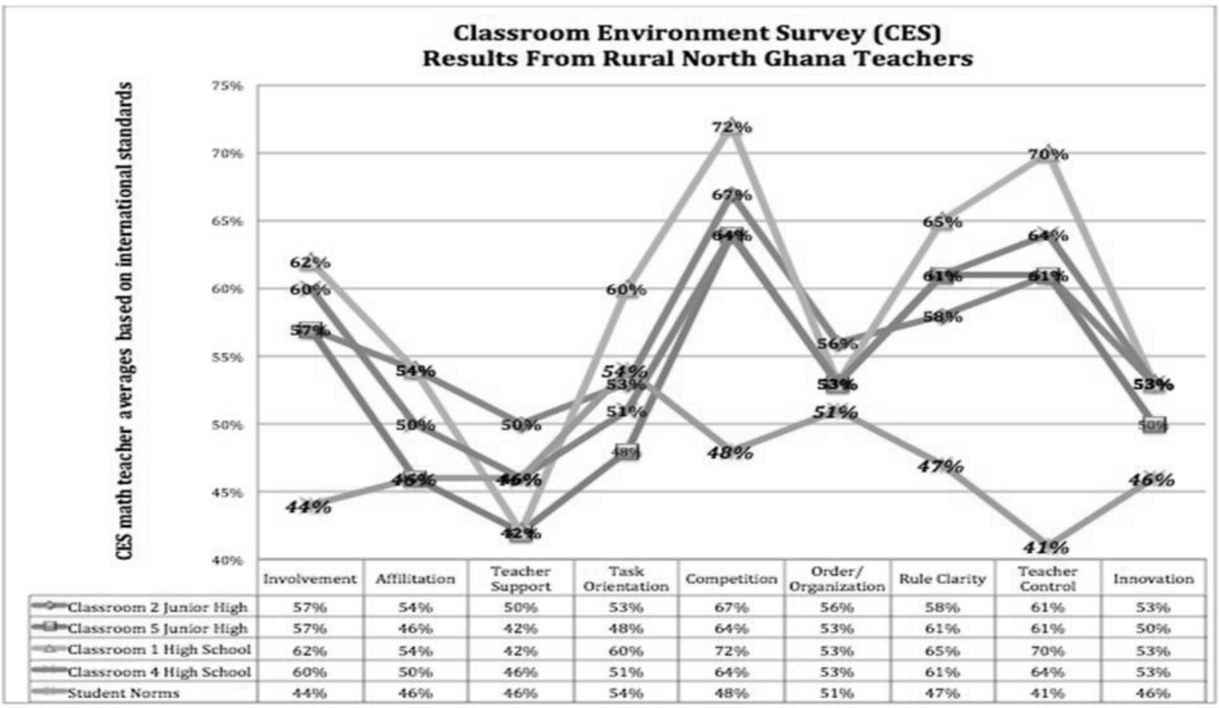
The classroom environments assessed by students scored average or above average mathematics classroom norms in the areas of affiliation, teacher support, and order/organization. (See Figures 2 & 3 below for observed classroom CES). The classrooms were substantially above average in the areas of involvement, competition, rule clarity, and teacher control. The classrooms were slightly above the average in innovation and both below and above the average in teacher support and task orientation (Johnson, 2015a). The science classroom assessed above norms for involvement, task orientation, competition, order/organization, and teacher control and below for teacher support and teacher control. Mathematics teachers assessed their classroom environments as average or above average in involvement, affiliation, task orientation, order/organization. The teachers assessed their classrooms above average in competition, teacher control, and innovation. Teacher support, task orientation, order/organization, and rule clarity ranged from below average, average, and above average. Competition and innovation were above average. The science teacher assessed their classroom above norms in involvement, affiliation, task orientation, completion, order/organization, rule clarity, and innovation and below on teacher support and teacher control.

From the students' perceptions, the classroom environments were characterized as highly competitive, with high teacher control enforcing rules, and a major emphasis on students following rules. Yet, there was a high level of student involvement, whereby students reported being attentive and actively engaged in class activities. Teachers also characterized their classrooms as competitive and high in teacher control. From CLASS observation measures, all the classrooms exhibited a positive climate reflecting a positive emotional connection between teachers and students and students and students. For example, teachers and students consistently showed respect for one another and there were frequent positive communication between teachers and students. The teachers communicated positive expectations to students (Johnson, 2015a). Positive affect was demonstrated via teacher enthusiasm, smiling, and laughter (Teachstone Training LLC, 2012; Trickett & Moos, 2002). The results reveal the context, whereby "both (students and teachers) shape and are shaped by the context in which they interact" (Carolan, 2014, p. 10).



	Involvement	Affiliation	Teacher Support	Task Orientation	Competition	Order/ Organization	Rule Clarity	Teacher Control	Innovation
Classroom 2 Junior High	57%	54%	50%	53%	67%	56%	58%	61%	53%
Classroom 5 Junior High	57%	46%	42%	48%	64%	53%	61%	61%	50%
Classroom 1 High School	62%	54%	42%	60%	72%	53%	65%	70%	53%
Classroom 4 High School	60%	50%	46%	51%	64%	53%	61%	64%	53%
Student Math Classroom Norms	44%	46%	46%	54%	48%	51%	47%	41%	46%

Figure 2. Classroom Environment Survey CES outcome graphs from aggregate student data



	Involvement	Affiliation	Teacher Support	Task Orientation	Competition	Order/ Organization	Rule Clarity	Teacher Control	Innovation
Classroom 2 Junior High	57%	54%	50%	53%	67%	56%	58%	61%	53%
Classroom 5 Junior High	57%	46%	42%	48%	64%	53%	61%	61%	50%
Classroom 1 High School	62%	54%	42%	60%	72%	53%	65%	70%	53%
Classroom 4 High School	60%	50%	46%	51%	64%	53%	61%	64%	53%
Student Norms	44%	46%	46%	54%	48%	51%	47%	41%	46%

Figure 3. Classroom Environment Survey CES outcome graph from teacher data

7. 2. Network Analysis

A network analysis of the four classrooms that were observed was conducted with Gephi network software for the second phase of the research. Data was collected from half hour classroom observation sessions after the CES survey had been administered. Compositionally, the network graphs of the four classrooms (see Figures 4, 5, 6, & 7 below) showed how heterogeneous agents of teachers and student can be represented by *teacher-student* interactions, *student-student* interactions, and *teacher-whole class* interactions. There could also be variations of students in varied groups and subgroups. Limited number of interactions during the observations sessions was student initiated. Structurally, the teachers topped of the hierarchy of the networks initiating and controlling most of the interactions communication-wise. The four classroom network graphs depicted simple structural and compositional components (Johnson, 2015a).

In contrast, the network graphs depicted functional complexity in varied degrees. Operationally, the dynamics in the temporal sequences of classroom processes fit into three separate networks (see Figures 4, 5, 6, & 7). The first graph on the left (of three) in each classroom network model includes *teacher to individual* student interactions and *student to student* interactions. The second middle graph in each classroom network shows the whole class responding to the teacher. The third graph on the right in each classroom network model depicts the processes the teacher engaged in with the whole class (Johnson, 2015a).

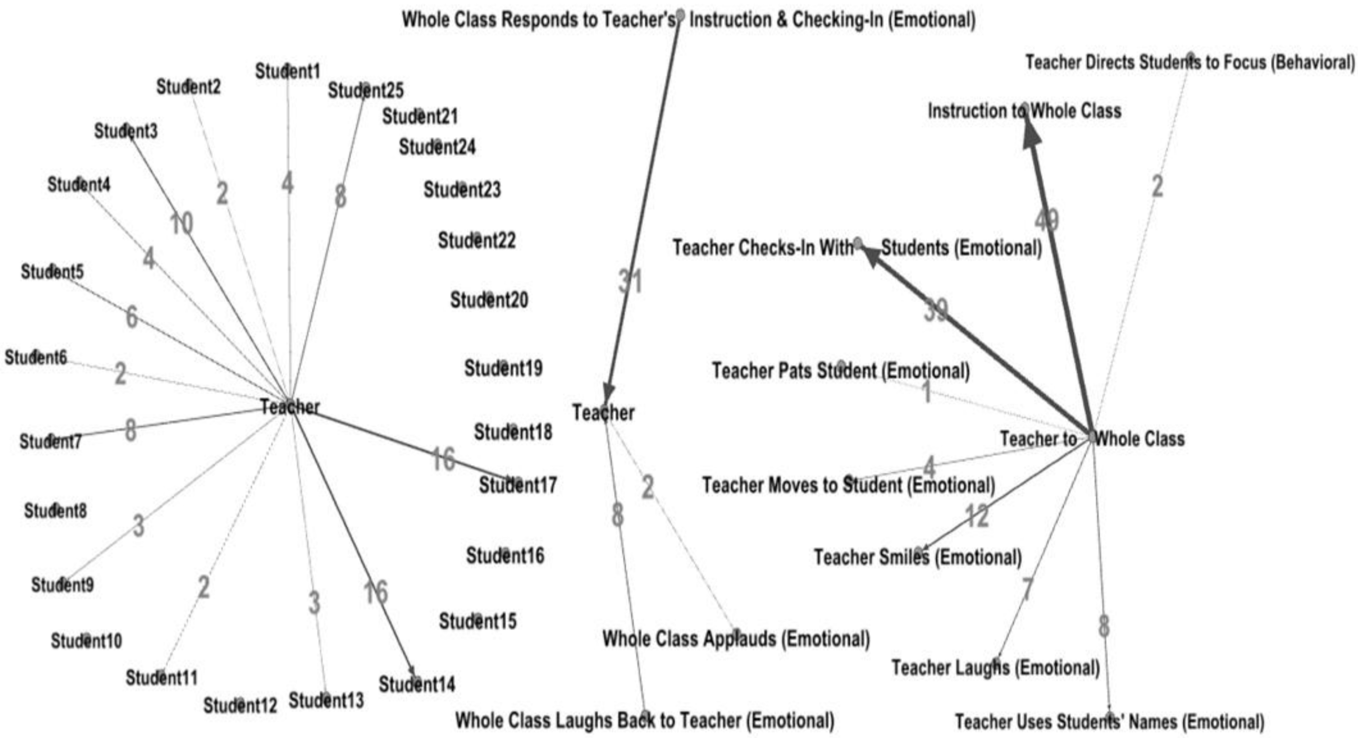


Figure 4. Classroom #2 junior high mathematics classroom

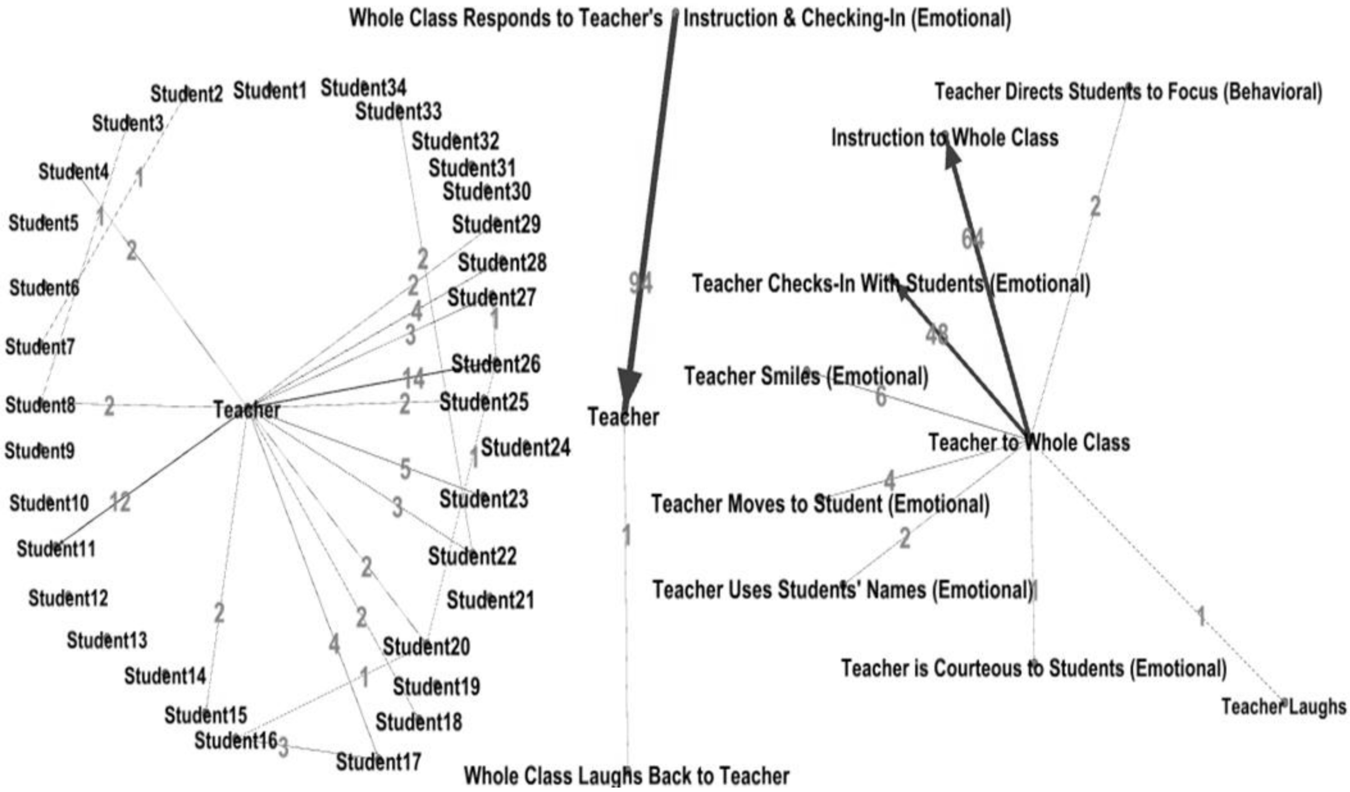


Figure 5. Classroom #5 junior high mathematics classroom

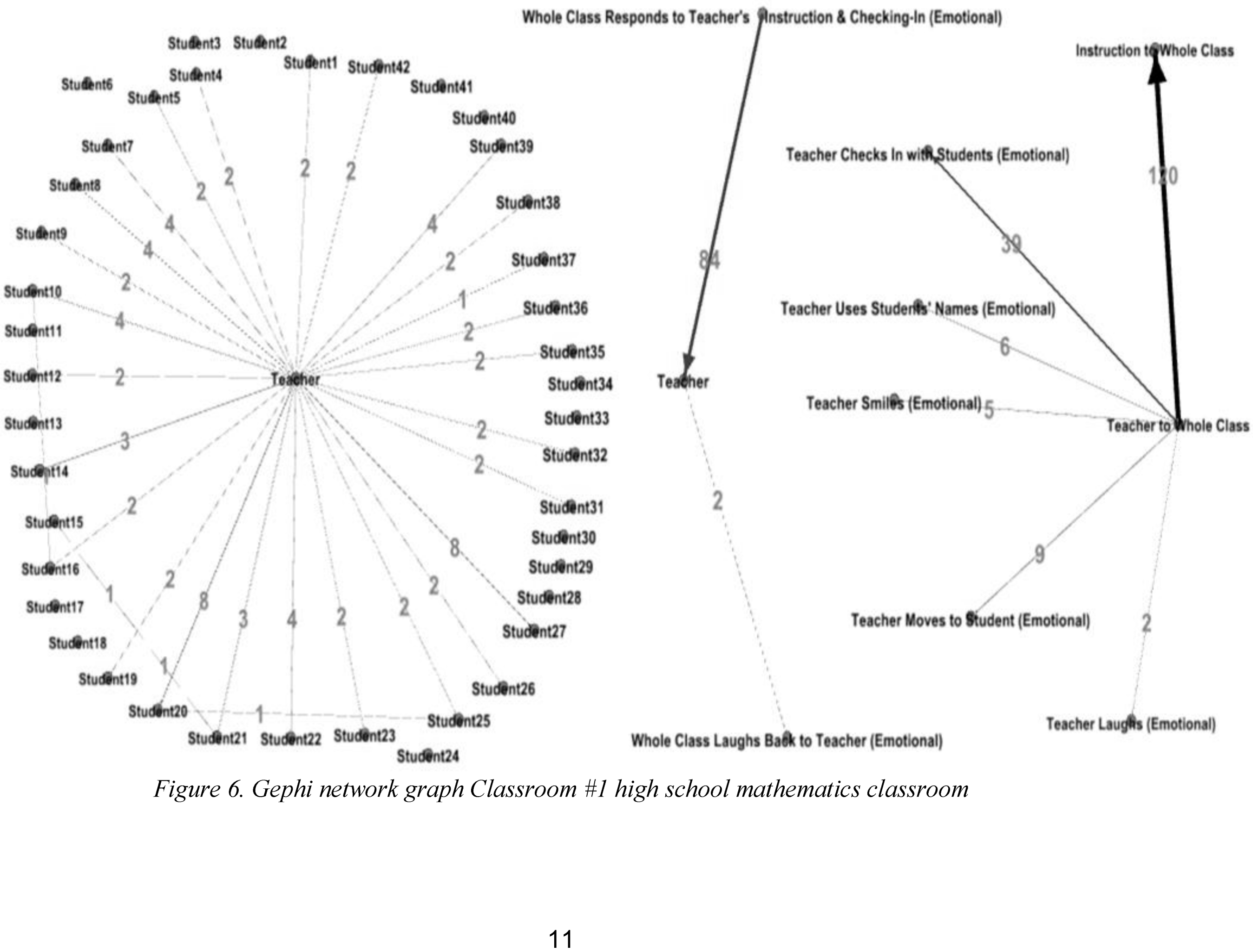


Figure 6. Gephi network graph Classroom #1 high school mathematics classroom

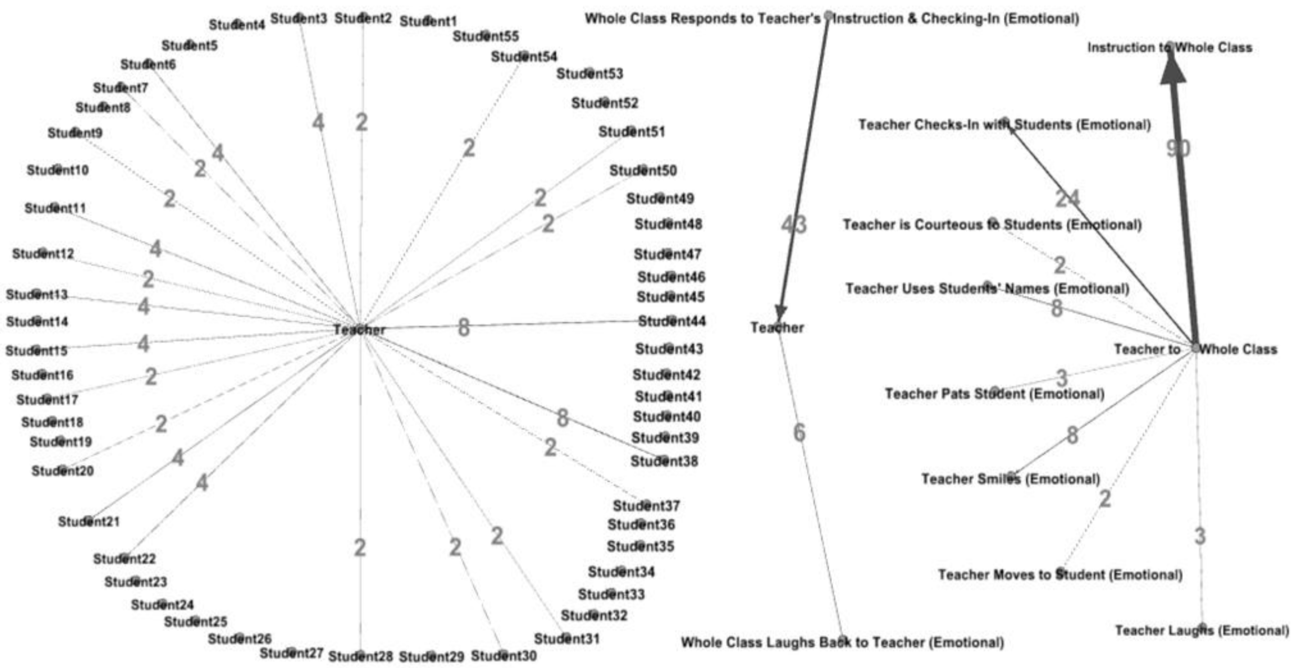


Figure 7. Gephi network graph Classroom #4 high school mathematics classroom

7.3. Phase Three Quantitative Analysis of Agent and Process Interactions from Observation Mapping

A quantitative analysis of agent and process interactions from the Observation Mapping Checklist (see Appendix A) of classroom interactions was conducted on four of the mathematics classrooms (Johnson, 2015a). Only two classrooms observed had behavioral management issues. The behavioral interactions accounted for just 1% to 3% of the total interactions. Two of the observed classrooms had *student-student* interactions, which ranged from 1% to 3% of total interactions. Teachers in the observed classes directed interactions for over half the 30-minute observation sessions ranging from 55% to 69% of the observation session. Instructional interactions varied from 67% to 85% of total interactions. Overall, the results demonstrated a majority of classroom interaction in each classroom was devoted to instruction, support of instruction, and sustaining a positive classroom environment (Johnson, 2015a). For more details on interaction see Appendix B.

1. High number of dynamic interactions.

Classroom #2 junior high had 245 total interactions.

Classroom #5 junior high had 289 total interactions.

Classroom #1 high school had 350 total interactions.

Classroom #4 high school had 263 total interactions.

2. Predominance of interactions were instructional from *teacher to whole class*.

Classroom #2 junior high had 49 *teacher to whole class* instructional interactions.

Classroom #5 junior high had 64 *teacher to whole class* instructional interactions.

Classroom #1 high school had 120 *teacher to whole class* instructional interactions.

Classroom #4 high school had 90 *teacher to whole class* instructional interactions.

3. High number of interactions were emotional checking-in from *teacher to whole class*.

Classroom #2 junior high had 24 *teacher to whole class* emotional support interactions.

Classroom #5 junior high had 48 *teacher to whole class* emotional support interactions.

Classroom #1 high school had 39 *teacher to whole class* emotional support interactions.

Classroom #4 high school had 24 *teacher to whole class* emotional support interactions.

4. High number of interactions were *whole class to teacher* responding to instructional and emotional checking-in.

Classroom #2 junior high had 31 *whole class to teacher* response interactions.

Classroom #5 junior high had 94 *whole class to teacher* response interactions.

Classroom #1 high school had 84 *whole class to teacher* response interactions.

Classroom #4 high school had 43 *whole class to teacher response* interactions.

7.4. Phase Four Regression Analyses

Preliminary regression analyses were run, though all student achievement scores were not fully available at the time of this report. The 2014 student achievement scores ranged from 13 to 74 in the mathematics and science classrooms. Ghanaian year-end achievement scores of 2014 served as the dependent variable. The independent variables included student gender, student age, a model index of all CES variables, and each CES variables separate (see Figure 1 above).

Model Index:	$p < .01$
Affiliation:	$p < .05$
Competition:	$p < .05$
Order and organization:	$p < .01$
Rule Clarity:	$p < .001$
Teacher Control:	$p < .05$

7.5. Phase Five Agent-Based Model

The data-drive ABM is in process and has been developed based on outcomes from the previous methodologies. The observations and network analysis allowed for the characterization of agent interactions into categories of instructional, emotional, and behavioral interactions between teachers and students, and students to students in varied groupings. The analysis of type and total number of interactions per observation session provided parameters. The ABM included demographic data from the surveys.

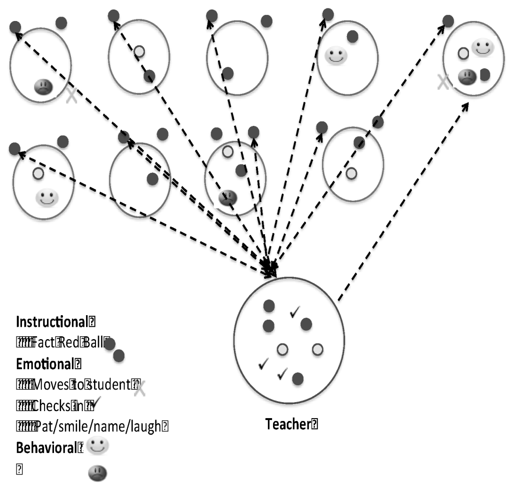


Figure 8. ABM interactions key

8. Discussion

Cultural context matters in research. The Ghanaian teachers exhibited a strict, serious, and militaristic style. Also, they were highly enthusiastic about the instructional review process and subjects of review. The teachers provided numerous and diverse instructional opportunities. Whether in mathematics or a science class, the teachers provided a proportionally high number of emotional interactions of positive affirmations and encouragement checking-in. The Ghanaian students knew what was required of them in an instructional session and were expected to respond individually or as a whole group when called on. There were minimal behavioral disruptions from students. Teachers were effective in redirecting students to refocus and stay on task. Additionally, the classrooms showed a positive climate with positive affect, positive communication, and respect that included interactions of laughing, smiling, clapping, patting, courtesy, use of student names, and teacher movement toward students. The environments were highly competitive (Teachstone Training LLC, 2012). Of note is 36% to up to 58% of the students in classrooms had no one-one one interactions with their teacher during the observed review session (Johnson, 2015a).

Overall, the positive climate and classroom network structures allowed for maximizing instructional time and active engagement by students. Teachers' instruction did not provide evidence of poor quality of instruction as reported by Akyeampong and Ananga (2014). The quantity of students in the classrooms in the sample ranged from 15 to 56, which is in contrast to national claims of overcrowding classrooms with 80 to 120 students (Akyeampong & Ananga, 2014). However, the observed teachers rigidly provided *all* the structure for the classes. Teachers did not provide adequate opportunities for student to engage in leadership and autonomy to meet the developmental and social needs of adolescents (Teachstone Training LLC, 2012). Ghanaian students in both mathematics and science classes reported their environment as highly competitive, which despite a positive climate can still contribute to student dropout (CASTLE, 2013).

From the regression analysis the data suggests older students perform lower on achievement scores as compared to younger students. The data suggests that females do not perform lower than their male counterparts on achievement, contrary to prevailing research results on Sub-Saharan female students (Dickerson, McIntosh, &Valente, 2012). The data suggests affiliation, competition, rule clarity, teacher control, and a combined index of all CES variables is significant in in relation to achievement. Some of the preliminary results seem to be in conflict with traditional thinking in educational research. For examples, teacher support and non-competitive classroom climates are critical to student success in achievement (CASTLE, 2013).

The research revealed that teachers had high expectations on student engagement and focus on comprehension. Conducting the research so close to achievement tests allowed for assessment of environments that were highly dynamic and instructionally driven. The research revealed there were numerous opportunities to learn from instruction and ask questions for further understanding. Yet there was no guarantee students comprehended, absorbed, or retained the instructional material. The Ghanaian teachers exhibited a high number of emotionally supportive interactions and *no* interactions that contribute to a negative classroom climate. Though teachers directed most of the interactions, they were encouraging, affirming, supportive, and enthusiastic through a variety of emotional strategies checking in with students. This further disputes a claim of poor quality in instruction, at least in the observed classrooms (Johnson, 2015a).

The combination of methodologies led to the development of a hybrid bipartite dynamic graph in its final form as an ABM. Preliminary outcomes from the ABM showed there is no way to discern if a student actually comprehends the instruction and can recall for an achievement test, yet students were exposed to as many 283 instructional interactions (81% of total interactions) in a half hour period with an additional 39 emotional interactions (11% of total interactions) checking in with students on instruction. Both interaction categories combined accounted for 92% of all interactions in a single classroom. Though there were no measures on the depth of content understanding during review sessions, there were more than adequate opportunities to learn and ask questions in a positive, dynamic, and supportive climate in all the classrooms. Overall, the positive classroom environment, the *type* of and *level* of instructional, emotional, and behavioral interactions impacted academic outcomes. Previous year-end achievement scores were not available, nor were attendance records.

9. Conclusions

In conclusion, there are a great number of challenges in Ghanaian classrooms to still be researched. The results did not provide adequate solutions for eradicating the Ghanaian drop out problem or improvement of academic scores. Additionally, the research was limited in sample size, generalizability, attendance records, and availability of achievement scores for all students.

However, this research provided a foundation for increased understanding, by researching classroom relations in regard to detailed interactions from both the students/teachers perceptions and observations. Assessing classroom climate and the proportion of instructional, emotional, and behavioral interactions can enhance a teacher's ability to empower and influence students in productive ways. The visual and simulation representations of classroom interactions can *unhide* some of the basic interrelationships between agents and processes, as demonstrated in this research and serve as a means to appraise and improve effectiveness in shaping the desired learning environment (Trickett & Moos, 2002; Johnson, 2015). Further development of the ABM is needed, as well as the means to provide a pre-test measurement to strengthen the model.

According to Brophy and Everston (2010), low social economic status (SES) students in the United States need over teaching, frequent teacher talk, patience from teachers, and frequent repetition to the point of over-learning. The low SES of rural northern Ghanaian students is important to consider in the context of a developing country, and the fact that in this research the Ghanaian teachers employed all the techniques from Brophy and Everston study to be successful with low SES students in a developed country.

References

Akyeampong, K., & Ananga, E. (2014). Reducing school dropouts through inclusive approaches to education in Ghana. Retrieved from http://www.eldis.org/index.cfm?objectid=132A5AA5-9E0B-0456-5BB24B4C5495B04E#.VKluimTF_0Q

Biology Online. (2005). Interaction, from http://www.biology-online.org/dictionary/Interaction

Brophy, J. E. & Everston, C. M. (2010). Teaching young children effectively. *Journal of Classroom interactions. (45)*1.

CASTL (2013). Measuring and improving teacher-student interactions in PK-12 setting to enhance students' learning. Retrieved November 20, 2013, 2013, from http://curry.virginia.edu/uploads/resourceLibrary/CLASS-MTP_PK-12_brief.pdf

Dickerson, A., McIntosh, S. & Valente, C. (2012). An analysis of the gender gap in mathematics in Africa'. Mimeograph, University of Bristol and University of Sheffield.

Federal Ministry for Economic Cooperation and Development. (2015). Ghana, from http://www.bmz.de/en/what_we_do/countries_regions/subsahara/Other-key-sectors-education-health-rural-development/index.html

GhanaWeb. (2014, September 26). Ghana is producing quantity education - Educationist. *Regional News, 327679*. Retrieved from http://www.ghanaweb.com/GhanaHomePage/regional/artikel.php?ID=327679

Glewwe, P. & Jacoby, H. (2003) Student achievement and schooling choice in low-income countries evidence from Ghana. The Journal of Human resources, XXXVIII:591-61.
Hadžikadić, M. (2014). *Network science lectures*. Charlotte, NC: University of North Carolina Charlotte.
Johnson, L. (2015a). A network context for observing and mapping of Ghana mathematics classroom. *International Journal of Humanities and Social Science*, *(5)*2.
Johnson, L. (2015b). Complexity modelling and application to policy research. In R. Geyer & P. Cairney (Eds.), *Handbook on Complexity and Public Policy*. Cheltenham Glos, UK: Edward Elgar.
Interaction [Def. 1]. (n.d.). *Merriam Webster Online*, Retrieved January 4, 2015, from http://www.merriam-webster.com/dictionary/interaction
O'Sullivan, E., Rassel, G. R., & Berner, M. (2008). *Research methods for public administrators*. New York, NY: Pearson.
Scott, J. (2009). *Social network analysis a handbook* (2nd ed.). Thousand Oaks, CA: Sage.
Solomon, W. (1984). *Scientific explanation and the causal structure of the world*. Princeton, NJ: Princeton University Press.
Teachstone Training LLC. (2012). *Secondary CLASS master code justifications*. Charlottesville, VA: Teachstone.
To Be Worldwide. (2012). A brief history of the Ghanaian educational system. Retrieved from http://www.tobeworldwide.org/index.php/component/rsfiles/download?path=General/history.pdf&Itemid=169
Trickett, E. J., & Moos, R. H. (2002). *A social climate scale* (3rd ed.). Menlo Park, CA: Mind Garden, Inc.
Westaby, J. D. (2012). *Dynamic network theory how social networks goal pursuit*. Washington, DC: American Psychological Association

Tables

INTERACTION TYPOLOGIES 25 Students Classroom #2 Jr. High Math	# Of Interactions In ½ Hour	% Of Total Interactions 245
INSTRUCTIONAL INTERACTIONS	164	67%
Teacher to all students INSTRUCTING	49	20%
All students RESPOND To Teacher	31	13%
Teacher to 1 student INSTRUCTING	44	20%
1 Student to Teacher REPSONDING	40	16%
EMOTIONAL INTERACTIONS	81	33%
Teacher checking-in with all students	39	16%
Teacher moving toward students	4	2%
Teacher pat student on shoulder	1	<1%
Teacher laughs	7	3%
Teacher smiles	12	5%
All students as group laugh	8	3%
All students as group clap	2	1%
Teacher use students names	8	3%
BEHAVIORAL MNG. INTERACTIONS by Teacher	0	0%
AGENT INTERACTIONS (no peer to peer interactions)	245	100%
Individual student to Teacher interactions	40	16%
All students as group interactions	41	17%
All teacher interactions	164	67%

Table 1 Classroom #2 junior high, agent and process interactions

INTERACTION TYPOLOGIES 36 Students Classroom #5 Jr. High Math	# Of Interactions In ½ Hour	% Of Total Interactions 289
INSTRUCTIONAL INTERACTIONS	217	85%
Teacher to all students INSTRUCTING	64	22%
All students RESPOND To Teacher	94	32%
Teacher to 1 student INSTRUCTING	34	12%
1 Student to Teacher REPSONDING	25	9%
EMOTIONAL INTERACTIONS	63	21%
Teacher checking-in with all students	48	15%
Teacher moving toward students	4	1%
Teacher laughs	1	<1%
Teacher smiles	6	2%
Teacher use students names	2	1%
Teacher uses courtesy	1	<1%
All students as group laugh	1	2%
BEHAVIORAL MNG. INTERACTIONS by Teacher	2	1%
AGENT INTERACTIONS	289	100%
Individual student to Teacher interactions	25	9%
All students as group to Teacher interactions	95	33%
Student to student interactions	7	3%
All teacher interactions	162	<55%

Table 2 Classroom #5 junior high, agent and process interactions

INTERACTION TYPOLOGIES 43 Students Classroom #1 Sr. High Math	# Of Interactions In ½ Hour	% Of Total Interactions 350
INSTRUCTIONAL INTERACTIONS	283	81%
Teacher to all students INSTRUCTING	120	34%
All students RESPOND To Teacher	84	24%
Teacher to 1 student INSTRUCTING	42	12%
1 Student to Teacher REPSONDING	37	10%
EMOTIONAL INTERACTIONS	63	18%
Teacher checking-in with all students	39	11%
Teacher moving toward students	9	2%
Teacher laughs	2	1%
Teacher smiles	5	1%
Teacher use students names	6	2%
All students as group laugh	2	1%
BEHAVIORAL MNG. INTERACTIONS by Teacher	0	0%
AGENT INTERACTIONS	346	100%
Individual student to Teacher interactions	37	14%
All students as group to Teacher interactions	86	24%
Student to student interactions	4	1%
All teacher interactions	223	61%

Table 3. Classroom #1 high school, agent and process interactions

INTERACTION TYPOLOGIES 56 Students Classroom #4 Sr. High Math	# Of Interactions In ½ Hour	% Of Total Interactions 263
INSTRUCTIONAL INTERACTIONS	203	77%
Teacher to all students INSTRUCTING	90	34%
All students RESPOND To Teacher	43	16%
Teacher to 1 student INSTRUCTING	37	14%
1 Student to Teacher REPSONDING	33	13%
EMOTIONAL INTERACTIONS	56	21%
Teacher checking-in with all students	24	9%
Teacher moving toward students	2	1%
Teacher laughs	3	1%
Teacher smiles	8	3%
Teacher use students names	8	3%
Teacher pats student on shoulder	3	1%
Teacher uses courtesy	2	1%
All students as group laugh	6	2%
BEHAVIORAL MNG. INTERACTIONS by Teacher	4	2%
AGENT INTERACTIONS	263	100%
Individual student to Teacher interactions	33	12%
All students as group to Teacher interactions	49	19%
All teacher interactions	181	69%

Table 4. Classroom #4 high school, agent and process interactions

Appendix A

Observation Mapping Checklist

Date__________ School______________________________ Teacher____________________ #Students______

STUDENTS

TEACHER

Emotional Support
Moves to/away from Student
Smiling
Laughing
+ Comments
Respectful language: please, Thank-you
Checks in with students: recognition & affirmation of effort
Individual support
Reassurance & assistance

Behavioral Support
Attention to positive
Anticipates possible neg. behavior
Irritability
Anger
Yelling
Threats
Punishment
Physical control
Teasing
Bullying
Humiliation/sarcasm
Exclusionary behavior
Inflammatory/discriminatory/derogatory
Language or behavior

Instructional Support
Open-ended questions
Follow-up questions
Assistance
Hints
Prompting
Expansion
Clarification
Specific feedback
Encouragement of persistence

Notes: on classroom context & time of day.

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.

Appendix B

Classroom Interactions

In the first graph on the left in Figures 4, 5, 6, and 7, the number of interactions between the teacher and one individual student range from 0 to 16.
In junior high classroom #2 48% of students had no one-on-one interactions with the teacher.
In junior high classroom #5 56% of students had no one-on-one interactions with the teacher.
In high school classroom #1 36% of students had no one-on-one interactions with the teacher.
In high school classroom #4 58% of students had no one-on-one interactions with the teacher. Students that had no one-on-one interactions with the teacher tended to be seated in the back of the classroom and on the right hand side of the classroom from a teacher's vantage point. Only two of the classrooms had *student-student* interactions. They occurred when the teacher was writing on the board faced away from the class. There were a limited number *student-student* interactions. They lasted briefly, seemed to be mostly focused on the instruction, and were not disruptive.

The middle graph shows the whole class' responses to the teacher, while the third graph on the right shows how the teacher interacted with the whole class. *Teacher to whole class* and *whole class to teacher* interactions fit into categories of instructional, emotional, and behavioral interactional processes. *Teacher to whole class* interactions include instructional, emotional, and behavioral management. The emotional processes in the *teacher to whole class* interactions included checking-in with students by means of encouragement and affirmation, using courtesy, using student names, moving to a student's desk, smiling, laughing, and patting student on shoulder. *Whole class to teacher* interactions include responses to the teacher's instruction and emotional checking-in, as well as clapping. There were not any additional observable group behaviors that showed any patterns.

Interaction types were weighted on the basis of the total number of interactions in a category in relation to all interactions. Overall, the graphs highlight the thickness of the weighted edges between *teacher to whole class* interactions of instruction and checking-in, as well as the edges between the *whole class to teacher* responses to teacher directed instruction and checking-in.

Additionally, the network graphs reveal some glimpses into unraveling the functional complexity in relation to nomic complexity in the classrooms. At least during end of school review time period, the interrelationships showed:

In the first graph on the left in Figures 4, 5, 6, and 7, the number of interactions between the teacher and one individual student range from 0 to 16.
In junior high classroom #2 48% of students had no one-on-one interactions with the teacher.
In junior high classroom #5 56% of students had no one-on-one interactions with the teacher.
In high school classroom #1 36% of students had no one-on-one interactions with the teacher.
In high school classroom #4 58% of students had no one-on-one interactions with the teacher. Students that had no one-on-one interactions with the teacher tended to be seated in the back of the classroom and on the right hand side of the classroom from a teacher's vantage point. Only two of the classrooms had *student-student* interactions. They occurred when the teacher was writing on the board faced away from the class. There were a limited number *student-student* interactions. They lasted briefly, seemed to be mostly focused on the instruction, and were not disruptive.

The middle graph shows the whole class' responses to the teacher, while the third graph on the right shows how the teacher interacted with the whole class. *Teacher to whole class* and *whole class to teacher* interactions fit into categories of instructional, emotional, and behavioral interactional processes. *Teacher to whole class* interactions include instructional, emotional, and behavioral management. The emotional processes in the *teacher to whole class* interactions included checking-in with students by means of encouragement and affirmation, using courtesy, using student names, moving to a student's desk, smiling, laughing, and patting student on shoulder. *Whole class to teacher* interactions include responses to the teacher's instruction and emotional checking-in, as well as clapping. There were not any additional observable group behaviors that showed any patterns.

Interaction types were weighted on the basis of the total number of interactions in a category in relation to all interactions. Overall, the graphs highlight the thickness of the weighted edges between *teacher to whole class* interactions of instruction and checking-in, as well as the edges between the *whole class to teacher* responses to teacher directed instruction and checking-in. (Johnson, 2015)

Data and Control Flow Visualization by Transforming Software into Schematic Diagrams

Nils Potthoff, Christoph Brandau, Carsten Gremzow, Dietmar Tutsch
University of Wuppertal
email: {potthoff,brandau,gremzow, tutsch}@uni-wuppertal.de

KEYWORDS

Computer software, software engineering, multiprocessor, software visualization, control and data flow

ABSTRACT

The development of software is still largely based on the sequential programming paradigm. This however does not fit modern multi-core architectures. For optimization of sequential code it is auxiliary to visualize the internal control and data flow. Well-Known data and control flow graphs are ineligible due to their complexity. Dynamic program characteristics which occur through use of pointers aggravate this problem. In this paper we propose a transformation of sequential source code based on dynamic software analysis. The program is divided into parts with static dependencies among one another. These parts can be visualized with electronic circuit diagram layout techniques. The quality of generated layouts is analyzed and multiple improvements are devised leading to a significant simplification of the resulting schematics.

INTRODUCTION

Multicore architectures are widespread and get more and more important in all kinds of devices. The number of cores will rise in the next years. The software development must therefore be adapted to benefit from the advantages these architectures provide. The most important programming languages do not bring sufficient support for concurrency. Development processes that are proven and familiar are still driven by the sequential programming paradigm.

To achieve efficient programming of simultaneous processes there is a need for knowledge regarding the architecture and parallel algorithms. For most time it is necessary to have an explicit programming of the synchronization and coordination of the single parts. Not only does software development reach a higher level of complexity, it also increases the potential for mistakes.

With optimizing code execution in mind it is very important to have knowledge about relations, especially the internal control and data flow. In order to determine important informations such as loop bounds or pointer accesses, it is not enough to rely on known compiler techniques. It is known that a static analysis cannot be sufficient. To allow every freedom in the design of sequential code, a dynamic analysis is unavoidable. In this regard visualization of the determined dependencies could support restructuring the program execution. Software visualization is a vast field with a wide range of applications. Depending on requirements the visualization extends from nice printed source code to static visualization of internal dependencies, and then to dynamic reproduction of execution sequences. Further the visualization of software evolution belongs to this field (e.g. used by seesoft in (Eick et al. 1992)).

The level of abstraction plays a major role to derive the greatest possible benefit. Visualization has the aim to reduce the amount of information to the essentials. It is often used in software maintenance to assist human comprehension. In the electrical industry, a schematic diagram is often used to describe the design of equipment. Schematic diagrams are also used for the maintenance and repair of electronic and electromechanical systems.

It is a usual although not universal convention that schematic drawings are organized on the page from left to right and top to bottom in the same sequence as the flow of the main signal or power path. For example, a schematic for a radio receiver might start with the antenna input at the left of the page and end with the loudspeaker at the right. Positive power supply connections for each stage would be shown towards the top of the page, with grounds, negative supplies, or other return paths towards the bottom.

Design flows are the explicit combination of electronic design automation tools to accomplish the design of an integrated circuit. Moore's law has driven the entire IC implementation register-transfer level (RTL) to graphic database system design flows (GDSII) from one which uses primarily stand-alone synthesis, placement, and routing algorithms to an integrated construction and analysis flows for design closure. The challenges of rising interconnect delay led to a new way of thinking about and integrating design closure tools. (French and Vierck 1975), (Martin et al. 2006)

The visualization of complex software systems facilitates within the range of software engineering in general. Our focus lies on visualization of combined control and data flow dependencies of complex software systems. One possible benefit can be that possible parallelism becomes evident. If the analysis software contains data independent regions, such a representation would show this. Complex software applications could not be visualized on instruction level due to the complexity. According to this an expedient level of granularity must be chosen. The details of the underlying model for our visualization is shown in section code decomposition. The presented model has characteristics that are alike combinational logic. This model allows visualizing the software dependencies with electronic circuit diagram layout techniques.

RELATED WORK

Software visualization consists of a range of diverse types. An overview of the techniques is given in the literature like (Koschke 2003) or (Diehl 2007). Behavioral representation does not fit this paper as it focuses on static structure visualization. The simplest kind of making additional program meta-information apparent is pretty printing. Source code indentation and syntax highlighting helps the developer in the programming process (Baecker 1988). In (Karrer et al. 2011) a new tool to support source code navigation was presented. It computes the call graph and visualizes relevant parts by augmenting the traditional code editor. The developer is capable to interactively traverse the graph.

The codification can be mapped to connectivity matrices or predecessor-successor tables. Later flow charts were used to represent this characteristic. (Allen 1970) describes the basic control flow relationships as directed graph. Besides the control flow also the data flow can be relevant (Orailoglu and Gajski 1986).

To push more structured programs the so called Nassi-Shneiderman diagrams were introduced (Nassi and Shneiderman 1973). The way vice versa is exemplary presented in (Landis et al. 1988). For maintenance a documentation of existing source code is produced. This representation has been replaced by UML activity diagrams (Rayner et al. 2005).

To find possible parallelism in software systems the data flow is essential. The used visualization must be well suited to the used abstraction level. The clarity and readability are important for usability. All the previously discussed techniques of graphical program visualization base mainly on the control flow. In our context the focus lies on the data flow dependencies. A directed graph can be used. Generating an acceptable layout in reasonable and finite time is prevented by the complexity in typical applications. (Sander 1999) deals with lay outing graphs and in addition with animation and interactive exploration.

Translating a behavioral specification to the Register Transfer Level and deriving a data and control flow graph is shown in (Namballa et al. 2004). The behavioral description is written in a hardware description language, instead of a high-level programming language. The interconnects between parts are structured differently than in this work.

One solution to the problem to find an appropriate level of abstraction with a coarse grained granularity was presented in (Wu et al. 2013). The ProgramCutter is an approach to automatically partition monolithic software using dynamic data dependency analysis. The constructed data dependency graph with functions as nodes and data dependency edges can only be used by the programmer for software refactoring. A mapping of the control flow and application behavior is not intended.

For this paper the needed analyses of intra-procedural and inter-procedural data flow in quality and quantity is based on the LLILA Framework (Gremzow 2007). The framework is based on the LLVM compiler framework. It has a language independent instruction set and type system (Lattner and Adve 2004). The instruction set of the internal representation uses a static single assignment form (SSA). The intermediate code is good for machine-independent optimizations as it is not bound to an architecture. Further, accessing the runtime environment is less complex in a virtual machine than in a physical target architecture.

To determine the control flow a static analysis is used. Exemplary figure 3 visualizes the possible control flow. LLVM provides bit code with block structure. Every block begins with a label and ends with a jump instruction (branch, switch, return). A correlation of labels and jump instructions yields the control flow on block or function level.

For quantitative data flow tracing the combination of sources and corresponding sinks must be determined. To realize a complete analysis even for pointer referenced objects the source code is augmented with profiling instructions. The enriched code is compiled and executed in the runtime environment (RTE). While runtime, the profiling information is written. The runtime profile and its analysis are used for extraction of the data flow which is not determined statically.

CODE DECOMPOSITION

Applying electronic circuit diagram layout techniques to control and data flow information of software need a suitable structure. The software must be decomposed into blocks with static data dependencies. This transformation is discussed in the following.

A function does not fulfill the constraint of static data dependencies. From the source code it is obvious where a function is called but a function can enclose branches. The processed data within the surveyed function is not static in either case.

A further known structure are basic blocks. A basic block has only one entry point and ends at one point

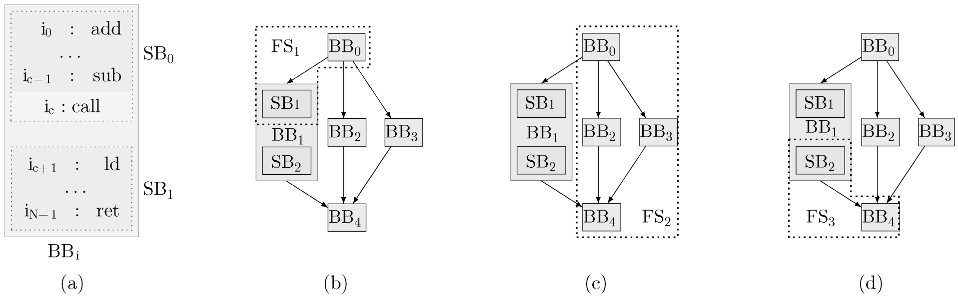


Figure 1: Creating function slices of a hypothetical function: (a) splitting a B into BS due to a call instruction; (b) - (d) grouping of Bs and BSs to function slices

with a jump instruction (branch, switch, or return). The consumed data is fixed, but the source or producer of them is not. Pointers and pointer arithmetic allows to alter data sources and control flow sources. In order to determine important informations such as loop bounds or pointer accesses, it is not enough to rely on known compiler techniques. For example a function

```
fun(char *data, int length){...}
```

can operate on data arrays with unknown length. The length is determined at runtime and passed to the function as second parameter. A subroutine call can modify pointers just as well; hence every call possibly changes the dependencies. It is conceivable that a basic block operates on one object (variable) but in fact reads or modifies different underlying objects. Instruction are executed several times. If the modified data were referenced by a pointer, the target can change in between. The source code indicates one object, at execution different memory areas are read or modified. A static code analysis cannot provide this information Hind (2001).

The solution to get units with postulated static data dependencies is to sub-divide basic blocks into sub-blocks at call instructions. A part of a basic block produced like this is called sub-block (SB). A basic block BB featuring N call instructions is thus subdivided into $N+1$ sub-blocks.

Figure 1a shows a hypothetical basic block featuring one call and the resulting sub-blocks $SB0$ and $SB1$. The shortest sequence of instructions between alternatively

- a basic block label
- a call instruction

and alternatively

- a call instruction
- a branch instruction
- a switch instruction
- a return instruction

is now called block B.

Compilers usually decompose programs into their basic blocks. Subdividing basic blocks into sub-blocks lead to a control flow graph like shown in figure 3. The code of the example application for which the division was applied here is shown in figure 2. It processes an input file by reading blocks of 1024 bytes. To every byte an

```
 struct stuff {
  int cnt;
  char buf[1024];
};

volatile void foo (char *buf_foo, int len_foo, char fac_foo) {
  int i;
  for (i = 0; i < len_foo; i++)
    buf_foo[i] *= fac_foo;
}

volatile void bar (char *buf_bar, int len_bar, char off_bar) {
  int i;
  for (i = 0; i < len_bar; i++)
    buf_bar[i] += off_bar;
}

volatile void foobar (char *buf_foobar, int len_foobar, \\
                      char fac_foobar, char off_foobar) {
  foo (buf_foobar, len_foobar, fac_foobar);
  bar (buf_foobar, len_foobar, off_foobar);
}


int main (int ac, char **av) {
  int fd;
  struct stuff st;
  char fac, off;
  fd = open (av[1], O_RDONLY);
  fac = (char)atoi(av[2]);
  off = (char)atoi(av[3]);
  while ((st.cnt = read (fd, st.buf, 1024)) > 0) {
    foobar (st.buf, st.cnt - 24, fac, off);
    write(2, st.buf, st.cnt);
  }
  close(fd);
}
```

Figure 2: Example application for linear data transformation – Source code.

offset is added and they are scaled with a factor. Even such a less complex application collapses into a multiplicity of sub-blocks. For further operations a reduction of complexity would be preferred. The characteristics which led to this structure must not be changed. It is noticeable that there are sequences of blocks within one function. If a sequence is not interrupted by another function the dependencies could not be changed externally. This means that no unpredictable modifications of data or pointers can take place. For example BB0-SB2 and BB1 of function foobar in figure 3 is such a sequence of two blocks. The required static data dependencies are still met if these blocks were combined to a new unit. This combination is called function slice (FS) and offer a coarse grained and less complex model for further handling.

A function slice is now defined as a set of blocks B. The combination is to maximize the unit of contiguous ex-

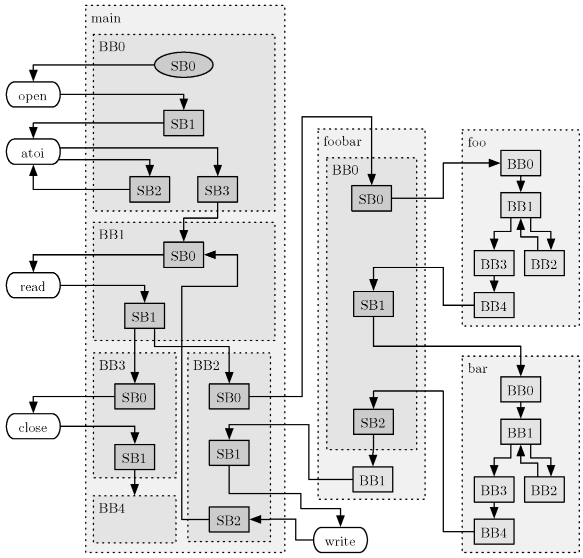


Figure 3: Example application for linear data transformation – Control flow on Sub-block level

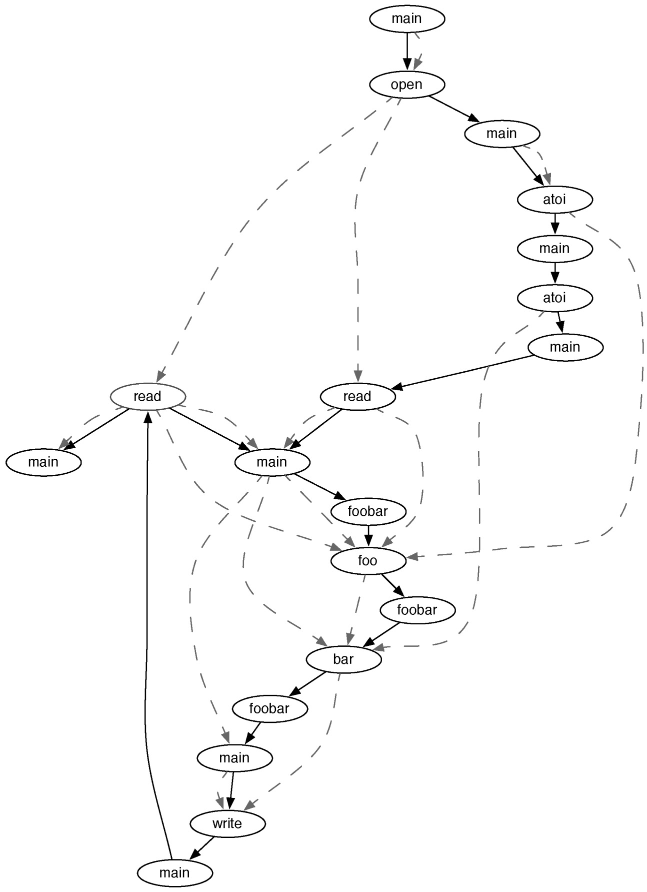


Figure 4: Example application for linear data transformation – Function slice graph; vertices: Function slices; solid arrows: control flow dependencies; dashed arrows: data flow dependencies

ecutable instructions without violating the condition of constant data dependencies. The characteristic of basic blocks to have only one entry and only one exit point must also apply to function slices. Basic blocks arising from a hypothetical function and their control flow relations are shown in figure 1 b) - d). Grouping the basic blocks and basic block sequences results in three function slices then (figure 1b-d). Every stringent sequence of blocks composes a function slice. The instructions of a slice lie between a label at the beginning and a branch, switch, return or call instruction at their end, too. Figure 4 shows an example function slice graph of the example application.

With this behavior a function slice resembles combinational circuits. The output only depends of the input data. The control flow corresponds to an enable-signal, which signalizes valid input data. Combinational circuits are stateless, even so function slices. Variables containing data similar to a state are mapped to a self referencing data flow edge.

A further step to reduce complexity of application can be function slice clusters (*FSC*). The start sequence (main) → (open) → (main) → (atoi) → (main) → (atoi) → (main) is executed only once and as one block with no interruptions. The abdication of back directed control flow edges indicates that this sequence can not be entered at any other point. No external modifications can occur. This sequence can be clustered and handles as a unit. Corresponding every sequence framed by back directed control flow edges can form a cluster.

SCHEMATIC GENERATION

To visualize the software model described before, graphical representations are needed. Function slices will be represented as rectangles with their id above and the function name below (figure 5a). Particular attention must be paid to the generation of the pins, these need locally unique names per slice. The data and control flow dependencies are visualized by connections between function slices.

Pins for control flows begin with the letter C and data flows with the letter D. An incoming flow gets an I at his name and will be placed at the left side of the symbol. By analogy is the letter for the outgoing flow an O and they will be placed on the right side of the slice. By dividing the input and output pins on various sides of the slices the readability is considerably increased. The last part of the name describes the slice from where the connection comes from. Pins for control flows are colored blue, the pins for data flows are colored yellow. Function slice clusters are exactly as Function slices represented (figure 5b), with the only change that her frame is orange and the function name is replaced with the Slice Cluster name.

After icons for all slices and slice clusters were defined, the local wiring of slices within a slice cluster can be

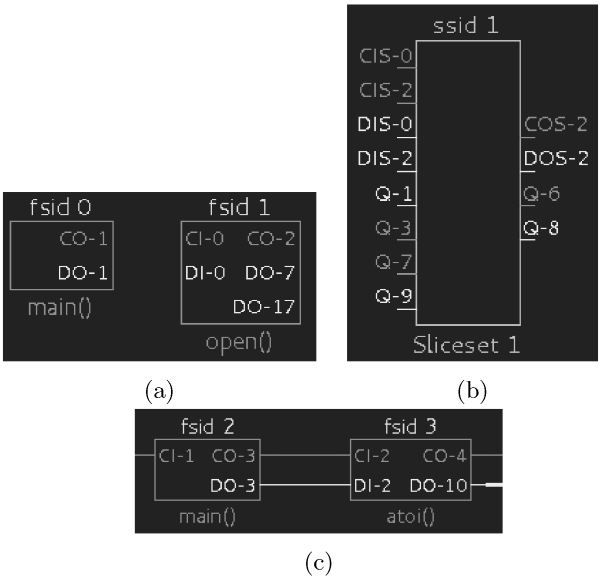


Figure 5: Schematic symbols for visualization
a) Slice with input- and output-ports
b) Slice Cluster with in- and output-ports
c) Data and control flow

done. The definition of the generating compounds results here directly from the slice flows that begin or end in the corresponding slice cluster. A simple example of local wiring is shown in figure 5 c). There are two slices, which are connected by control and data flow. Flows are shown as connecting lines between one outgoing and one incoming pin.

The problem in automated schematic generation is to find a placement and routing algorithm to place the elements in a grid. First the function slices must placed into a column- and row-system. The columns are numbered from left to right. The sequence of the module within its column is referred as the row position. Each (column, row) pair can be seen as a position on a logical grid, and cannot be occupied by more than one module. The slices with outgoing edges only will be placed in the first column and Slices with incoming edges only will be placed in the last column. The column placement for the rest can be calculated by getting the depth from a starting-slice to the current slice.

Once each module is assigned to a column on the logical grid, the row position can be calculated. There are different methods to calculate the row position, based on the optimization criteria and the choice of heuristic. By minimizing the number of crossovers, you might increase the number of signal line bends and the total wiring required. One solution for finding an existing path between two slices is the Lee-Router. The algorithm consists of propagating a wavefront from a source point towards a destination point. When the destination point is reached, the path is traced back to the source point.

After this logical placement, the elements must placed geometrical. First problem is calculating the exact coordinates of each Slice on the drawing surface of the schematic. This is usually done per column of Slices, and from left to right, starting with the input Slices. Alternative strategies include starting with the column containing the most modules. Second is calculating the exact coordinates of each connection on the drawing surface of the schematic diagram. This involves calculating the begin and end coordinates of the different line segments forming a single connection. Except for fan out stems and branches this step is fairly straightforward. The fan-out-problem can be solved by getting a second layer over the first one and use this for branching.

Nlview is state-of-the-art in automated schematic generation and will be used for this work to get the best achievement. This tool provides automated generation of schematic diagrams for different levels of electronic circuits. The schematic layout can be modified and controlled by human intervention to get the best human readable results. Nlview provides a tool chain to create big circuit schematics with many elements in a for humans readable description. (see (Lageweg 1998), (Koster et al. 1989))

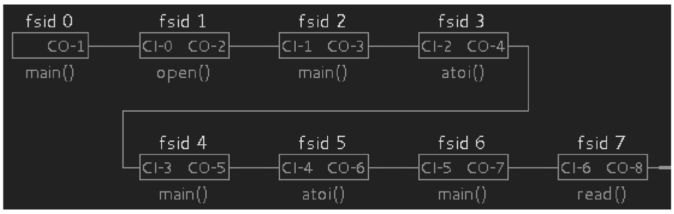
Complexity reduction

Applying the model and schematic layout techniques to complex software systems leads to schematics with increasing complexity. Some attempts can reduce the graphical complexity without degrade the readability.

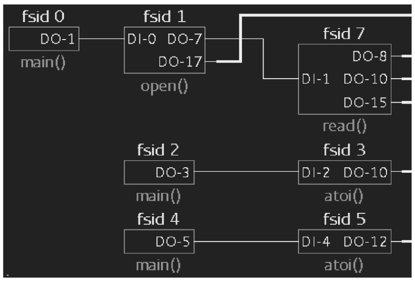
The use of hierarchical levels is the first improvement. It splits the graph into subgraphs of less complexity. The hierarchical level existing is to visualize function slice clusters. In the folded state they are just represented as black box.

The second approach deals with the connections. In complex software systems an increasingly high number of connections dominates the schematic. If there is only one pin per slice for incoming control flow and another pin for outgoing data flow the number of generated pins and connections is reduced significantly. The connection-“wire” then splits to reach all connected slices.

Pins for control flows can obtain a label that bears the name of the function that is the destination point of the control flow, on the experience gained in the slices Information. For data flows would be the same procedure possible, but offers here yet another way. The outgoing pins can be connected to the incoming pins of the slices and labeled with details of the transmitted data. This approach facilitates the localization of the relevant data in the source code of a program considerably.



(a)



(b)

Figure 6: Analysis example application slice cluster 0 a) Slice Cluster 0 with control flow; b) Slice Cluster 0 with data flow

ANALYSIS

The analysis of the previously described techniques consists of two parts: describes slice cluster 0 from the example application and illustrates an example of the MPEG-2 decode data and control flow.

Slice Cluster 0

For an analysis function slice 0 of our sample application is considered. The control flow in Slice Cluster 0 is visualized in figure 6 a), reflecting the sequential execution of the program in the generated diagram. This Slice Cluster has no branches and we can see the control flow moves straight forward till the end.
A pure control flow representation does not provide any guidance in determining of parallelizable program sections, it simply reflects the structure of the software in a graphical representation.
When analyzing the data flow shown in figure 6 b), there is a significant deviation from the control flow sequence. The data flow consists of three independent sequences, that suggests of potential parallelization processes taking place. The function main() respectively forms the starting point of operations, so it only acts as a producer of data in appearance. This behavior can also be derived directly from the example application in figure 2, since data from the argument vector of the main() function can be read.
As the first parallelizable scope arises from the diagram, the order fsid0 → fsid1 → fsid7 where a date returns by the main()-function is passed to the open()-function, and this provides a date to the read()-function. The scopes two and three are the same functions in principle. As seen in Figure 2 the atoi()-function is called two times to read out offset and scaling of the transfer parameters.

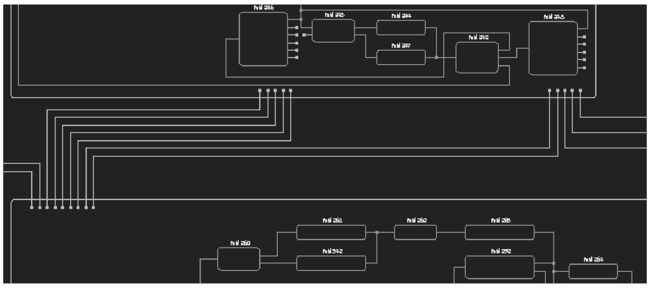
	example app	MPEG-2	coefficient
Slice-Cluster	2	38	19
Slices	19	1471	77
control flow	19	3752	197
data flow	20	15809	790
loading time	0,1s	0,5s	5
optimization	0,005s	0,025s	5

Table 1: Comparison Example Application and MPEG-2

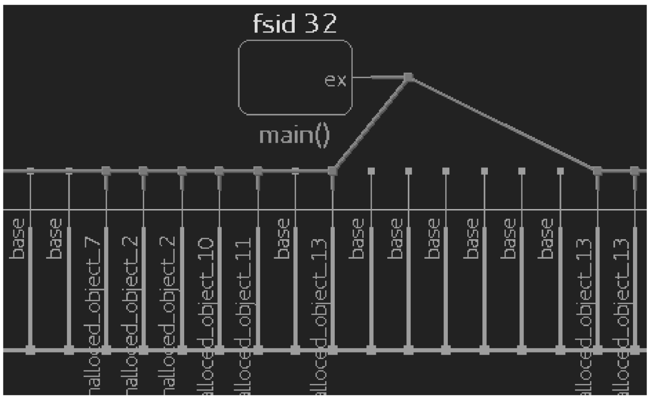
MPEG-2 Decode

After testing the switching system generation using the example application, and the implementation of measures to reduce the complexity of the logic systems, the visualization is now used on a much larger program. It is the function slice graph of a MPEG-2 decoder that executes the decoding of video and audio in the MPEG-2 format.
A direct comparison of the number of elements of the function-slice graph of the example application and the MPEG-2 decoder is shown in Table 1. In terms of the total number of elements of the two graphs increases the complexity of the generating circuit approximately by a factor of 351. The time needed for loading and optimization of the internal data structure of the graph needed only the factor of five longer. Overall, the time required is significantly less than a second, serious performance problems when reading large Function-slice graphs are can not be expected because of these reference values.

Control flow analysis
For the analysis of the control flow in the MPEG-2 decoder the view with all flows is used and slice clusters will be shown as hierarchy. Figure 7 shows a part of the control flow representation. Particularly evident recognizable are the external control flows shown between the slice clusters. The number of external connections is distinctly higher than in the previously considered cases. But for the Function-slice graph of the MPEG-2 decoder is a maximum of 13 incoming or outgoing external control flows per slice set. The schematic diagram remains is readable despite significantly increased wiring complexity. The beaming of external control flow connections to a bus system is possible but not necessary.
In the absence of the hierarchy the layout is primarily based on the of the software and shows clear analogies to the already known example application. Structures that are already in the analysis of the control flow in the example application have shown.



(a)



(b)

Figure 7: Mpeg2 analysis; a) Mpeg2-decode example control flow; b) Mpeg2-decode example data flow

The layout generator arranges the slices in the control flow in long sequences, some of which have returns and branches. Similarities with known structures can be attributed to the limited number of available control structures attributed that used in the C language.

The complexity of the control flow representation is directly related to the Structure of the source code. Even with extensive examples, only increases the number of elements and connections, the connection of the elements to networks will follow the already known structures.

Data flow analysis

The generated diagrams of the MPEG-2 decoder are far too complicated for a comprehensive analysis by hand. In particular the large number of data flows on the outside of slice sets makes the presentation confusing.

In figure 7 b) the described problem is clear. The red marked connecting are the data flows from the main()-function to the pins on the edge of the slice clusters branched at numerous other slices in the other slice clusters. A modified method for hierarchy formation could potentially get better results here. The use of slice clusters appears not the best representation of large function slice graphs.

In the absence of the hierarchy formation, the problem of external data flows is eliminated. This results grow

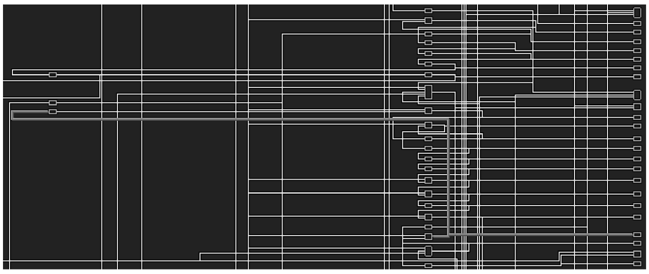


Figure 8: Mpeg2-decode data flow and slices only

mainly in the vertical direction in the overall presentation of a network structure. Figure 8 illustrates this observation using a section of the flow representation of the MPEG-2 decoder. In the course of generating the netlist are slices that have no incoming data flows, referred as input elements. They will be placed in the schematic diagram in the first column. Slices without outgoing data flows will be classified as output elements and placed in the last column.

CONCLUSION

For this work were various possibilities for the visualization of function slice graphs discussed. The fundamental problem of the automatic generation of layouts for schematic diagrams was presented and a tool for the layout synthesis was presented. We created an application that is able to efficiently transform function slice graphs in schematic diagrams. Based on the insights gained in the analysis of the example application we have been developed and implemented methods which lead to a significant reduction in the complexity of the schematics. The resulting process for transforming function slice graph in schematics has been tested with reference to the MPEG-2 decoder, and the resulting problems are illustrated.

The mapping of function slice graph on schematic diagrams by transformation to netlists proves to be suitable visualization ability. The generated diagrams are a valuable tool in the analysis of control and data flows . The result is a well structured representation whose can convince easy viewing and handling. The introduction of hierarchies reduces the complexity of the schematics. It allows the targeted considering the local behavior of the individual diagram blocks without interfering external influences. The deficits initially occurring in the representation by the display of automatically generated descriptions of the pins in slices were in control flows are largely eliminated through the use of existing information. At data flows was an extension of with additional information relating to the transmitted data added, which produces significantly better understandable schematics. The beaming of data flows at the outputs of slices and the hiding of external connections decrease the wiring overhead in the generated schematics significantly. The readability is further improved.

REFERENCES

Allen F.E., 1970. *Control Flow Analysis. SIGPLAN Not*, 5, no. 7, 1–19. ISSN 0362-1340. doi:10.1145/390013.808479. URL http://doi.acm.org/10.1145/390013.808479.

Baecker R., 1988. *Enhancing program readability and comprehensibility with tools for program visualization.* In *Software Engineering, 1988., Proceedings of the 10th International Conference on.* 356–366. doi:10.1109/ICSE.1988.93716.

Diehl S., 2007. *Software Visualization - Visualizing the Structure, Behaviour, and Evolution of Software.* Springer. ISBN 978-3-540-46504-1.

Eick S.; Steffen J.; and Sumner E.E. J., 1992. *Seesoft-a tool for visualizing line oriented software statistics. Software Engineering, IEEE Transactions on*, 18, no. 11, 957–968. ISSN 0098-5589. doi:10.1109/32.177365.

French T.E. and Vierck C.J., 1975. *Engineering Drawing and Graphic Technology.* McGraw Hill. ISBN 0-07-022157-X.

Gremzow C., 2007. *Compiled low-level virtual instruction set simulation and profiling for code partitioning and ASIP-synthesis in hardware/software co-design.* In *Proceedings of the 2007 summer computer simulation conference.* Society for Computer Simulation International, San Diego, CA, USA, SCSC. ISBN 1-56555-316-0, 741–748.

Hind M., 2001. *Pointer analysis: haven't we solved this problem yet?* In *Proceedings of the 2001 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering.* ACM, New York, NY, USA, PASTE '01. ISBN 1-58113-413-4, 54–61. doi:10.1145/379605.379665.

Karrer T.; Krämer J.P.; Diehl J.; Hartmann B.; and Borchers J., 2011. *Stacksplorer: Call Graph Navigation Helps Increasing Code Maintenance Efficiency.* In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology.* ACM, New York, NY, USA, UIST '11. ISBN 978-1-4503-0716-1, 217–224. doi:10.1145/2047196.2047225. URL http://doi.acm.org/10.1145/2047196.2047225.

Koschke R., 2003. *Software Visualization in Software Maintenance, Reverse Engineering, and Re-engineering: A Research Survey. Journal of Software Maintenance*, 15, no. 2, 87–109. ISSN 1040-550X. doi:10.1002/smr.270. URL http://dx.doi.org/10.1002/smr.270.

Koster G.; Stok L.; and of Technology. Faculty of Electrical Engineering E.U., 1989. *From Network to Artwork: Automatic Schematic Diagram Generation.* EUT report / E: EUT report. Faculty of Electrical Engineering, Eindhoven University of Technology. ISBN 9789061442196.

Lageweg C.R., 1998. *Designing an Automatic Schematic Generator for a Netlist Description.*

Landis L.D.; Hyland P.M.; Gilbert A.L.; and Fine A.J., 1988. *Documentation in a software maintenance environment.* In *Software Maintenance, 1988., Proceedings of the Conference on.* 66–73. doi:10.1109/ICSM.1988.10141.

Lattner C. and Adve V., 2004. *LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation.* In *Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization.* IEEE Computer Society, CGO '04.

Martin G.; Lavango L.; and Scheffer L., 2006. *Electronic Design Automation For Integrated Circuits Handbook.* Taylor and Francis. ISBN 0-8493-3096-3.

Namballa R.; Ranganathan N.; and Ejnioui A., 2004. *Control and data flow graph extraction for high-level synthesis.* In *VLSI, 2004. Proceedings. IEEE Computer society Annual Symposium on.* 187–192. doi:10.1109/ISVLSI.2004.1339528.

Nassi I. and Shneiderman B., 1973. *Flowchart Techniques for Structured Programming. SIGPLAN Not*, 8, no. 8, 12–26. ISSN 0362-1340. doi:10.1145/953349.953350. URL http://doi.acm.org/10.1145/953349.953350.

Orailoglu A. and Gajski D.D., 1986. *Flow Graph Representation.* In *Proceedings of the 23rd ACM/IEEE Design Automation Conference.* IEEE Press, Piscataway, NJ, USA, DAC '86. ISBN 0-8186-0702-5, 503–509. URL http://dl.acm.org/citation.cfm?id=318013.318093.

Rayner M.; Hockey B.A.; Chatzichrisafis N.; and Farrell K., 2005. *OMG Unified Modeling Language Specification.* In *Version 1.3, 1999 Object Management Group, Inc.*

Sander G., 1999. *Graph layout for applications in compiler construction. Theoretical Computer Science*, 217, no. 2, 175 – 214. ISSN 0304-3975. doi: http://dx.doi.org/10.1016/S0304-3975(98)00270-9. URL http://www.sciencedirect.com/science/article/pii/S0304397598002709. ORDAL'96.

Wu Y.; Sun J.; Liu Y.; and Dong J.S., 2013. *Automatically partition software into least privilege components using dynamic data dependency analysis.* In *Automated Software Engineering (ASE), 2013 IEEE/ACM 28th International Conference on.* 323–333. doi:10.1109/ASE.2013.6693091.

NETWORKS AND LOGISTICS SIMULATION

PERFORMANCE EVALUATION OF VIDEO CONFERENCING OVER AD HOC WIRELESS NETWORKS USING DIFFERENT ROUTING AND QUEUING TECHNIQUES

Azeddien M. Sllame
Faculty of Information Technology
University of Tripoli
Aziz239@yahoo.com

Abdelrahim Raey, Basher Mohamed, Abdelhafed Alagel
Computer Network Department
Faculty of Information Technology
University of Tripoli
Tripoli, Libya

KEYWORDS
Ad Hoc Networks, Routing, Multimedia, Queuing, QoS

ABSTRACT

In this paper we present a comparative study using OPNET simulation tool for video conferencing over wireless ad hoc networks. The study performed using three different routing protocols (DSR, AODV, TORA) and three queuing techniques (FIFO, PQ, WFQ). Different scenarios exploiting a combination of routing and queuing are produced. A quality of service is measured by means of packet delay variation, packet end-to-end delay and network throughput.

INTRODUCTION

Recently, the usage of the Internet has been experienced huge growing all over the world. The Internet is becoming an essential thing in people's life via using multimedia communications everywhere (e.g. VoIP, video on demand and video conferencing, and e-commerce). Multimedia streaming over the Internet requires greater transmission bandwidth than any other communication types. That is due the huge data included into multimedia continuous flows. In another side, modeling and simulation is one of the most main assessment and validation techniques that are used in exploring communication systems due to the complexities of such systems and because of the higher cost values of constructing such systems. Communication systems include variety of architectures, standards, protocols and technologies for wire, wireless, satellite networking. Ad hoc wireless networking is one of such emerging technologies. Ad hoc wireless network is a set of randomly located mobile devices that have neither organized structure nor centralized control, but it may have access to stationary infrastructure such as central network or the Internet (Azzedine 2009).
This paper is organized as follows: ad hoc networks are briefly described in section 2. Routing in ad hoc networking is illustrated in section 3. Multimedia streaming over networking is outlined in section 4. Section 5 briefly describes quality of service (QoS) with multimedia. Finally, practical simulation results are given in section 6.

WIRELESS AD HOC NETWORKS

Wireless ad hoc networks can be realized by different wireless communication technologies such as Bluetooth, IEEE 802.11, and Ultra-Wide Band (UWB) (Azzedine 2009). Wireless ad hoc networks can be deployed quickly so it can be used in disaster recovery, scientific conferences and military operations. In an ad hoc network, links between nodes are typically made temporarily according to individual node operations and nodes may be added or removed either according to the network's requirements. Wireless ad hoc networks differ from other networks by the following features: decentralized control, each node has wireless interface, mobility of nodes, network topology changes, the nodes have limited resources (power, memory, etc.), nodes participate in routing by forwarding data for other nodes, and data forwarding is made dynamically based on the current status of the network connectivity.

ROUTING IN WIRELESS AD HOC NETWORKS

The most significant characteristic of the wireless ad hoc networks is the dynamic topology that resulted from node mobility. Nodes mobility imposes routing protocols to quickly respond and adapt to topology changes. In wireless ad hoc networks the routing made to destinations through a series of nodes making-up a path to the specified destination. Routing protocols for wireless ad hoc networks can be classified into two main categories: *proactive or table-driven routing protocols* and *reactive or on-demand routing protocols*. Other classifications do exist; the intended reader can find more in (Azzedine 2009; Kai-Wei 2006; Sunil 2004).

A- Table driven routing protocols (proactive)

In this kind, each node continuously maintains up-to-date routes to all other nodes in the network and each node required to maintain one or more tables to store that routing information. Routing information is periodically transferred all over the network to keep routing tables consistent and reliable (Anuj 2010). This enables these protocols to quickly respond to topology changes by propagating updates throughout the network. Some proactive routing protocols are: Destination-Sequenced Distance Vector (DSDV), Wireless Routing Protocol (WRP), Global State Routing (GSR) and Cluster-head Gateway Switch Routing (CGSR).

B- On-demand routing protocols (reactive)

In reactive protocols, route search is necessary for every unknown destination and nodes need to retain the routes to all active targeted nodes. However, this is done by each node by initiating a route discovery process all over the network, once it requested to send data to another node. Then, once the node determined its selected effective route; the route is then kept by the maintenance process until the desired route is no more needed either because of the transfer is finished or the route is no more applicable. Some reactive protocols are: Dynamic Source Routing (DSR), Ad hoc On-Demand Distance Vector (AODV), and Temporally Ordered Routing Algorithm (TORA) (Anuj 2010).

DSR ROUTING PROTOCOL

DSR is a reactive, source-routed routing protocol designed for extremely dynamic networks. In DSR, each node preserves route collection information that keeps all routing paths from the node itself to all other nodes in the network. Basically, in DSR routing the source node introduces the "Route Discovery" method in order to discover a routing path to a required node when there is no entry about that node in the route collection. Moreover, DSR uses "Route Maintenance" technique to retain the routes when there are connection failures in the routes or when any topology changes happened; thus; renews broken routes rapidly to make the node reachable by other nodes. However, when a route is found, the sending node sends packets with headers holding full path information toward the receiving end. Moreover, each node along that path forwards packets to the next node in the network according to control information contained in the packet's header. DSR maintains only routes for nominated nodes, and does not make any periodic advertisements with other unwanted nodes. DSR as source routing protocol can make the usage of discovery messages information dissemination unpractical due to node routing cache overhead in large networks. Therefore; DSR is not scalable for large networks (Kai-Wei 2006).

AODV ROUTING PROTOCOL

AODV is a reactive, distance-vector routing protocol suitable for highly dynamic networks. By deploying the AODV, every node in AODV holds a routing table that contains only active routing entries. Furthermore, the AODV builds and preserves routes in the similar way of DSR protocol. However, the AODV keeps only its local connections with close neighbors and it is not using periodic advertisement and it keeps only routing information about needed routing entries in which it takes the advantage of DSR protocol (Kai-Wei et al 2006). In another words, AODV searches route entries for nods only when needed and are kept in route cache only as long as they are necessary for current communication. Therefore, AODV does not play any role when the endpoints of the current communication have effective routes to each other. AODV protocol has loop freedom feature and when link failures occurs immediate notifications is issued to the set of affected nodes only. This in turn, reduces the number of routing messages in the network significantly (Anuj 2010).

TORA ROUTING PROTOCOL

TORA is a reactive, greatly adaptive distributed routing protocol designed to operate in a dynamic multihop networks. TORA is based on link reversal algorithm and it uses a directed acyclic graph (DAG) to generate multiple routing paths upon requests from sender to receiver. TORA is designed to reduce reaction to topologies changes. One important thing with TORA is that control messages are normally localized to a minimal set of nodes; this will assures that all routes are loop-free and offers multiple routes for any two communicating nodes. TORA provides only routing task and rest on Internet MANET Encapsulation Protocol (IMEP) for other underlying functions; which further increases the overhead to the protocol (Anuj 2010).

MULTIMEDIA STREAMING

Multimedia streaming contains audio and video content ("continuous media"). In practice, networks supporting video conferencing services should typically be designed for very close to zero percent packet loss for both the VoIP and video streams (Evans 2007; Balakrishnan 2008). Moreover, QoS mechanisms and suitable capacity planning procedures may be engaged to ensure that no packets are lost due to congestion, with the only actual packet loss being due to layer one bit errors or network element failures. When packet loss occurs, the impact of the loss on voice streams should be reduced to acceptable levels using concealment techniques. The loss rates tolerated for video conferencing are likely higher than those acceptable to broadcast video services. The end-to-end delay from streaming server to the client is the significant delay metric in the case of video streaming. Digital video decoders used in streaming video receivers need to receive a synchronous stream, typically with jitter tolerances of only +-500 ns, in order to decode without visible impairments (Evans 2007; Balakrishnan 2008). However, in IP networks which transfer VoIP and video broadcasting, the jitter tolerances are not feasible. Therefore, de-jitter buffers are usually used in receivers to remove delay variation caused by the network elements; to make it capable of meeting VoIP and video broadcast services. Jitter is defined as the delay variation between two consecutive packets belonging to the same traffic stream. Although queuing is the main cause of traffic jitter, lengthy reroute propagation delays and additional processing delays can also affect traffic jitter. Packet delay is defined as the difference in the time at which the packet enters the network and the time at which it leaves the network; from synchronized sender to destination (Evans 2007; Balakrishnan 2008)

QUALITY OF SERVICE

A network that offers QoS is a network that provides definite guarantee level for delivering the transmitted packets in steady and safe way; especially in multimedia streaming. In a wireless networking, the quality may include packet transfer delay (one-way end-to-end delay), delay variation (Jitter), and packet loss ratio. Nowadays, all Internet service providers are estimated to provide personalized media-rich application services and migrating toward offering all multimedia-intensive applications over a single infrastructure composed of mixed wire/wireless networks. To apply a QoS model, many QoS features are required such as traffic classification, queuing and buffering, scheduling, rate limiting, policing, marking, and traffic filtering (Evans 2007; Balakrishnan 2008). Bear in mind, packets flowing through a node may wait before being serviced by a scheduler toward their corresponding destinations. However, waiting in networking referred as queuing delay or latency. Moreover packets belonging to different classes of services are queued in distinguished queues. The packet belonging to a high priority traffic class is assured of buffering space. On the order hand, overflow may occur in the queues assigned to low priority traffic classes. Schedulers and queues are used together to conform the required stream's bandwidth while keeping queueing delays at the desired values for the given application. However, scheduling function is applied within a node where it decides the order in which queues are serviced and how data-streams allocated to different forwarding classes (Evans 2007; Balakrishnan 2008). First in first out (FIFO), priority queuing (PQ) and weighted fair queuing (WFQ) are considered in this paper as a measure for QoS. FIFO is one of the simplest techniques which consists of buffering and forwarding of packets in the same order of their arrival. With PQ, packets are classified into a definite priority classes; then; packets that belonging to higher priority class are sent before all lower priority traffic. In turn, this guarantees their delivery in timing and prevents packets loss as much as possible. In WFQ, the service is set according to the queue weight, i.e. each queue is given a slice from the link proportional to its prearranged weight. WFQ employs sorting and interleaving of individual packets by flow and then queue each flow based on the volume of traffic in this flow. However, by using this technique, larger flows are prevented from consuming network's bandwidth. However, WFQ is max-min fair technique and it provides some QoS control, and it is used in some industrial routers, but it is relatively complex to realize and it involves heavy computational overhead per packet in the flow (Evans 2007; Balakrishnan 2008).

SIMULATION RESULTS

This section gives an overall description of a wireless ad hoc network case study as seen in figure (1) using OPNET 14.5 (OPNET 2004). OPNET Modeler is the industry's leading is a discrete-event simulator specialized for supporting network research and development. The model network consists of 15 wireless nodes distributed in 1000mx1000m area; and the video application configuration as follows: frame size information (bytes) equal to 128x120 pixels with frame inter-arrival time information with 10 frames/sec, simulation time is 3600 second, and with heavy traffic load introduced.

- Results of comparing video conferencing for the three routing protocols (AODV, DSR, TORA) with FIFO queuing is illustrated in figures (2) till figure (4). It is clear from figure (2) that the TORA has the worst packet delay variation value and from figure (3) TORA has also the worst packet end-to-end delay value; while AODV has the lowest packet end-to-end delay value. Furthermore, the AODV showed the highest throughput than the other two protocols with FIFO queuing; as seen in figure (4).
- Results of applying PQ queuing with the video conferencing application using three different routing protocols for wireless ad hoc networks are seen in figures (5) through (7). From figure (5) we can conclude that AODV has the lowest packet delay variation, and from figure (6); AODV has also the lowest packet end-to-end delay with PQ queuing; while TORA has the worst value (much higher than the others) for both packet delay variation and packet end-to-end delay. In this case also the AODV protocol granted the highest throughput than TORA or DSR as seen in figure (7).
- Results of using three different routing protocols with video conferencing application using WFQ queuing technique are shown in figures (8), (9) and (10). In this case, DSR protocol has the lowest both packet delay variation and packet end-to-end delay; but, AODV comes in the second place with values less that 1; while TORA has much worse values. AODV has got the highest throughput value with WFQ queuing as seen in figure (10).
-

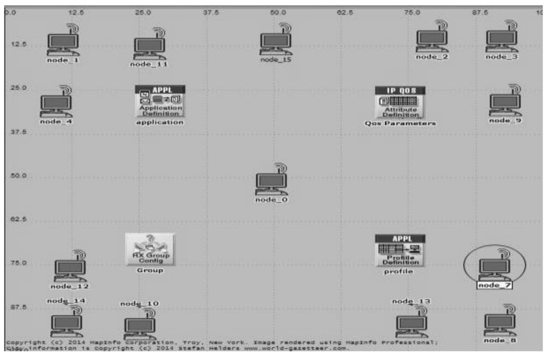


Figure 1: wireless ad hoc network model

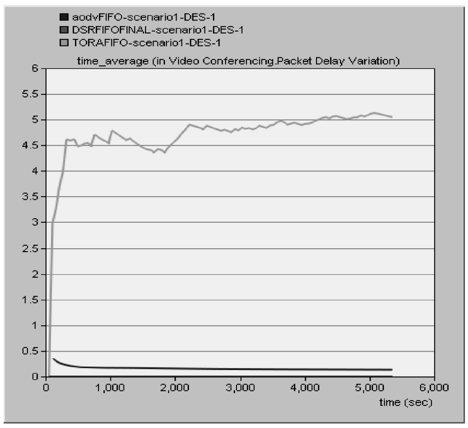


Figure 2: Video conferencing packet delay variation for FIFO queuing; Red: AODV, Blue: DSR, Green: TORA

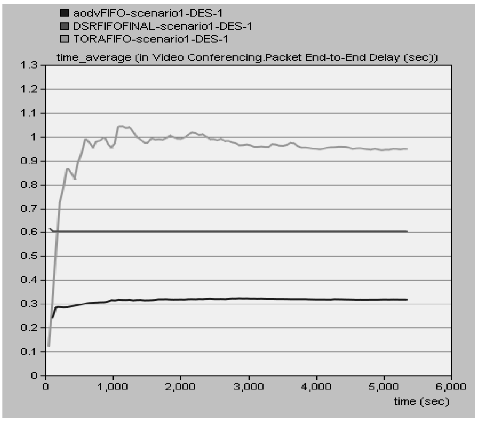


Figure 3: Video conferencing packet end-to-end delay for FIFO queuing; Blue: AODV, Red: DSR, Green: TORA

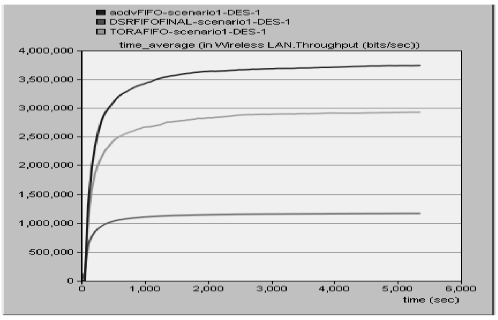


Figure 4: Video conferencing; wireless LAN throughput with FIFO queuing; Blue: AODV, Red: DSR, Green: TORA

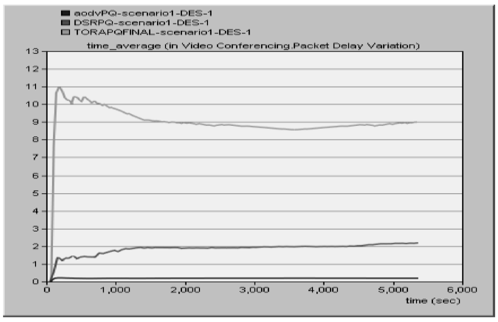


Figure 5: Video conferencing packet delay variation for PQ queuing; Blue: AODV, Red: DSR, Green: TORA

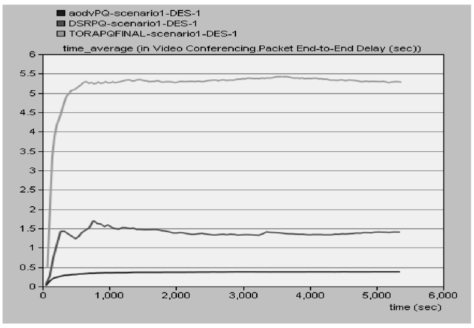


Figure 6: Video conferencing packet end-to-end delay for PQ queuing; Blue: AODV, Red: DSR, Green: TORA

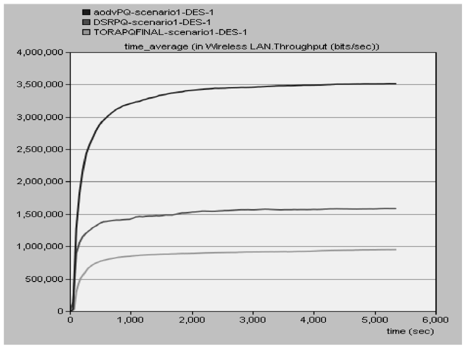


Figure 7: Video conferencing; wirelessLAN throughput with PQ queuing; Blue: AODV, Red: DSR, Green: TORA

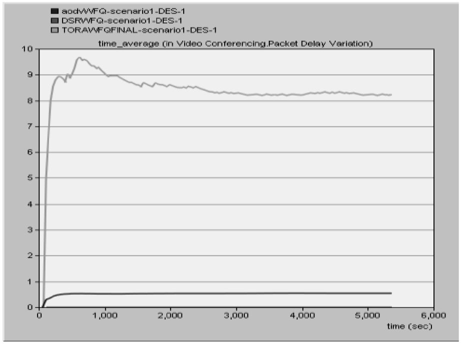


Figure 8: Video conferencing packet delay variation for WFQ queuing; Up: AODV, Middle: DSR, Down: TORA

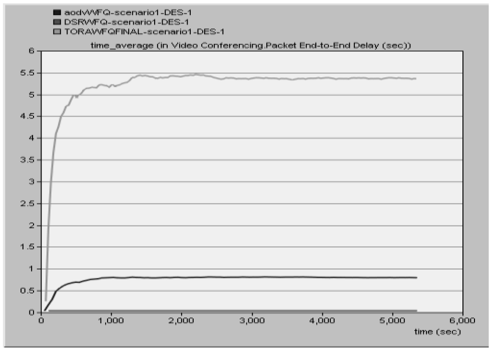


Figure 9: Video conferencing packet end-to-end delay for WFQ queuing; UP: AODV, Middle: DSR, Down: TORA

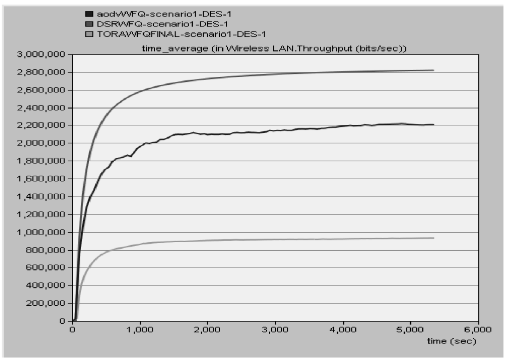


Figure 10: Video conferencing; wireless LAN throughput with WFQ queuing; Blue: AODV, Red: DSR, Green: TORA

CONCLUSION

In this paper the effect of routing (DSR, AODV and TORA), and queuing (FIFO, PQ and WFQ) for video conferencing over wireless ad hoc networks has been studied using OPNET tool. The results showed that AODV presented good results with different scenarios.

REFERENCES

Azzedine Boukerche et al. 2009. "Algorithms for Mobile Ad hoc Networks", In Algorithms and Protocols for Wireless and Mobile Ad hoc Networks", Edited by: Azzedine Boukerche, John Wiley and Sons International Publisher, USA.

Kai-Wei Fan, Sha Liu, Prasun Sinha. 2006. "Ad Hoc Routing Protocols", In Handbook of Algorithms for Wireless Networking and Mobile Computing, Edited by: Azzedine Boukerche, Chapman and & Hall/CRC Publisher, USA.

Anuj K. Gupta, Harsh Sadawarti, Anil K. Verma. 2010. "Performance analysis of AODV, DSR & TORA Routing Protocols". In IACSIT International Journal of Engineering and Technology, Vol.2, No.2, pp.226-231.

Sunil Taneja and Ashwani Kush. 2010. "A Survey of Routing Protocols in Mobile Ad Hoc Networks". International Journal of Innovation, Management and Technology, Vol. 1, No. 3, pp.279-285.

Evans J. and C. Filsfils, Deploying IP and MPLS QOS for Multiservice Networks: Theory and Practice, Morgan Kaufmann Publishers, USA, 2007

Balakrishnan R., Advanced QoS for Multi-Service IP/MPLS Networks, Wiley Publishing, USA, 2008

Kurose J. F. and K. W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 2012

OPNET Product Documentation v.11.0.A, OPNET Technologies, Inc., Bethesda, MD, 2004.

HYBRID MODELLING OF INTEGRATED CARE SYSTEMS

Tillal Eldabi
Brunel Business School
Brunel University London
Uxbridge
Middx
E-mail: tillal.eldabi@brunel.ac.uk

Keywords
Hybrid simulation, Combined simulation, Complex Systems

ABSTRACT

Improved life expectancy and prosperity lead to an increased number of aged people resultin in more need for elderly care coupled by the necessity to integrate health and social care. To cope with such complexity, many modelling methods have been employed to assist in capturing health and social care problems. However, such methods are mainly used separately and without any coordination. Several techniques that have been employed mostly to model the care integration can be found in literature. Literatures also suggests that there are some challenges which still persist when modelling integrated care. One of the main challenges is the complexity of integrated systems. Therefore, this paper attempts to explore the main reasons behind these challenges. The paper shows that the combined application of OR/Simulation methods would make use of the strengths of individual techniques, while reducing their limitations. In this paper an example of joint application is given (SD-DES) to show how combined modelling would help model complex and integrated systems.

INTRODUCTION

The continuous increase in world population is widely attributed to developments in science and technology which improved quality as well as quantity of life. Thus the life expectancy for humans has increased dramatically worldwide. As a result, elderly services needs have increased along with demands for elderly people services. Integrated care (IC) is a system type that is rapidly making grounds as an alternative way to provide such services. The high demand for integrated care along with the complexity of such systems have attracted the interest of many researchers in an effort to develop the best methods to model integrated care (Mur-Veeman et al., 2008). However, and despite having models of integrated care being developed, these models have not been able to represent the real nature of IC. Therefore, this article paper attempts to address these questions by reviewing existing IC models and proposing a modelling approach – based on the combined used of Discrete Event Simulation And System Dynamics - to meet such complexities.

MODELS OF INTEGRATED CARE

There are a number of studies related to modelling and simulation of integrated care systems. These are considerably less than what is found in other areas and applications such as productions, management and other areas. Discussion from Zulkepli and Eldabi (2011) concluded that there is no single tool that can be used for modelling integrated care to assist decision making both in the short and long term. They go to express that Discrete Event Simulation (DES) has the capability of providing and assisting short term decision making as it is used to model the problem operationally and tactically. On the other hand, System Dynamics (SD) has the capability of assisting long term decision making as it is used to model the problem from a strategic perspective (Chahal and Eldabi, 2008). Therefore, to improve the integrated care model into a viable model, it is suggested that the two techniques should be combined, namely SD and DES as a hybrid simulation technique. By combining these techniques, the limitations in the DES can be covered by the SD technique while limitations in the SD can be covered by the DES. Figure 1 shows how the hybrid technique can help in improving the IC model and therefore, improve the decision making process.

COMBINED SD-DES SIMULATION

SD and DES are two simulation techniques that have been applied in a hybrid context in many fields, for example, Desai et al. (2008), Campbell et al. (2001), Katsaliaki et al., (2005), Wolstenhome et al. (2004). However, developing an IC model using a single simulation technique may lead to losing some of the important features associated with the system. For example, using DES alone would not capture any *feedback loop* features. In addition to that DES models have the tendency to rapidly grow in complexity making it difficult to understand (Chahal and Eldabi, 2008). On the other hand, using SD alone would not capture individual details (of patients, clinicians, clinics, etc.). Therefore, it is suggested that combining SD and DES will benefit from their strengths, while reducing their limitations. Having said that, we find that only a handful of studies have examined the applicability of hybrid SD-DES techniques for modeling systems.

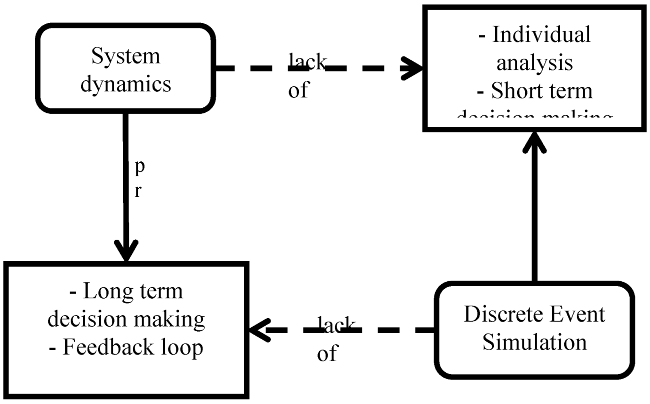


Figure 1: How Hybrid Techniques can Improve the IC Model

AN EXAMPLE OF HYBRID TECHNIQUE - COMBINED SD-DES MODELLING

In this section we provide an example for modelling an integrated care system to demonstrate the benefits related to using hybrid approaches. There are several problems related to IC that have been highlighted in literature, e.g., Mur-Veeman et al. (2008), Glendinning et al. (2005), and Van Raak et al. (2005) and among them is the problem of late transfer to social care. Lack of integrated planning leads to issues such as to bed blocking, where patients stay in the hospital due to the lack of social care. In order to address this problem, *intermediate care* facilities are proposed to provide temporary placement to the patients while waiting for social care. In this case, the hybrid model consists of three models – a DES hospital model (Critical Care Module), a DES intermediate care model (Intermediate Care Module), and an SD model of critical care, intermediate care and social care (IC module).

The Critical Care Module (DES model)

The Critical Care Module models the patient flow through a critical care facility, e.g., a hospital (Figure 2). It starts with the patient being admitted to the ward. The patient is then assessed by the doctor and will undergo some tests. Following the doctor's initial assessment some patients will be discharged, whereas others may require further assessment. A patient that needs to undergo a surgical procedure may be admitted to a surgical ward while waiting for the surgery. Subsequent to the operation the patient will be transferred to a recovery room for postoperative recovery. The patient will then be transferred to the normal ward and will remain here until medically fit to be discharged. There are two alternate postoperative pathways for the patient from this point on, (a) either the patient will be discharged, and (b) the patient will require continuing care services..

The Intermediate Care Module (DES Model)

The Assessment/Intermediate Care starts when patients are ready for discharge. The assessment is conducted by occupational therapists and psychiatric nurses. This assessment ensures that the care placements to be allotted to the patients will meet their medical needs. Subsequent to this, the care manager will assess whether the patients require personal care and whether they can carry out household tasks. Following this, a care package will be created and an appropriate care placement must be found. Thus, the assessment and intermediate care processes are considered as one process and only one DES model is considered.

The IC Module (SD Model)

SD is used for modeling intangible variables that relate to those providing care services to the patients (e.g., the nurses and the clinician). Examples of such variables include performance, motivation and stress level; these variables are influenced by the level of patients flow. The DES output is the total number of patients entering acute care; this will, in turn, influence the stress level of the staff, leading to increased patient discharge to the intermediate care. Whilst in intermediate care, the increase in patients will increase assessment time, influencing the stress level of staff leading to an increase in discharge rate and patient readmission. When this situation occurs, it will create bed blocking and will consequently increase the stress level of the clinicians. To make resources available again in critical care, it is arguable that the discharge rate to intermediate care will be increased.

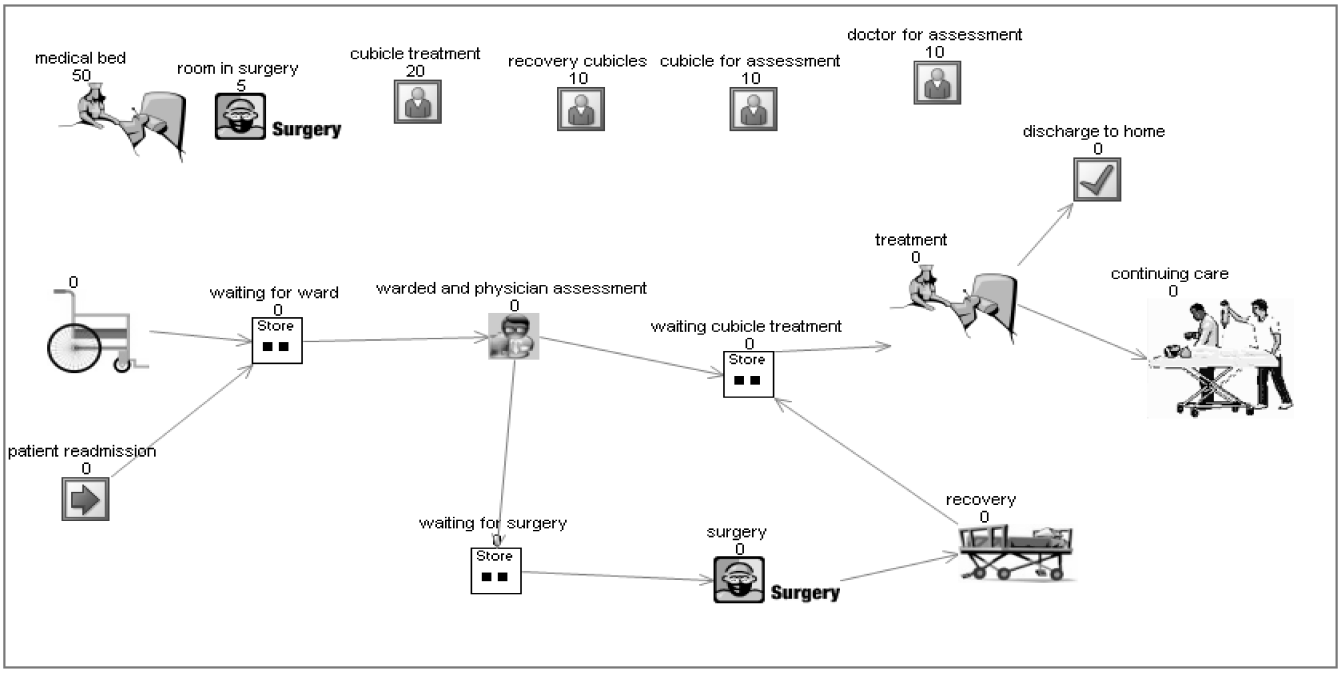


Figure 2: DES Model of the Critical Care Module

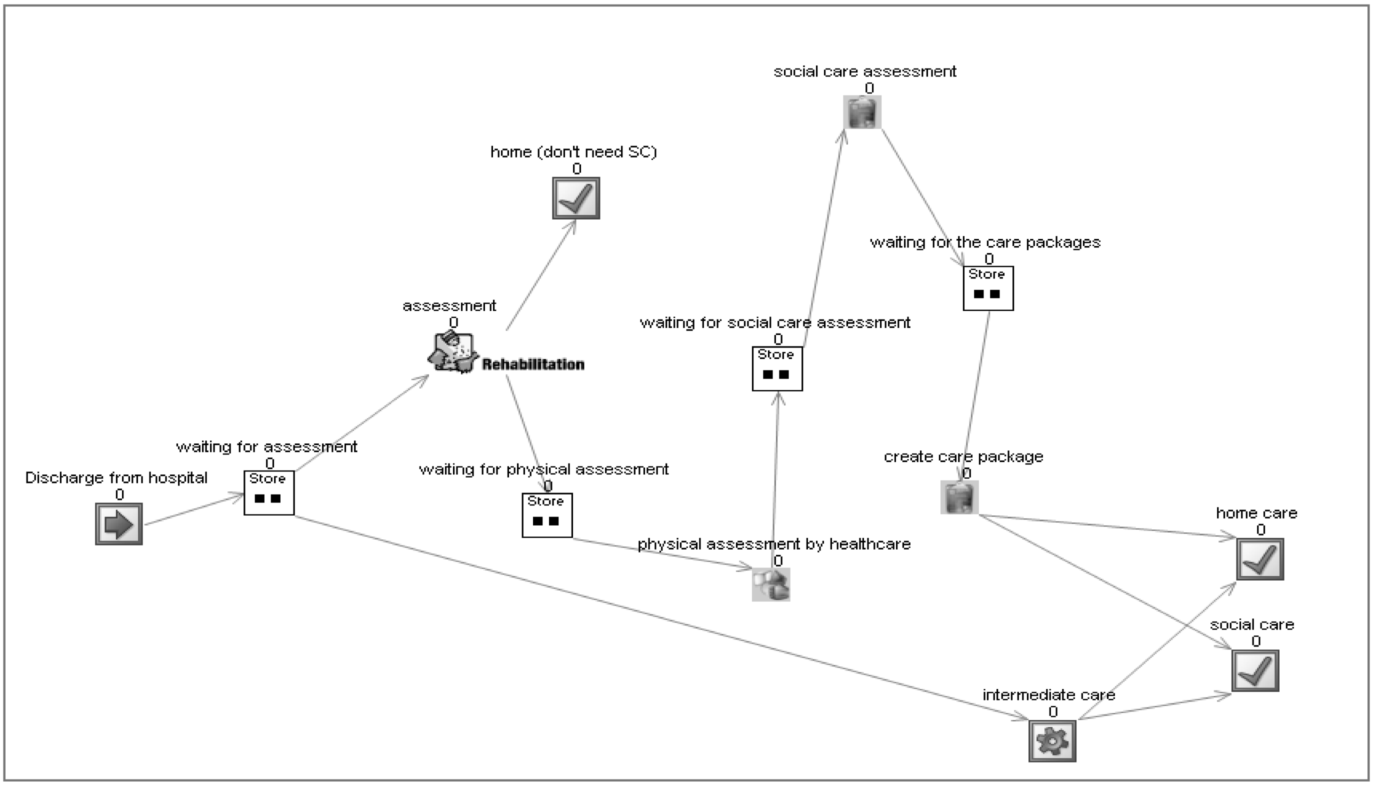


Figure3: DES Model of the Assessment/Intermediate Care Module

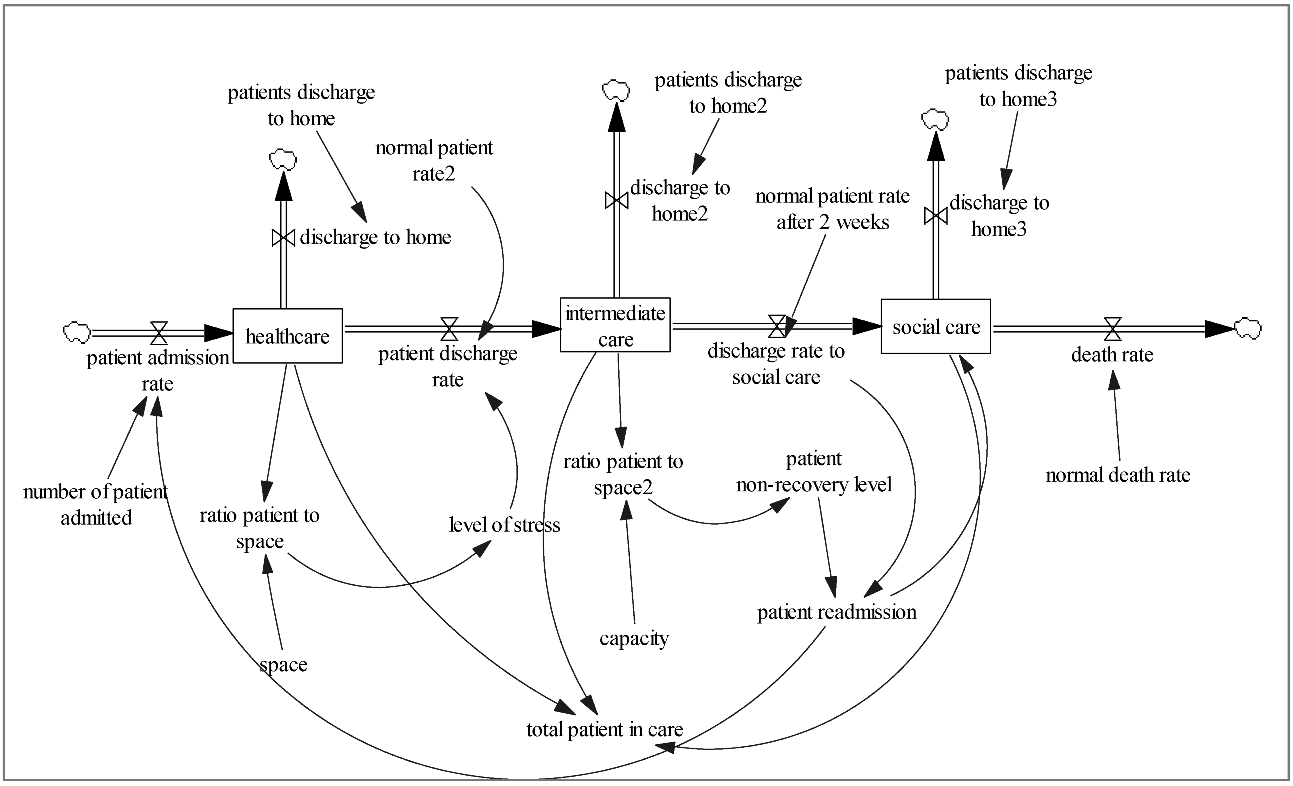


Figure 4: SD Model of the Integrated Care Module

MODEL INTEGRATION

The hybrid SD-DES approach necessitates the identification of relationships among the variables present in DES and SD models. In this case, the DES variable "number of patient admitted" from the Critical Care Module influences the SD variable " stress level" in the Integrated Care Module, wherein an increase in patient admission will result in an increase in the stress level among the healthcare professionals. Elf and Putilova (2005) argued that physical spaces influence patients' health. This scenario might create another condition wherein patients may be readmitted to critical care, since, due to the stress level increase, the professional judgment pertaining to patients' health may be wrong. Thus, we identify the realtionship between the following sets of variables:

Patient admission → stress level → patient discharge rate
Patient admission → stress level → patient readmission

The source DES model is executed first, the output of which will be used as inputs for the SD model. The SD model will then be executed where its output "number of patients" will be inserted back into the DES model. The DES model will be executed again to provide new results after considering intangible factors, i.e. stress level and non-patient recovery level. The result of the combined SD-DES simulation should be considered as a whole in order to find the best result for implementation.

CONCLUSIONS

Health and social care integration are complex systems as they combine many stakeholders, different processes, procedures and pathways for every patient as well as different agencies involved in handling the same patient. In order to reduce the complexity, modelling is used to depict the system for better understanding. An important element in developing hybrid SD-DES models is to make sure that both models are balanced in their inputs/outputs. The need for developing system models using hybrid SD-DES techniques will depend on several factors, such as, its impacts on the current systems, the need for individuality analysis, and whether the system has a feedback element. The researchers are planning to run further experiments on more complex situations; this will provide more realistic assessments of this approach. Furthermore, the researchers are planning to automate the process of values transferred between models to speed up the overall process, this in addition to using other techniques such as ABS. It must be noted that this proposed approach is not a substitute to existing technologies, rather provides a structured framework for modeling IC environments without starting from scratch.

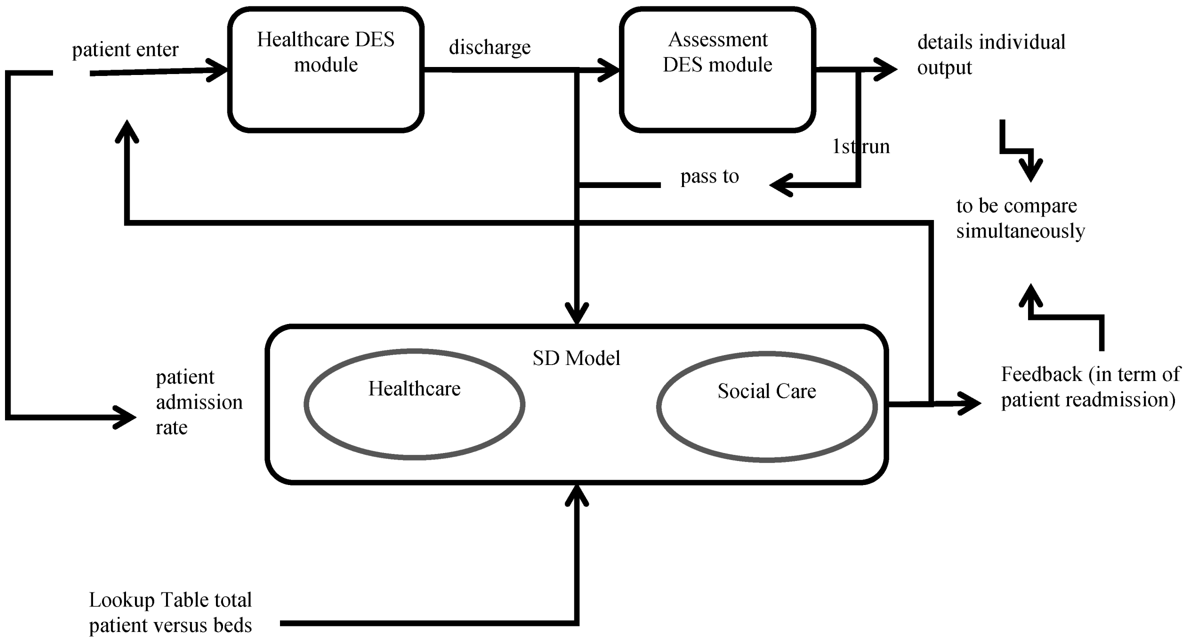


Figure 5: Models Communication

REFERENCES

Brailsford, S. C. 2008. "System dynamics: what's in it for healthcare simulation modellers". In: Mason, S.J., Hill, R. R., Monch, L., Rose, O., Jefferson, T., Fowler J.W. (eds). *Proceeding of the 2008 Winter simulation conference.* pp 1478 – 1483.

Campbell, H., Karnon J. and Dowie R. 2001. "Cost analysis of a hospital-at-home initiative using discrete event simulation". *Journal of health and service research policy*. 6 (1) pp 14 – 22.

Chahal K. and Eldabi, T. 2008. "Applicability of hybrid simulation to different modes of governance in UK healthcare". In: Mason, S.J., Hill, R. R., Monch, L., Rose, O., Jefferson, T., Fowler J.W. (eds*). Proceeding of the 2008 Winter simulation conference.* pp 1469 – 1476.

Chahal K., Eldabi T. and Mandal A. 2009. "Understanding the impact of whiteboard on A&E department using hybrid simulation". *Proceeding of 27th International Conference of the system dynamics society*. Albuquerque, New Mexico, USA

Desai M. S., Penn M. L., Brailsford S. and Chipulu M. 2008. "Modelling of Hampshire adult services – gearing up for future demands". *Journal of health care management sciences*. 11 (2) pp 167 – 176.

Elf, M. and Putilova M. 2005. "The care planning process – a case for system dynamics". *Proceedings of the 25th International Conference of the System Dynamic Society*. 25(1) pp 1 – 18.

Glendinning C., Hudson B. and Means R. 2005. "Under strain? Exploring the troubled relationship between health and social care". *Journal of public money and management*, 25(4) 245 – 251.

International Data Base 2008. "World population information". Available from http://www.census.gov/cgi-bin/ipc/idbagg [Accessed 23rd April 2008].

Katsaliaki K., Brailsford S., Browning D. and Knight P. 2005. "Mapping care pathways for the elderly". *Journal of health organization and management.* 19 (1) pp 57 – 72.

Kodner D. and Kyriacou C. 2000. "Bringing managed care home to people with chronic disabling conditions: challenges and prospects for policy, practice and research". Commissioned paper for the home care research initiative at the Visiting Nurse Service of New York.

Mur-Veeman I, van Raak A. and Paulus A. 2008. "Comparing integrated care policy in Europe: does policy matter?" *Journal of health policy*, 85 (2) 172 – 183.

McClean S. and Millard P. 2007. "Where to treat older patient? Can Markov models helps us better understand the relationship between hospital and community care?" *Journal of the operational research society*. 58 (2): 255 – 261.

Van Raak A., Paulus A. and Mur-Veeman I. 2003. "Why do health and social care providers co-operate." *Journal of health policy*, 74 (1) 13 – 23.

Wolstenhome, E., McKelvie, D., Smith, G. and Monk, D. 2004. "Using system dynamics in modelling health and social care commissioning in the UK". *Proceedings of the 22nd International Conference of the System Dynamic Society*. 24(1) pp 1 – 20.

Xie, H., Chaussalet, T. J. and Millard, P. H. 2005. "A continuous time Markov model for the length of stay of elderly people in institutional long-term care". *Journal of the Royal Statistical Society: Series A (Statistics in Society)*. 168 (1) pp 51 – 61.

Zulkepli, J. and Eldabi, T. 2011. "Technique for improving care integration model." *European, Mediterranean & Middle Eastern Conference on Information Systems 2011* (EMCIS 2011), May 31, 2011, Athens, Greece.

BIOGRAPHY

TILLAL ELDABI is a senior lecturer at Brunel University. He has B.Sc.in Econometrics and M.Sc. and Ph.D. in Simulation Modeling in Healthcare. His research is into aspects of healthcare modeling and simulation. He developed a number of models and bespoke packages to support health economists and clinicians to decide on best treatment programs. Dr Eldabi has published widely in the field of modeling in healthcare and edited a number of special issues in highly respected journals. Dr. Eldabi's email address is tillal.eldabi@brunel.ac.uk.

AUTHOR LISTING

AUTHOR LISTING