

Smart STEM-Driven Computer Science Education

Vytautas Štuikys • Renata Burbaitė

Smart STEM-Driven Computer Science Education

Theory, Methodology and Robot-based
Practices



Springer

Vytautas Štuikys
Department of Software Engineering
Kaunas University of Technology
Kaunas, Lithuania

Renata Burbaitė
Department of Software Engineering
Kaunas University of Technology/Panevėžys
Juozas Balčikonis gymnasium
Kaunas/Panevėžys, Lithuania

ISBN 978-3-319-78484-7 ISBN 978-3-319-78485-4 (eBook)
<https://doi.org/10.1007/978-3-319-78485-4>

Library of Congress Control Number: 2018939185

© Springer International Publishing AG, part of Springer Nature 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG part of Springer Nature.

The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

At least in this decade, the acronym STEM (Science, Technology, Engineering, and Mathematics) was among dominating terms and themes in the literature on educational research and practice. In recent years, however, the interest in STEM education has grown even more intensively in both schools and universities, as well as among responsible bodies and organizations, including governments. The signs of that are many reports, the ever-growing stream of supporting initiatives in the USA, Europe, and research activities worldwide in the field. This interest arises from the urgent need to respond to the economic, social, and technological challenges of the twenty-first century. The rapid development of science and technology raises new challenges for labor markets and education. Nowadays, as never before, the modern economy requires the workforce to be equipped with the interdisciplinary knowledge based on using high-tech equipment and methodologies. STEM is, in fact, an interdisciplinary approach to learning. It brings rigorous academic concepts coupled with real-world lessons and tasks as students apply science, technology, engineering, and mathematics in contexts that make connections between school, community, work, and the global enterprise. That enables the development of STEM literacy and with it the ability to compete in the new economy.

The other important research stream focuses on the efforts to improve Computer Science (CS) education. It is difficult to overestimate the role of CS in the twenty-first century as leading science, because practically all sciences use CS approaches. The educators and scientists commonly agree that CS education in high school must consist of two parts. The first part should focus on learning knowledge and skills of ICT and its devices. Typically, we define that as *Digital Literacy*. However, these skills are short-living knowledge, because they are changing with technology. The second part should focus on learning the fundamentals of CS that are essential to understanding our digital world that advances extremely rapidly. This field is also known as *Informatics*, especially in Europe, and brings the long-living knowledge, which lasts forever and does not change with technology.

There is yet another exciting research area the concepts of our book deals with directly, i.e. educational robotics. Robots play an extremely significant role in

education due to their “smart functionality” and the ability to engage students in the learning process directly. It is so because robotics entails the possibility to demonstrate various tasks and situations taken from the real world and in this way to enforce the motivation to learn. Indeed, educational robots are smart devices and therefore are so widely used in the field. Robots are able to perform mechanical jobs with little to no human interference. On the other hand, robots contain units to perform computations and control tasks to support mechanical actions. Therefore, it is possible to treat the robot as a dedicated computer with memory and processor, the main units of a conventional computer. For a long time in education, computers have been in use in the mode *use-as-is* (except perhaps specific electronics courses). Similarly, in many cases, we use educational robots in that mode. However, typically robots have a modular structure enabling teachers and students to assemble the whole architecture from the available parts. This activity, in fact, means the use of the pedagogical approach known in the literature under a generic term *learning-by-doing*, though there are other terms such as a *constructivist* approach or *inquiry-based learning* in terms of STEM.

The above-mentioned fields, i.e. STEM, CS education, and educational robotics, are highly heterogeneous in their own rank. Now a myriad of approaches exists to deal with those paradigms separately. However, we still know little about how it would be possible to combine those paradigms seamlessly into a coherent methodology in order to gain the synergistic benefit of all. This book is just about that. The basic idea we discuss in this book is how to project the extremely high variability of STEM pedagogy, STEM and CS-related content onto high school CS curricula, using modern technologies of two types. The first type is educational robotics used as direct facilities of learning in a high school. The second is supporting technologies to enable achieving goals of automation in the preparation, design, and use of the content and learning processes. The supporting technologies include high-level modelling and feature model transformations using meta-programming techniques. Note that we are able to integrate seamlessly those technologies with the conventional software.

In this book, we present the approach that distinguishes from known ones by the following attributes. Firstly, we focus on integrative aspects of the STEM by introducing it into the CS education courses, such as Programming, at the high school level. Secondly, we apply robotics as the relevant means to CS and STEM education. Thirdly, we introduce smart approaches to support this integration. By smart approaches, we mean those that rely on automation, generativity, context-awareness, automated adaptation, and application of knowledge-based approaches such as agent-based in designing and use of the learning content and scenarios. Finally, we present the overall material as a coherent methodology that covers methodological aspects, theoretical aspects, and practices. The latter includes multiple use cases, evaluation procedures, and figures taken from the real teaching setting at the one high school.

This monograph is a result of our intensive research in the field of automation of the educational processes and approaches. To some extent, this book is a continuation of the previous one published by Springer in 2015 (we mean “Smart Learning

Objects for Smart Education in Computer Science: Theory, Methodology and Robot-Based Implementation”). Though it has the same theoretical background as this one (i.e. high-level feature-based modelling, and transformation of those models using heterogeneous meta-programming), this book presents new findings. The novelty of this book covers multiple aspects. Firstly, it includes the explicit integration of STEM concepts into CS courses. This integration requires a new vision, new frameworks, and extended approaches, we have described in this book. Secondly, the book presents a new vision and models of representing smart content and smart scenarios. Thirdly, this book provides new concepts of the Personal Generative Library and Smart educational environments. Finally, this book discusses the relationship of STEM education with emerging paradigms such as the Internet-of-Things and MOOC.

What is the structure of this book? We have divided the content into five parts. Part I is the introductory part and, in fact, is the wide context of the remaining parts. Part II represents the methodological and theoretical background. Part III deals with the design and use of the smart content to support STEM and CS education. Part IV deals with the infrastructure of our approach. Part V extends the vision of our approach.

Part I includes Chaps. 1, 2 and 3. Chapter 1 is about the challenges of STEM-driven Computer Science (CS) education. Chapter 2 discusses a vision for introducing STEM into CS education at school. Chapter 3 deals with educational robotics and smart devices as a way of obtaining the interdisciplinary knowledge.

Part II includes Chaps. 4 and 5. Chapter 4 focuses on learning variability as a methodological background of STEM-driven reuse-enhanced CS education. Chapter 5 provides a theoretical background consisting of feature-based modelling to specify and represent the STEM learning variability and basics of heterogeneous meta-programming to implement the proposed STEM-driven approaches.

Part III includes Chaps. 6, 7, 8 and 9. Chapter 6 presents a vision for understanding smart learning objects (SLOs) for STEM-driven CS education. Chapter 7 describes the design and redesign of STEM-oriented generative learning objects (GLOs). Chapter 8 deals with the design of STEM-oriented smart learning objects (SLOs). Chapter 9 describes agent-based smart learning objects for the STEM.

Part IV includes Chaps. 10, 11, 12 and 13. Chapter 10 presents the concepts and implementation of the personal generative library for STEM-driven educational resources. Chapter 11 deals with the design of STEM-driven generative scenarios for CS education. Chapter 12 analyses the smart STEM-driven environment for smart CS education at high school. Chapter 13 summarizes the educational practices of smart STEM-driven CS education at one high school.

Part V includes Chaps. 14 and 15. Chapter 14 introduces the Internet-of-Things (IoT) vision in CS education and gives an idea for the remote STEM-driven environment. Chapter 15 provides a finalizing discussion, outlines other educational paradigms (such as MOOC and STEM for universities), and formulates open issues.

Who could be the potential reader of the book? We dedicate the book in the first place to the researchers in STEM-driven CS education, CS researchers, especially to

those who are interested in using robots in learning and teaching, course designers, and educational software and tools developers. The CS teachers should also be highly interested not only in reading, but in studying the adequate chapters as their advanced teaching material. We hope that the content of the book will be understandable to anybody who has enough skill in programming. Therefore, students studying CS-related courses, especially master-level and PhD students, are also potential readers. As the book includes the wider context (e.g. reusability aspects in technology-enhanced learning and the educational research activities in STEM and CS), the other e-Learning community members might be interested in reading the book as well (especially the modelling of CS education and the integrative aspects of technology and pedagogy).

How should one read the book?

There is no specific algorithm for selecting and prioritizing the chapters for reading. Our writing priority was to present the content so that it would be possible to read a separate chapter independently from others. Nevertheless, some selecting and sequencing seem to be helpful for a deeper understanding. This depends on the reader's status, previous knowledge, and his or her intention. The title of a particular chapter, for example, can be an indicator to make a selection. Our recommendation is the following scheme. The senior researchers and policymakers should first read Chap. 1 and, perhaps, all introductions in each chapter and then to move to the ending sections in each chapter. After that, the readers will have the possibility to make the relevant choice for the in-depth studies of the full material within the chapters. Experts and knowledgeable researchers first could read the introduction and concluding parts of each chapter, or some selected chapters depending on the reader's flavor. If they will find interesting ideas, they could study a particular chapter more intensively. The readers who will select some material for their own research topic should also go through the relevant references.

What about the CS teachers and students? Those readers might use the book's content differently. Chapters 12 and 13 are mainly dedicated to the secondary (high) schoolteachers and students. For example, Section 13.4 presents *the full scenarios* on how to use smart LOs and smart educational robot-based environments to teach and learn STEM-driven CS topics at the school level. The university-level educators and students should use the book content with regard to their teaching/learning topics. For example, Chap. 4 is relevant to teach and learn the feature-based modelling methodology. Chapters 5, 6, 7, 8, 9, 10 and 11 better fit for teaching and learning topics related to model transformations. The educators of CS teachers' should use the book entirely.

We hope that the book will be the beneficial methodological instrument (due to the use of multiple illustrative examples and case studies) for those educators who are ready to provide the innovative models and methods in CS education on the STEM paradigm.

Kaunas, Lithuania
Kaunas/Panevėžys, Lithuania

Vytautas Štuikys
Renata Burbaitė

Acknowledgments

Firstly, we look at the prehistory of writing this book. We could hardly have written this monograph without the aspiration we have experienced in preparing the EU project-proposal on similar topics in 2015. This activity has brought to us a clear understanding of how topics on STEM and CS education are important for huge communities worldwide. The next impact was our continuing research and accumulated practice of the second author in teaching CS based on using robotics in the real setting. Researching and working from day to day in the classroom with students from 2011 onward, we have accumulated a tremendous amount of educational data. The teacher was able to discuss the issues with students, to evaluate, and to carry out the research work in the classroom, as well as to enforce and present the outcomes for a wider audience. Students' contribution to the activities and outcomes in collecting and generalizing data was noticeable. We would like to thank them all. Some of the students became apprentices of the teacher. Especially we would like to mention Adomas Paulauskas, a winner of the special prize of the best computational project in the EU contest of young scientists in Tallinn, in 2017.

We also thank our team members, especially professor R. Damaševičius, and the administration of Informatics faculty at Kaunas University of Technology and the administration of Panevėžys Balčikonis Gymnasium (Lithuania) for promotion and support in writing this book. Finally, we thank anonymous reviewers for their comments on the fragments of the manuscript.

We dedicate this book to the prominent date – the centenary jubilee of restoration the State of Lithuania. On the 16th of February 1918, the Council of Lithuania proclaimed the Act of Reinstating Independence of Lithuania. . .

Nowadays we believe in the talent and creativity of our young generation and the rest of the world. We hope this book will enrich by new ideas the educational researchers, schoolteachers, students, and other creators of our digital future.

Contents

Part I

Introductory Part: Motivation, Challenges, and Conceptual Vision of STEM-Driven CS Education Based on Robotics

1

Challenges of STEM-Driven Computer Science (CS) Education . . .

3

1.1

Introduction

3

1.2

Motivation of Our Approach

5

1.3

Smart Education and STEM

9

1.4

Ten Top Challenges in STEM Education

10

1.5

CS Teaching Challenges Without STEM Context

13

1.6

Challenges of STEM-Driven CS Education

15

1.7

The Book’s Objectives and Research Agenda

20

1.8

The Topics This Book Addresses

21

1.9

Summary and Concluding Remarks

23

References

.

23

2

A Vision for Introducing STEM into CS Education at School

31

2.1

Introduction

31

2.2

Related Work

33

2.2.1

STEM-Based Education Challenges

34

2.2.2

The Role of CS in STEM-Oriented Education

34

2.2.3

The Role of Smart Devices and Educational Robotics in
STEM-Driven CS Education

35

2.2.4

The Role of Context in Analysis and Design of Educational
Systems

36

2.3

A General Description of Our Approach

38

2.3.1

A Conceptual Model of STEM-Driven CS Education

38

2.4

A Framework to Implement the Proposed Conceptual Model

42

2.5

Basis for Implementing Our Approach: A Process-Based
Vision

45

2.6	STEM-Driven Learning Processes as a Problem Domain	48
2.7	Summary, Discussion and Overall Evaluation	50
2.8	Conclusion	52
	References	52
3	Smart Devices and Educational Robotics as Technology for STEM Knowledge	57
3.1	Introduction	57
3.2	Related Work	58
3.3	Introducing Robotics in STEM-Driven CS Education	61
3.4	Educational Robot Generic Architecture	64
3.5	Conceptual Model of STEM-Driven Environment	65
3.6	Discussion and Conclusion	66
	References	66
 Part II Methodological and Theoretical Background of Approaches to Implement the Proposed Vision		
4	A Methodological Background for STEM-Driven Reuse-Enhanced CS Education	71
4.1	Introduction	71
4.2	Related Work	73
	4.2.1 Variability Research in SWE	73
	4.2.2 Variability in Learning	77
4.3	Explicit Representation of Variability: A Motivating Example . . .	81
	4.3.1 Capabilities of Feature Diagrams in Learning Object Domain	85
	4.3.2 Limitations of Feature Diagrams in Learning Object Domain	86
4.4	A Framework to Implement Learning Variability in STEM Paradigm	86
4.5	Motivation of STEM-Driven Research Topics	88
4.6	Two Approaches of Dealing with Variability in STEM	89
4.7	Summary, Evaluation and Extended Discussion	91
4.8	Conclusion	94
	References	95
5	Theoretical Background to Implement STEM-Driven Approaches	99
5.1	Introduction	99
5.2	Motivation and Methodology of Describing the Background . . .	100
5.3	Related Work	102
5.4	Background of Feature-Based Modelling	107

5.4.1	A Vision for Researching STEM-Driven CS Education	107
5.4.2	Basics of Feature Modelling	108
5.4.3	Formal Definition of Features and Constraints	111
5.4.4	Static and Dynamic Feature Models	112
5.4.5	Mechanisms to Support Dynamicity for STEM	114
5.5	Meta-programming as Solution Domain	116
5.5.1	Meta-program of Type 1	119
5.5.2	Meta-program of Type 2	121
5.5.3	Meta-program of Type 3	123
5.5.4	Meta-program of Type 4	124
5.6	Data Transfer Modes Formal Definition	126
5.7	Summary, Evaluation and Conclusion	126
	References	128

Part III Design, Re-design, and Use of Smart Content for STEM-Driven CS Education

6	Understanding of Smart Content for STEM-Driven CS Education	135
6.1	Introduction	135
6.2	Related Work	136
6.3	GLO/SLO Evolution Curve	138
6.4	A Framework to Define and Understand SLOs	140
6.4.1	Learner's Vision	141
6.4.2	Teacher's Vision	144
6.4.3	Designer's Vision	145
6.5	SLO Evolution Vision: Researcher's Perspective	148
6.6	Summary, Evaluation and Conclusion	152
	References	153
7	Model-Driven Design and Redesign of Smart STEM-Driven CS Content	157
7.1	Introduction	157
7.2	Related Work	159
7.3	Two Conceptual Models and Two Approaches to Design SLOs	161
7.4	Problem Statement for STEM Content Design and Redesign	162
7.5	Model-Driven Framework to Design CS-Based SLOs	165
7.5.1	Understanding of Context in Our Approaches	167
7.5.2	Model-Driven SLO Design at the Top Level	169
7.5.3	SLO Design at the Intermediate Level	173
7.5.4	SLO Design at the Low (Coding) Level	176
7.6	Theoretical Background of the Approaches	177

7.7	Generic Transformation Rule	180
7.7.1	Generic Rule	181
7.8	Summary, Overall Evaluation and Conclusion	183
	References	185
8	Stage-Based Smart Learning Objects: Adaptation Perspective	189
8.1	Introduction	189
8.2	Related Work	191
8.2.1	Context-Related Issues in TEL	191
8.2.2	The Term Stage and Relevant Methodologies	192
8.2.3	Adaptation in e-Learning	193
8.3	Motivation of the Approach	194
8.4	Categories of Learning Objects to Support STEM	195
8.5	A Framework and Tasks to Develop SB GLOs	197
8.6	A Background of the Approach	198
8.6.1	Basic Assumptions	198
8.6.2	Definition of Basic Terms	199
8.6.3	Basic Properties	200
8.7	Staging and Context Awareness	203
8.8	Integration of STEM Concepts into SB Model	204
8.9	Case Study	206
8.10	Stage-Based Adaptation Processes and Scenarios	207
8.11	Analysis of Capabilities of the SB Model	210
8.11.1	Designer's Perspective	210
8.11.2	Teacher's Perspective	211
8.11.3	Student's Perspective	212
8.12	Summary and Concluding Remarks	213
	References	214
9	Agent-Based GLOs/SLOs for STEM	217
9.1	Introduction	217
9.2	Related Work	219
9.3	GLOs and SW Agent Domains Analysis: Problem Statement	220
9.4	Robot-Oriented Agent-Based Educational Environment: Architecture and Processes	223
9.5	Implementation of Software Agent: A Case Study	225
9.5.1	How to Integrate Technological Agent into Our Vision of STEM?	225
9.6	Evaluation and Conclusion	227
	References	227

Part IV Infrastructure to Support STEM-Driven CS Educational Practice

10 Personal Generative Library for STEM-Driven Educational Resources	233
10.1 Introduction	233
10.2 A Concept of Personal Generative Library and More	234
10.3 Related Work	236
10.4 Tasks in Creating PGL for STEM-Driven Education	239
10.5 Basic Idea of the Approach	239
10.6 Background of the Approach	240
10.6.1 Definitions of Basic Terms and Relationships	241
10.7 A Detailed Description of the Approach	244
10.8 A Methodology of Experiments and Case Study	246
10.8.1 Results of Modelling	246
10.8.2 A Case Study: Results Obtained by the Generated Programs	247
10.9 Adaptation of PGL Concept to STEM Library	248
10.10 Summary, Discussion and Evaluation	252
10.11 Conclusion	253
References	254
11 A Methodology and Tools for Creating Generative Scenario for STEM	259
11.1 Introduction	259
11.2 Related Work	261
11.2.1 Part A: STEM Context Issues	261
11.2.2 Part B: Educational Scenarios Review	263
11.3 Research Tasks and Methodology	265
11.3.1 A Framework for Creating Scenarios	265
11.3.2 Architectural Aspects	267
11.3.3 Design Processes to Develop Generative Scenario	270
11.4 Methods Used	272
11.5 Discussion and Summarizing Evaluation	273
11.6 Conclusion	276
References	276
12 Smart STEM-Driven Educational Environment for CS Education: A Case Study	279
12.1 Introduction	279
12.2 Related Work	281
12.3 A Framework for Creating Smart Educational Environments: Principles and Requirements	285
12.4 Part 1: Architectural and Functional Aspects of STEM-Driven SEE	288

12.5	Part 2: Main Features of the STEM-Driven SEE for CS Education	289
12.5.1	Communication Processes: A User Perspective	291
12.5.2	Structure of the Server Part	293
12.6	Part 3: Evaluation of Smart Educational Environments	294
12.7	Discussion, Summary and Conclusion	296
	References	300
13	Practice of Smart STEM-Driven CS Education at High School	305
13.1	Introduction	305
13.2	Related Work	306
13.3	Curriculum of Programming Basics to Support Our Approach	307
13.4	Case Studies	312
13.4.1	Case Study 1	313
13.4.2	Case Study 2	314
13.4.3	Case Study 3	317
13.5	Evaluation	321
13.6	Conclusion	323
	References	324
 Part V An Extended Vision to STEM-Driven CS Education		
14	Internet-of-Things: A New Vision for STEM and CS Education	327
14.1	Introduction	327
14.2	Related Work	328
14.2.1	Stream A	329
14.2.2	Stream B	330
14.3	An Architecture of IoT System for STEM-Driven CS Education	334
14.4	A Framework to Consider Educational Tasks in Relation to IoT	337
14.5	Case Study: Line Following Task	339
14.6	Summary and Conclusion	342
	References	343
15	A Finalizing Discussion and Open Issues	347
15.1	Introduction	347
15.2	A Summary and Evaluation of the Proposed Approach	348
15.2.1	Capabilities of Smart Learning Objects	350
15.2.2	Capabilities of Generative Learning Scenario	351
15.2.3	How Does Our Approach Support Computational Thinking?	353

15.2.4	Drawbacks of the Proposed Approach	355
15.2.5	Difficulties from Teacher's Perspective	355
15.2.6	Drawbacks from Learner Perspective	356
15.3	Applicability of the Approach at the University Level	357
15.4	STEM in Industry: A New Way	357
15.5	MOOC and STEM-Driven CS Education	359
15.6	Open Issues	361
	References	363
	Glossary	365
	Index	367