

Enterprise Information Systems

Boris Shishkov

Enterprise Information Systems

A Modeling Approach



Author

Boris Blagovestov Shishkov

Bulgarian Academy of Sciences / IICREST

53, Iv. Susanin Str., AT. 4

1618 Sofia, Bulgaria

b.b.shishkov@iicrest.org

ISBN 978-619-90124-3-7

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the right of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks.

iicrest.org

© IICREST Sofia 2017

Printed in Bulgaria

To my Dad

PROLOGUE

*We arrive at the truth, not by the reason only,
but also by the heart.
Blaise Pascal*

Businesses changed when computers first appeared on the scene and enterprises had to accommodate (partial) automation of their business processes. Businesses changed again when web services and cloud infrastructures appeared, and enterprises faced the challenge of making (some) internal business processes external. All those changes have been justifying more and more the necessity of bringing together enterprise modeling and software specification – an enterprise engineer alone is insufficiently capable of grasping the technical complexity of the enterprise’s IT system and its reach outside through software services, while a software engineer would have only superficial enterprise-specific domain knowledge. However, enterprise engineering and software engineering have developed separately, and hence the alignment between enterprise modeling and software specification is still uncertain.

Bridging that gap is thus considered important with regard to Enterprise Information Systems - EIS. For this reason, our EIS modeling approach has conceptual foundations that represent a common ground for creating models, no matter if they are enterprise-driven or software-driven. Further, that common ground has been extended to accommodate modeling co-creation featuring: (i) technology-independent enterprise models rooted in social theories; (ii) technology-specific software rooted in computing paradigms. On top of that, the approach’s guidelines and related notations further grease the integrated enterprise-software modeling, by facilitating modeling generations and transformations – starting from unstructured business information, coming through enterprise models, and reaching as far as specification and implementation of software. Finally, we provide a case study and illustrative examples for the sake of ‘grounding’ our studies and demonstrating some strengths and limitations of the proposed EIS approach.

This book is not about the approach itself, it is about telling you how to *bring together enterprise modeling and software specification* (maybe also by using another approach and/or other notations) such that an *enterprise-engineering-driven software generation is achieved*.

In carrying out those studies, I have been driven by inspiration brought forward by Colleagues from Delft University of Technology, University of Twente, University of Reading, and IICREST, to whom I am happy and honored to extend my compliments and gratitude. I am privileged as well to be leading a prestigious relevant scientific event, namely the international symposium on Business Modeling and Software Design – BMSD (www.is-bmsd.org). Finally, it is so very special for me that it would have been my Dad's 80th birthday today, he used to be a wonderful father and a brilliant scientist, always inspiring!

I hope that you will find the current book interesting and will consider it a helpful reference with regard to *enterprise information systems*.

Sofia, 6 April 2017
Boris Shishkov

CONTENTS

1	Introduction	1
2	Systems and Context	15
2.1	Systems	16
2.2	Enterprise Systems	18
2.3	Enterprise Information Systems	26
2.4	Ontological Systems and Function	31
2.5	Normalized Systems	36
2.6	Context-Awareness	38
3	Social Theories	55
3.1	Human Relativism and TOA	57
3.2	LAP and Enterprise Ontology	61
3.3	Organizational Semiotics	75
4	Computing Paradigms	81
4.1	Component-Based Development	87
4.2	Service-Oriented Architecture	97
4.3	Model-Driven Engineering	105
4.4	Cloud Computing	110
4.5	Aspect-Oriented Software Development	112
5	The SDBC Approach	115
5.1	SDBC Outline	127
5.2	SDBC Notations	133
6	Case Study and Examples	139
6.1	Background	140
6.2	Icomp	142
6.3	Applying SDBC	143
6.4	Enabling Service-Orientation	182
6.5	Other Examples	187
	Epilogue	195
	Bibliography	197

