



HAL
open science

COTS software integration for simulation optimization coupling: Case of ARENA and CPLEX products

Valeria Borodin, Jean Bourtembourg, Faicel Hnaien, Nacima Labadie

► To cite this version:

Valeria Borodin, Jean Bourtembourg, Faicel Hnaien, Nacima Labadie. COTS software integration for simulation optimization coupling: Case of ARENA and CPLEX products. *International Journal of Modelling and Simulation*, 2018, 10.1080/02286203.2018.1547814 . emse-01687555

HAL Id: emse-01687555

<https://hal-emse.ccsd.cnrs.fr/emse-01687555>

Submitted on 19 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

COTS software integration for simulation optimization coupling: Case of ARENA and CPLEX products

^{1,2,3}Valeria BORODIN, ²Jean BOURTEMBOURG, ³Faïcel HNAÏEN, ³Nacima LABADIE

¹Ecole des Mines de Saint-Etienne, CMP Georges Charpak, CNRS UMR 6158 LIMOS, F-13541, Gardanne, France

²CD-LOSI, UMR CNRS 6281, Université de Technologie de Troyes, 12 rue Marie Curie, CS 42060, 10004 Troyes Cedex, France

³Agricultural Cooperative Society in the Region of Arcis-sur-Aube, Industrial Zone of Vilette 10700, Vilette-sur-Aube, France

February 2017

Working Paper EMSE CMP–SFL 2017/01

This paper focuses on coupling of simulation and optimization techniques, both representing an integrated part of modern design and supervision practices in management sciences, industry and engineering. After discussing the different possible connotations of the concept of simulation optimization coupling, we explicitly describe the integration of two commercial off the shelf (COTS) industry-standard software tools ARENA and CPLEX, respectively. Being written in the continuity of the earlier published work by Vamanan et al. (2004), the contribution of the present article is threefold: (i) clarifying the meaning of the coupling of simulation and optimization, considered as two complementary decision helping approaches, (ii) explaining explicitly the software integration of the latest versions of ARENA and CPLEX, (iii) pointing out the undeniable usefulness of such tools tandem.

CMP - SITE GEORGES CHARPAK

CENTRE MICROELECTRONIQUE DE PROVENCE



COTS software integration for simulation optimization coupling: Case of ARENA and CPLEX products

Valeria Borodin^{1,2,3}, Jean Bourtembourg², Faicel Hnaïen³ and Nacima Labadie³

¹ École des Mines de Saint-Étienne, CMP Georges Charpak, CNRS UMR 6158 LIMOS, F-13541, Gardanne, France
valeria.borodin@emse.fr

² Laboratory of Industrial Systems Optimization (LOSI), Charles Delaunay Institute (ICD),
University of Technology of Troyes (UTT), ICD-LOSI, UMR CNRS 6281,
12 rue Marie Curie - CS 42060, 10004 Troyes, France
{faicel.hnaïen, nacima.labadie}@utt.fr

³ Agricultural Cooperative Society in the Region of Arcis-sur-Aube (SCARA),
Industrial Zone of Vilette 10700, Vilette sur Aube, France
j.bourtembourg@scara.fr

(Last update: January 19, 2018)

Abstract

This paper focuses on coupling of simulation and optimization techniques, both representing an integrated part of modern design and supervision practices in management sciences, industry and engineering. After discussing the different possible connotations of the concept of *simulation optimization* coupling, we explicitly describe the integration of two commercial-off-the-shelf (COTS) industry-standard software tools ARENA and CPLEX, respectively. Being written in the continuity of the earlier published work by [Vamanan et al. \(2004\)](#), the contribution of the present article is threefold: **(i)** elucidating the coupling of simulation and optimization, considered as two complementary decision helping approaches, **(ii)** explaining explicitly the software integration of the latest versions of ARENA and CPLEX, **(iii)** pointing out the undeniable usefulness of such tools tandem.

Keywords: simulation optimization coupling; software integration; ARENA; CLPEX; OptQuest

1 Introduction

Simulation and optimization represent two prominent and powerful tools widely exploited in the framework of a panoply of industrial and engineering applications. On the one hand, simulation refers to the behavioural reproduction of real-world processes or systems over time ([Kelton et al., 2014](#)), while optimization seeks to find the best element (solution) from a given definition domain with regard to some criteria. In the last couple of years, their use in tandem is becoming increasingly spread through different fashions, namely: **(i)** simulation-based optimization, **(ii)** simulation for optimization, and **(iii)** simulation and optimization ([Figueira and Almada-Lobo, 2014](#); [Fu, 2002](#); [Juan et al., 2015](#)).

In compliance with this upward trend, various dedicated and highly specialized optimization and simulation software tools have appeared on the market in the last two decades. Among these, IBM ILOG CPLEX Optimization Studio (often referred to simply as CPLEX) is the most-widely used optimization software package, being currently the market leader in solver technology ([Donoghue, 2015](#)). Meanwhile, ARENA is a general-purpose discrete event simulation (DES) and automation software ([Altiok and Melamed, 2007](#); [Kelton et al., 2014](#)). Both being commercial available off-the-shelf products, their integration and combined use are not officially supported.

Given the large spectrum of its real-world applicability, appropriateness and relevance, the simulation is catalogued the second most widely used technique in the field of Operations Research and Management Sciences, after the *modelling* approach ([Jahangirian et al., 2010](#)). On the flip side, simulation is often reproached because of a lack of optimization functionality. Not literally involving an optimization procedure *per se*, simulation is not able to find/make optimal decisions in an admissible domain of solutions, allowing only the obedience to a predefined tree of possible choices or rules under a set of system observations. As already [Tekin and Sabuncuoglu \(2004\)](#) have reported, the optimization expertise embedding into the simulation systems may not only

overcome this major simulation limitation, but also provide additional valuable opportunities for simulation and optimization with reference to new potential application areas and research possibilities.

The work presented in the current article was motivated by a real-life need occurred within the framework of an agricultural supply chain application. In practice, a simulation-based aiding tool often does not satisfactorily meet all the managers expectations. An optimization expertise turns out to be often required to support the decision making process in the context of a simulation-based system. The central purpose being to illustrate the integration of ARENA and CPLEX, we will not overfill this paper by presenting completely our experience, which has inspired this work. For further details on this case study, refer to our previous publications (Borodin et al., 2013, 2014, 2015).

In the framework of the above described context, the current article intends: (i) to clarify the concept of *simulation optimization* coupling, (ii) to provide guidance on how to integrate the latest versions of CPLEX and ARENA software, which takes its place in the continuity of the previously published work by Vamanan et al. (2004), and (iii) to shed light on the great need and usefulness of coupling the simulation optimization tools.

The remainder of this paper is structured as follows. In the next section, the concept of *simulation optimization* is meticulously addressed in order to avoid any confusion between multiple possible usages distinguished in the related literature. After that, in Section 3, both COTS products ARENA and CPLEX are firstly introduced, and then their software embedding is detailed in Subsection 3.3. In order to emphasize the difference between simulation(-endogenous) optimization(-exogenous) coupling and simulation(-exogenous) optimization(-endogenous) coupling, the optimization tool OptQuest is subject to scrutiny in Subsection 3.4. Finally, this paper ends in Section 4 with some concluding remarks including several suggestions concerning current software requirements.

2 Simulation optimization: A unified macroscopic classification

Still dating from several decades ago, the idea of the combined use of simulation and optimization approaches is not recent. On the one hand, looking for effective (production, distribution, manufacturing, logistic, etc.) system configurations, practitioners frequently use optimization tools in order to better tune their simulated systems. On the other hand, it is in majority the scientific community that employs simulation within the framework of stochastic programming for the purpose of uncertainty handling.

Hence, the same expression *simulation optimization* can be found in the related literature to refer to different possible usages. In this sense, it is worth mentioning the only recently attempt of Figueira and Almada-Lobo (2014) to propose a taxonomy of simulation optimization methods, with a view to creating a standard for a better communication within the scientific community. In order to avoid any ambiguity and misunderstanding and by supporting the proposal of Figueira and Almada-Lobo (2014), let us elucidate the multiple methodological connotations of the concept of *simulation optimization*, that can be discerned in the related literature according to the simulation purpose, namely: (I) solution evaluation, (II) analytical model enhancement, and (III) solution exploitation.

Being quantitative, dynamic and/or stochastic, simulation models are approached experimentally. The time path of *simulation-dependent* variables (outputs) are determined given: (i) the initial state of simulated system, and (ii) the instantiation of *simulation-independent* variables (inputs) (Kleijnen, 2005). On the contrary, the solutions (outputs) corresponding to optimization models are calculated analytically based-on: (ii) mathematically techniques, and (ii) the values of input data.

Therefore, in order to better understand the interaction between simulation optimization, let us push forward the classification of Figueira and Almada-Lobo (2014) by distinguishing the following elementary cases of simulation optimization coupling: (I) simulation(-exogenous) optimization(-endogenous) coupling, (II) simulation(-exogenous) optimization(-exogenous) coupling, and (III) simulation(-endogenous) optimization(-exogenous) coupling.

- (I) **simulation-based optimization:** (also known as *optimization via simulation* or *optimization for simulation*) this type of combined simulation optimization consists in optimizing the simulated systems, whose procedure is illustrated in Figure 1.

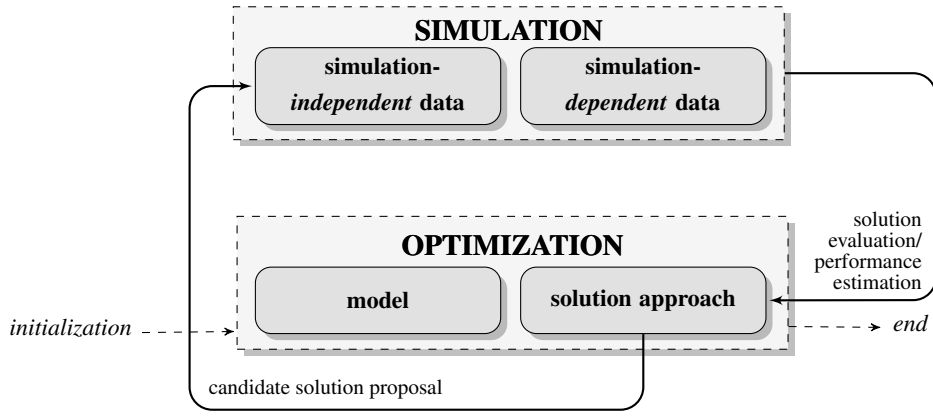


Figure 1: Optimization via simulation: *simulation(-exogenous) optimization(-endogenous) system*

Specifically, the simulation is used to evaluate successively the simulated systems configured in accordance with the solutions proposed by the optimization routine, i.e. it assists optimization algorithms in better solutions searching. In other words, optimization routine exploits the simulation model as a black-box evaluation function (Fu, 2002; Kleijnen and Wan, 2007).

The main optimization techniques of simulated systems are: meta-heuristics (genetic algorithms, simulated annealing, tabu search, scatter search, *etc.*), memory-based meta-heuristics (swarm intelligence methods, artificial neural network based approaches, ant colony optimization, *etc.*), random search, gradient surface methods, metamodel-based methods, sample-path optimization, statistical selection methods, *etc.* (Bianchi et al., 2009; Chan and Schruben, 2008; Figueira and Almada-Lobo, 2014; Melouk et al., 2014). It is interesting to note the newly introduced term of *simheuristic* by Juan et al. (2015) that designates any algorithm, which integrates simulation (in any of its variants) into a meta-heuristic driven framework for solving complex stochastic combinatorial optimization problems. In their review paper, Juan et al. (2015) proposed a comprehensive classification of simulation optimization methods and contextualized simheuristics inside this classification.

Looking to configure more flexibly simulated systems, the scientific community continues to invest in this subject, focusing in particular on the development of dedicated optimization algorithms (Bachelet and Yon, 2007), multi-objective (Person et al., 2008) or non-linear procedures (Angün et al., 2009; Kleijnen et al., 2010). Being able to parameter simulated systems, optimization modules functioning in this manner (cf. Figure 1) are optionally integrated in many simulation software as *add-ons* e.g.: AutoStat (AutoMod), OptQuest (Arena, Crystall Ball, AnyLogic, *etc.*), OPTIMIZ (Simul8), SimRunner (ProModel), Optimizer (WITNESS), *etc.*

- **(II) simulation for optimization:** is an active area in the field of stochastic programming, where the relationship between approaches can be regarded conversely, i.e. as simulation for optimization approach (cf. Figure 2). In this connection, the simulation is generally used: either **(i)** to generate scenarios in accordance with the probability distributions data, or **(ii)** to reproduce different system configurations subject to randomness, thus playing the role of validation/diagnostic framework for solutions proposed by the optimisation module (Alferi and Matta, 2012; Domenica et al., 2007; Figueira and Almada-Lobo, 2014; Shapiro et al., 2009).
- **(III) simulation and optimization:** by this kind of coupling, we understand a combination of two complementary self-sustaining approaches (cf. Figure 3):
 - *optimization:* dealing with optimal solution search in an admissible domain under an optimization criterion and subject to a set of constraints, which not necessarily needs any simulation feedback;
 - *simulation:* able to handle complex system structure or dynamics, uncertainty, non-linearity or any other concerns, which are difficult to be analytically expressed.

As already outlined above, simulation is not able to take optimal decisions within the simulated system. Notwithstanding its practical importance, to the best of our knowledge, only few studies are devoted to

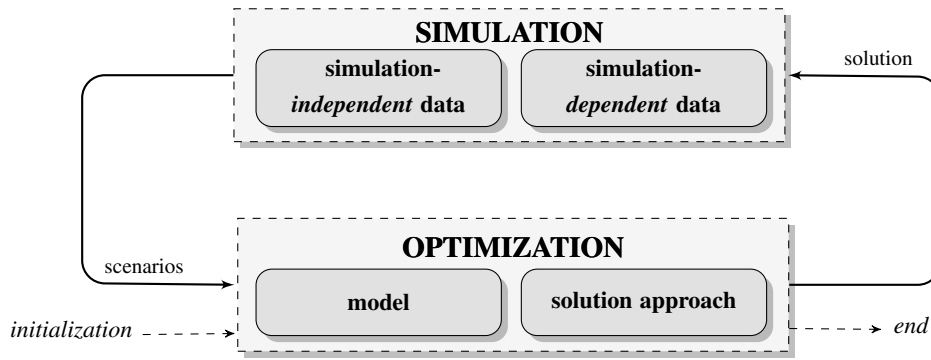


Figure 2: Simulation for optimization: *simulation(-exogenous) optimization(-exogenous) system*

addressing this simulation drawback. A relevant example in this sense represents the work of [Almeder et al. \(2009\)](#). By raising the question "Are simulation and optimization alternative or complementary approaches?", these authors proposed a general framework for the combination of an optimization model and discrete-event simulation in the context of supply chain planning. The interest of the article of [Almeder et al. \(2009\)](#) resides in iterative combination of simulation and optimization allowing to gain the advantages of both optimization (optimal solution finding) and simulation (handling of system nonlinearities, complex structure, stochasticity).

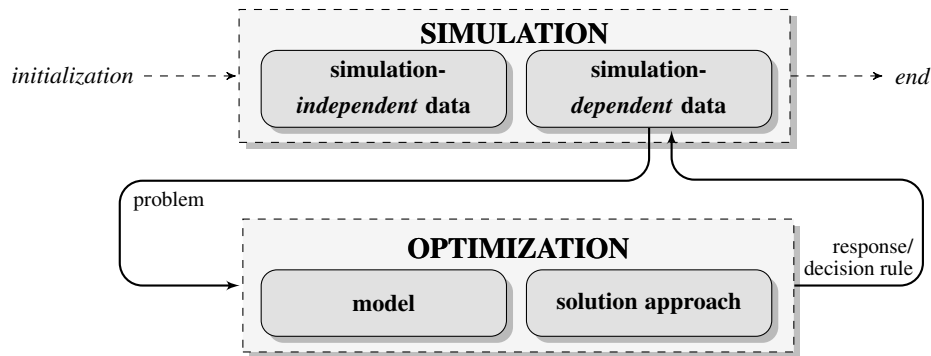


Figure 3: Simulation and optimization: *simulation(-endogenous) optimization(-exogenous) system*

From the point of view of software imbrication, [Almeder et al. \(2009\)](#) have made a simple data exchange between AnyLogic (the simulation tool) and Xpress (the optimization solver) via an external MS Access database. Further efforts in this regard have been conducted by [Vamanan et al. \(2004\)](#), who described the mechanics of the COTS integration of ARENA and CPLEX software products for an inventory/logistic problem. Nowadays, what has been done in the latter article cannot be reproducible because of lack of sufficient non-obsolete details, required to integrate the ARENA and CPLEX software COTS products.

In what follows in this paper, the third kind of simulation(-endogenous) optimization(-exogenous) coupling is discussed. In fact, given the lack of details related to simulation optimization software integration and being confronted with a real-world application requesting a such of coupling, we give explicitly in the next section the instructions, necessary for embedding the latest versions of ARENA and CPLEX products. Taking into consideration the numerous potential applications of such coupling, we argue this will be helpful for both practitioners and researchers.

3 Simulation optimization coupling: Software aspect

In this section, the software integration of two actual industry-standard COTS products, ARENA and CPLEX, is explicitly described after introducing them in the following two subsections.

3.1 ARENA

ARENA¹ is a discrete event simulation and automation industry-oriented general-purpose software, applied in a diversity of sectors (supply chain, manufacturing, healthcare, logistics, military, etc.) for addressing various business and industry challenges (Rockwell Automation, Inc., 2012a). In the context of ARENA, systems are described from the point of view of entities that are crossed by flows using available resources. Its models are structured in a hierarchical and modular way. Primary modelling components, called *modules*, are selected from template panels, such as *Basic Process*, *Advanced Process*, and *Advanced Transfer*, and placed on a canvas in the course of model construction (Rockwell Automation, Inc., 2010, 2012a). Hence, ARENA makes use of a hierarchical architecture for simulation modelling, i.e. modules are defined using other modules. As far as models construction and execution are concerned, a collection of panels is provided.

At the language level, ARENA has a graphical user interface (GUI) around the SIMAN simulation programming language. The SIMAN statements consist of two classes of objects: *Block* (basic logic constructs that represent operations) and *Elements* (objects that represent facilities). Thus, a module is a high-level construct, composed of SIMAN blocks and/or elements. By the way, since ARENA modules are just sub-programs written in SIMAN, simulation models can be built using SIMAN constructs from the *Blocks* and *Elements* template panels alone (Altiok and Melamed, 2007). After a program definition, SIMAN code is not directly executed on the computer. Before running the SIMAN code, it is checked for syntax and other errors, and then it is translated to an executable form in C/C++ parser as depicted in Figure 4 (Fishman, 2001; Panneerselvam and Senthilkumar, 2013; Rossetti, 2009).

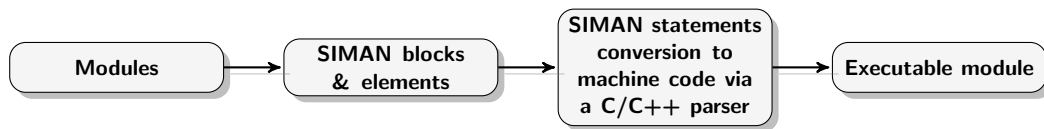


Figure 4: Programs construction and execution with ARENA (Fishman, 2001; Rossetti, 2009)

In addition, the ARENA environment interoperates with two Microsoft technologies designed to improve desktop applications, namely: (i) Active X Automation, which provides control facilities within and across applications *via* a framework platform accessible through a programming language, and (ii) the programming language Visual Basic for Applications (VBA).

By using VBA, it is possible to write procedural code to enhance visual programming facilities, or to automatize simulation models if specific algorithms are required. Being an implementation of Microsoft event-driven macro programming language, VBA allows also the users to take advantages of the integrative capabilities supported by Microsoft technologies, and easily interacts with such well-known applications as Excel, Access, Visio, etc., which are afforded by Active X Automation (Altiok and Melamed, 2007).

ARENA makes available two paid software versions: *commercial* (Professional and Standard editions) and *academic* (Academic Lab Package, Research Edition and Student Edition). As far as the academic version is concerned, only the Student Edition is free, being perpetual but limited in functionalities and model sizes.

3.2 CPLEX

IBM® ILOG® CPLEX® Optimization Studio² is an analytical decision support toolkit dedicated to optimization models development and deployment using mathematical and constraint programming. As illustrated in Figure 5, it incorporates an integrated development environment (IDE) with the powerful Optimization Programming Language (OPL) and high-performance ILOG CPLEX optimizer solvers based on Concert Technology. By employing the CPLEX callable library, Concert Technology represents a set of C++, Java, and .NET class libraries providing an application programming interface (API) that includes modelling facilities allowing the programmer to embed CPLEX optimizers in C++, Java, or .NET applications (cf. Figure 6).

¹<http://www.arenasimulation.com/>

²<http://www.ibm.com/developerworks/downloads/ws/ilogcplex/>

The architecture of the CPLEX callable library (C API) supports user-written applications in C and other programming languages callable from C. The CPLEX core is configured by CPLEX callable library in conjunction with the CPLEX internals.

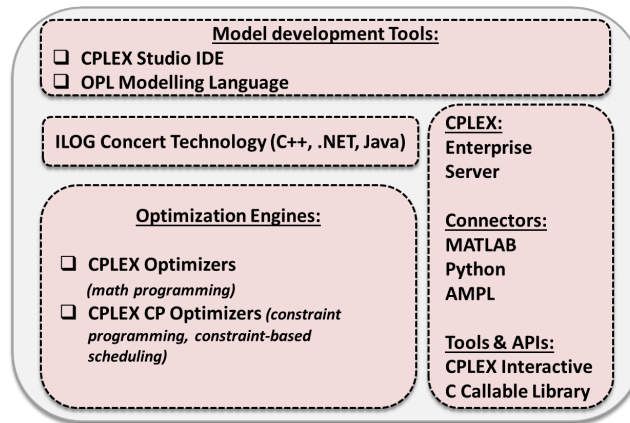


Figure 5: CPLEX Optimization Studio components (IBM Inc., 2014)

The modelling and optimization parts of a user-written application program are represented by a group of interacting C++ objects created and controlled within the application. For the sake of clarity, Figure 6 offers a picture of an application using CPLEX together with Concert Technology to solve optimization problems.

Despite the fact that its commercial use requires paid licenses, IBM also provides academic licenses for non-commercial research at no-charge. Note that, full versions of the latest releases of the IBM ILOG CPLEX Optimization Studio are available at no-charge, as well as professionally-developed course-ware, to registered members of IBM Academic Initiative for teaching and non-commercial usage.

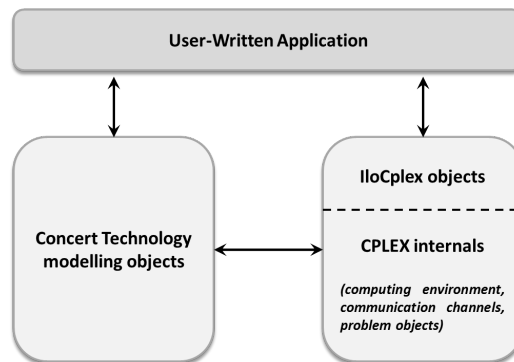


Figure 6: CPLEX design with Concert Technology (IBM Inc., 2014)

3.3 ARENA and CPLEX integration: Simulation(-endogenous) optimization(-exogenous) coupling

Incorporating COTS software into other COTS-based systems proves to be a challenging task. Although, it may procure many advantages, the lack of access to source code requires a different treatment of COTS software reuse than traditional one. COTS software cannot be directly adapted inside by modifying its source code. Hence, software adjustments must be made from the outside, e.g. via wrappers or glue code, which can be very precarious for the integrated system reliability (Egyed and Balzer, 2006). However, notwithstanding this risk, let us describe the integration of two COTS products CPLEX and ARENA in what follows in this subsection. Note that, COTS integration and computational experiments have been carried out on an Intel(R) Core(TM) i7-2720QM CPU 2.20GHz workstation running under Microsoft Windows 7 Professional operating system.

To begin with, let us present the latest available versions of the two software products in question:

- ❑ ARENA: As of June 2014, ARENA is in version 14.7. Note that, it is a 32-bit Windows desktop application that also runs on 64-bit operating systems.
- ❑ CPLEX: Since January 2014, its stable release is 12.6.

Besides a trivial external user-written application, a more proper integration of ARENA and CPLEX can be realized through VBA blocks³, which support the users interactivity with the simulation elements. Indeed, they allow to insert general-purpose procedural customer code into the simulation model, directly *via* the Visual Basic Editor.

The connection between ARENA and VBA is based on the event-oriented nature of VBA. A *simulation event* is a procedure that is called (executed) when a particular condition (event) occurs. There are a number of VBA events that are predefined within the model interaction of ARENA with VBA. We will not reproduce them in this paper, but the readers are encouraged to consult the books of Rossetti (2009) and Altiook and Melamed (2007) for further information on this matter.

However, let us remainder the three categories of ARENA built-in VBA events, the time sequencing of which are provided in Algorithm 1:

1. ARENA-generated events that occur before a simulation run;
2. ARENA-generated events that occur during a simulation run;
3. user-generated events that occur during a simulation run.

Algorithm 1 : Time sequencing of ARENA VBA events

- | | | |
|----|---|---|
| 1: | RunBegin → simulation run is going to start | |
| | ▷ module data is available | |
| | ▷ SIMAN data is not available | |
| 2: | RunBeginSimulation | } |
| 3: | →RunBeginReplication | |
| | ▷ ARENA runs replication | |
| 4: | ~RunEndReplication | |
| | ▷ Simulation run data is available | |
| | ▷ SIMAN data is available | |
| 5: | RunEndSimulation | |
| | ▷ module data is available | |
| | ▷ SIMAN data is not available | |
| 6: | RunEnd → simulation run is completed | |
-

To integrate optimization capabilities into a simulation model during its progress, the events belonging to the third category must be exploited. While an ARENA model is running, it allows the user to activate VBA user code by calling one of the ARENA VBA events:

- ❑ ModelLogic_VBA_Block_Fire is called when an entity reaches a VBA block in the simulation model.
- ❑ ModelLogic_OnKeyStroke: is called whenever the user strikes a key during a simulation run (except the *Esc* key).
- ❑ ModelLogic_OnClearStatistics: is called whenever statistics are reset.

Hence, for a VBA block with an identifier number N assigned by ARENA for its identification, the user can provide VBA code for the corresponding `VBA_Block_N_Fire` event. When an entity enters into a VBA block, ARENA fires the corresponding event to execute the user-provided VBA code. Once the execution is completed, the entity exits the VBA block and proceeds to traverse the simulation model in the usual manner. The implementation code in VBA is given in Appendix A.

³Note that, the VBA blocks are only available in the Professional Edition of ARENA.

In order to embed optimization features into a simulation process, let us take advantage of the both opportunities: **(i)** on the one hand, ARENA allows VBA user-code integration *via* the VBA blocks, and **(i)** on the other hand, CPLEX callable library is a C application programming interface of CPLEX that allows to embed CPLEX optimizers in applications written in languages that can call C functions. In particular, CPLEX⁴ provides functions that permit to run CPLEX from Visual Basic.

The CPLEX callable library is provided as a Dynamic Link Library (DLL). As any other DLL, for calling CPLEX from Visual Basic, it is necessary to:

- ❑ **Access the CPLEX DLL:** for each CPLEX callable library routine, a similar function declaration in Visual Basic must be created, for instance:

```
Declare Function CPXopenCPLEX Lib "cplex1260.dll" Alias "_CPXopenCPLEX@4" (ByRef status As Integer) As Long
```

The above code fragment recognizes CPXopenCPLEX as an exported function of the DLL calling by Visual Basic. In this way, posterior calls from Visual Basic can be made by using the name CPXopenCPLEX. The clause Alias serves to identify the correct format for the declared function in the Visual Basic domain. Note that, in cases where a DLL function (e.g. CPXopenCPLEX) has a name that is not a legal identifier, it is necessary: **(i)** firstly, to declare the function with a legal name, **(ii)** secondly, to use the Alias clause to reference the procedure real name.

- ❑ **Handle C structures inside Visual Basic:** note that pointers to two types of C structures, specific to CPLEX, are indispensable to handle, namely: *environment* and *lp* pointers. To do this, the type Long in Visual Basic can be used to represent 32 bit pointers, since both correspond to 4 byte integers. Consider another example:

```
Declare Function CPXmipopt Lib "cplex1260.dll" Alias "_CPXmipopt@8" (ByVal env As Long, ByVal lp As Long) As Integer
```

To recuperate environment pointer (env) and lp pointer (lp), it is sufficient to call CPLEX routines which have them as arguments (e.g. CPXmipopt) and pass these pointers in ByVal using a 4 byte Long.

- ❑ **Handle C pointers inside Visual Basic:** as far as pointers of standard C types (e.g. int, double, char, etc.) are concerned, in Visual Basic it is possible to assign them values and pass them by reference to the functions in DLL, as done in the following code fragment for the third function argument.

```
Declare Function CPXgetobjval Lib "cplex1260.dll" Alias "_CPXgetobjval@12" (ByVal env As Long, ByVal lp As Long, ByRef obj As Double) As Integer
```

Moreover, it is worth noticing that a NULL pointers is passed for an argument in one of the CPLEX routines by ByVal 0&. General requirements about the interface between Visual Basic, C/C++ and Windows API can be found for example in the book of Bai (2003).

In a nutshell, the integration of ARENA and CPLEX software products can be realized *via* VBA Blocks, through which CPLEX optimizers are callable if properly accessed and handled. Given the important number of practitioners and scientists who use ARENA and CPLEX in their applications, we argue that the findings exposed in the foregoing paragraphs would be instructive and helpful. The optimization layer thus added complements the simulation major drawback to not be able to make optimal decision during its running.

3.4 ARENA and OptQuest: Simulation(-exogenous) optimization(-endogenous) coupling

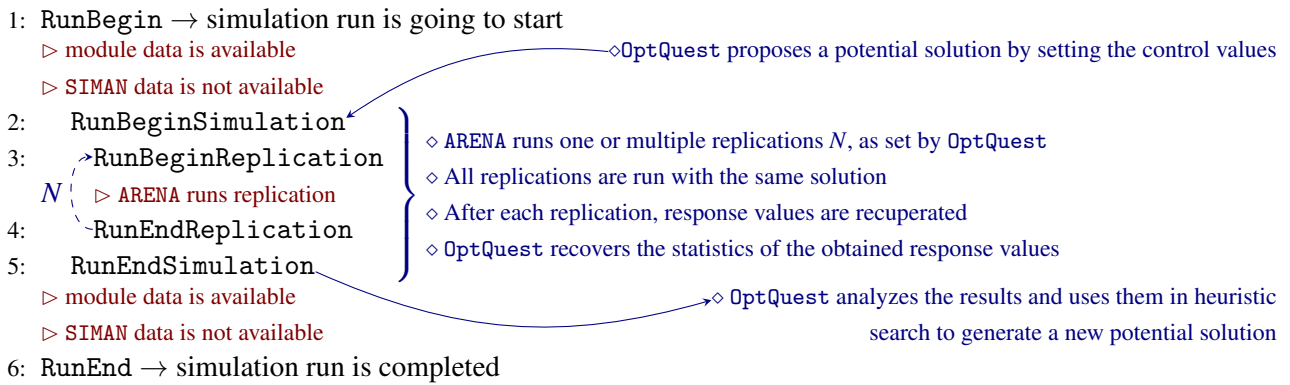
OptQuest is an optimization *add-on* for a number of simulation software products (in particular, for ARENA) used to optimize the control variables (simulation-independent variables) of simulated systems. It is worth mentioning that such optimization tools as OptQuest⁵ are not applicable to solve problems occurring inside a simulation, i.e. involving simulation-endogenous information.

⁴CPLEX 9.0 and later versions include a .NET API.

⁵<http://www.opttek.com/>

Presupposing the optimization of simulated systems, it is appropriate only to configure a simulation system looked at holistically. In order to argue this statement, let us analyse its manner of functioning and interaction with ARENA displayed by Algorithm 2. Hence, OptQuest behaves towards ARENA as a black-box, observing only the inputs and outputs of the simulation model. A solution proposed by OptQuest refers to the *control* values (inputs for ARENA), and the resulting values to the *responses* (ARENA outputs) measures the simulation model performance. By running multiple simulations, OptQuest searches to optimize a user-defined (deterministic) model⁶ via a combined heuristic approach (meta-heuristics, mathematical optimization, and neural network) (Rockwell Automation, Inc., 2012b). For commercial reasons, this search approach remains unknown. That being so, OptQuest optimization engine does not interact with Arena simulation, as a complementary approach. It is appropriate for simulation systems tuning, and not for its piloting.

Algorithm 2 : OptQuest for ARENA



To sum up, one of the concluding recommendations is that the OptQuest engine usage is justified when it is applied to optimize complex systems at the strategical even tactical levels. On the contrary, if operational problems are to be addressed during the simulation piloting, it is more reasonable to embed an optimization solver inside simulation. Fortunately, such an integration can be performed with CPLEX and ARENA software products. Obviously, this can burden the resolution and affects the computational time complexity, but it is already another price-efficacy question.

4 Conclusion

This paper discusses the simulation optimization coupling, considered as two complementary self-sustaining approaches. A such of tandem exploitation allows to overcome the limitations of both these approaches. On the one hand, simulation is able to handle the system features which cannot be analytically modelled, and by that, to circumvent this optimization lack. On the other hand, optimization intervenes with its optimal solution search expertise, thereby improving the simulation potential in responding to the expectations and needs of real-world applications. We argue that the procured advantages of this kind of simulation optimization coupling is very attractive and useful for a large variety of applications fields.

In addition, this paper describes explicitly the integration of the latest versions of two commercial available off the shell products, namely CPLEX and ARENA (both being nowadays industry-standard software tools). We hope the software embedding mechanism will be instructive for the industry and academia communities, which are confronting with related topics.

⁶OptQuest supports optimization models with constraints. A linear constraints define linear relationships among controls, while non-linear constraints can be only formulated using simulation responses (i.e. ouptput variables, whose values are determined by the black box evaluator) (Laguna, M., 2011). On the other words, uncertainties or other non linearities cannot be explicitly defined into a optimization model, their handling is possible only via a simulation evaluator.

Acknowledgement

The authors are grateful to Dr. Claude Yugma for its helpful suggestions and valuable discussions about the simulation optimization coupling.

A Using VBA programming in ARENA for the CPLEX integration

```
Option Explicit

' Acces the CLEX DLL

Private Declare Function CPXopenCPLEX Lib "cplex1260.dll" Alias "_CPXopenCPLEX@4" (ByRef
    status As Integer) As Long
Private Declare Function CPXcreateprob Lib "cplex1260.dll" Alias "_CPXcreateprob@12" (ByVal
    env As Long, ByRef status As Integer, ByVal name As String) As Long
Private Declare Function CPXreadcopyprob Lib "cplex1260.dll" Alias "_CPXreadcopyprob@16" (
    ByVal env As Long, ByVal lp As Long, ByVal file As String, ByVal filetype As String) As
    Integer
Private Declare Function CPXmipopt Lib "cplex1260.dll" Alias "_CPXmipopt@8" (ByVal env As
    Long, ByVal lp As Long) As Integer
Private Declare Function CPXgetobjval Lib "cplex1260.dll" Alias "_CPXgetobjval@12" (ByVal env
    As Long, ByVal lp As Long, ByRef obj As Double) As Integer
Private Declare Function CPXgetx Lib "cplex1260.dll" Alias "_CPXgetx@4" (ByVal env As Long,
    ByVal lp As Long, ByRef obj As Double) As Integer
Private Declare Function CPXcloseCPLEX Lib "cplex1260.dll" Alias "_CPXcloseCPLEX@4" (ByRef
    env As Long) As Integer
Private Declare Function CPXtxtsolwrite Lib "cplex1260.dll" Alias "_CPXtxtsolwrite@12" (ByVal
    env As Long, ByVal lp As Long, ByVal name As String) As Long
Private Declare Function CPXsolwrite Lib "cplex1260.dll" Alias "_CPXsolwrite@12" (ByVal env
    As Long, ByVal lp As Long, ByVal name As String) As Long
Private Declare Function CPXwriteprob Lib "cplex1260.dll" Alias "_CPXwriteprob@16" (ByVal env
    As Long, ByVal lp As Long, ByVal name As String, ByVal name As String) As Long
' ...

Private Sub VBA_Block_1_Fire()

Dim env As Long
Dim lp As Long
Dim status As Integer

' Initialize the CPLEX environment

env = CPXopenCPLEX(status)
If (env = 0) Then
MsgBox "Error " & status & " in opening DLL", vbOKOnly, "Error!"
GoTo TERMINATE
End If

' Create a CPLEX problem object in the CPLEX environment

lp = CPXcreateprob(env, status, "<name_problem>")
If (status <> 0) Then
MsgBox "CPXcreateprob failed: " & CStr(status)
GoTo TERMINATE
End If

' Define a CPLEX problem object or instantiate a CPLEX file in LP format
' <problem definition> ...

' Read an LP file into the CPLEX problem object

status = CPXreadcopyprob(env, lp, "<file_path>", "lp")
If (status <> 0) Then
MsgBox "CPXreadcopyprob failed: " & CStr(status)
GoTo TERMINATE
End If

' Find a solution for the CPLEX problem object

status = CPXmipopt(env, lp)
If (status <> 0) Then
MsgBox "CPXlpopt failed: " & CStr(status)
GoTo TERMINATE
```

```

End If

' Retrieve solution information

Dim obj As Double
status = CPXgetobjval(env, lp, obj)
If (status <> 0) Then
MsgBox "CPXgetobjval failed: " & CStr(status)
GoTo TERMINATE
End If

' MsgBox "Objective value: " & CStr(obj)

status = CPXsolwrite(env, lp, "<file_path>")

If (status <> 0) Then
MsgBox "CPXsolwrite failed: " & CStr(status)
GoTo TERMINATE
End If

' Free all of the data structures associated with CPLEX

TERMINATE:

If (env <> 0) Then
status = CPXcloseCPLEX(env)
End If

End Sub

```

References

- A. Alfieri and A. Matta. Mathematical programming formulations for approximate simulation of multistage production systems. *European Journal of Operational Research*, 219(3):773 – 783, 2012. ISSN 0377-2217.
- C. Almeder, M. Preusser, and R. Hartl. Simulation and optimization of supply chains: alternative or complementary approaches? *OR Spectrum*, 31(1):95–119, 2009.
- T. Altiok and B. Melamed. *Simulation Modeling and Analysis with Arena*. Academic Press, 2007.
- E. Angün, J. P. C. Kleijnen, D. den Hertog, and G. Gürkan. Response surface methodology with stochastic constraints for expensive simulation. *JORS*, 60(6):735–746, 2009.
- B. Bachelet and L. Yon. Model enhancement: Improving theoretical optimization with simulation. *Simulation Modelling Practice and Theory*, 15(6):703 – 715, 2007. ISSN 1569-190X.
- Y. Bai. *Applications Interface Programming Using Multiple Languages: A Windows Programmer's Guide*. Prentice Hall Press, Upper Saddle River, NJ, USA, first edition, 2003. ISBN 0-13-100313-5.
- L. Bianchi, M. Dorigo, L. Gambardella, and W. Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8(2):239–287, 2009. ISSN 1567-7818.
- V. Borodin, F. Hnaien, N. Labadie, and J. Bourtembourg. A discrete event simulation model for harvest operations under stochastic conditions. In *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference*, pages 708–713, 2013.
- V. Borodin, J. Bourtembourg, F. Hnaien, and N. Labadie. A decision support system for efficient crop production supply chain management. *Lecture Notes in Computer Science*, 8583 LNCS(PART 5):775–790, 2014.
- V. Borodin, J. Bourtembourg, F. Hnaien, and N. Labadie. A multi-step rolled forward chance-constrained model and a proactive dynamic approach for the wheat crop quality control problem. *European Journal of Operational Research*, 246(2):631 – 640, 2015. ISSN 0377-2217.
- W. Chan and L. Schruben. Optimization models of discrete-event system dynamics. *Operations Research*, 56(5):1218–1237, 2008.
- N. D. Domenica, G. Mitra, P. Valente, and G. Birbilis. Stochastic programming and scenario generation within a simulation framework: An information systems perspective. *Decision Support Systems*, 42(4):2197 – 2218, 2007.
- J. R. Donoghue. Comparison of integer programming (IP) solvers for Automated Test Assembly (ATA). *ETS Research Report Series*, 2015. ISSN 2330-8516.
- A. Egyed and R. Balzer. Integrating cots software into systems through instrumentation and reasoning. *Automated Software Engineering*, 13(1):41–64, 2006. ISSN 0928-8910.

- G. Figueira and B. Almada-Lobo. Hybrid simulation-optimization methods: A taxonomy and discussion. *Simulation Modelling Practice and Theory*, 46(0):118 – 134, 2014. ISSN 1569-190X.
- G. Fishman. *Discrete-Event Simulation: Modeling, Programming, and Analysis*. Springer-Verlag New York, 2001.
- M. C. Fu. Feature article: Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3):192–215, 2002.
- IBM Inc. *IBM ILOG CPLEX Optimization Studio Getting Started with CPLEX (Version 12 Release 6)*. IBM Corporation, 2014.
- M. Jahangirian, T. Eldabi, A. Naseer, L. K. Stergioulas, and T. Young. Simulation in manufacturing and business: A review. *European Journal of Operational Research*, 203(1):1 – 13, 2010. ISSN 0377-2217.
- A. A. Juan, J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, 2(0):62 – 72, 2015. ISSN 2214-7160.
- D. Kelton, R. Sadowski, and N. Zupick. *Simulation with Arena*. McGraw-Hill Professional, 2014.
- J. P. Kleijnen. Supply chain simulation tools and techniques: a survey. *International Journal of Simulation and Process Modelling*, 1(1-2):82–89, 2005.
- J. P. Kleijnen and J. Wan. Optimization of simulated systems: Optquest and alternatives. *Simulation Modelling Practice and Theory*, 15(3):354 – 362, 2007.
- J. P. Kleijnen, W. van Beers, and I. van Nieuwenhuysse. Constrained optimization in expensive simulation: Novel approach. *European Journal of Operational Research*, 202(1):164 – 174, 2010. ISSN 0377-2217.
- Laguna, M. *OptQuest: Optimization of Complex Systems*. OPTTEK SYSTEMS, INC., 2011.
- S. H. Melouk, B. A. Fontem, E. Waymire, and S. Hall. Stochastic resource allocation using a predictor-based heuristic for optimization via simulation. *Computers & Operations Research*, 46(0):38 – 48, 2014. ISSN 0305-0548.
- R. Panneerselvam and P. Senthilkumar. *System Simulation, Modelling and Languages*. PHI Learning Private Limited, Delhi, 2013.
- A. Person, H. Grimm, and A. Ng. A metamodel-assisted steady-state evolution strategy for simulation-based optimization. In *Trends in Intelligent Systems and Computer Engineering*, volume 6 of *Lecture Notes in Electrical Engineering*, pages 1–13. 2008. ISBN 978-0-387-74934-1.
- Rockwell Automation, Inc. *User's Guide*. Allen Bradley - Rockwell Software, 2010.
- Rockwell Automation, Inc. *Template developer's guide*. Allen Bradley - Rockwell Software, 2012a.
- Rockwell Automation, Inc. *OptQuest for Arena: User's Guide*. Allen Bradley - Rockwell Software, 2012b.
- M. D. Rossetti. *Simulation Modeling and Arena*. Wiley, 2009.
- A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming: Modelling and Theory*. Philadelphia: Society for Industrial and Applied Mathematics, 2009.
- E. Tekin and I. Sabuncuoglu. Simulation optimization: A comprehensive review on theory and applications. *IIE Transactions*, 36(11): 1067–1081, 2004.
- M. Vamanan, Q. Wang, R. Batta, and R. J. Szczerba. Integration of COTS software products ARENA & CPLEX for an inventory/logistics problem. *Computers & Operations Research*, 31(4):533 – 547, 2004. ISSN 0305-0548.