KNOWLEDGE BASED EXTENDABLE SIMULATION OF PIG IRON ALLOCATION

Diego Díaz
Francisco J. Lago
M. Teresa Rodríguez
Arcelor Research Knowledge Innovation
Apdo 90, E–33400 Aviles (Asturias, Spain)

Sergio Rodríguez
I-R&D Arcelor Mittal
Apdo 90, E–33400 Aviles (Asturias, Spain)

Abstract—

In a site with one set of blast furnaces that must feed two steel mills separated by over 13 kilometers, allocation of hot pig iron is a complex subject owing to the long cycle times and the large number of parameters involved in the decision making. Simulation of the transportation appears as an ideal solution to tackle this complexity. However, allocation priorities are stategic issues that change in time, as does, if less often, the available railway layout.

A simulation based decision support system has been developed to integrate in the current torpedo tracking application that works around these difficulties by generating the simulations dynamically from three sources: a database model of the transportation network, the dynamic database that holds the current state of the torpedo wagons (through the tracking application), and a shared, updatable knowledge base made up of several sets of rules and priorities, and conditional paths of ruleset selection. The knowledge base is now being built on expert knowledge and current strategies, and will provide a framework for user-accessible update.

KEYWORDS

Knowledge-based simulation, decision support systems, applied simulation.

I. INTRODUCTION

In this paper the building of a simulator of material flow in a complex, constrained, discrete-event productor – consumer environment is described. The aim of such a simulator is to provide support for the system coordinator in decision making and to gather current knowledge on the matter for the future. Part of the complexity of the system arises from the fact that both the environment and the priority criteria change in time, so that the solution must provide the capability to update them easily.

The simulated system consists of one or more sources (productors) which produce material that is needed in several sinks (consumers) with different requirements; between them lie a transportation system and a set of facilities that perform some transformations on the material —related to consumer requirements—, with their own constraints on timing, throughput, etc.

In particular, this paper describes the solution applied to this problem in the shape of allocation of pig iron from one set of blast furnaces to two steel mills, with additional limitations owing to its industrial nature.

Rule-based systems, also known as production systems, were found to be the most suitable knowledge representation scheme under the stringent constraints of the project and given the way in which the coordinators are able to express their knowledge of the domain (Konar 1999; Mitchell 1997).

Simulation has long been used in decision support systems (Power 2002), either as an underlying model or as an interactive tool to try different scenarios. Augmentation of the capabilities of simulation include extension by integration of different aspects (Fishwick 1994), including knowledge systems (Anjewierden et al. 2002; Williams et al. 1996) and the use of simulation for planning (Lee and Fischwick 1994). However, this has been done mostly in academic environments; some examples exist nevertheless of a rule-based simulation approach in applied projects, such as (Cheng 1998), where ILOG Rules is used for the inclusion of an embedded rule engine into an object-oriented simulation of apron traffic planning.

The solution finally undertaken combines these different approaches and additionally keeps a crisp separation of the different aspects of the system. The central part is a simulator that reproduces the material flow in the system, so that the solution provided is not just a planning of the flow, but also a description of the state of the system if this planning is followed. The simulator builds its system model from two outside sources: a database that describes the physical layout and a rulebase that provides the logic necessary for decision making. These two sources are independent, and can be taken care of by different agents.

In the next section, **Description of the problem**, an in-depth review of the problem is given, defining what is expected of the solution. Next, in **Additional specifications**, a further set of constraints is defined that applies to the solution rather than to the problem; these are related to the final in-use features and user requirements. Then in **Design**, the design that was adopted for the solution is put forth, together with a review of the constraints and how the chosen design manages them. In **Current state** the degree of completion at the time of writing is stated. **Conclusion** summarises the results, and **Future work** indicates the next steps to be taken.

Proceedings 21st European Conference on Modelling and Simulation
Ivan Zelinka, Zuzana Oplatková, Alessandra Orsoni ©ECMS 2007
ISBN 978-0-9553018-2-7 / ISBN 978-0-9553018-3-4 (CD)

II. DESCRIPTION OF THE PROBLEM

Blast furnaces reduce iron ores and output hot pig iron, an iron-carbon alloy with much higher carbon content than steel. Steel mills then oxidise this pig iron in the BOF (a.k.a. LD furnace) in order to obtain the right concentration of carbon. Depending on the characteristics of the pig iron, an intermediate desulphuration process might be neccessary.

Transport from the blast furnace to the steel mill is usually by railway, using special wagons called *torpedo wagons*. These torpedo wagons consist of a cast iron shell lined with refractory material —the torpedo— set on top of a cargo wagon. Several torpedo wagons cycle from the blast furnace to the steel mill (carrying the pig iron) and back (empty). Depending on the evolution of the processes in the blast furnace and in the steel mill, the number of torpedo wagons may vary: there has to be enough to take away the production of pig iron from the blast furnace, but too many would build long queues in the steel mill to be unloaded, so that the pig iron would be colder in the BOF, which means that more energy has to be consumed to process the steel. Plus, if the temperature loss is greater, the pig iron might be rendered unusable; in this case, it has to be dumped, which is even more expensive (if the pig iron solidifies inside the torpedo, it would have to be scrapped).

The system has high inertia: on the one hand, the production rate of the blast furnace can change only very slowly, and the torpedoes must be kept heated to avoid thermal shock and to prevent temperature loss when tapping the pig iron into them. Thus, torpedoes that are not to be used for a relatively short time must have their temperature maintained at a heating facility, and the torpedoes that enter the cycle must be heated to working temperature before being loaded. All of this makes complex long term planning necessary for managing the whole cycle.

The typical layout consists of one or several blast furnaces connected by a relatively short stretch of railway to a steel mill, and a desulphuration unit on the way, as shown on the left side picture in figure 1. Additional facilities are needed: a dumping area, cleaning facilities, torpedo heating facilities, garage, etc.

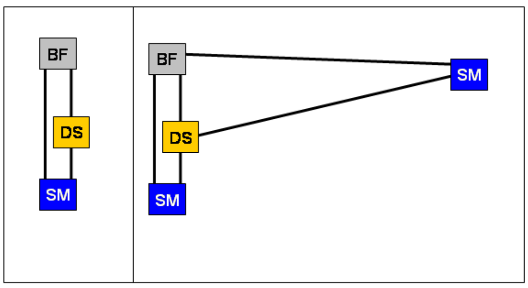


Fig. 1. Layouts of blast furnace and steel mill. Left: typical layout with a single steel mill near the blast furnace. Right: Special case with two steel mills far apart fed by the same set of blast furnaces

Arcelor Mittal's site in Asturias is a special case which does not comply with this typical layout due to historical reasons. Instead, two steel mills are provided with pig iron by a single set of blast furnaces. One of the steel mills lies near the blast furnaces, but the other one is over 13 kilometers away from them. The right side picture in figure 1 shows the basic structure, but, as above, additional facilities are needed, and most of them are duplicated around each steel mill.

Alongside the layout, the steel mills have different priorities depending on production schemes, customers, etc. All of this makes the problem of managing pig iron allocation at this site specially complex. The coordination has been traditionally carried out by two people (per shift), assisted by a text console torpedo tracking application that, together with information gathered via telephone from the facilities involved in the process, helped them know the state of the system. Recently, a project was started to renew the torpedo tracking application. The new application has a graphical user interface and provides improved communications to centralize all the information. Thus, the application's database is an accurate representation of the current state of the system and is also used for reporting. With the new tools, a single person per shift can do the coordination.

Integrated with the new application, a knowledge based simulator has been developed that can work in two modes:

Pure simulator The system evolves in time and the user is asked to make the decisions where more than one path is available, f.i. after desulphuration the user has to decide to which steel mill the torpedo will go. This allows to try different what-if scenarios.

Decision support system The system makes all the decisions by itself relying on expert knowledge gathered from the coordinators. This will provide a kind of *playground* for testing new strategies and a learning and helping tool for new coordinators.

III. ADDITIONAL SPECIFICATIONS

The solution to this problem is heavily constrained in every dimension:

Technical constraints Since the simulator should be integrated in an existing framework, the technical means employed must be supported by the framework, a .NET web application built on top of a relational database. Thus, the following aspects were imposed:

- Programming language: C#.
- Persistence: relational database.
- Communications – framework integration: relational database and remoting.

End user constraints The simulator must *explain* the decisions it makes for the coordinators to take it into account. Black-box algorithms and mathematical optimization techniques were right out of consideration. The simulator must also behave in the way the coordinators are used to working: marking the passing of each torpedo wagon by a series of checkpoints along the circuit; these checkpoints correspond mainly to entering or exiting a facility, the beginning or end of a process such as loading or unloading, and significant railway forks.

Knowledge constraints The simulator must encompass the knowledge of the current coordinators, who boast many years of experience; this experience is a precious asset of the company, and it must be capitalized. Therefore, the resulting simulator should somehow integrate their knowledge.

Dynamicity constraints The system is dynamic both in layout and strategy. The facilities undergo revampings from time to time and their performance evolves, and occasionally one of them can be closed or a new one built, and new railway lines may become available; the simulator must be able to tackle all these events without needing to recode the whole model. The strategy with regard to the priorities for allocation is also dependent on higher level strategy, and must be updatable by non-programmer users who will later take care of the rulebase.

Reporting constraints In order to be able to assess the quality of the decisions made by the simulator, it is necessary to define some kind of indicators.

IV. Design

A. Overview

Figure 2 shows the general architecture of the system, and the communication links among them. The different elements are explained hereafter.

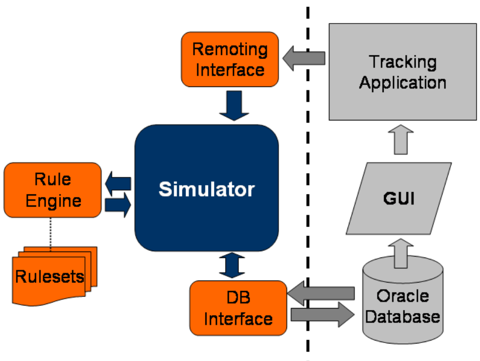


Fig. 2. General Overview of System Architecture

B. Simulator

The simulator has been designed to be modular and event-driven, and the workload has been distributed among the different elements.

The torpedo wagon consists basically of a set of state variables, plus a few methods to manipulate them: loading and unloading of pig iron, including mixing pig iron from different heats, for instance.

The facilities receive the torpedo wagons and perform one or more operations on them, taking some time to do so. In order to homogenise the behaviour, these operations are divided into atomic processes that perform a single operation each. Each process has an associated time. A facility may perform one or more consecutive processes. When the torpedo wagon has been processed, it is returned to the simulator, with the corresponding changes to its state variables.

For instance, in a steelshop there are four processes: entrance, queuing, unloading and exit; whereas in a railway stretch there is only one process which only modifies the torpedo's state by increasing the time (which means that the pig iron temperature decreases).

The simulator is initialized with the available set of facilities and torpedo wagons at the initial time, and loops the following cycle until a given simulator time limit is reached:

1. The simulator polls the facilities for events. Each facility will answer with the time when the next torpedo wagon will finish its process.
2. The simulator then sets the simulation time to the earliest of the events, and all the facilities update their state accordingly and return any torpedo wagons that must exit at this time.
3. The simulator then makes the decision as to where the torpedo wagon should go next and sends it to that facility.

The way in which the decision is accomplished varies with the mode.

In decision support system mode, all happens automatically and the decisions are made by the embedded rule engine. In the end, a time line of future decisions and the evolutions of the system has been defined for the coordinator to consult. A change in the real-world state will trigger the simulation to restart with the new conditions.

In pure simulator mode, each loop is triggered by the user pushing a *Next step* button. Decisions are made by prompting the user to choose from the possible destinations. Furthermore, the user can step back and forth along the time line and revoke one of the decisions, choosing an alternative one to compare the results.

C. Database

The DSS uses the database in three ways:

1. For model building: The parameters of the simulator model are stored in a section of the database. These include: each of the torpedo wagons, each of the facilities, the processes contained in each facility, the times and operations associated to each process. This allows to modify the simulation model by updating the database; for instance, this will have to be done in the near future because the railway layout is going to be changed.
2. For initialization: the state of the facilities and torpedo wagons at the initial time is retrieved from the database. This information is constantly updated by the tracking application.
3. As a user interface driver: The GUI for the DSS displays the information that the DSS writes in a specific section of the database.

D. Remoting Interface

The remoting interface allows the DSS to be notified of GUI events —such as simulation commands, destination decisions or options (un)checked— and database changes.

E. Rules

The rulebase is currently being developed. It is built on expert knowledge from the pig iron allocation coordinators, quality procedures and the existing constraints. There are three main rulesets:

Model rules These rules define the constraints for the simulator model, and mostly determine which the possible destination facilities for a torpedo wagon are in a given moment, considering the state of the system. Some of the facilities can only be followed by one other facility; these are not accounted as decisions, and are never presented to the user. Instead, they are always made by the rule engine, both in DSS and Simulator mode. Changes to these rules are only needed if the model must change (a new facility or railway line started or an old one closed).

Strategy rules These apply only to DSS mode; these rules make the decisions on destination facility when there are several possibilities, according to the knowledge described above. These rules define the strategy and are meant to be dynamic, so they can embody the better strategy at any time, or they can be used to test different strategies. They will be available in a repository where they can be modified and loaded into the DSS rule engine.

Torpedo number rules These rules evaluate the need to have torpedoes enter or exit the cycle. Too many of them would slow down the cycle, leading to longer times and greater pig iron temperature drops. Too few involve a risk of not having an empty one available in time for tapping the blast furnace. The optimal number varies with system conditions. As is the case for strategy rules, these will be available in a repository where they can be modified and loaded into the DSS rule engine.

The chosen rule engine was ILOG Rules. This rule engine integrates itself in C# and allows to define the rules directly on the objects used in the program; furthermore, it provides an annotation mechanism whereby the variables and methods of an object can be assigned parameterized strings to be shown in the rule editor. In this way, the rules appear to be written in natural language.

Rule edition is done via templates with drop-down menus, so that it is intuitive for anyone to use if the annotation is good.

F. Review of the Additional Specifications

Technical constraints The simulator and support modules were written in C#; the rule engine, ILOG Rules, is implemented as a C# object. Remoting and the database provide all the interfaces needed by the simulator.

End user constraints The definition of each rule includes a priority level and a "reason". The simulator selects among all the rules that are triggered that with the highest priority, and shows its reason; but it also shows all other rules that were triggered, but discarded because of their lower priority level, providing the whole reasoning process. The event-driven nature of the simulator allows to match the coordinators' expectations by defining each process as the passing from one checkpoint to another.

Knowledge constraints The use of a rule engine makes it possible to include the knowledge of the coordinators.

Dynamicity constraints Changes in layout are dealt with by adding, removing or modifying the affected facilities in the database. The changes are loaded the next time the simulator is launched, since the internal representation of the system is dynamically built from the database. The ruleset that specifies default decisions must also be updated to let the system know how the facilities are interconnected. Changes in strategy imply modification of the rulesets. ILOG Rules provides a shared rule repository based on Sharepoint for this, and since the rules are expressed in near-natural language the end users can work on them. The rules are loaded from the repository at every run of the simulator, updating any changes made.

Reporting constraints Two reports have been designed, which allow to calculate three indicators. The first report saves the first movement proposed by the simulator and the first movement of the same torpedo actually performed by the coordinator. The second report saves at a given interval the full proposal of the simulator (the actual results are already being stored by the tracking application). The three indicators that are built from these reports are:

1. Percentage of follow-up of the simulator: how many times (in %) the decision actually carried out is the one that was proposed by the simulator. This is an indicator of how much the simulator is used and trusted.
2. Pig iron temperatures (1): when the decision carried out is the one proposed by the simulator, the comparison (MSE) of the pig iron temperatures measured and those forecasted by the simulator provide an indicator of simulation accuracy.
3. Pig iron temperatures (2): if the decision is different —once the simulator is considered accurate— the comparison of the actual and forecasted temperatures (statistical values) provides an indicator of the performance of the ruleset against that of the coordinator.

V. CURRENT STATE

At the time of writing, all the simulator framework has been built and integrated with the tracking application. Tests have been made using a dummy ruleset; these tests show that the simulator works as expected and only lacks minor debugging and optimization for performance.

The rulebase is now being built from the gathered documentation and by interviewing the coordinators. Advances in the ruleset are validated with them before being committed. Several interviews are needed in order to take into account different cycle conditions (f.i. breakdown of a facility).

VI. CONCLUSION

A simulator has been built that meets the demanding constraints of the project (see above). Its modular architecture allows it to provide each of its final users with their requirements: a GUI that matches the *jargon* of the coordinators; easy maintenance of the physical model through the database and a specific ruleset of connections; an accessible, modifiable rulebase to define the pig iron allocation strategy; and a set of indicators for measuring system performance.

This separation of functionality, physical model and knowledge not only improves maintainability, but has also been a major bonus during development: the different modules could be built sequentially, and thoroughly tested before taking the next step. The addition of the subsequent layers has been the proof of extendability of the system: from a prototype simulator working with a dummy set of three facilities, to a full-fledged one dealing with the real model; from a dummy ruleset with little more than default decisions to the multiple ruleset with conditional selection approach.

VII. FUTURE WORK

The most immediate step now is to finish building the knowledge base and fine tune it in real-world conditions. Next, implementation and management of the indicators will be tackled. Finally, the system will be made somewhat adaptive by making it periodically update the parameters of the facilities by analysing the data from the tracking database.

BIBLIOGRAPHY

Anjewierden, A.; Shostak, I. and de Hoog, R. 2002 "KMsim: A Meta-Modelling Approach and Environment for Creating Process-Oriented Knowledge Management Simulations". *13th European Conference on Knowledge Acquisition, Management and Modelling (EKAW)*, Sigüenza, October 2002.

Cheng, Y. 1998 "" *Knowledge-Based Systems* Volume 10, Issue 4 , January 1998, pp. 225–236. Elsevier Science B.V.

Coats, P. K. 1990 "Combining an expert system with simulation to enhance planning for banking networks". *SIMULATION*, Vol. 54, No. 6, pp. 253–264, Simulation Councils Inc.

Fishwick P. A. 1994 "Computer simulation: Growth through extension". *Proc. European Simulation Multiconference*, 1994. 20 Modelling and Simulation of Uncertain Dynamics.

Konar A. 1999 *Artificial Intelligence and Soft Computing. Behavioral and Cognitive Modeling of the Human Brain*, Chapter 3: Production Systems, CRC Press.

Lee J. J. and Fishwick P. A. 1994 "Real-Time Simulation-Based Planning for Computer Generated Force Simulation", *Simulation*, pp. 299–315. http://citeseer.ist.psu.edu/lee94realtime.html.

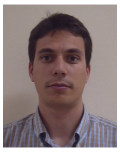
Mitchell, T. M. 1997 *Machine Learning* (International Edition). McGraw-Hill International Editions, Computer Science Series.

Power, D. J. 2002 *Decision Support Systems.* Quorum Books division Greenwood Publishing.

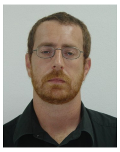
Williams, M. D.; Gorman E. C. and Shiva, S. G. 1996 "Design of a Knowledge-Based Simulation Environment". *SIMULATION*, Vol. 67, No. 2, pp. 121–135, Simulation Councils Inc.

AUTHOR BIOGRAPHIES

DIEGO DÍAZ was born in Avilés, Spain and studied industrial engineering in the University of Oviedo, finishing in 2002. He worked there as a researcher for the department of systems automation for over three years, during which he did his postgraduate studies. Since 2004 he works for the Knowledge Innovation research center of Arcelor Mittal in the fields of simulation, knowledge management and optimisation. His e-mail address is: `diego.diaz@arcelor.com`.



SERGIO R. CUESTA received the M.S.E.E degree in 2003 and worked for one year as a researcher in the Electrical, Electronics and Computer Engineering Department at the University of Oviedo where he is currently completing PhD studies. His main research interests are Data Mining, Information Visualization, Neural Networks and Signal Processing. Since 2004, he is working at Arcelor Mittal Research Centre in Spain. His e-mail address is: `sergio.rodriguez-cuesta@arcelor.com`.



FRANCISCO J. LAGO was born in Avilés, Spain. He has a degree in Applied Mathematics and Computation. He has worked for several years as a software analyst and developer in telecommunications companies. Since 2004 he works for the Knowledge Innovation Research Center of Arcelor Mittal in the fields of simulation, knowledge management and optimisation. His e-mail address is: `francisco-javier.lago@arcelor.com`.

M. TERESA RODRÍGUEZ graduated in Applied Mathematics (1998) and holds a Ph.D in Mathematics for the application of data mining techniques to industrial processes (2003) from the University of Oviedo, Spain. She spent seven years teaching graduate and undergraduate courses at the University of Oviedo. Since 2005 she works in the for the Knowledge Innovation Research Center of Arcelor Mittal. Her research interests include data mining, transport logistics and business process modelling. Her e-mail address is: `mayte.rodriguez@arcelor.com`.