An Anticipatory Control Based on On-Line Real-Time Simulation for Supporting Rescheduling of Complex Industrial Plants with High Automation System

Roberto REVETRIA [a,1], Alessandro TESTA [a], Roberto MOSCA [a], Alessandro BERTOLOTTO [b] and Marco DE LEO [b]

[a] DIME, Università degli Studi di Genova, Via Opera Pia 15, 16145 Genova (Italy)
roberto.revetria@unige.it, alessandro.testa@unige.it, roberto@itim.unige.it

[b] ISO SISTEMI srl, Piazza della Vittoria 11A/8, 16121 Genova (Italy)
marco.deleo@isosistemi.it, alessandro.bertolotto@isosistemi.it

Abstract

The goal of this project is to create a real-time based virtual plant for an easier automated re-scheduling of production plan. Consider a real system plant (steelmaking plant for our case study), with complex logistic for machines placement: the system needs a production order list and the initial plant status, then an initial optimized production planning is generated to satisfy orders. During the production, accidents or other contingencies are possible and an immediate production planning re-scheduling is needed. Introducing a virtual plant all significant events that modify the planned production story introducing delays (i.e. increasing the lead time), we can see the plant status in real-time and for all stored possible events (particularly accidents) the system calculates and shows a new optimized re-scheduled production plan. At the same time the proposed system is able to provide a cost reduction over the energy purchase by interacting actively with the free market.

Keywords: real-time, drools expert, rule engine, steelmaking, simulation.

1 Introduction

The industrial companies (mainly operating high automation level plants are characterized by process strongly linked to technical and operational decisions of production managers) are often forced to operate in a highly stressful environment, characterized by uncertainty for both the completion of production cycle and quality of the product. Although process plants are operated by an automation system, decisions of shift manager or production planning manager are continuously required to ensure compliance with appointments, obtain high quality level products and find an immediate solution when failures occur during the cycle. This work today is organized on the basis of the experience of technicians and supervisors, creating operational practices not easy to implement. Supervisor finds an "impromptu" solution starting by the actual plant status and little "predictable" information about scheduled production target. Increasing the plant automation level, the production rate and objectives of quality and "just in time" delivery, the time to elaborate the best solution is strictly reduced until to become the main problem: this reduced time availability for decider may cause potential dangers for workers, in terms of security and stress.

2 General Architecture

So, consider the virtual system composed by n processes P_1, P_2, ... P_n performing specific operations (heat treatments, chemical analysis, blowing, etc.): the virtual system must be connected to the real system taking in real-time the significant events that modify the scheduled scenario. In **Figure 1** we can see the operating system: the baseline scenario is composed by a scheduled Gantt Diagram without unexpected events; the simulated scenario considers unexpected events (process delays, bad chemical analysis, etc.) and the system produces various proposed scenarios to validate.

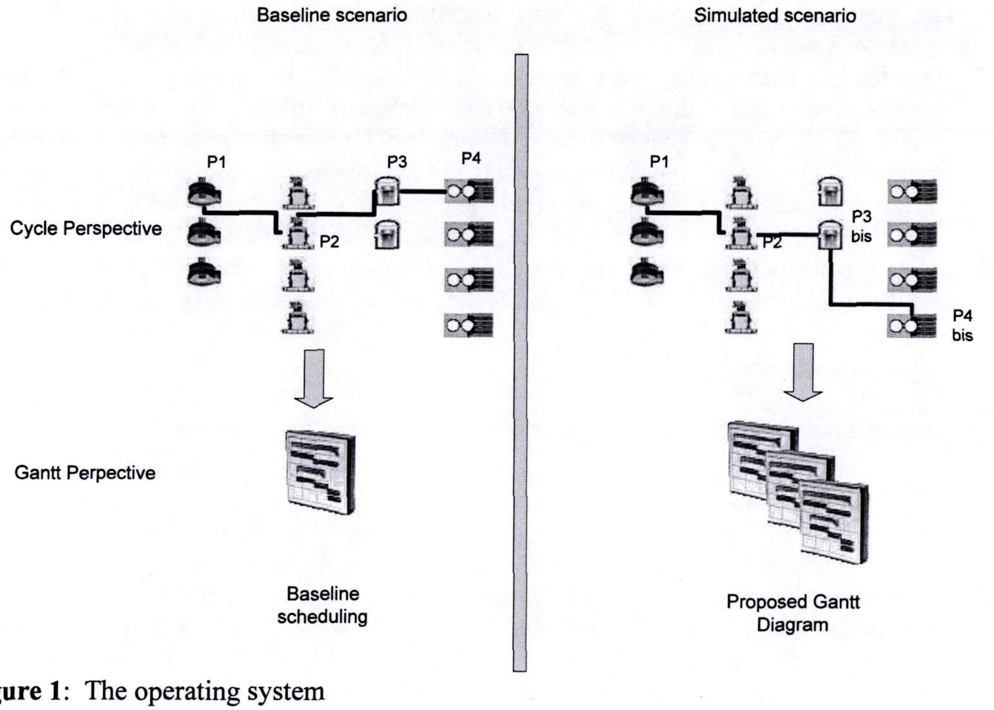

Figure 1: The operating system

System Architecture

The system is composed by three modules: the real time module has the duty to construct and present an overview on the current state of facilities and resources of plants, the forecast map module allows for the representation of the future events by a "on-line script" as planning and the current state of facilities, the simulation module finally starting from baseline scenario, produces the simulation of a set of various possible different scenarios on-line.

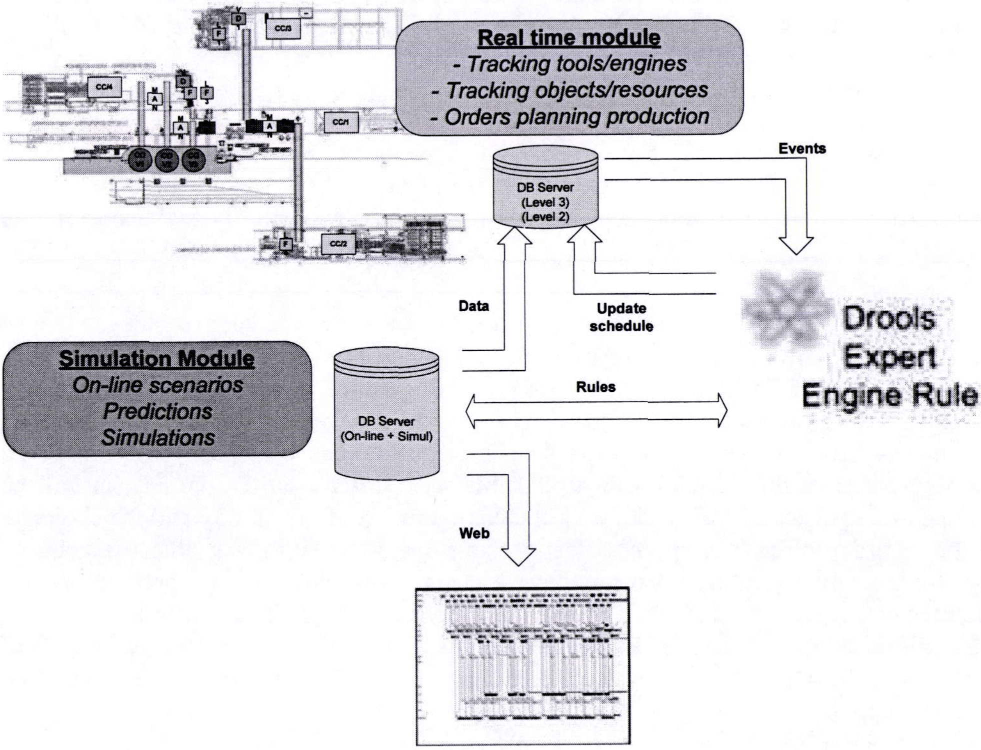

Figure 2: System Architecture

The Real Time Module

This module receives the update of the status of facilities, objects and parameters of the plant. Each object/resource is attached to the previous history and may obtain various information for each object. For example, in steelmaking application, for a steel ladle it will be possible to obtain the list of flows that have been contained, the temperature of the refractory state of the porous layers and drawer as we noted in the various inspections. A first graphical display will see at a glance, the current state of objects, see the previous history and that provided on the basis of current planning (online scenario). The representation will be shown by a Gantt Diagram in which the vertical line indicates clearly the "time-now", on the left of this line will take place the events already happened, on the right of the line are represented scenarios of future events possible.

The Forecast Map Module

This module, by photographing the current situation, updated moment by moment of the first module, based on the standard process parameters and the current status of planning controls, generates a sequence of future events concern: facilities, object/resources, the position of maintenance, handling system and products to be made in the immediate future. The forecast horizon can reach up to one week's time now. The display will be graphic and table, Gantt, in which it will be possible to clearly see a vertical bar indicating the time now, ongoing operations, the expected dates of accomplishment (for the work already started) and dates planning for subsequent events (for activities not yet begun). The display has a color code to assist the operator in identifying the sequence of operations organized.

The Simulator Module

This module will create a simulation from the current installation configuration, a configuration installation future (planned) or configuration of any installation and will create a production scenario, formally similar to the forecast module, but based on assumptions determined by the same user in order to achieve a specific objective. To assist the user in the definition of a plausible scenario simulation, the system will be equipped with a set of "rules" defining the configuration of the facility and the choice to make, in order to reach or approach the possible one of the following objectives chosen by the operator: production increasing, reducing the product/semi product stocks, maintenance operation definition, fixing new delivery dates for a portfolio of orders. The interface will be shown like a vertical bar indicating the "time-now"; start of simulation, the activities already started, and completed the "simulated" on a supposed scenario application. The set of "rules" is managed by a dedicated module as the Rule Engine System: this management is developed by Drools Expert. The next chapter is dedicated to the Rule Engine Architecture.

Rule Engine Architecture

Artificial Intelligence (A.I.) is a very broad research area that focuses on "Making computers think like people" and includes disciplines such as Neural Networks, Genetic Algorithms, Decision Trees, Frame Systems and Expert Systems. In order to manage the Real-Time System the last one discipline is chosen: Expert Systems use Knowledge representation to facilitate the codification of knowledge into a base knowledge which can be used for reasoning; we can process available data using base knowledge to infer conclusions. Expert Systems are also known as Knowledge-based Systems and Knowledge-based Expert Systems and are considered to be "applied artificial intelligence". The process of developing with an Expert System is Knowledge Engineering. EMYCIN was one of the first "shells" for an Expert System, which was created from the MYCIN medical diagnosis Expert System. Whereas early Expert Systems had their logic hard-coded, "shells" separated the logic from the system, providing an easy to use environment for user input. Drools is a Rule Engine that uses the rule-based approach to implement an Expert System and is more correctly classified as a Production Rule System. The term "Production Rule" originates from formal grammars where it described as "an abstract structure that describes a formal languages precisely, i.e., a set of rules that mathematically delineates a (usually infinite) set of finite-length strings over a (usually finite) alphabet". The brain of a Production Rules System is an Inference Engine that is able to scale to a large number of rules and facts. The Inference Engine matches facts and data against Production Rules to infer conclusions which result in actions. A Production Rule is a two-part structure using First Order Logic for reasoning over knowledge representation.

```
when
        <conditions>
then
        <actions>;
```

The process of matching the new or existing facts against Production Rules is called Pattern Matching, which is performed by the Inference Engine. There are a number of algorithms used for Pattern Matching by Inference Engines including: Linear, Rete, Treat, Leaps: Drools implements and extends the Rete Algorithm The Drools Rete Implementation is called ReteOO, signifying that Drools has an enhanced and optimized implementation of the Rete Algorithm for Object Oriented systems.

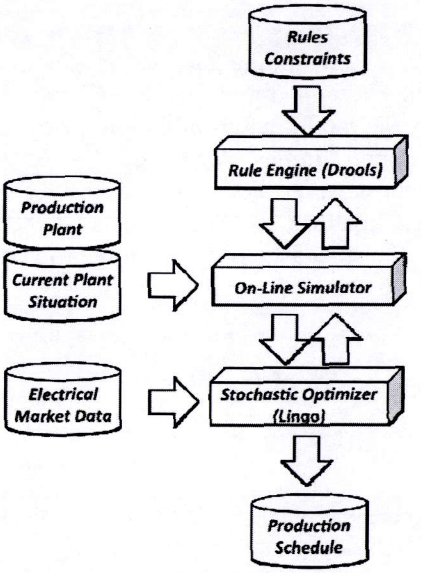

Figure 3: High Level View of the anticipation/optimizing control system.

Figure 3 presents a general view of the implemented anticipation/optimization control, starting from the current plant situation and the production plan the systems, using the on-line simulation, computes the next periods events list providing an evaluation of the fulfillment of the production schedule. Based on the evaluation of such schedule the rule engine suggests possible corrections to the current production schedule in order to:

reducing wasting of time, increasing utilization coefficient, solve issues about defects, re-routing productions and all other actions normally carried out by a typical production supervisor. Since one of the key issues in many production facility is the energy cost: the proposed systems is able to interact actively with the market in order to proper manage the purchases of energy over the free market. For this purpose a stochastic optimization engine (Lingo) is used. The overall production plan is divided in three groups:

1. Mandatory production: is the production that has to be executed in order to ensure fulfillment of the incumbent orders, this production must be executed at any energy price, the energy is generally purchased well n advance on the forward market.

2. Elective production: is the production that could be postponed when the energy price will reach a convenient value but not after a deadline. The energy is purchased on the spot day-before market.

3. Opportunity production: is the production that will be executed only at a minimum energy price, the energy will be typically purchased on the spot-intraday market.

The system will provide, based on the production plant, the profile of the energy purchases to be executed over the three markets (Forward, Spot-Day-Before, Spot-Intraday) and, eventually, the modification of the production schedules in order to take advantages of the various opportunities. For the purpose of this function the systems is acquiring in real time the market data (hi-lo, closures) over the three markets, processing the time-series, creating the distributions of the prices variables over the market prices and solving the stochastic programming problem.

3 A first case study: the Steel Industry

Steelmaking process description

The entire steel product cycle is typically composed by 3 macro areas: hot metal making (or scrap yard preparation if electric steelmaking plant), steelmaking and rolling mill. For steelmaking process description, consider the picture below:

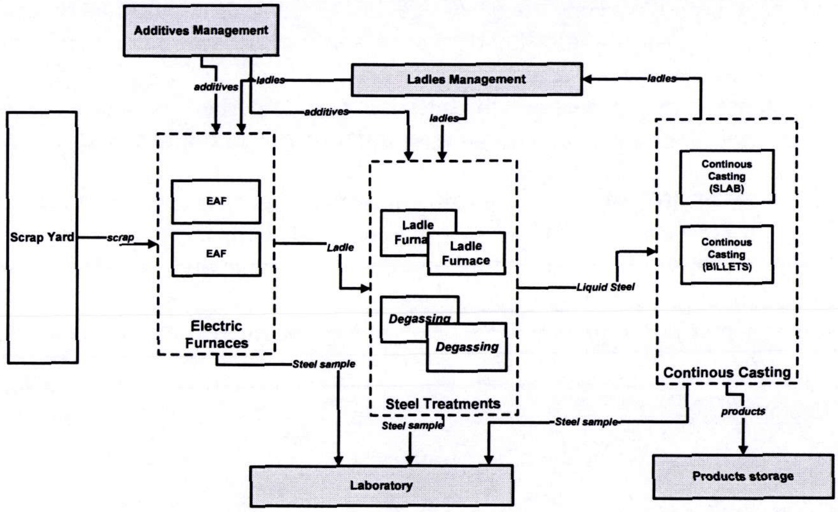

Figure 4: Typical detailed steelmaking cycle

The steel production cycle is typically composed by 4 macro areas: steelmaking flow areas, ladles management, analysis laboratories and products storage management. In this paper we consider the only steelmaking cycle (**Figure 4**). In the following a brief description of the steel process is presented. Scrap metal is delivered to a scrap bay: a particular scrap recipe list composes the steel grade to product, so the steel grade defines the future process cycle of ladles. Scrap generally is loaded into large buckets called baskets. The scrap baskets are then taken to the melt shop (Electric Furnaces Area), the roof is swung off the furnace, which is charged with scrap from the basket. A lot of energy is generated by multiple tons of falling metal; any liquid metal in the furnace is often displaced upwards and outwards by the solid scrap, and grease and dust coating the scrap are ignited when the furnace is hot, resulting in a fireball erupting out of the top of the furnace and the slag door. In some twin-shell furnaces, the scrap is charged into the second shell while the first is being melted down, and pre-heated with off-gas from the active shell.

After fusion, steel is tapped into a ladle, then moved by crane or transfer car to the steel treatments area (only if the laboratory has produced positive results for the EAF steel analysis sample sent before, typically during the end of fusion); to reach the expected steel grade the ladles passes through the Ladle Furnace process, that maintains high temperature of steel (by electrodes lowered onto the liquid steel) adding particulars ferroalloys: during this process, some steel samples are analyzed to control the chemical composition of the steel grade. Another possible steel treatment is the vacuum degassing (typically for steel grades for harder railheads): the ladle is inserted into a cylinder vessel, then the roof is swung off the place and air and gases are removed from the internal environment, by creating the vacuum. Vacuum degassing causes hydrogen to diffuse and separate from the liquid steel so as to prevent hydrogen-induced defects.

Then, the steel liquid is ready to be casted in the continuous casting machine: this complex process transforms the liquid steel to solid steel bars (named slab, billets or blumes). Ladle is charged onto a double place-tower, attending the end of previous ladle (casting the same casting sequence scheduled): is very important for successive ladles to respect this appointment to maintain the continuity of continuous casting sequence. From the bottom of the ladle the steel liquid flows into a tundish dividing the main liquid flow into N different flows (casting lines): for each line, steel passes through ingot moulds (one for each line) necessary to cool and shape the steel (number of lines N depending by the semi product type casted: N=2 (typically) for slabs, N=4/6 for billets and blumes). Steel finally is pulled across the line and cut into slabs or billets (by oxygen-cut machine). In **Figure 5** the scheme of the described continuous caster.

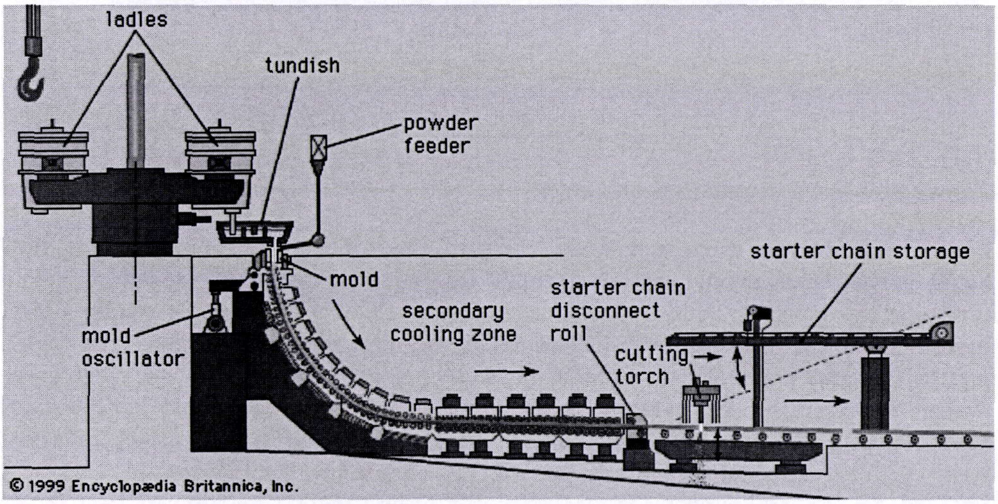

Figure 5: Continuous Caster (Frontal View)

The prototypal system

The entire prototypal system, named SimProcess™,is composed by a number of virtual machines representing a specific block. These blocks are: Real-time machine contains the Real-Time Module composed by a Real-Time simulation model (named Real Time Plant) developed using a discrete-event simulation tool (Plant Simulation™), but the model run in real-time only (not used with accelerate simulation time). The Real-Time Plant gets the current status from Automation System (Level 2) and current Production Planning (Level 3). The current status is updated each 60 seconds and the Real-Time Plant may generate events as Real-Time events occurs in a real plant, for example: break out of a continuous caster that interrupt the production in a specific line or, more often, an out-of-range chemical analysis. When Real-Time Plant generates anomalies, the rule-engine block in the Simulation Model verifies which rule is possible to indicate to the Virtual Simulation Model as the best solution to resolve the anomaly. Then, the

Virtual Model generates different scenarios with fine-tuning of controllable input parameters in order to obtain the best solution for decision making (by user); if the solution is good, the rule-engine block actuate the solution, rebuilding the planner module with the new updated production plan. A web-based module carries out the updated Gantt Diagram (consumptive + provisional) as production plan user interface. The rule-engine prototype is composed by two classes of rule containers: Chemistry_Drools and TimeDrools. Each rule container is a .drl text file that contain the rule list: Chemistry_Drool contains the chemical analysis rules according with the data collection activity used for this case-study. For example: at the end of EAF process there is the SAMPLING phase. At the sampling phase a steel sample is taken for chemical analysis in the laboratoty. During the analysis percentage of particular elements as phosphorus (P) and copper (Cu) are evaluated. Now, consider P_m and Cu_m as measured values by laboratory and P_{max} and Cu_{max} the maximum values. If $P_m > P_{max}$ a re-blowing operation for process EAF is needed and if $Cu_m > Cu_{max}$ the only need is to change the Steel Grade. Otherwise, if the anomaly is evaluated for long time processes, the TimeDrools rule container is called. For example: time transport for a ladle (entity in general) from a resource (Ladle Furnace LF) to the next one scheduled (Continous Casting, CCO) is expected in 20 minutes but, for unknown problems, the entity arrives in the designated resource with a large delay (over 10 minutes): in this case the TimeDrools rule container is called to find the new best cycle solution without Steel Grade quality change. At the same time the model is evaluating the electric workload (MW every minute) providing an estimate of the next period energy consumption. At this point the consumption is compared with the already purchased (over the Forward market) energy for the corresponding period is energy volume is consistent with this forecast the production schedule is confirmed otherwise an action is taken. Based on the rule energy the minute per minute gap between energy needs and energy purchased/sale or swap are evaluated: in energy is not sufficient the market is checked to verify if an opportunity of purchase or swap are possible (energy purchase rule, energy swap rule) or if a production change is allowed (production rule). In both case the Drools Rule Engine is responsible for such choice. In case of excess of energy purchased respect to the consumption estimate (i.e. in case of a plant failure the marked is checked for an energy sales opportunity (energy sales rule) or an energy swap opportunity (energy swap rule). All the market transaction are supposed to take place OTC. Over-the-counter (OTC) or off-exchange trading is done directly between two parties, without any supervision of an exchange. It is contrasted with exchange trading, which occurs via these facilities. An exchange has the benefit of facilitating liquidity, mitigates all credit risk concerning the default of one party in the transaction, provides transparency, and maintains the current market price. In an OTC trade, the price is not necessarily made a public information, so such information will remain into the system. This last part is provided as a web-service from the system, each SimProcess™ has an OTC module that provide access to the free market where the manager can set purchase/sales order plus an internal OTC swap module that exposes swap deals to all the other SimProcess™ installations belonging to the same organization. In such way two plants may exchange energy contracts if one is on a long position and the other is on a short position.