INFO TO DECISIONS - APPLICATION OF DECISION NETWORKS IN MINE MANAGEMENT SYSTEMS

Steven Chung
Senior Mining Consultant
Modular Mining Systems
Port Coquitlam, BC

Sergey Seroukhov
Product Line Architect
Modular Mining Systems Inc.
Tucson, Arizona

Abstract

Decision Network Modelling in a mining operation represents a complex process environment that is inherently more variable than a processing plant or factory. The uncertain availability of loading locations, haulage paths, loading and hauling equipment, along with inconsistent operator behaviour, cause a relatively large number of upset events that must be recognized, communicated and dealt with by the appropriate management authority. Traditional process maps that represent a deterministic model of the mining process can quickly break down in the face of such a volatile environment. An alternate modelling strategy, composed of decision networks, focuses on the interactions between goal-oriented entities and their environment, allowing system integrators to explicitly determine the information to be exchanged while providing feedback mechanisms that improve system flexibility and decision quality. This ensures a consistent framework for incrementally increasing system automation.

Introduction

In the modern world, the amount of data that mining companies generate and need to analyse is increasing at an accelerating rate. From real-time mine management systems and mine planning systems, to enterprise management systems, plant management systems, accounting systems, stakeless surveying systems, ERP systems, and OEM interfaces, the amount of data seems endless.

Information overload can quickly overwhelm the ability for humans to effectively process and make decisions. To help humans make sense of the raw data, mining information systems automate some of the analysis and decision-making process. However, humans must still bridge gaps and fill in holes between systems.

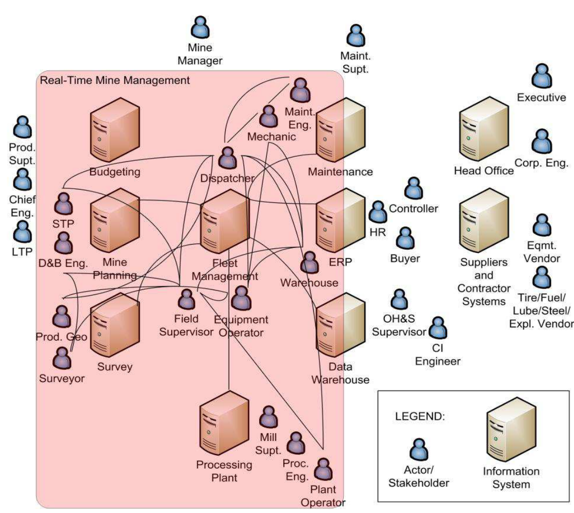

Figure 1. Complex interconnection of mine management system and stakeholders

Several major mining companies have published their vision for the future as a digital, or fully automated, mine (Albanese, 2008 and Orellana, 2007). The number of stakeholders, components and dependencies is large. The cost of traditional integration methods can expand exponentially as system scale and variety increases. More importantly, it may be necessary to integrate systems that have been designed based on different principles, with a focus on different objectives. Where system components are conceptually misaligned, no amount of investment can deliver seamless integration. This challenge must be addressed to create the system of systems required to deliver a digital mine.

Evolution of the Digital Mine

Since 1979, Modular Mining Systems Inc. has led the mining fleet management industry, with nearly 200 systems installed at some of the world's largest mining operations. Since its inception, Modular has been dedicated to bringing cutting-edge technology and solutions to its customers and to removing technical hurdles whenever possible.

Until fairly recently, there was no off-the-shelf unified framework to unite real-time control, communications, database and reporting operations, so Modular built proprietary tools from the ground up. With the release of the Microsoft .net 3.5 framework, Modular began a product line migration into commercial off-the-shelf technology in order to reduce the technological gaps associated with integration.

With technological gaps well-addressed by existing tools and methodologies, connectivity issues are for the most part no longer a problem. However, simple connectivity does not guarantee seamless integration. The problems of logical gaps and conceptual misalignment remain (McElman and Seroukhov, 2009):

Logical Gaps/Conceptual Misalignments	**Limitations of Existing Solutions**
Structural Mismatch	Conversion between formats is possible using ontological models, but only to the extent that concepts are compatible.
Incomplete Information	Difficult to discover and combine information from multiple systems if a single source is incomplete. Simple business transactions require gathering of information from multiple sources. Production metrics do not contain links to the decisions. This limits the ability to understand why things happened.
Duplicated Information	Multiple parallel copies of information can reduce data integrity. Difficult for middleware to reconcile multiple data sources.
Different Levels of Granularity	If information has already been aggregated, it may be impossible to decompose to finer detail.

Table 1. Logical gaps and limitations of existing solutions

Traditionally, the Logical Gaps have been dealt with on a case-by-case basis, leading to a patchwork of data transforms and bridging logic. Therefore, achieving a high level of integration requires increasing costs and complexity (Paige & Inbar, 2008).

To formulate a unified solution to the problem of Logical Gaps and Conceptual Mismatches, we have to understand the source of divergent requirements of the users:

In order for a large organization to operate efficiently, the various functions of the organization need to be distributed and delegated to different departments, workgroups and individuals. This allows specialization of individuals, which increases timeliness and efficiency when various tasks are executed.

The departmentalization of the organization allows for the executives to abstract broad or strategic plans on a company or department level, and those plans are decomposed into individual tasks and decisions as the strategic plans are turned into tactics. Ideally, the results of the individual tasks and decisions are aggregated back into strategic plan feedback, allowing managers to determine how the strategic plan is progressing and to make changes where required.

This type of organization forces each department to work in independent "silos" with crossover only possible where the organizational hierarchy allows. While this allows for effective use of parallel resources, it causes communication bottlenecks for inter-departmental communications, as well as for diverging objectives and requirements for enterprise level integration.

Traditional "business process" modeling techniques, such as IDEF0, Flow Chart, or Business Process Definition Language, often aggravate this problem. They are usually actor-centric and require extensive detailing of the actor-biased rule-based decision logic. The information exchanged is implicit in the model, and tends to isolate decision automation inside departmental silos.

In order to achieve "Enterprise Level Integration and Automation", a new toolset is required. This toolset should explicitly express the information to be exchanged as well as the relationships between the systems and the people that use them, decreasing workflow and increasing worker effectiveness.

Conquering Integration through Command and Control

While investigating possible modeling methodologies to align and integrate the objectives of a real-time mine management system with other business and planning systems, Modular found that the military sphere has dedicated considerable research effort into defining and managing volatile environments according to scientific principles (Alberts & Hayes, 2006). It became apparent that that the principles of Military Command & Control (C2) can be converted and extended to industrial domains. This subset of principles is described as “Industrial Command and Control” to differentiate it from the original source (McElman and Seroukhov, 2009). The modelling method revolves around the OODA loop model:

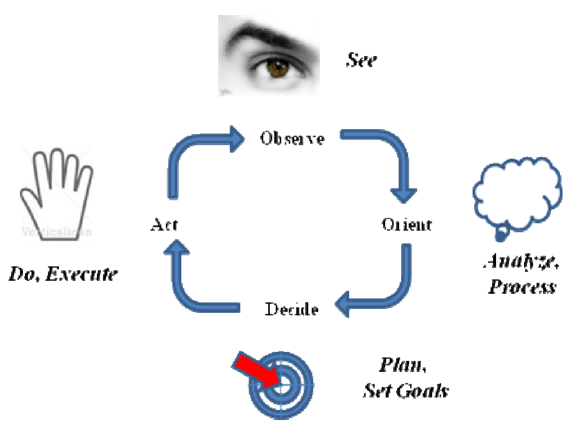

Figure 2. OODA loop

In the late 1970’s, US Air Force Colonel John Boyd created the Observe-Orient-Decide-Act (OODA) model of real-time decision making from his observations of dogfights between F-86 Sabres and MiG-21s during the 1950-52 Korean War. Boyd was himself an outstanding fighter pilot with service in the Vietnam War. Despite its origins, the OODA model is restricted neither to jet fighters nor to military operations. Over the course of time, OODA has been adopted by other military services, both inside and outside the USA, and by several large commercial organisations. (T.J. Grant, 2005)

The OODA loop model is executed by "Decisions Making Entities" (DME) that continuously evaluate the OODA loops and interact with the other DMEs to create a "Decision Network". For example, a DME could be a CEO making a strategic corporate decision, and a Decision Network would describe how the decision is propagated to specific departments and individual operators.

The Decision Network Concept

The fundamental unit of the Decision Network is the Decision Making Entity. A Decision Making Entity is any person, process or group that makes decisions and conducts actions based on observable information. This could be a decision made by a program, individual, department or even a company:

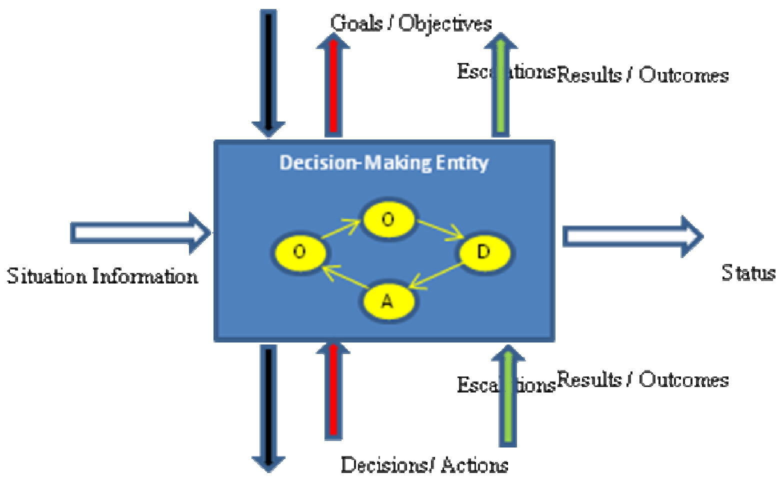

Figure 3. Decision-Making Entity

In isolation, the DME interacts with its surrounding environment based on the following inputs/outputs:

- Goals/Objectives - There must be a reason for any action to be taken. The goal can be specific to the decision, such as "arrive at this location", or broader in scope where the decision brings the Decision Making Entity one step closer to its goal (e.g. Move 1000 tons from this bench to this dump).
- Situation Information - Relating to the Observe portion of the OODA loop, the situation information relays all parameters that are relevant to the decision making process and the various constraints that need to be applied to those parameters. This could be the current grade of material received at a dump and the range of grades accepted at that dump.
- Decision/Actions - The corollary of Goals/Objectives are Decisions/Actions. The Decision Making Entity continuously monitors the situation information to first determine the current situation and any operating constraints, and then plans a course of action that brings it closer to achieving its goal.
- Status - Updates to the situation information that come as a result of the action taken.
- Escalations - If a decision is blocked due to lack of information, non-compliance with constraints or inability to execute an action, the decision should be escalated to the next level.

- Results/Outcomes - In order to provide feedback on the decision quality, the observable outcome of the decision is traced back to the decision.

Each of these interactions has a time associated with it (i.e. Situation Information has sample rate, Decisions require processing time and Status information has an update frequency). The time it takes for the DME to detect the Situation Information and make a Decision is the DME agility.

Linking Decision Making Entities creates a Decision Network. The Decision Network is also a goal-oriented system that combines individual interactions of the DMEs and unites then to achieve the goal of the highest level DME.

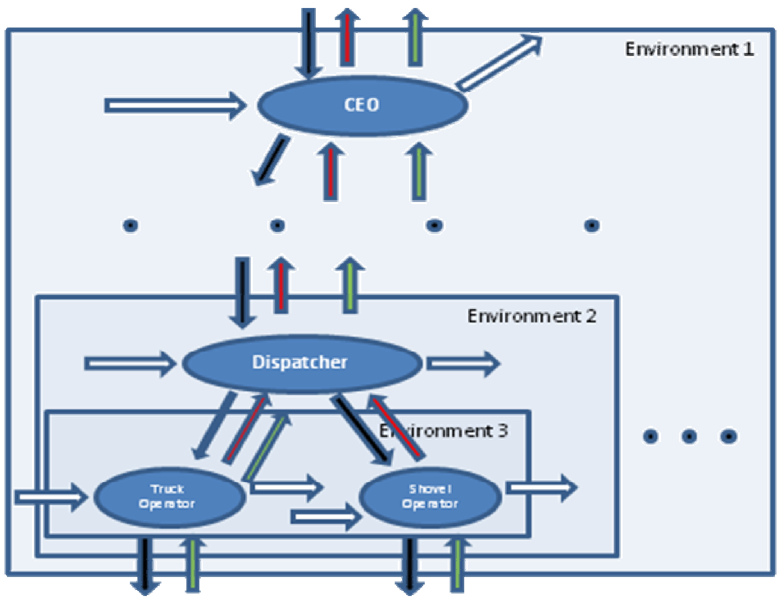

Figure 4. Decision Network

A Decision Network concentrates on the relationships between each DME. Mapping these relationships allows for different concepts to be applied:

- Composition and Sub-division - In Figure 4, there are four Decision Making Entities with three different Environment Boxes. DMEs can be aggregated by the Environment Box in which they reside , allowing the system to be summarized to the level of detail required (i.e. Dispatcher, Truck Operator and Shovel Operator in Environment 2 can be combined under "Haulage Operations" when propagating daily production plans).
- Decision-Goal relationships - The decision of one DME could be the goal of another (e.g. the decision of the Dispatcher to move a truck to another production circuit becomes the goal of the Truck Operator to haul in that circuit).
- Indirect interactions - Decisions are often a result of monitoring a set of variables in the environment (sensor) rather than a direct result of another DME (e.g. watering a road)

Decision Network Example

The following section provides an example of a road cleanup workflow that compares a traditional Business Process Diagram with Decision Making Entities.

Business Process Diagram

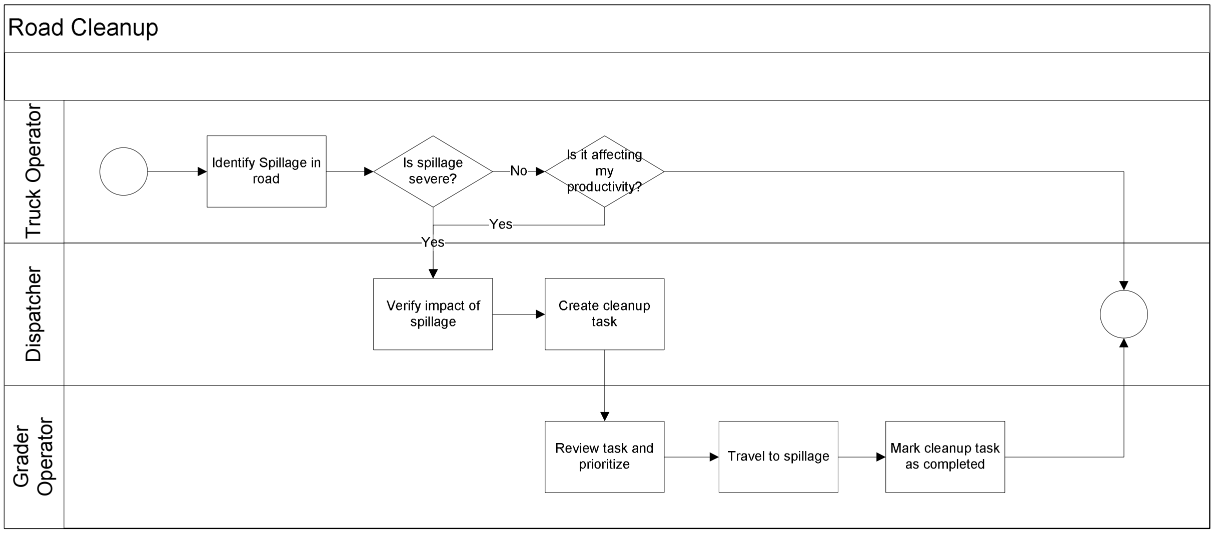

Figure 5. Business Process for Road Cleanup

In the diagram above, the Truck Operator identifies a spillage on the road and notifies the Dispatcher based on its severity. The Dispatcher then creates a task for the Grader Operator, who then cleans up the spillage and marks the task as completed.

The Business Process "swimlane" diagram provides an overview of the overall process; however, it does not provide detail on the exact information exchanged by the actors in the process. In order to provide further detail, a Suppliers/Inputs/Process/Outputs/Customers (SIPOC) diagram is required.

SIPOC

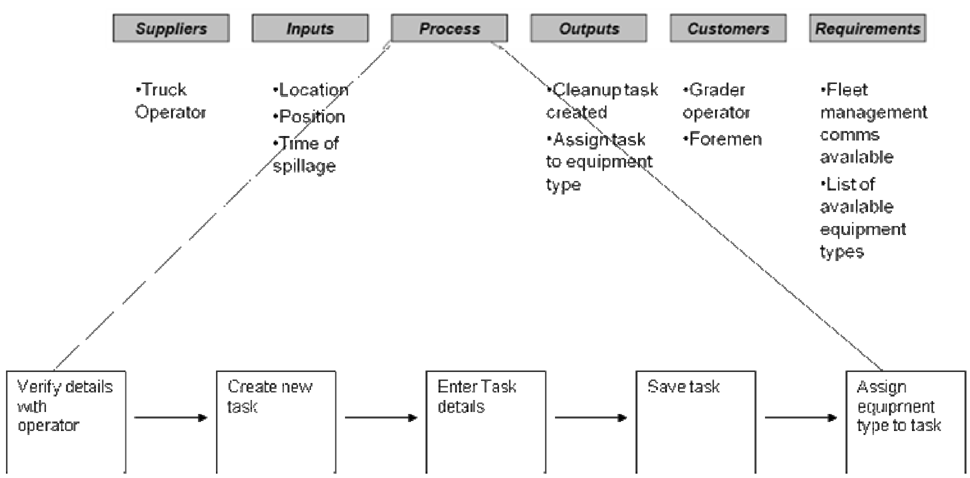

Figure 6. SIPOC diagram for Create cleanup task

To gather greater detail about the business process, each step of the process needs to be elaborated in a SIPOC diagram. In this example, the SIPOC diagram elaborates the "Create cleanup task" set in the overall "Road Cleanup" Business Process Diagram.

Thus, the combination of the Business Process Diagram and the SIPOC diagram provides a mechanism for detailing a deterministic process execution.

Decision Making Entity Diagram:

Because Command and Control needs to operate in environments that are in a constant state of flux (e.g. a battlefield), the primary objective of the actors is not to execute a process, but to achieve a goal.

Therefore, the Decision Making Entity diagrams focus more on the information requirements to achieve a goal and the situation information surrounding barriers to accomplishment.

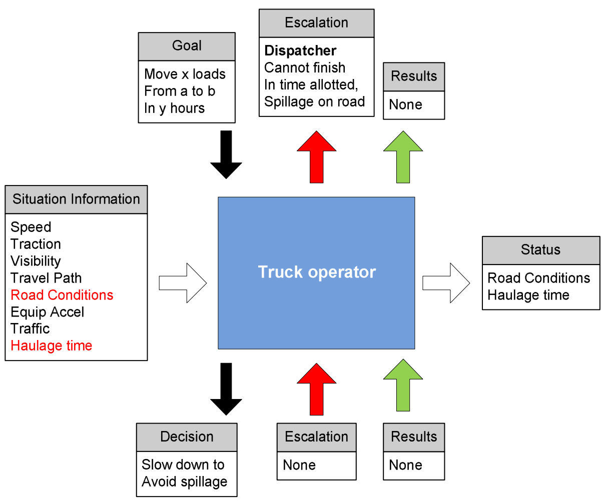

Figure 7. Truck Operator DME diagram

In the DME diagram for a Truck Operator, the primary objective is to move a set number of tons from the blast to the dump in a set amount of time. A spillage in the road is seen as a barrier to accomplishing that goal as it would increase the truck's haul time when the operator slows down to avoid the spillage.

This barrier is escalated to the Dispatcher to either adjust the goal or allocate other resources.

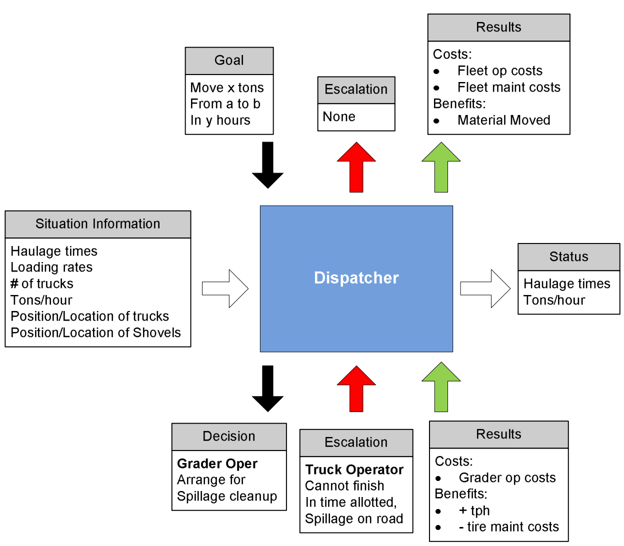

Figure 8. Dispatcher DME diagram

In the DME diagram for the Dispatcher, the primary goal is to move a set tonnage from various locations in a set amount of time. While the situation information would eventually indicate that there is a problem on one of the haulage paths through decreased productivity, the Truck Operator is able to directly escalate the spillage to the Dispatcher because of the reporting hierarchy.

The Dispatcher can then decide whether the costs of allocating grader resources justify the benefits of increased productivity and decreased maintenance. If so, grader operating costs, the resulting increase in truck productivity, and the estimated decrease in maintenance (probability and cost of a tire cut) can be fed back into the decision to determine the quality of the decision for future reference.

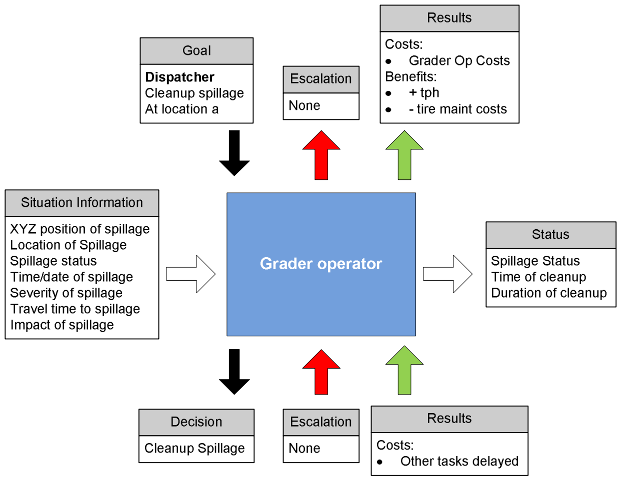

Figure 9. Grader Operator DME diagram

Once the decision has been made by the Dispatcher, it becomes the goal of the Grader Operator to complete the cleanup.

The Grader Operator will take the situation information of the spillage and decide when the cleanup should occur based on its priority relative to other tasks. Once the task has been completed, the system can calculate the opportunity costs associated with delaying the other tasks as opposed to the benefits of completing the spillage task immediately.

Business Process vs. Decision Networks

The examples above take two different approaches to documenting how a particular process takes place. The Business Process documents the process chain and how each step is executed. The Decision Network focuses on the goals of the Decision Making Entities and the circumstances in which decisions were made to achieve each goal.

This difference in approaches means that Business Processes Diagrams tend to plot out "one way" deterministic flows where there exists a distinct start and end to the process. Decision Networks, however, imply monitoring and feedback loops that allow for broader-based objectives and higher degrees of flexibility, while enforcing accountability for each of the decisions made. This flexibility and feedback allows the DME's in the Decision Network to compensate for unforeseen events and still achieve the overall system goal.

Other repercussions of this change in philosophy occur when integration architecture is developed, eliminating functionality "chains" which, when crossing multiple systems, can to a "spaghetti code" type of integration. Instead, functional "cells" pass information to and from other "cells" using common infrastructure (such as databases). This enables both direct and indirect interaction, while simplifying deployment and support of system-wide integration.

But this does not imply that Decision Networks are better than Business Processes Diagrams. Decision Networks describe how functions interact with each other, but not the functions themselves. This means that Business Process Diagrams are still required for individual actors when describing system functionality.

Conclusions

To improve productivity and efficiency, mining companies are becoming increasingly dependent on computerization and automation. But with increased technology adoption comes increased complexity of integration. And without some sort of cohesive framework to organize system integration and automation, the administrative, support and switchover costs of a fully digital mine will outweigh the productivity gains made through integration and automation.

The solution comes from the Military domain of Command and Control. This framework focuses on the actors' goals, the situation information required and the monitoring of roadblocks and feedback loops to track progress. Individuals or groups are given the flexibility to continue to operate in the face of unexpected events, and structured communications allow a group to self-organize without predetermined rigid procedures. A Decision Network framework focuses on "cellular" functionality grouping, providing actors all the information required to accomplish their goals. Interactions can be explicitly mapped out so that each "cell" is fully integrated into the organization as a whole.

If used as a requirements tool, Decision Networks automatically propagate the principles of Industrial Command and Control to software architecture, because the mechanism focuses on the nature of the interactions between actors and the information that is exchanged. Decision Networks do not replace Business Process Diagrams, but rather complement them by providing situation information that describes the environment in which these processes are executed.

In producing a "cellular" software architecture, Industrial Command and Control effectively enables systems integration projects to be broken into smaller sub-projects. This allows mines to realise the benefits of integration and automation more quickly as sub-projects are completed. But more importantly, it maintains the overall integrity of operation of enterprise management systems, and the architectural framework required for the eventual fully integrated and automated system. Much like Industrial Command and Control itself, the key to successful integration and automation is the ability for projects and sub-projects to provide feedback as quickly as possible.

REFERENCES

ALBANESE, T. (2008). Rio Tinto chief executive unveils vision for „mine of the future‟. Retrieved January 6 2010, from http://www.riotinto.com/media/5157_7037.asp

ALBERTS, D.S. & HAYES, R.E. (2006). Understanding Command and Control, Retrieved January 6 2010, from http://www.dodccrp.org/files/Alberts_UC2.pdf.

ATKINSON, S.R. (2005). The Agile Organization. Retrieved January 8 2010, from http://www.dodccrp.org/files/Atkinson_Agile.pdf.

BOYD, J.R. (1987). Organic Design for Command and Control. Retrieved January 15 2010, from www.ausairpower.net/JRB/organic_design.ppt.

Enterprise Application Integration. Retrieved January 8 2010, from http://en.wikipedia.org/wiki/Enterprise_application_integration.

GRANT, T. J. (2005). Unifying Planning and Control using an OODA-based Architecture. Retrieved January 14 2010, from http://www.c2coe.org/beta/files/Reports/TJ.Grant_NLDA_%20Unifying%20Planning%20and%20Control%20using%20an%20OODA-based%20Architecture.pdf.

MCELMAN, C. & SEROUKHOV, S. (2009). From Buttons to Bits - Achieving Level 3 Integration. Paper 4383, APCOM 2009 conference, Vancouver BC, Oct 2009

ORELLANA, M. (2007). Digital Codelco: Codelco's Strategy in the Use of Information & Communication Technology & Automation. Retrieved January 11 2010, from http://downloads.gecamin.cl/cierre_eventos/apcom2007/prsntcns/00048_00263_pr.pdf.

PAIGE, R., INBAR, D. (2008). Reality Check: The Costs of Data and Application Integration. Retrieved Jan 11 2010, from http://www.information-management.com/infodirect/2008_99/10002234-1.html.

RICHARDS, C.W. (2001). A Swift, Elusive Sword. Retrieved January 12 2010, from http://www.cdi.org/mrp/swift_elusive_sword.pdf.

SIMON, K. SIPOC Diagram. Retrieve February 10, 2010, from http://www.isixsigma.com/library/content/c010429a.asp.

WHITE, S.A (2005). Using BPMN to Model a BPEL Process. Retrieved Feb 4 2010, from http://www.businessprocesstrends.com/publicationfiles/03-05%20WP%20Mapping%20BPMN%20to%20BPEL-%20White.pdf.

WILSON, C. (2004). Network Centric Warfare: Background and Oversight for Congress. p. 15 Retrieved June 18 2009, from http://www.globalsecurity.org/military/library/report/crs/33858.pdf.