

# Adapting to Uncertain and Evolving Enterprise Requirements

The case of business-driven business intelligence

Eric Yu, Alexei Lapouchnian, Stephanie Deng

University of Toronto

Toronto, Canada

**Abstract**—Information systems today are expected to function in an increasingly dynamic world with many uncertainties. System development is seldom a linear progression from well-defined, fully-specified requirements to finished products that fully meet the initial requirements. More likely, there are ongoing cycles of exploration, design and implementation, taking into account evolving needs and capabilities, as well as lessons from earlier cycles. Existing requirements modeling and analysis techniques largely presume application settings that are stable and predictable. Can these techniques be used to support analysis in the new dynamic environment? Scenarios from the recent surge in demand for business intelligence capabilities in enterprises provide an interesting setting for examining organizational and IT responses to the challenges of high uncertainty and rapid change. In this paper, we apply existing requirements modeling techniques to these scenarios in order to uncover their inadequacies, and to identify research challenges.

**Keywords**—requirements modeling; evolution; adaption; ongoing change; design cycles; business intelligence;

## I. INTRODUCTION

A major challenge for information systems (IS) engineering today is the need to cope with ongoing change. Consider the pressures faced by enterprise IT professionals integrating social media, or enabling mobility for workers and customers, or achieving environmental sustainability. In each of these areas, the change cannot be accommodated by a single episode of system redesign and implementation. Each area opens up an enormous range of possibilities, which are explored and selectively acted upon over extended periods of time. There are numerous unknowns and many uncertainties. Change occurs at many levels of granularity, and over many cycles of iteration. Change initiatives – large and small, long-term or short-term – will encounter varying degrees of success, with lessons learned feeding back into subsequent iterations.

The challenge has prompted a call for fundamentally new approaches to requirements, and new conceptual framework and theories for understanding requirements [1]. It is recognized that requirements activities need to deal simultaneously with the social and the technical, as people and technology interact in emergent ways.

IS engineering has long relied on modeling techniques to support requirements analysis. Process models and data models

are used extensively, while intentional and social modeling techniques have been introduced more recently [2][3]. Nevertheless, most modeling techniques developed to date presume application settings that are fairly stable and predictable. Can these techniques be used to support analysis in the new, highly dynamic environment?

To better understand the capabilities and inadequacies of existing requirements modeling techniques in the context of highly dynamic environments, we consider the recent rapid uptake of business intelligence (BI) technologies in organizations. BI, analytics, and big data have come to be seen lately as critical for advancing business performance and competitiveness [4][5]. The sociotechnical trajectory of how BI took shape in organizations provides a microcosm of the challenges of uncertain and evolving requirements. The trajectory reflects the increasingly common situation where people and technology each have to adapt to the other. Multiple rounds of adaptation may occur, as responses are often found to be inadequate, and renewed attempts are made. Furthermore, these adaptation responses often introduce technology innovations that aim to reduce the cycle time for responding to new changes. These dynamic and adaptive phenomena in today's enterprise IT environments present new challenges for modeling and analysis.

Section II reviews the relevant literature. In Section III, we introduce recent developments in business-driven BI (BDBI). As each stage of development unfolds, we examine to what extent an existing technique for goal-based social modeling ( $i^*$ ) is able to support analysis of the pertinent issues. Inadequacies are noted as items for future research. In Section IV, we consider the multiple layers of change in dynamics environments. In attempting to model multiple layers of change processes, we opted for the widely-known BPMN [6] notation. Again, we note its limitations, and introduce some improvised extensions for our purpose. In Section V, we summarize modeling issues we encountered and outline a set of desired features for a more comprehensive modeling framework to deal with the challenges of conceptualizing requirements in an increasingly dynamic world.

## II. RELATED WORK

Requirements change has long been of concern in information systems requirements engineering (RE) research (e.g., [7][8]). Software evolution has also been a major preoccupa-

tion in software engineering (e.g., [9]). These studies have largely focused on software engineering artifacts and processes, and not so much on dynamics in the usage organizational environment. They have not considered mutual adaptation or co-evolution between business users and IT from a sociotechnical perspective.

Iterative optimization of organizational processes is the key feature at the highest level of maturity in the CMMI maturity model [10]. Agile methods in software development also aim to be highly adaptive. However, they do not encompass adaptation in the user environment.

There is much recent work in the self-adaptive software systems area, conceptualizing adaptation in terms of feedback loops and cycles [11][12]. Requirements have an important role in these adaptive systems [13][14]. These works address adaptation that occurs largely within software systems. Our concern in this paper is adaptation and evolution that involve sociotechnical interactions among business users, the IT organization, and technology systems.

There is extensive work on agility and adaptiveness in enterprises. This includes proposals and studies on strategies and tactics for adaptive enterprises. For instance, Haeckel [15] advocates for a sense-and-respond approach to adaptive enterprise. Lee [16] argues that supply chains should strive to be agile, adaptive, and aligned, and proposes methods to achieve each characteristic in the supply chain domain – e.g., deferring decisions to take advantage of latest market data, as practiced by apparel companies such as H&M and Zara. Agile enterprise ideas and concepts can also be found in the manufacturing industry, e.g., [17], while several studies looked at the role of IT in business agility, e.g., [18]. However, these studies have not considered the mutual iterative adaptation between business and IT.

### III. MODELING THE ONGOING EVOLUTION OF BUSINESS-DRIVEN BI

To examine the link of ongoing change, we consider the current rapid spread of business intelligence technologies. Despite the much touted business benefits of data analytics, the

implementation and adoption of BI has not been straightforward. The needs and requirements of business users are uncertain and evolve in hard-to-predict ways. High expectations and business pressures have created unprecedented demands on IT departments. Industry observers and analysts have noted several rounds of innovation and adaptation in enterprise BI [19][20][21][22]. For our illustration, we draw on Eckerson’s analysis of “business-driven BI” [21], which is based on a survey of 234 BI practitioners, with respondents from business areas as well as IT across multiple industries. Similar experiences have been reported by other observers [22] and by our contacts in industry.

In the following subsections, we will examine a series of sociotechnical configurations in the evolution of business-driven BI. At each stage of evolution, we focus especially on the pressures of change, and how they prompt a transition to a new configuration. We attempt to use models to reveal pertinent issues and to support analysis. From amongst existing modeling techniques, we have elected to use  $i^*$  [23][24], a modeling notation that was created to help analyze information systems in social contexts. Given that each stage of evolution involved technical as well as organizational changes, a social modeling technique such as  $i^*$  might be expected to offer suitable support for analysis. We use the case of business-driven BI to uncover the limitations of the social modeling technique when applied to highly dynamic sociotechnical settings.

#### A. Traditional BI

BI technologies are designed to facilitate organizational decision-making. A *traditional BI* environment is usually implemented with data sourced from a multitude of transactional databases. The data is collected, cleansed and massaged through ETL (Extract, Transform, and Load) processes, then stored in the corporate Data Warehouse (DW). The cleansed data are then used to produce business reports, which become the basis for managers to make decisions in daily operation or for planning.

To address the information needs of Business Users (BUs), the corporate BI team would develop standard reports for monitoring tracked KPIs and to answer pre-determined busi-

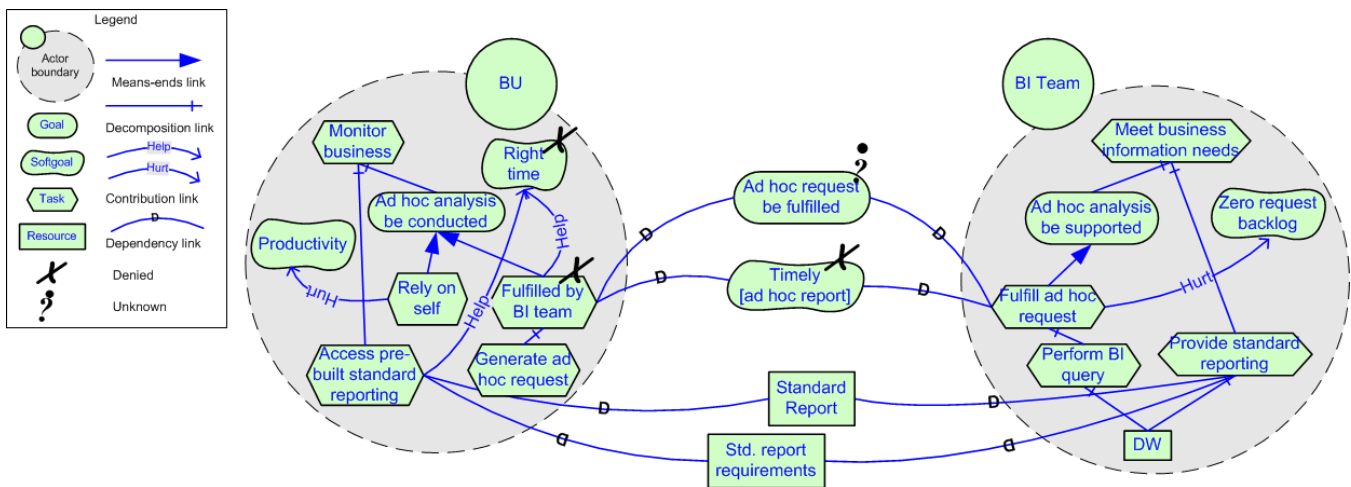


Fig. 1. Social model for traditional BI

ness questions. Pre-built standard reports can be generated by BUs without further help from IT. However, BUs often need additional (ad hoc) reports, which they expect the BI team to deliver promptly so they can have the information they need at the right time. With the explosive growth of data and the perceived value of analytics, BI teams increasingly struggled to deliver ad hoc reports in a timely fashion. BU's activities are impeded by their dependence on the BI team.

To better understand and eventually resolve the problematic situation, one would like to ask: what specifically does BU depend on BI team for? Why is BI team not able to deliver what BU wants? Does BU have other options than to depend on BI team?

$i^*$  is a goal-oriented requirement modeling approach, with special focus on how actors depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished. Within each actor, means-ends links from tasks to goals can be used to answer "why" questions. Multiple tasks for achieving the same goal represent alternative choices available to that actor. Softgoals (or quality goals) serve as criteria for choosing among alternatives. The notion of softgoal in  $i^*$  draws upon the treatment of non-functional requirements (NFRs) in software engineering [25][26]. An actor may also depend on another actor to attain a softgoal.

In traditional BI (Fig. 1), BU depends on BI team for Standard report. To Provide standard reporting, BI team depends on Standard report requirements from BU. (The Arial font is used to refer to objects depicted in the model.) In the BU, Access pre-built standard reporting is only one sub-task in Monitor business. For business questions not covered by standard reporting, BU needs to Conduct ad hoc analysis. One way to achieve this goal is to have it Fulfilled by BI team (right means-ends branch in BU). This task in turn depends on BI team to achieve the goal Ad hoc request be fulfilled. In addition, the Ad hoc report needs to be Timely – a softgoal dependency.

It turns out that this softgoal is not met by BI team (marked by X), leading to goals in BU not being met, thus affecting BU's ability to make timely decisions (Right time). As an alternative to depending on BI team, BU could choose to Rely on self to achieve Ad hoc analysis be conducted, but that would hurt her Productivity.

A social, goal modeling approach as exemplified by  $i^*$  supports the analysis of goal achievement in actors, and factors that contribute to it or impede it. With the aid of the model in Fig. 1, one can see that in traditional BI, the long-established "design precedes access" [20] approach works well for structured and repeated analysis (e.g., monthly sales reports, profitability and financial analysis). However, as business becomes faster paced and uncertainties increase, decisions makers are not able to predetermine what questions they need answered at what times. They increasingly rely on ad hoc analysis to support decision-making under unanticipated conditions. Challenges for the BI team are exacerbated as data volumes increase, data sources multiply, and new types of analytics are introduced. The gap between the rate of changes in analytical requirements and the time for report delivery widens, producing an insurmountable backlog for the BI team. Frustrated users are tempted to bypass the BI team. However, even when

users have the ability to rely on self for ad hoc analysis, their productivity suffers.

To address the unmet goal of timely ad hoc reports, the BI team could look for alternative ways for achieving Ad hoc analysis be supported. Indeed, the BI team creates a new option to achieve this goal in the next approach.

While a goal-based modeling approach can help reveal driving forces for change, such as unmet goals, it is limited in a number of ways. The  $i^*$  model of Fig. 1 is a snapshot and fails to capture the dynamics of the scenarios. Speeds and rates of change cannot be expressed. Thus the bottleneck resulting from BI team's inability to catch up with rapid changes can only be indirectly inferred. Frequencies of occurrences are unclear. For example, the Standard report requirements dependency is infrequent, whereas the Ad hoc report dependency is increasingly frequent.

### B. Moving to Self-Service BI

To address the ad hoc challenge, some organizations adopt self-service BI tools with the aim to resolve problem of meeting BUs' ad hoc needs in a timely way. Such tools have been increasingly embraced by organizations because they allow direct access to business data for the users without the IT departments being the intermediaries. It can be viewed as an attempt to move the design cycle (for ad hoc reports) from the BI team to the report users, thus shortening it. Business users are given the freedom and are expected to rely on themselves, so the BI team can be freed from fulfilling the never-ending stream of ad hoc requests requests..

In terms of modeling, in this iteration of the BI solution, we want to capture the shift of responsibilities for ad hoc report creation from the BI team to the BUs themselves and see how it affects their objectives, including timeliness, and the need for the tool to be both easy and powerful. Is it possible to show that the report design cycle is now shorter?

In Fig. 2, the BI team decides to implement the self-service BI tool (Provide Single Self-Service BI Tool) to help achieve its goal (Ad Hoc Analysis be Supported). At the same time, this new option is hoped to result in Zero Request Backlog (for ad hoc reports). With the self-service tool, what used to be handled by the BI team (Formulate and Execute BI Query) is now in the hands of BUs. Since the users are doing the ad hoc analysis themselves at the time they need it, the obstacle to Right Time is overcome by removing the dependency on others.

However, the single tool solution is unable to satisfy varying user needs. There are actually two types of users who are capable of performing different types of queries, and therefore they require different level of freedom (in terms of reports, tools, and data access). Casual Users (including executives, managers and frontline workers) are domain experts, who use information to do their job. They primarily use standard reports, but increasingly need just-in-time ad hoc analysis to produce decisions within short timeframes. In contrast, Power Users (including data scientists, statisticians and business analysts) are hired to analyze information. Their objective is to explore the potential value of data through creative and iterative

ad hoc analysis. Therefore, they usually have strong analytical aptitude, good tool proficiency and reasonable domain knowledge. We model this challenge in Fig. 2 as two competing tool-related softgoals (Easy to Use and Powerful) failing at the same time. From the requirements perspective, this problem is indicative of the issue of the existence of different usability requirements and of whether such differences can be recognized upfront (i.e., before a solution is implemented).

As we can see, unfortunately, the intended objective is not achieved. A single self-service tool is simply unable to meet BUs' wide-ranging usability needs. The one-size-fits-all solution restricts the ones who want powerful tools (e.g., statisticians running complex queries), while overwhelming the ones who want easy to use tools (e.g., frontline workers running simple queries). This results in a fallback to the old way – asking the BI team to do the ad hoc analysis.

The complexity of the tool on the one hand and its inflexibility on the other are the main barriers to the success of the single self-service tool. This problem arises from our failure to account for several user groups with diverse objectives and capabilities/skills in the design and in the models in this solution. Since tools are usually designed to meet certain usability needs, we argue that how to split the tool user base into user categories is a design decision about granularity. Separating them upfront or delaying the split for a later time may yield different results in terms of addressing the changes in usability requirements. Therefore, it is important to have the modeling capability to represent and reason about design granularity. This includes capturing goals, criteria, and choices that differentiate user groups, product types, time intervals, etc., thus possibly necessitating separate designs for them.

In terms of modeling, we see that  $i^*$  can somewhat show that the ad hoc report design is now “closer” to BUs since the query design is now in their hands, though the dynamic behaviour is still hard to analyze. Cyclical behaviours and differences between cycles (as in III.A and III.B) cannot be shown. User capabilities/skills have not been explicitly modeled in this iteration, but an attempt to account for them is made in the next version of the solution.

In the next iteration, to recognize the different needs of

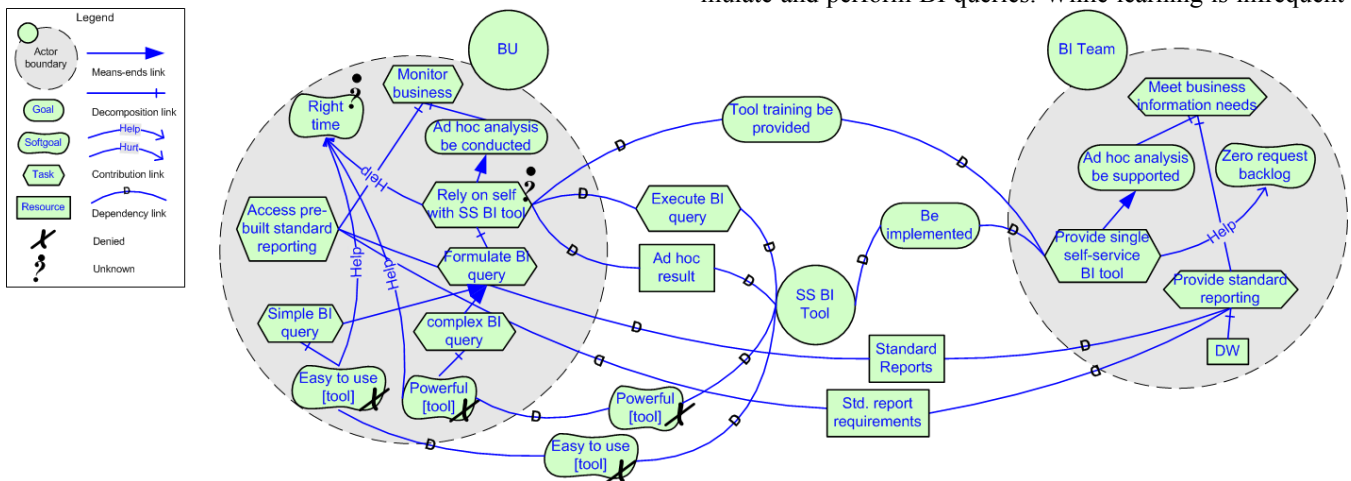


Fig. 2. Social model for the move to self-service

BUs, tailored solutions are created for the two types of users. The majority of business users are considered as Casual Users (CUs) – executives, managers and frontline workers, while data scientists, statisticians and business analysts are categorized as Power Users (PUs). The difference between them is that a PU, hired specifically to do advanced information analysis, typically has a high level of analytical and tool proficiency.

### C. Tailoring for the Casual User

In recognizing the different needs of BUs, a refined approach is to provide solutions that are tailored to the needs of specific user groups. Here, the self-service tool is tailored to CUs to help them compile ad hoc reports.

Because of the focus on CUs, our goal is to produce a model reflecting their point of view. What are the assumptions that we make about these users when designing the ad hoc reporting tool for them? What do CUs want from the tool? Can the design satisfy this user group?

The ad hoc authoring tool is intended for CUs to be Quick to Learn and Easy to Use to compile their own ad hoc reports (see Fig. 3). It has long been understood by the BI team that training is essential for users to adopt a new technical tool. Therefore, the BI team provides ample tool training to CUs when implementing the tool (see the dependency Tool Training be provided and the task within BI Team fulfilling it), expecting that casual users would be able to Learn tool, thus enabling them to formulate and execute BI queries by themselves.

Unfortunately, adopting the authoring tool turns out to be difficult. Fig. 3 helps us visualize several obstacles to tool adoption inside the CU. First, the learning curve is insurmountable for many casual users. Even with ample training, learning the tool is harder than what they expected (with the softgoal Quick to learn failing in CU). The longer the learning, the slower the user adopts new tool features. The slow adoption then becomes a barrier to get quick answers (Right time) for frequently changing business questions. In addition, without proper motivation and incentives, CU might lack interest in spending time and effort to learn the tool.

Second, CUs must Learn Tool before they are able to formulate and perform BI queries. While learning is infrequent (a

one-time effort for a particular feature), using the tool to perform queries happen frequently. The users still do not think the authoring tool is easy enough for their use. Therefore, the soft-goal of having an Easy to use tool fails, having a negative effect on getting results in short amount of time and continuing to learn the tool.

Furthermore, to be able to Formulate BI Query, one needs to be analytically inclined. While tool knowledge can be acquired and tool skills can be trained, Analytical Aptitude is what many casual users lack (the resource Analytical Aptitude fails), which requires a longer time to develop. Often this can be the real reason why CUs do not succeed in running queries themselves and thus ask for BI team's help.

In  $i^*$ , there are distinctions between physical actors (agents) and logical actors (roles). A capability is a property of an agent. The challenge here is that physical agents may fail to possess or to acquire the skills required for playing logical roles. As illustrated by Analytical Aptitude, we model human capabilities as resources within the actor boundary. Failure to obtain such a resource indicates the unavailability of the corresponding capability.

This raises the questions of what is considered to be Easy to Use for the tool and what amount of time spent on learning it is Quick to Learn? As user capabilities (knowledge and skills) improve with learning and experience, the acceptable easiness and quickness of learning are changing as well. Thus, in addition to the changing information needs of users, another important ongoing change is user capabilities, which the BI team needs to monitor so as to adapt the tools accordingly.

In the analytical environment, tools for business users are designed to meet information needs rather than functional needs. As the needs change, the functionality required to deliver the right information also changes based on the question to be answered, the data sets involved and their respective environments, and the analysis tasks to be performed. Thus, due to the variety of relevant uncertainties listed above, addressing changing information needs requires flexible solutions. In addition, one needs to deal with the evolution in both tool the functionality and human capabilities (i.e., with their co-evolution). While requirements engineering approaches in software engi-

neering and business process management are traditionally focused on abstract actors, requirements methods prevalent in human computer interaction, user-centered design, and user experience concentrate on agents. We argue for the need to integrate both perspectives to deal with changing and evolving requirements in sociotechnical systems.

In this solution, the tool adoption is not as easy as anticipated. CUs feel that the tool is not simple to use and takes a lot of effort to learn. Moreover, many lack the analytical aptitude needed to perform BI queries (e.g., some CUs do not know how to select the appropriate data elements for their analysis or how to interpret the data), which points to the overly optimistic assumptions about CUs' skills embedded in the design. However, users capabilities are not static. As their knowledge and skills evolve (e.g., through learning and tool use), the users may be ready and willing to use more advanced features. This indicates the need for continuous analysis of user needs and capabilities, another important ongoing change in the BI environment.

In this iteration of the approach, we see that while certain  $i^*$  features, such as the ability to model requirements from the point of view of a particular user group, to represent both roles and concrete agents, and to capture the required user capabilities, are extremely useful,  $i^*$  again lacks the capacity to model the gradual learning and improvement of user skills as well as the evolving tool functionality to match those skills.

#### D. Leveraging the Super User

Super Users (SUs) are leveraged in the next iteration to alleviate the problems that plagued the previous solution – the challenge of CUs' tool adoption. A SU is a CU who has good technical skills and has become the departmental go-to-person for tailored information. The BI team provides the ad hoc report authoring tool to SUs rather than CUs. Instead, CUs are given interactive standard reporting tools, which allow them to personalize their views to some degree. A configurable tool is used to enable on-demand feature provisioning (for both CUs and SUs) to deal with the uncertainty and evolution in user capabilities and demands.

The approach has multiple levels of design to meet the

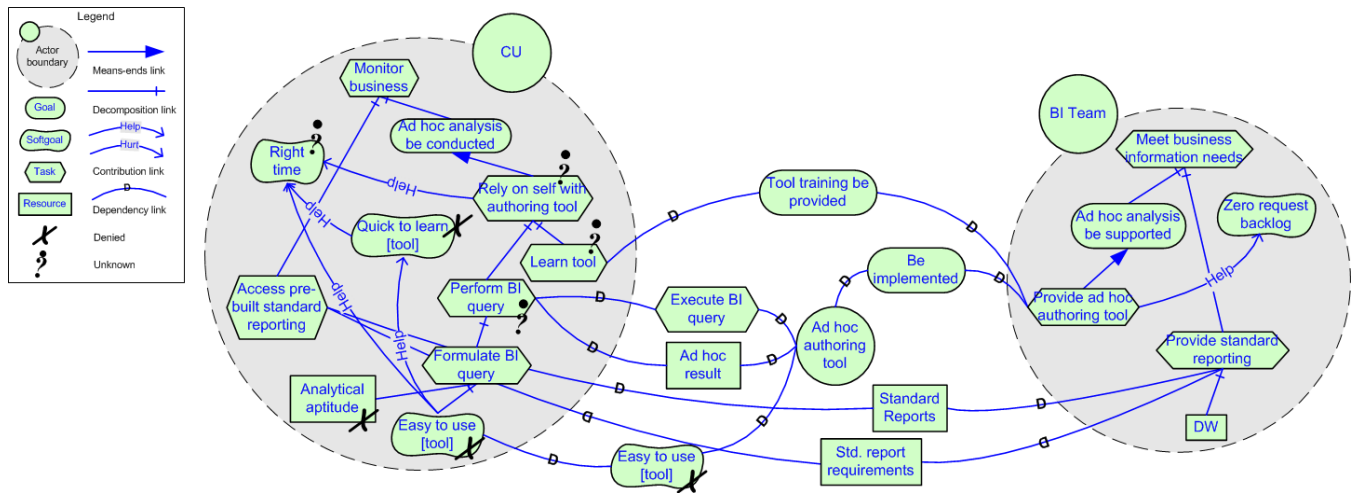


Fig. 3. Social model for the tailoring approach for the casual user

timeliness objective. CUs can tailor their information views on their own quickly and easily through interactive standard reporting. Then SUs can help them with the ad hoc analysis when it goes beyond simple customizations. The BI team publishes new standard (departmental or corporate) reports on a regular basis as the report requirements continue to evolve.

The model produced for this iteration should be able to help verify that with appropriate assumptions about their skills, SUs are, in fact, able to deliver ad hoc reports to CUs on time. Moreover, we want to be able to analyze if SUs, who are business users themselves, are motivated enough to help CUs in addition to fulfilling their own objectives and if the sense of belonging to the BI team can be fostered in them. Similarly, the model should help determine if this 3-tier BI organization works in terms of meeting its objectives. Finally, can the multiple report design and tool improvement cycles be made to work in the new approach?

As shown in Fig. 4, the BI team provides the ad hoc report authoring tool to SUs rather than CUs, and CUs now depend on SUs to deliver timely ad hoc reports. A SU represents a social component added to the technical solution. Unlike the previous approaches, the new intermediary is again placed between the user and the data. The design cycle for ad hoc reports is now moved to a SU. The dramatic reconfiguration of actor relationships is clearly visible in the social model of Fig.

4.

This change in the social landscape is to overcome the CUs' lack of personal interest or tools skills that caused problems in the previous approach. Compared to a CU, a SU is more tech savvy and analytically inclined (the Technical Skills and Analytical Aptitude resources in SU, resp.), while gravitating to using the BI tools. They quickly become proficient with the report authoring tool. Moreover, by fulfilling ad hoc requests for other casual users in the department, SUs are known for their Tool Expertise which makes them feel Empowered. Instead, CUs are given interactive standard reporting (Standard Reports and Report Interactivity dependencies between CU and the BI team), which allows them to personalize their views.

The On-demand Tool Feature dependency is fulfilled by the BI team both for CUs and for SUs. It offers the flexibility needed to deal with changes in user capabilities and needs. The sensing mechanism is implemented through actor dependencies. In order to fully understand the users' needs (Accurate Requirements in BI team), the BI team not only Gathers Input from the users, but also Senses and Analyzes the User Behaviour. So, the features can be adjusted accordingly (not overwhelming the user and yet not limiting him).

This approach has multiple levels of iterative design or adaptation cycles to meet the required responsiveness (Right

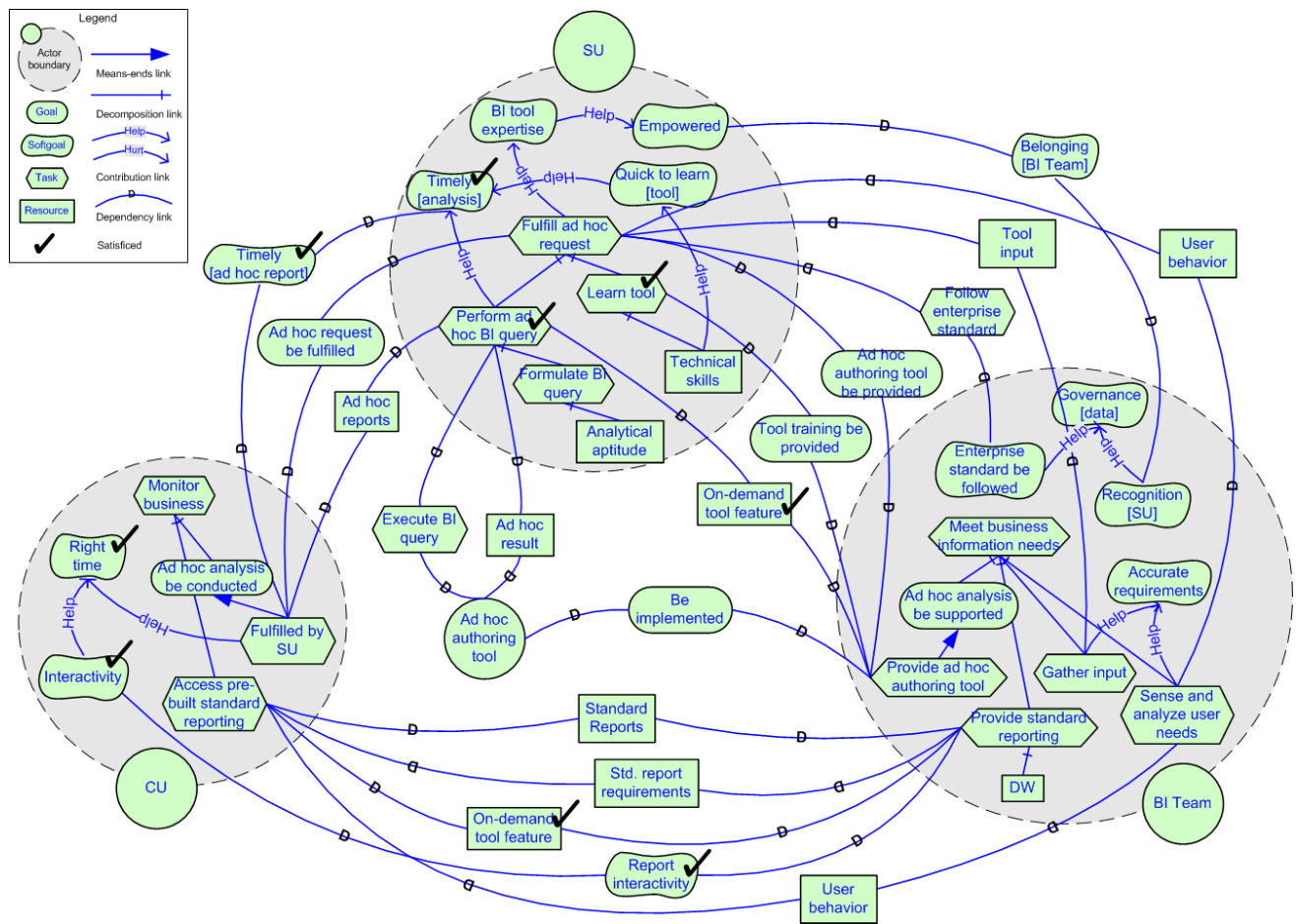


Fig. 4. Social model for leveraging the super user

time in CU). CUs can tailor analysis according to their changing information needs quickly and easily through interactive reports (e.g., predefined drill-down, personalization of colour and font). It achieves timeliness through a nearly instantaneous adaptation cycle. When changes beyond simple customizations are needed, they are deferred to SUs, who have the tool and the skills to support a fast adaptation cycle for per-user changes. Standard reports that are meant to serve a large user base (departmental or corporate), run on a relatively longer design cycle. However, the social model cannot show these levels easily as well as express how they are related.

While this approach seems to be solving the prominent issues that emerged in the analytical environment, there are potential alignment problems that should be handled carefully. How does one ensure that personal interests of the user are aligned with the new SU role when it is created? How do we properly align SUs to the interest of the BI team (i.e., have them Follow the Enterprise Standard)? Incentives need to be created to prevent misalignment between the physical agent and the role it is playing, and between different actors in the BI landscape. E.g., some BI managers consider incorporating SUs into the BI team (the Belonging dependency between the SU and the BI team) by recognizing their value, which will make them feel Empowered (in SU) and therefore better align their work with the corporate standards.

The above analysis shows that, as expected, the model in Fig. 4 can capture the dramatic change in the social landscape of the solution, with the lighter load on CUs and heavy involvement of SUs. It also supports the analysis of whether personal goals of SUs (e.g., Empowered) can be met and how. Similarly, SUs' skills and their effect on SUs' objectives and the success of the approach can be explicitly represented. However, while some elements of design cycles can be modeled (e.g., sensing as resource dependencies), the relative length and other properties of these cycles are hard to analyze.

#### E. Catering to and Reining in the Power Users

While lacking space to present models for further iterations of the approach, here we discuss further developments and challenges in BDBI. The initial tailored solution for PUs is to address the problem that one-size-fits-all BI tool constraints PUs' analytical work too much. Replacing the single self-service tool, an analytical toolbox is given to PUs to support their creative data exploration. Power users can now select tools suitable for the analysis they want to do.

Although this approach gives PUs the tool flexibility, two challenging issues remain. To run comprehensive data analysis, a PU also needs unconstrained data access. The as-is way to access data (i.e., sending query to the BI team or creating PU's own offline data sets) limits the possibility for iterative analysis and results in data maintenance issues. The isolated data shadow systems, indeed, jeopardize data consistency and make data governance more difficult. In addition, due to limited computing resources on desktop machines and local servers, PUs often experience poor system performance for their complex queries.

The refined approach for PUs is to deploy the analytical sandbox environment to fully address their needs. Sandbox is a

powerful environment where PUs can merge and manipulate data efficiently. By centralizing the analysis, the solution is intended to satisfy both the PUs and the BI team, and to solve the problems that remain in the distributed environment.

The sandbox solution makes a PU a first class citizen in the corporate BI environment. The sandbox's powerful computation capability dramatically improves query performance. Easy access to data is achieved through online data access and the ability to upload and merge local data. As a result, the maintenance difficulties of offline data sets are alleviated. In turn, it helps the BI team achieve data governance and improve data consistency. In addition, the centralized analytical environment helps to promote collaboration and reuse between PUs.

The business-driven BI setting is highly dynamic and yet still emerging. To unfold and to resolve the complexities illustrated above, organizations need to experiment with different business intelligence solutions. They are iterative adaptation processes with co-evolution between the technical systems and the human systems. The solutions are not limited by the illustrated approaches, as uncertainty and ongoing changes still reign. E.g., is the new role, the SU, well aligned with the interests of others? How does one support a PU's creative data exploration when a complex analytical job is run differently every time and the data needed to answer unexpected questions is not yet in the system?

The challenges we see here are not unique to the BI setting. In recent and ongoing developments, like the spread of web-based IS, enterprise 2.0 and social media, document to content management, enterprise search, mobility, BYOD (Bring Your Own Device), as well as infrastructure changes such as cloud, SOA (Service-Oriented Architecture), uncertain and evolving requirements are a pressing challenge to be answered by the IS engineering community.

Overall, the  $i^*$  notation showed its suitability for capturing and analyzing the social aspects of solutions, including actors with their objectives and skills assumptions, networks of dependencies among them. Goals that are not met, and thus the driving forces for change, can also be modeled. However, these models are snapshots and cannot express the dynamics of solutions or convey cyclical, recurring patterns of handling change.

## IV. MODELING TEMPORAL AND ITERATIVE ASPECTS

As we have seen in Section III,  $i^*$  models are not able show temporal change (even though they can help reason about the motives and forces that drive change), and therefore adaptation loops and design cycles. Here, we look at the multiple layers of change in dynamics environments and, while noting its limitations, attempt to use the BPMN notation with extensions to model multiple change processes.

### A. Multiple Levels of Adaptation Processes

The  $i^*$  social models presented in Section III are able to capture actor intentions and relationships. However, they cannot show any dynamics. In particular, one cannot recognize that there are changes that take place within different time frames or scales. For example, the main BI activity of monitoring business performance that leads to a management action

occurs at a different time frame than the transactional activity that it is monitoring. The design and redesign of tools to support the BI activity occurs at yet another (less frequent) time frame. To depict and better understand these different processes and their relationships we look to process modeling.

The process model (see Fig. 5) shows these related processes, each operating within its own time cycle. Sales transactions (Operational Process) can be started anywhere from once every few seconds to once every few minutes. While monitoring the sales performance (Business Monitoring), a BU can have a sense of the tracked KPI values within minutes by looking at dashboards. However, it may take more time (hours) for him to find out why sales in certain areas have declined or what products are projected to be the most profitable in the next quarter. Scheduled Standard Reports provide answers to known business questions. In case when answers to new questions are needed, business users often rely on Ad Hoc Analysis to find answers for particular queries.

How do these processes interact with each other? What is the nature of the flows between them? From the story in Sec-

tion III, we note that Business Monitoring is a change process with respect to the Operational Process. The output data from the Sell Products Online activity is, in fact, a sensing operation that extracts information (via BI infrastructure such as a data warehouse) about the Sell Products Online activity in order to improve it. This information gets compiled into reports and is used by Business Monitoring. The result of the change process is the Managerial Action that alters the way Sell Products Online operates. For the purpose of enterprise adaptiveness, it is important for the analyst to note the existence of such a monitoring and change process for Sell Products Online so that the effectiveness of this feedback adaptation loop can be evaluated, e.g., regarding timeliness, frequencies, accuracy, effectiveness of the managerial action, etc.

The Managerial Action flow is not a normal transactional data input transformed by every instance of the Sell Products Online process. Rather, it is a “control” input, which prescribes or modifies how the Sell Products Online process operates. It varies infrequently compared to the transaction frequency of the operational process.

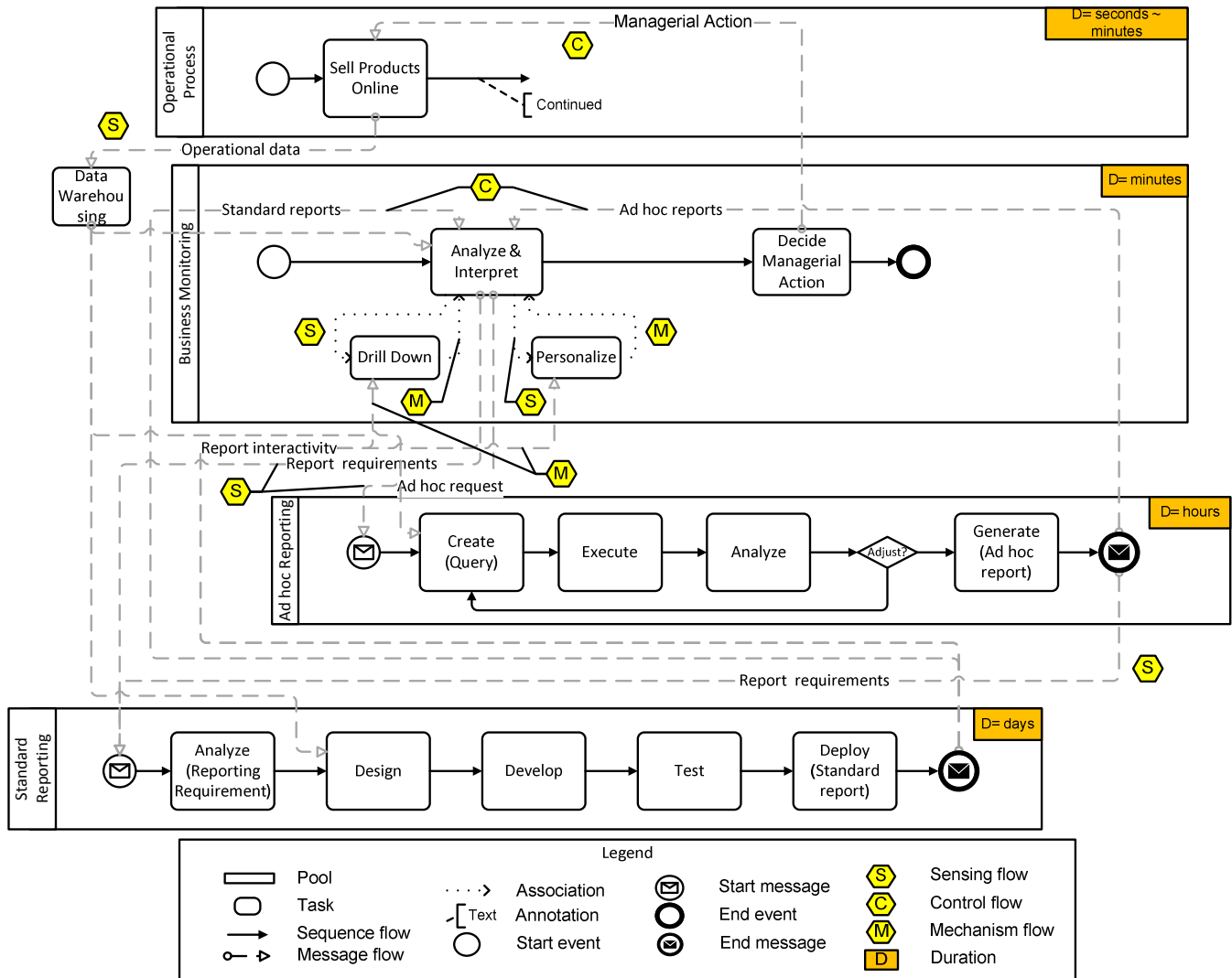


Fig. 5. Modeling sensing, control and mechanism flows in multiple design cycles



In order to interpret business performance properly, business users drill down or personalize reports as they see fit (sensing operation). The outputs of these activities do not directly “control” the process (Analyze and Interpret) that they aim to improve. Rather, they produce better tools to enable or support their target processes. The tool here helps tailor the information by changing its scope or format. Similarly, these features are enabled by the Report Interactivity capability.

While in the model of Fig. 5, we attempted to employ the BPMN notation as it is widely adopted today, to convey the relationships among the different kinds of change processes, we encountered the need for adaptations and extensions. Thus, we have applied some annotations to differentiate the newly identified flows from the normal inputs and outputs. A sensing operation (e.g., Operational Data) is a message flow with the <S> annotation, showing as an output from the bottom of the activity. Control (e.g., Managerial Action) is represented as a message flow adorned with the <C> annotation, showing as an input to the top of the activity. And the <M> annotation is used to represent supporting capabilities, a mechanism (an input to the bottom of the activity, e.g. Report Interactivity). The terminology of “control” and “mechanism” flows are borrowed from the IDEF0 language [27]. In this paper, BPMN was selected over IDEF0 due to the former being a newer, more widely used and a more flexible notation.

The Report Requirements and Ad Hoc Request are annotated as sensing outputs, as they provide inputs to the change processes (Standard Reporting and Ad Hoc Reporting resp.), which will deliver new reports for the next use cycle. Reports provide a way to look at available information (upon some processing), which is something different from the raw data itself. Hence, we consider reports (Standard Reports and Ad Hoc Reports) as control inputs that limit users in the way they can analyze and interpret the information.

Pools, which can represent abstract roles in BPMN, are conveniently used to show a multiplicity of processes that operate relatively independently of each other. Each of the processes is provided with a typical duration to provide a rough sense of their relative timing. The types of inter-process connections rather than sequence flows are of particular interest to us. In addition to annotating the different inputs and outputs, we expand the use of a “message” flow between pools (BPMN concept) to represent artifacts and capabilities (e.g., requirements, reports, and tools).

### B. Adaptation Loops and Feedback

By differentiating control and mechanism inputs from normal data inputs as well as sensing from normal outputs, we are able to locate adaptive loops (<S>-<C> loop and <S>-<M> loop). When analyzing these loops, it is not the iterative repetitive executions of the same activities that are of concern, but the fact that there is information flow back to the next iteration.

Adaptive loops reveal special relationships between processes. A higher-order process is considered to be a design (change) process with respect to its lower-order processes (the use or execution process). It senses how well its lower level process operates and may change the way it operates. The change is either through a “control” flow constraining the pos-

sible options for the target process at runtime or through a “mechanism” flow that changes the space of options for its target process by creating new capabilities. Hence, there is a hierarchy among processes that reflects their relative control order.

Such a hierarchy is illustrated in Fig. 6, where we draw horizontal boundaries between processes to operate at different design levels. The collapsed process element from BPMN is used to represent a “collapsed” pool (i.e., with its internal details hidden). Thus, we can focus on the relationships between them.

The Operational Process is improved by managerial actions from Business Monitoring. Standard Reporting provides standard reports to support Business Monitoring and to help answer known/anticipated questions. When this basic analysis is not sufficient, Ad Hoc Reporting is triggered to provide ad hoc analysis to address additional information needs. However, ad hoc reports can only leverage existing data. Hence, if the business question requires new data to be analyzed, Data Exploration is then triggered to meet the need. While Tool Development builds the analytical infrastructure and tools (leveraged by the other processes), Feature Provisioning aims to provide on-demand features based on user capabilities and needs.

In order to identify these loop patterns, we have adopted a directional convention for the inputs and outputs similar to IDEF0 (as shown in the legend of Fig. 6). At the top right, process B (Business Monitoring) and process A (Operational Process) form a sense-and-control loop. Process B is at a higher level as it is a design process for A (boundary 1). Moving downward, both process D (Standard Reporting) and process C (Ad Hoc Reporting) are report design processes for process B. Therefore, a level boundary (boundary 2) is placed between processes C and B, meaning that process B is a lower-level process with respect to processes D and C. Yet, process E (Data Exploration) sources new data and evolves the semantic layer for C. So, process E is a higher-level process with respect to C (boundary 3). At the bottom left, another boundary (boundary 4) is drawn between process F and those above it. Processes F (Feature Provisioning) and G (Tool Development) produce analytical tools for several processes above them in the model. The two operate at different time frames, as process G runs on a much longer cycle (months) than process F (hours to days). Similarly, process C (hours) is intended to deliver more frequent and faster results than D (days).

We have introduced levels in Fig. 6 to help recognize the design relationship that exists between processes. Its visualization is aided by distinguishing control (arrow at the top of process elements) and mechanism (arrow entering at the bottom) inputs and sensing outputs (arrow exiting from the bottom) from regular inputs (arrow entering from left) and outputs (arrow exiting from the right). The need for a design activity that is distinct from the execution of a process arises when a change to the process cannot be readily accomplished at “runtime”. However, if runtime adaptation capabilities are built-in (e.g., certain options and/or configurations), then adaptation can occur within the same level, without going through another design cycle, thus achieving better agility and responsiveness. For example, a CU is able to drill down and personalize prede-

financed reports (in Fig. 5), which is a self-adaptive loop to help tailor information analysis at runtime.

## V. DISCUSSION

In this section, we discuss our experiences in attempting to

apply social and process modeling notations to capture and analyze a complex and dynamic problem facing many enterprises today, namely the problem of creating a sociotechnical solution for business intelligence that is adaptive, resilient, effective, efficient, timely, flexible, user friendly and empowering for all the concerned roles. Below we identify some aspects

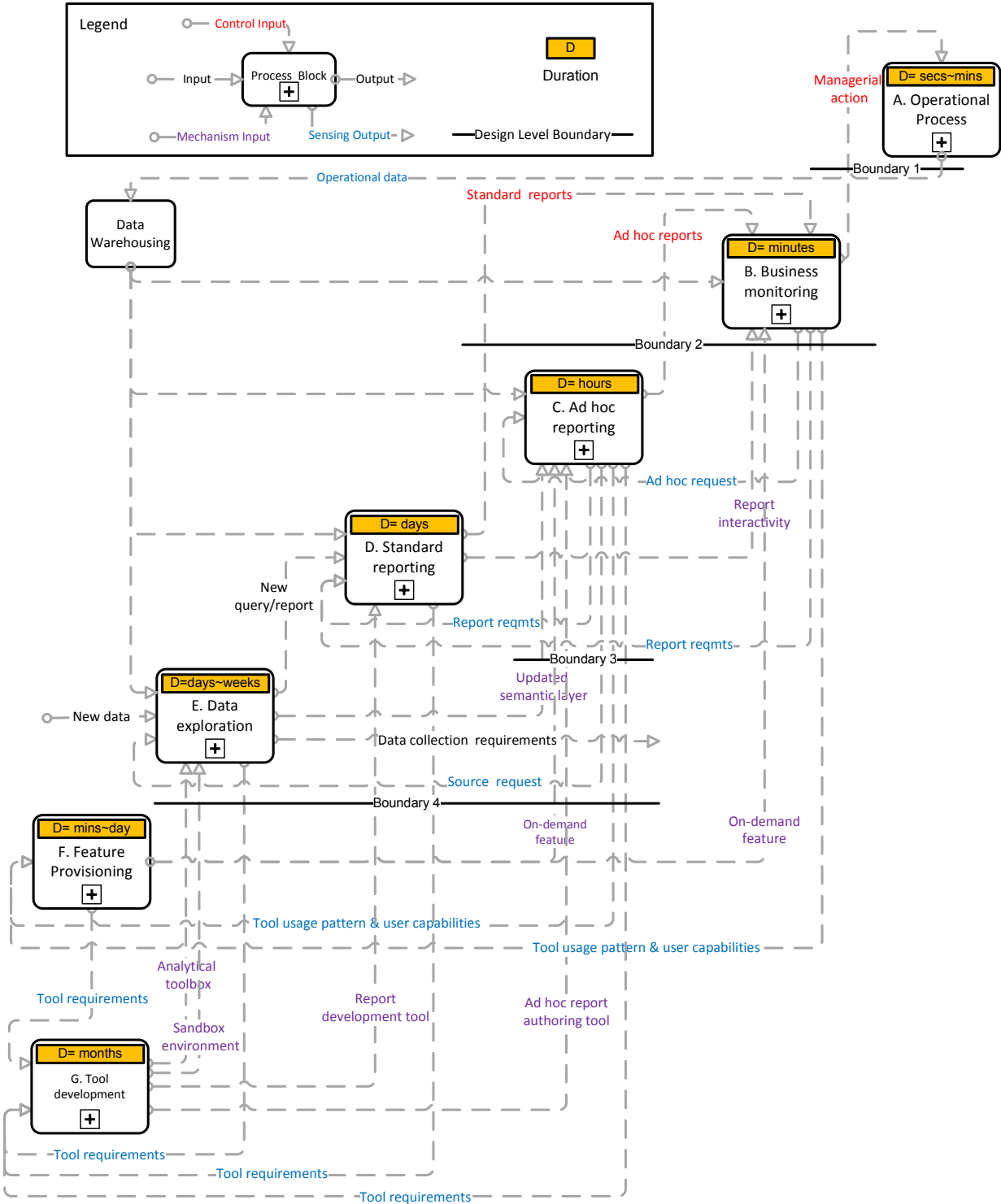


Fig. 6. Modeling multiple design cycles in the BI analytical environment (extended BPMN model)

of modeling and analysis that we found important to support when investigating adaptive and flexible enterprise designs.

**Variability modeling and binding.** Support for modeling and analysis of intentional variability (the variability in the way system objectives can be met), the criteria for selecting among the available choices, as well the implementation of behavioural variability is important for designing flexible, robust, adaptable/adaptive organizations and systems.  $i^*$  is an intentional modeling notation well capable of supporting means-ends analysis of goals, while explicitly representing the corresponding selection criteria. Process modeling with BPMN is capable of capturing varying behaviour and flows, while lacking the ability to model quality criteria for choosing the appropriate configuration.

Further influencing or limiting alternatives selection are constraints and barriers – requirements in terms of resources, skills, capabilities, production/computing capacity, inflexibilities due to the legal and other constraints associated with particular variants. Analysis of these barriers is thus important as they influence enterprise flexibility/adaptivity and can trigger processes aimed at overcoming the barriers (e.g., activities aimed at learning the reporting tool in our case study). Certain kinds of barriers can be modeled as resources in  $i^*$  models (e.g., the SU's analytical aptitude in Fig. 4), but a more detailed representation of these constraints in  $i^*$  as well as in process models needs to be explored.

**Social modeling.** Business enterprises are social systems and therefore, to support flexibility in the face of changing and unknown requirements, one necessarily needs to look at socio-technical solutions to enterprise objectives.  $i^*$  is a social modeling notation that has extensive capabilities to capture and analyze social aspects of enterprises. Crucial aspects of such analysis include means-ends analysis done from the point of view of individual actors, with actor-specific quality criteria (NFRs) guiding the selection of suitable alternatives, the ability to explore alternative dependency configurations and responsibility assignments (e.g., the responsibility for developing ad hoc reports in our example) and the ability to distinguish logical and physical actors and thus to model and determine if the skills of individual agents match the requirements of the roles they have to play (as illustrated by our case study) – otherwise this becomes a barrier to change. In addition, social alignment issues, such as conflicting actor goals and conflicts involving personal goals of agents and the goals of the roles those agents are playing can be analyzed. In our case study, making sure the SUs feel empowered in their positions is an attempt to align their individual interests with those of the enterprise. However, certain aspects, such as the gradual increase in the level of report tool usage skills of BUs are not easy to model in  $i^*$  due to their temporal properties.

**Multiple levels of design.** One of the important lessons learned from the BI case study is that in dynamic domains with unpredictably evolving requirements design decisions should remain open for possible revisions, while new solutions should be able to be added to the set of existing ones (which is, for example, supported in  $i^*$  via the openness of its means-ends decomposition). Thus, the overall space of configurations includes not only the ones currently implemented and enabled at

runtime, but also the configurations that result from doing some appropriate level of enterprise redesign. For example, when the custom report creation bottleneck is identified in the original BI scenario, a significant enterprise redesign ensues, with new tools and/or actors being introduced. Overall, we have a hierarchy of design levels, with a particular level using the design provided to it by a higher-up design level, while itself making design decisions to be used by the lower design level. Different design levels usually operate on different time cycles (e.g., the frequent design of ad hoc reports by SUs vs. the rare provisioning of on-demand tool features by the BI team, as in Fig. 4). Moreover, different design levels have different decision-making criteria, which we are able to capture in  $i^*$  models. When handling change, adaptation can happen both within the same design level (i.e., through switching to an already implemented alternative) or across multiple design levels, reflecting the situation where the variability within the original design level is not sufficient to successfully handle the change (e.g., when none of the standard reports is able to handle the new data analysis requirement, thus needing an ad hoc report).

**Feedback.** An enterprise operating in a dynamic, unpredictably evolving domain, needs to adapt to the changing environment (context), evolving requirements, failures to achieve its objectives (or opportunities to improve on them). This requires sense-and-response behaviour, one example of which is a feedback loop that is illustrated in the BI case study. A feedback loop consists of a controller and a target system. The controller monitors its target system and can modify it if the monitored output deviates from the expected value. A modification of a target system by its controller represents adaptation. Thus, feedback loops feed information to the controller in order to improve the next iteration cycle of their target systems. In annotated process models, we are able to capture feedback loop details through the use of <M>, <S>, and <C> flows (with control flow roughly corresponding to adaptation, while mechanism flow amounts to evolution). Feedback loops can operate within a single design level, with that level exemplifying a self-adaptive system, or across multiple design levels. To analyze enterprise adaptiveness, one needs to map the paths of various (often nested) adaptive loops criss-crossing the organization to determine whether they are achieving the desired adaptive behaviour. Positive and negative feedback should be analyzed. Dynamic systems analysis techniques may be relevant (e.g., time-domain response, rates and frequencies of change). Since an adaptive enterprise will encounter transformational and disruptive change, the modeling framework will need to encompass nonlinearities. A proposal in [28] identifies awareness requirements (requirements about the success/failure of other requirements, as in “Ad Hoc Analysis be Conducted should never fail”) as the requirements that feedback loops are designed to achieve. Capturing these requirements can potentially support the integration of multiple, possibly conflicting feedback loops at the intentional level. As illustrated in [28], appropriate annotations will be needed to be added to  $i^*$  to support this type of requirements.

**Modeling temporal and dynamic aspects.** Since  $i^*$  has poor support for quantities, frequencies and temporal progression, one cannot distinguish between failures on a per-instance

transactional level, or an inherently invalidated option. Appropriately changed process models, as shown in the paper, can be used to capture process durations and their frequencies. Similarly, the rate of change and the length of the corresponding design cycle can be shown, leading to the possibility of bottleneck identification and improved coordination. To avoid overloading the notation, we have not distinguished between control/adaptation that is applied to a particular process instance (i.e., at the instance level only) or to all process instances hence (i.e., type level). For more detailed analysis of the temporal or real-time properties of various solutions, specialized (e.g., queuing) models can be utilized. Unlike  $i^*$ , process models, with their support for events and messages, can be used to capture change triggers and specify change propagation within the enterprise. Still, intentional models are invaluable in representing variability in handling change and the rationale for selecting particular alternatives.

## VI. CONCLUSION

As the pace of change accelerates and complexity increases, there is a need for new modeling techniques to support requirements activities in the new setting. In the case of recent developments in BI adoption, we found that the co-evolution of the social and the technical can be modeled and analyzed to some degree using  $i^*$  modeling. However since  $i^*$  cannot convey dynamics, a series of models had to be used, each representing a snapshot in time. To complement the  $i^*$  social goal models, we used process modeling with some extensions to depict multiple levels of design cycles. A number of modeling challenges and issues were uncovered as a result of the analysis, suggesting the need for a more comprehensive rethinking of requirements modeling techniques in the new setting. These new modeling techniques will be an important basis for a model-based approach to adaptive enterprise architecture [29].

Limitations of our study include use of scenarios from a single domain setting, and the application of only two types of modeling techniques. We expect these limitations to be addressed in future work.

**Acknowledgement.** This research was supported by the NSERC (Natural Sciences and Engineering Research Council of Canada) Business Intelligence Network.

## REFERENCES

- [1] M. Jarke, P. Loucopoulos, K. Lyytinen, J. Mylopoulos, and W. Robinson, "The brave new world of design requirements," *Inf. Syst.* vol. 36(7), pp. 992-1008, November 2011.
- [2] J. Mylopoulos, "Information modeling in the time of the revolution", *Inf. Syst.*, 23(3-4):127-155, 1998.
- [3] S. Nurcan, C. Salinesi, C. Souveyet, and J. Ralyté (Eds.), *Intentional perspectives on information systems engineering*, Springer, 2010.
- [4] T.H. Davenport, J.G. Harris, and R. Morison, *Analytics at Work: Smarter Decisions, Better Results*, Harvard Business Press, 2010.
- [5] J. Manyika, et. al., "Big data: the next frontier for innovation, competition, and productivity," McKinsey Global Institute, pp. 1-137, 2011.

- [6] Business Process Model and Notation (BPMN) Version 2.0, <http://www.omg.org/spec/BPMN/2.0/PDF/>
- [7] C. Rolland, C. Salinesi, and A. Etien, "Eliciting gaps in requirements change," *Requirements Engineering*, 9(1):1-15, 2004.
- [8] N. Nurmiliani, D. Zowghi, and S. Powell, "Analysis of requirements volatility during software development life cycle", *Software Engineering Conference Proceedings*, 2004.
- [9] N. Madhavji, J. Fernandez-Ramil, and D. Perry, *Software Evolution And Feedback: Theory And Practice*, John Wiley & Sons, 2006.
- [10] CMMI Version 1.3 Information Center, Software Engineering Institute, <http://cmminstitute.com/cmmi-solutions/cmmi-version-1-3-information-center/>
- [11] J.O. Kephart, and D.M. Chess, "The vision of autonomic computing," *IEEE Computer*, vol. 36(1), pp. 41—50, 2003.
- [12] B.H.C. Cheng, R. de Lemos, H. Giese, P. Inverardi, J. Magee, et al., "Software engineering for self-adaptive systems: a research roadmap," In *Software Engineering for Self-Adaptive Systems*, LNCS Vol. 5525, pp. 1–26, Springer-Verlag, Berlin Heidelberg, 2009.
- [13] P. Sawyer, N. Bencomo, J. Whittle, E. Letier, and A. Finkelstein, "Requirements-aware systems: a research agenda for re for self-adaptive systems," 18th IEEE International Requirements Engineering Conference, pp. 95–103, 2010.
- [14] V. Souza, A. Lapouchnian, and J. Mylopoulos, "Evolution requirements for adaptive systems," In *Proc. SEAMS 2012*, Zurich, Switzerland, June 2012.
- [15] S.H. Haeckel, *Adaptive Enterprise: Creating And Leading Sense-And-Respond Organizations*, Harvard Business Press, 1999.
- [16] H.L. Lee, "The triple-a supply chain," *Harvard Business Review*, pp. 102-112, October 2004.
- [17] R. Dove, *Response Ability - The Language, Structure, and Culture of the Agile Enterprise*, Wiley, New York, 2001.
- [18] L. Mathiassen, and J. Pries-Heje, "Business agility and diffusion of information technology," *Eur. J. Inf. Syst.*, 15(2):116-119, 2006.
- [19] W. Eckerson, "Analytics in the era of big data: exploring a vast new ecosystem," *TechTarget*, 2012.
- [20] W. Eckerson, "Big data and its impact on data warehousing," *TechTarget*, 2012.
- [21] W. Eckerson, "Business-driven bi using new technologies to foster self-service access to insights," *TechTarget*, 2012.
- [22] B. Evelson, "The Forrester wave: self-service business intelligence platforms, Q2, 2012", *Forrester Research*, 2012.
- [23] E. Yu, P. Giorgini, N. Maiden, and J. Mylopoulos, *Social Modeling for Requirements Engineering*, MIT Press, 2011.
- [24] E. Yu, *Social Modeling and  $i^*$* , In Borgida et al, eds. *Conceptual Modeling - Foundations and Applications*, Springer, 2009.
- [25] L. Chung, B. Nixon, J. Mylopoulos, and E. Yu, *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishing, 2000.
- [26] J. Horkoff, E. Yu, "Interactive analysis of agent-goal models in enterprise modeling", *Int. J. Info. System Modeling & Design*, 1(4):1-23, 2010.
- [27] *Integration Definition for Function Modeling (IDEF0)*, <http://www.idef.com/pdf/idef0.pdf>.
- [28] V. Souza, A. Lapouchnian, W.N. Robinson, and J. Mylopoulos. "Awareness requirements for adaptive systems," In *Proc. 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2011)*, Honolulu, Hawaii, USA, May 2011.
- [29] E. Yu, S. Deng, and D. Sasmal, "Enterprise architecture for the adaptive enterprise - a vision paper," *Workshop on Trends in Enterprise Architecture Research*, Open Group Conference, Barcelona, Spain, October 2012.