# Bringing Model-based Systems Engineering Capabilities to Project Management: an Application to PRINCE2

Diana Coppola[1], Andrea D'Ambrogio[2], and Daniele Gianni[1]

[1] Dept. of Applied Sciences and Technologies
Guglielmo Marconi University
Rome, Italy
dianacoppola@virgilio.it, d.gianni@unimarconi.it

[2] Dept. of Enterprise Engineering
University of Rome Tor Vergata
Rome, Italy
dambro@uniroma2.it

*Abstract*—**PRINCE2 is arguably one of the most adopted process-based methods for project management. Currently, PRINCE2 is defined in a textual specification, which describes the principles, the themes, and the processes that project managers should apply in their management activities. Although the specification is well structured and mature, the specification does not provide a browsable digital representation that can be interactively used for learning and/or for the specification application during project management activities. This paper aims to overcome these limitations with the application of a model-based systems engineering approach to represent the PRINCE2 specification in a model-based format. This can bring several benefits to the specification, including the availability of a graphical, comprehensive and digitally browsable visualization of the PRINCE2 processes, their inputs/outputs, and the constituting tasks. The model-based format has been obtained by a top-down mapping of the PRINCE2 specifications, beginning with the process architecture in IDEF0 down to the individual tasks, roles, and tools in BPMN 2.0. Besides supporting PRINCE2 understanding and application, the model-based format can also serve as a baseline for further exploitations, such as consistency verification of the PRINCE2 specification and model-based process simulation for the governance of the PRINCE2 processes and of the project management activities.**

***Keywords—project management; business process; model-based; systems engineering.***

## I. INTRODUCTION

Project management methodologies have been increasingly adopted to structure, monitor, control, and execute temporary and cross-functional organizational activities in order to support managers with the information, tools, and processes to increase predictability in the delivery of the expected output. These methodologies have also been shown to increase the overall efficiency in terms of resource optimization, risk management, and cost management [1]. More tangibly, in the last decades, several studies have provided evidence about the link between the enterprise performance indicators and the maturity level of the adopted project management methodology [2].

PRINCE2 is arguably one of the most adopted standard project management methodologies in various systems engineering domains. PRINCE2 has been more and more adopted since 2009, for three reasons:

1. the overall trend of business to use a project-based approach to develop products or transformations within increasingly collaborative contexts with multiple partners;
2. the overall trend of capitalizing on knowledge of best practices in project management;
3. the inherent PRINCE2 characteristics, such as the general approach (i.e. application/domain independent), the product-based planning or the product breakdown structuring.

However, PRINCE2 is still relatively complex to learn and to apply as the specification sequentially presents all the PRINCE2 elements (particularly the processes), which are highly interleaved during a PRINCE2 project execution. As a consequence, the project manager is required to build a detailed mental map of the interconnections to directly access the relevant parts of the specifications. Expert project managers, who have been using PRINCE2 for ten or more years, have likely mastered all these interconnections. However, younger and aspiring project managers may require more assistance and time to become fully familiar with these interconnections. In the wider systems engineering community, model-based approaches have often been introduced to represent document-based specifications with the implicit objectives of providing an automated processing, a direct (i.e. non linear access to the individual parts), and an integration with other supporting tools for decision making [1].

In this paper, we apply a model-based systems engineering approach to the representation of PRINCE2 in order to facilitate the communication, the understanding, and the application of the PRINCE2 method. Our model-based approach consists of: IDEF0 diagrams for the representation of the PRINCE2 process architecture; BPMN diagrams for the detailed description of the PRINCE2 processes; a set of supporting tables to further assist the project manager in reading the model and in following the development of the project in PRINCE2.

The paper is structured as follows. The background section provides the fundamental terminology used in PRINCE2 and in business process modeling. The related work section positions this paper's contribution with respect to the state-of-art. The method section illustrates and motivates the model organization, from the architecture to the details of PRINCE2. Finally, the conclusion section provides closing remarks and directions for further developments.

## II. BACKGROUND

This section briefly recalls the main concepts introduced by project management methodologies (i.e., PRINCE2), modeling methodologies (i.e., IDEF0) and modeling languages (i.e., BPMN).

### A. PRINCE2

PRINCE2 has been introduced in 2006 by the UK's Department of Commerce, and is now developed by AXELOS [6]. PRINCE2 is a best practice for the organization, management, and control of projects of any size within any organizational context. Although PRINCE2 is of general purpose application, it is based on a structured method which guides the project manager in the application of the best practices for the project definition and execution. As further advantages, PRINCE2 offers means to standardize the communication among the actors, to focus on the product to be delivered, and to ensure responsibilities assignment to the actors, and also to embed agile approaches within the PRINCE2 specification.

PRINCE2 consists of the following elements: seven principles—which define the fundamental rules satisfied by the following elements; seven themes—which define the areas of concerns in a project; and seven inter-linked processes—which define the activities to be performed during the project life-cycle, seven themes, and seven processes; nine roles (e.g. project manager, supplier, customer, executive, team, etc.)—which define the responsibilities in the project; and 26 internal project products (business case, project plan, risk register, etc.)—which inherently define the dependencies among the processes. Fig. 1 shows the seven processes along the four stages of a PRINCE2 process and the three levels of actions (directing, managing, executing).

The *Pre-project* phase regards the preliminary evaluation of the validity and the convenience of project. In this phase, the process SU develops two key documents: *Project Brief* and *Plan for the Beginning of the Project*. These documents are taken as input by the process DP, in which the project board assesses the plan to decide whether or not to authorize the project. In the positive case, the project execution moves to the *Initiation stage* phase, in which the processes IP and SB are implemented. The former produces several internal project documents, such as detailed business cases (from the initial project brief) and guidelines to assess the achievement of the expected benefits. Differently, the process SB produces a detailed planning for the execution of the following stage.

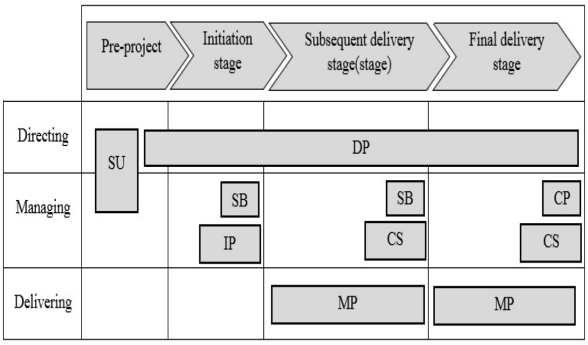


**Fig. 1. The PRINCE2 processes—Starting Up a Project (SU), Directing a Project (DP), Initiating a Project (IP), Managing a Stage Boundary (SB), Controlling a Stage (CS), Managing Product Delivery (MP), Closing a Project (CP).**

### B. IDEF0

IDEF refers to a family of modeling methodologies widely adopted in the field of systems and software engineering [9]. Specifically, *IDEF0* is a function modeling methodology for describing manufacturing functions. IDEF0 provides a functional modeling language consisting of three elements:

1. *Diagrams*, which represent the structure of the process architecture in terms of activities (boxes) and their dependencies (arrows).
2. *Text*, which are labels that can provide further information on the elements in a diagram (e.g. name of the activity, etc.).
3. *Glossary*, which define extensively all the labels, names, and acronyms defined in the diagrams.

However, IDEF0 also allows the use of supplementary material, also known as FEO (For Exposition Only) pages, which can be used to represent flow diagrams or technical design using any formalism.

Graphically, a diagram consists of a frame, and one or more boxes and arrows. Asides from delimiting the area for the drawing of boxes and arrows, the frame provides also a preassigned location for the diagram name. For the top-level (or context) diagram defining the overall process, the name must be "A-0". This diagram "declares" the process and also indicates the parameters related to the model purpose and scope, which are needed to check the consistency and appropriateness of child diagrams. The first child diagram must be named "A0" and defines the composition of overall process. This diagram must contain between 3 and 6 activities, which boxes are to be located on the upper-left/lower-right diagonal of the diagram's frame, following a temporal order in the development of the activities. Further child diagrams can be introduced by recursively detailing an activity already defined in one diagram. For each child diagram, a DRE (*Detail Reference Expression*) is to be used as part of a diagram name to link the diagrams. This expression corresponds to the unique and hierarchically-structured numeric identifier placed at the bottom-right corner of the box representing the activity. Concerning the arrows, these can be used to represent dependencies of input/output, on external conditions, or on resources. Graphically, these dependencies lead to the following type of arrows (Fig.2):

- *Input*: incoming arrow into the left side of the box. This arrow represents data or objects that are transformed by the activity;
- *Output*: outgoing arrow from the right side. This arrow represents the output produced by the activity.
- *Controls (or Constraints)*: incoming arrow from the top of the box. This arrow represents external conditions (e.g. norms, procedures, regulations, etc.) that are required for the activity to be successfully developed.
- *Mechanisms* (or *Resources*): incoming arrow from the bottom of the box. This arrow represents the means or resources (machinery, information systems, humar resources, etc.) needed to perform the activity.
- *Recall* (o *Call*): outgoing arrow from the bottom of the box. This arrow represents that the activity relies on an activity already defined somewhere else in the model or in another model.

In an IDEF0 model, each arrow should also be associated to a label which indicates the nature of the dependency. Moreover, arrow lines are enforced either to maintain a straight horizontal line or to break into horizontal segments that are connected by perpendicular lines.

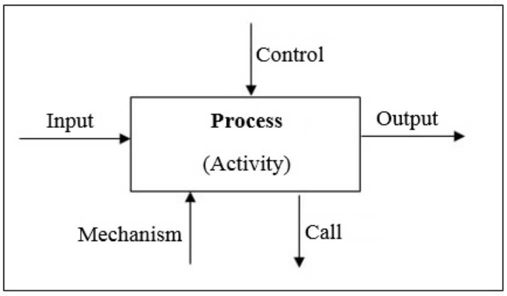


**Fig. 2. Roles and positions of the arrows**

### C. BPMN

The BPMN notation has been defined to introduce a modeling language that could be easily understood by business users (e.g. managers, process owners, business analysts, etc.), as well as by technical users (e.g. IT analysts, IT developers, etc.) [10]. The notation builds on concepts defined by popular modeling languages, such as UML, IDEF, and the classical workflow language. BPMN provides three types of diagrams, however only the *Business Process Diagram* (*BPD*) has been used in this work.

A BPD is a directed graph composed of *flow objects* and *connecting objects*. A flow object is the main describing elements within BPMN, and can be an *event* (something that happens), an *activity* (process step) or a *gateway* (the divergence and convergence of execution flows). A connecting object is a line that connects BPMN flow objects to specify the flow of activities in a process.

To organize activities in a process model, BPMN provides the *swimlane* primitive, which is used to represent roles, responsibilities and/or organizational structures by means of *pools* (organizations) and *lanes* (structures within an organization).

Finally, the BPMN notation also provides means to represent information. *Artifacts* represent information relevant to the model but not to individual elements within the process. Actually, they are used to augment and describe a BPMN process. There are three artifact types: *annotations* (to represent additional flow parts of the model), *groups* (to organize tasks or processes), and *data objects* (to specify input, output or stored data).

Fig. 3 summarizes the main elements of a BPD specified by use of BPMN.

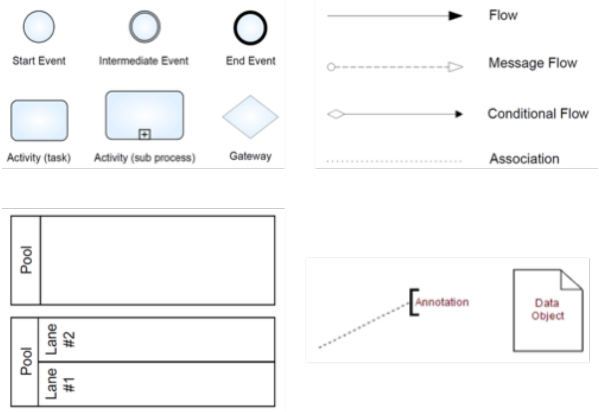


**Fig. 3. BPMN base elements**

## III. RELATED WORK

Model-based systems engineering (MBSE) approaches have been widely applied to various systems engineering domains and activities as these approaches have shown increasingly benefits with respect to the conventional document-based approaches [1]. For example:

- Communication improvement through a visual, shared, and unambiguous representation of the final system for the entire engineering team;
- Risk reduction in the development process through the progressive and continuous validation of requirements and of the verification of proposed solution (for example via integrated simulation or formal verification)
- Quality improvement through a rigorous and computer-managed traceability of the specification integrity and consistency over the entire development process

- Productivity improvement, through a prompt availability of traceability and impact analysis scope, including also the increased reuse of software and system components from the model specification level already, and the automatic generation of code for software systems.

Leveraging on the above benefits, model-based systems engineering approaches have been applied to business processes engineering, which can be radical (Business Process Reengineering, BPR) or incremental (Business Process Improvement, BPI). Various works can be found in literature about case studies, industrial applications, technologies, and methodologies related to model-based approaches. More rarely, these approaches have been applied to support project management, which has historically focused on the collection of lessons learned and in the definition of tools for, e.g., planning, estimation and risk management. This paper bridges the areas of model-based systems engineering and project management with the application of a model-based approach to the specification of a project management method consisting of roles, principles, project artefacts, and processes. Although the application of MBSE is not new to support project management (see, e.g., the model-based approach to support the integration of aircraft systems described in [4]), to the best of our knowledge no contributions can be found dealing with the application of MBSE to the formulation of project management methods, particularly related to PRINCE2.

## IV. MODEL-BASED REPRESENTATION OF PRINCE2

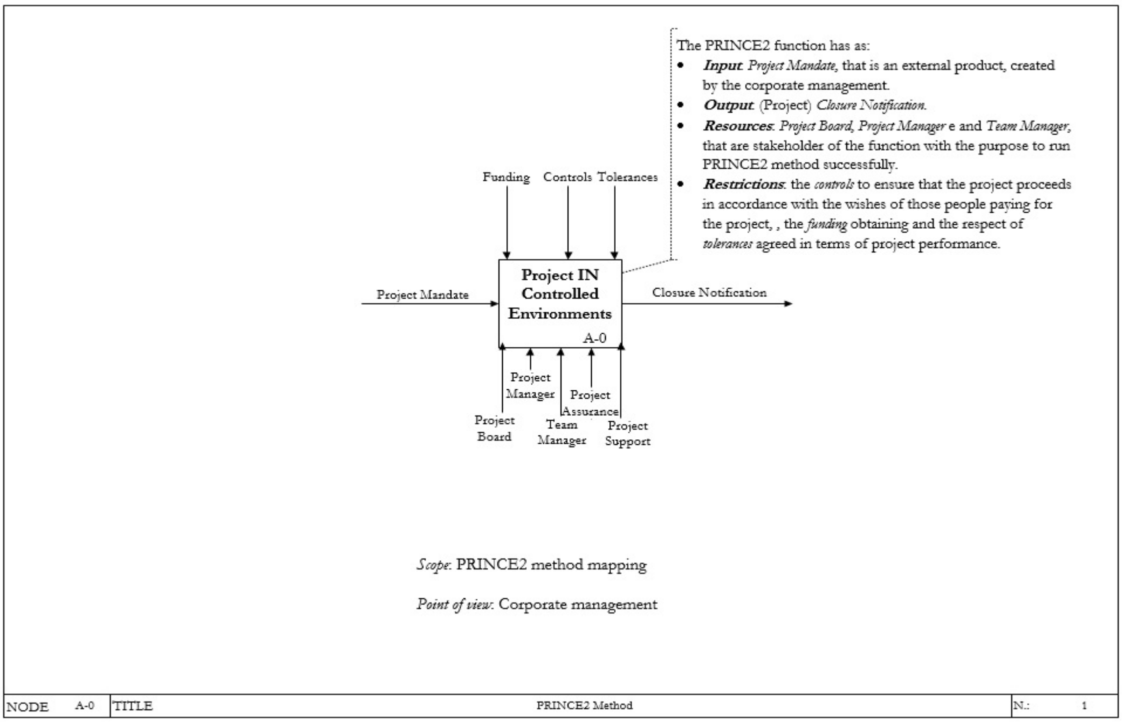
The model-based representation has been developed top-down in three steps:

1. Development of a *process architecture* in IDEF0, to identify the high-level processes and their interrelationships.
2. For each top-level process, one or more BPMN diagrams were developed to show the activities, the internal documents produced, and the communication and synchronization with other processes.
3. Development of tables and matrices for an immediate indexing of all the diagrams, to further ease the use of the model-based specification of PRINCE2.

### *A. Process Architecture in IDEF0*

IDEF0 was selected for the representation of the process architecture as this language allows the organic and systematic representation of models with an increasingly level of details. Moreover, this language inherently highlights the functional dependencies rather than the temporal dependencies, which can conversely be better represented in BPMN. Within this work, the objectives of the process architecture are to:

- provide the users with a synthetic and conformant representation of PRINCE2
- highlight the various high-level activities performed in a PRINCE2 project
- formalize the process interdependencies to provide an overarching framework that could provide a "starting point" for the detailed modeling and an "end point" for overall consistency verification.
- define the responsibility areas uniquely
- understand how resources are used
- identify key controls



The PRINCE2 function has as:

- ***Input***: *Project Mandate*, that is an external product, created by the corporate management.
- ***Output***: (Project) *Closure Notification*.
- ***Resources***: *Project Board*, *Project Manager* e and *Team Manager*, that are stakeholder of the function with the purpose to run PRINCE2 method successfully.
- ***Restrictions***: the *controls* to ensure that the project proceeds in accordance with the wishes of those people paying for the project, , the *funding* obtaining and the respect of *tolerances* agreed in terms of project performance.

*Scope*: PRINCE2 method mapping

*Point of view*: Corporate management

**Fig. 4. IDEF0 A-0 diagram of PRINCE2**

For completeness, we first represented the PRINCE2 function as the IDEF0 A-0 diagram, which is shown in Fig. 4.

Next, the process architecture has been developed in three steps and two levels of details (phases and processes). The first step deals with the functional breakdown of all the PRINCE2 phases and processes (see Fig. 5), the second with the representation of phases and the third step with the representation of processes.

Fig. 5 illustrates the diagram representing the following PRINCE2 project phases: *Pre-project* (A1), *Initiation stage* (A2*)*, *Subsequent delivery stage(s)* (A3*)* e *Final Delivery stage* (A4).

Fig.6 instead represents the four PRINCE2 phases and their interrelations according to a temporal decomposition (i.e. each block groups the lower level processes by a temporal proximity rather than a functional similarity).

Although this approach is theoretically debatable, it offers the most intuitive representation for project managers, who are primarily concerned with a temporally sequential access to the model.

The IDEF0 diagram also highlights the required resources in each phase and the expected conditions needed for their successful implementation. Consequently, with only one figure, the project manager can quickly acquire a more comprehensive view on the respective part of the PRINCE2 specification.

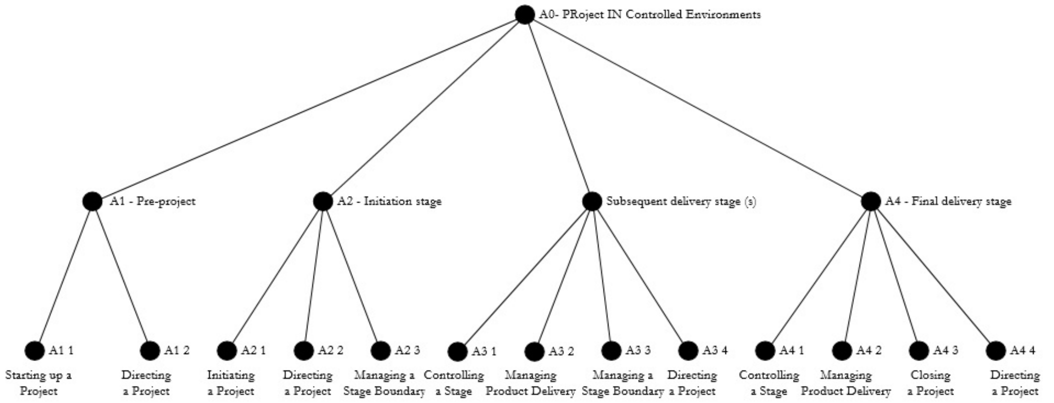


**Fig. 5. Functional breakdown of all the PRINCE2 phases and processes**

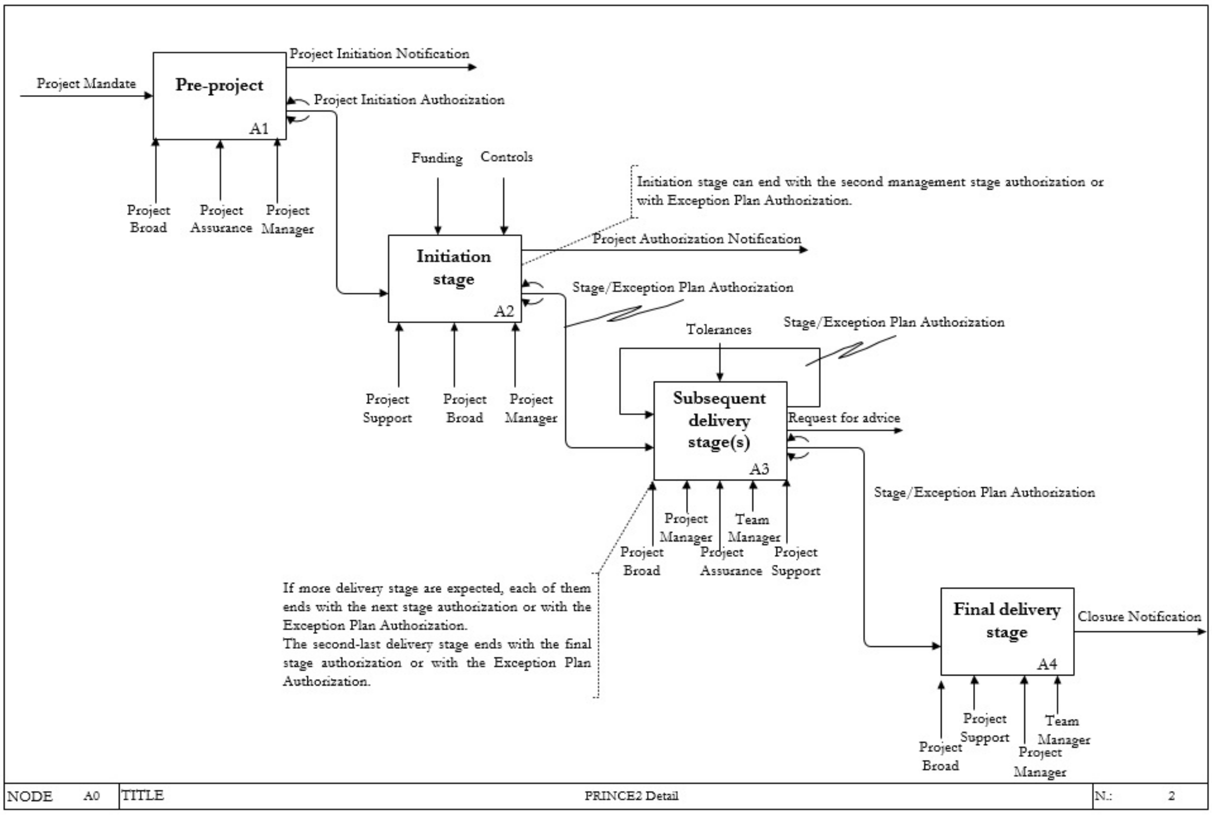
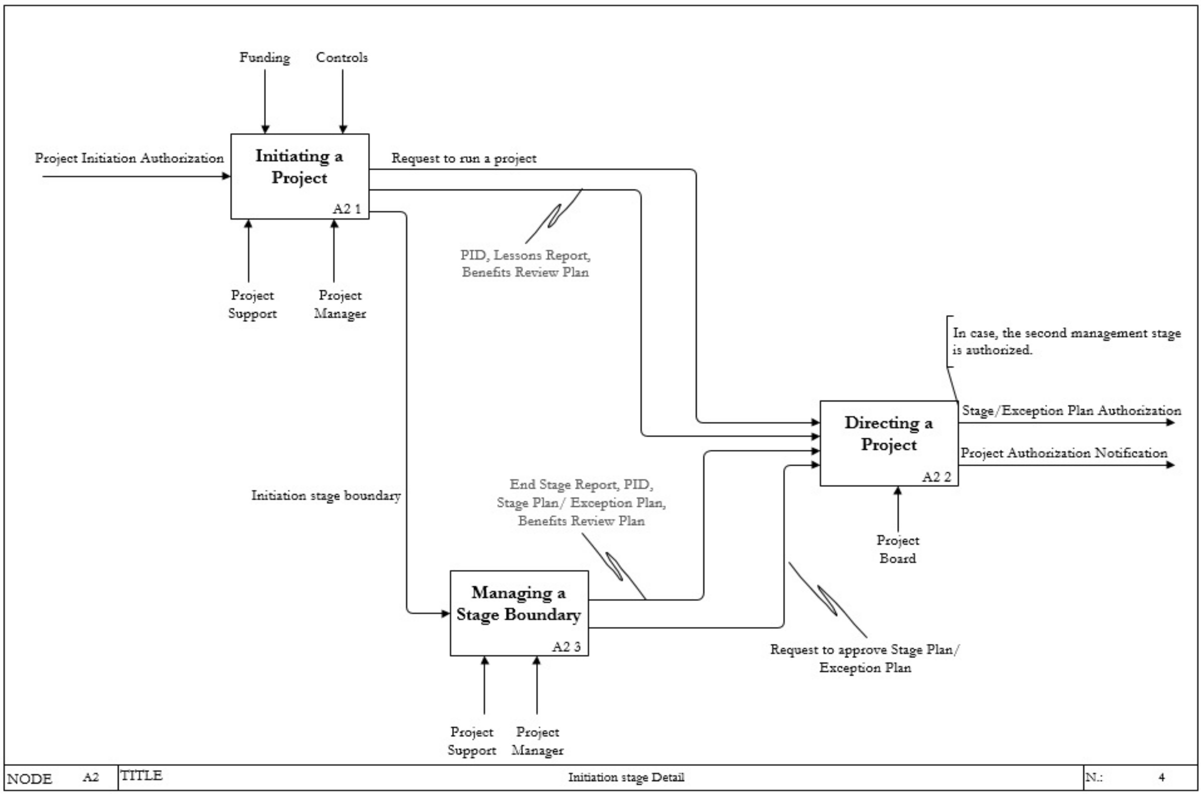
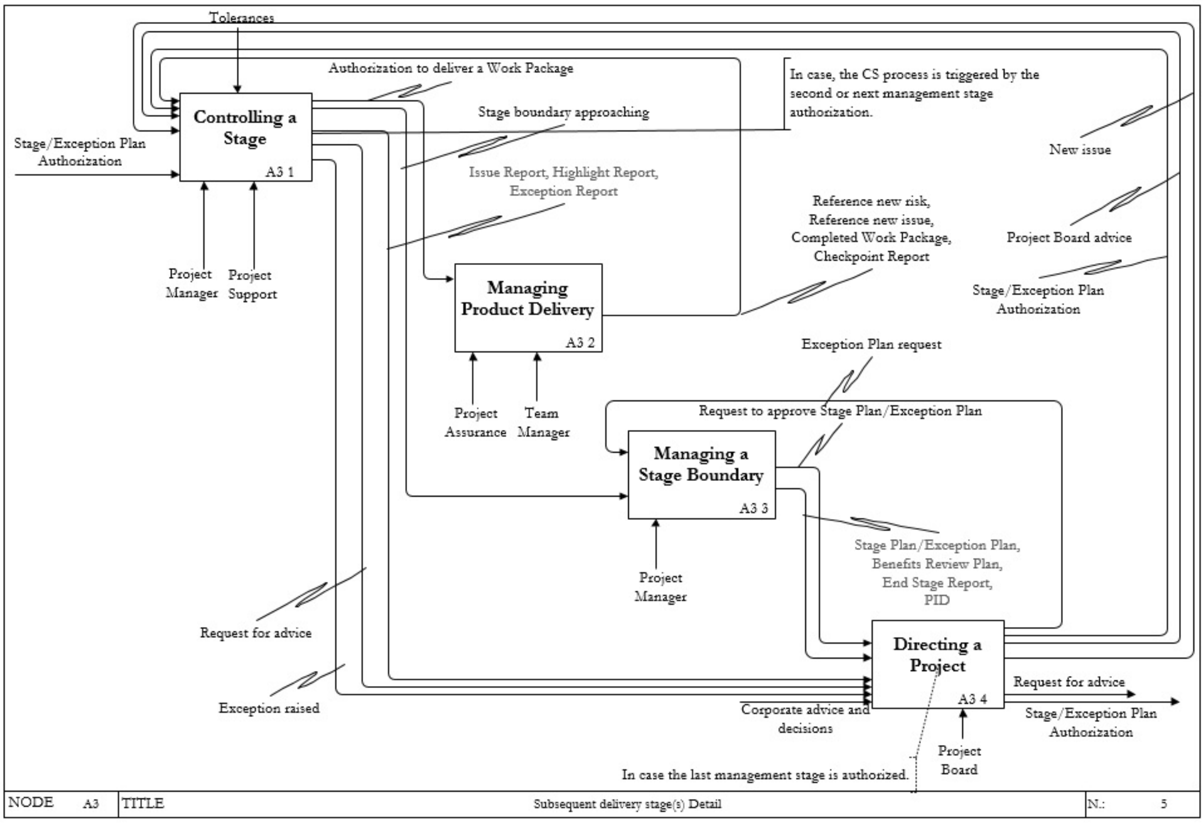

Project Mandate
Pre-project
A1
Project Initiation Notification
Project Initiation Authorization
Project Broad
Project Assurance
Project Manager
Funding
Controls
Initiation stage can end with the second management stage authorization or with Exception Plan Authorization.
Initiation stage
A2
Project Authorization Notification
Stage/Exception Plan Authorization
Project Support
Project Broad
Project Manager
Tolerances
Stage/Exception Plan Authorization
Subsequent delivery stage(s)
A3
Request for advice
Stage/Exception Plan Authorization
Project Manager
Team Manager
Project Broad
Project Assurance
Project Support
If more delivery stage are expected, each of them ends with the next stage authorization or with the Exception Plan Authorization.
The second-last delivery stage ends with the final stage authorization or with the Exception Plan Authorization.
Final delivery stage
A4
Closure Notification
Project Support
Team Manager
Project Broad
Project Manager
NODE A0
TITLE
PRINCE2 Detail
N.: 2

**Fig. 6. IDEF0 First Level – PRINCE2 Phases**



Funding
Controls
Project Initiation Authorization
Initiating a Project
A2 1
Request to run a project
PID, Lessons Report, Benefits Review Plan
Project Support
Project Manager
In case, the second management stage is authorized.
Directing a Project
A2 2
Stage/Exception Plan Authorization
Project Authorization Notification
Initiation stage boundary
End Stage Report, PID, Stage Plan/ Exception Plan, Benefits Review Plan
Project Board
Managing a Stage Boundary
A2 3
Request to approve Stage Plan/ Exception Plan
Project Support
Project Manager
NODE A2
TITLE
Initiation stage Detail
N.: 4

**Fig. 7. IDEF0 Second Level – Expansion of A2 Process (Initiation Stage)**



Tolerances
Authorization to deliver a Work Package
In case, the CS process is triggered by the second or next management stage authorization.
Controlling a Stage
A3 1
Stage boundary approaching
Stage/Exception Plan Authorization
New issue
Issue Report, Highlight Report, Exception Report
Reference new risk, Reference new issue, Completed Work Package, Checkpoint Report
Project Board advice
Project Manager
Project Support
Managing Product Delivery
A3 2
Stage/Exception Plan Authorization
Exception Plan request
Project Assurance
Team Manager
Request to approve Stage Plan/Exception Plan
Managing a Stage Boundary
A3 3
Stage Plan/Exception Plan, Benefits Review Plan, End Stage Report, PID
Project Manager
Request for advice
Directing a Project
Exception raised
A3 4
Request for advice
Corporate advice and decisions
Stage/Exception Plan Authorization
Project Board
In case the last management stage is authorized.
NODE A3
TITLE
Subsequent delivery stage(s) Detail
N.: 5

**Fig .8. IDEF0 Second Level – Expansion of A3 Process (Subsequent Delivery Stage(s))**

Each A0 activity block is then detailed into a more refined IDEF0 diagram. For the sake of conciseness, only activity A2 (Fig.7) and A3 (Fig.8) are detailed in this paper. These diagrams represent the interrelationships among the PRINCE2 processes activated in the initiation and subsequent delivery stages. As both stages involve the invocation of the process "Directing a Project", the respective block activity is included in both diagrams. Inherently, this double presence also explains the complexity in managing the lower level BPMN diagram: the same PRINCE2 process is activated in various points by different events. The supporting tables will ensure that the project manager is provided with full guidance for the identification of the lower level diagrams. With a basic understanding of the IDEF0 formalism, all the above diagrams can be immediately read also by a novice IDEF0 user, and therefore their accurate narration is left to the interest reader.

### B. *Detailed Process Definition*

The seven PRINCE2 processes are represented by a set of BPDs in BPMN. The BPD allows for a higher accuracy in the semantics of the activities to be performed, as well as of the interdependencies. Moreover, BPMN also provides symbols to represents events and data, both essential for the definition of process interdependencies. BPD diagrams have been specified according to the modeling guidelines introduced in [10], while aiming to maintain diagrams relatively simple and readable within an A4 page format. The process mapping has progressed hierarchically through the analysis of the PRINCE2 specification and by applying the mapping guidelines described in [11].

BPMN intrinsically ensures the traceability by use of hyperlinks provided by modeling tools (ADONIS in this paper case [12]). However, with regard to the IDEF0 model, the traceability has been achieved by including textual notes indicating the respective IDEF0 activity block(s) and diagram into the top-level BPDs. Concerning the model consistency, the internal consistency is similarly ensured by the modeling tool using the model syntax validation rules. Differently, for the consistency between the BPMN diagrams and the IDEF0 diagrams, it is manually ensured by using the following mapping [13]:

**Table. 1. Mapping between IDEF0 and BPMN.**

| IDEF0 | BPMN |
|---|---|
| Activity | Process |
| Input/Output | Internal Process Data Object and inherent Flow |
| Control (Condition) | Event |
| Mechanism | External Data Object or BPMN Actor |

Eventually, the whole BPMN model resulted into a hundred diagrams. Fig. 9 and Fig. 10 show the BPDs of the processes "Initiating a Project" (A21) and "Directing a Project" (A12, A22, A34, A44), respectively. These processes are synchronized through the event "Request to Run a Project", which is triggered by the former process and catched by the latter one. This key integration aspect can be immediately inferred in both diagrams by noticing the respective relationships between the process pools and the triggering/catching event symbols.

### C. *Supporting Tables and Matrices*

The above IDEF0 and BPMN diagrams completely represent the whole PRINCE2 specification. As these diagrams are in a machine readable format, they can be opened with the relevant modeling tools and be directly navigated through the hyperlinks inherently associated to the model elements. Currently, most of the project management tools focus on the definition of project plans or on the KPI-based monitoring/tracking of project performance. Differently, the project execution requires a pervasive human intervention in the implementation of the tasks, and therefore it has received less attention for automation opportunities. Consequently, when using the paper-based version of the model, a project manager can further benefit from indexing means to identify which diagram(s) are to be retrieved. The indexing is particularly needed for two reasons. The first reason is that the IDEF diagrams provide a functional breakdown that is to be linked to the lower level time-oriented BPMN diagrams. The second reason is that the BPMN diagrams are numerous and are not accessed sequentially. For example, the project manager may want to know which project actors are to be involved in a process, or which processes should be started as a consequence of an event occurring, or which process(es) produces or consumes an internal product of the project activity. In particular, the following types of tables and matrices are made available along with the PRINCE2's IDEF0 and BPMN diagrams:

- an *Overview Table*, which describes the key features of the process, including the purpose, the triggering event, the involved actors, the list of the activities, the triggered processes, and the produced/consumed internal project output. The project manager may refer to this table for consultation when preparing a process execution.
- an *Event Traceability Matrix*, which identifies the processes that can trigger an event and the processes that should be activated when that event. The project manager can consult this matrix to identify which process should be executed when an event occurs.
- a *Product-Process/Activity Traceability Matrix*, which identifies the process(es) that creates, confirms, updates, or reviews an internal project product. This matrix also provides references to the actors performing the activity, offering therefore a comprehensive view on all the actions performed on the internal products. For example, the business case is created in the process "Starting a Project", updated in the processes "Begin the Project" and "Manage Stage Boundary", and finally it is confirmed in the process "Directing a Project".
- an *Activity Summary Table*, which shows the list of input and output items for each activity.

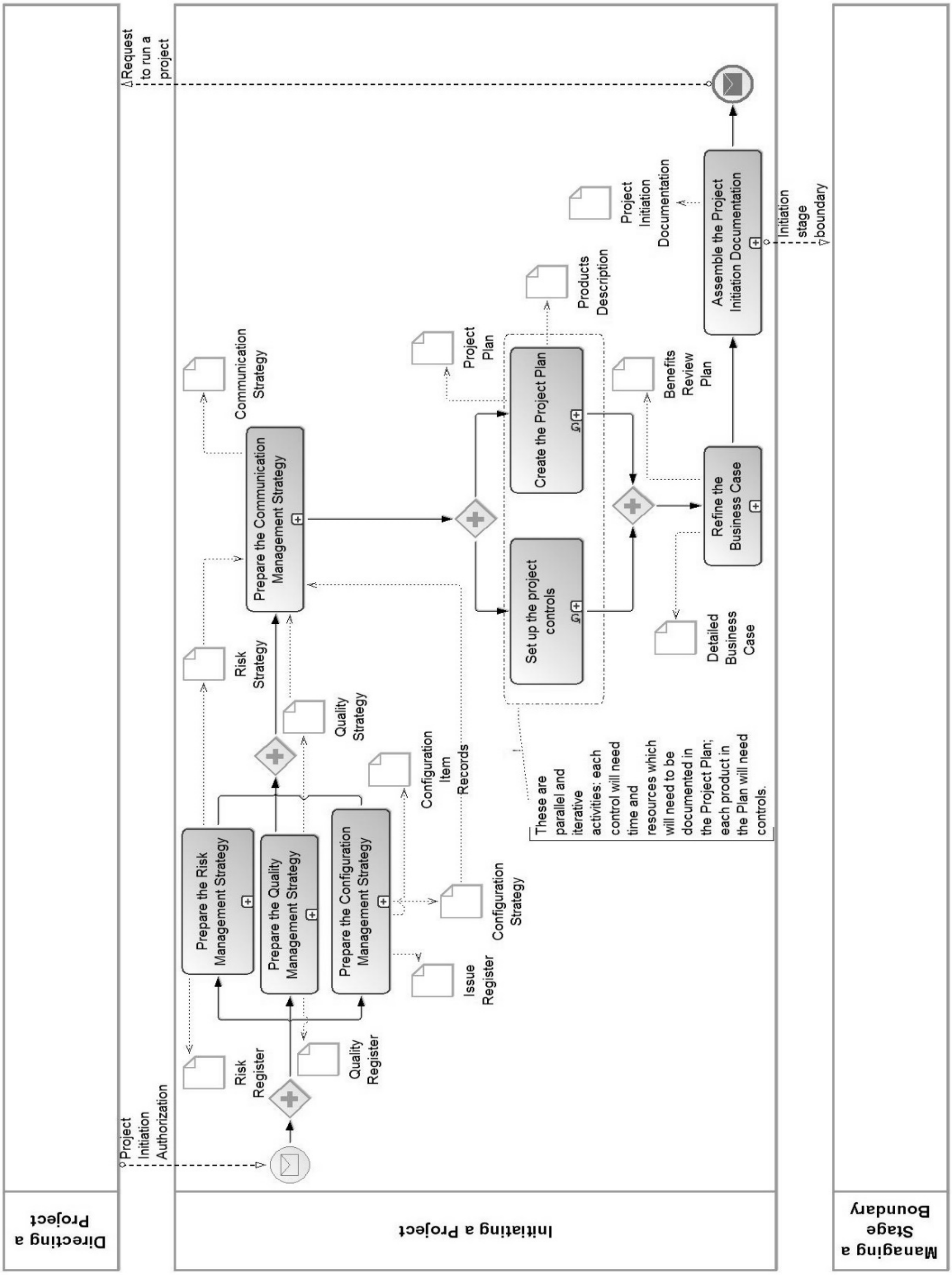

Directing a Project
Initiating a Project
Managing a Stage Boundary
Project Initiation Authorization
Request to run a project
Risk Register
Quality Register
Issue Register
Prepare the Risk Management Strategy
Prepare the Quality Management Strategy
Prepare the Configuration Management Strategy
Configuration Strategy
Configuration Item Records
Risk Strategy
Quality Strategy
Prepare the Communication Management Strategy
Communication Strategy
Set up the project controls
Create the Project Plan
These are parallel and iterative activities: each control will need time and resources which will need to be documented in the Project Plan; each product in the Plan will need controls.
Project Plan
Products Description
Refine the Business Case
Detailed Business Case
Benefits Review Plan
Assemble the Project Initiation Documentation
Project Initiation Documentation
Initiation stage boundary

Fig. 9. BPD of the process “Initiating a Project”

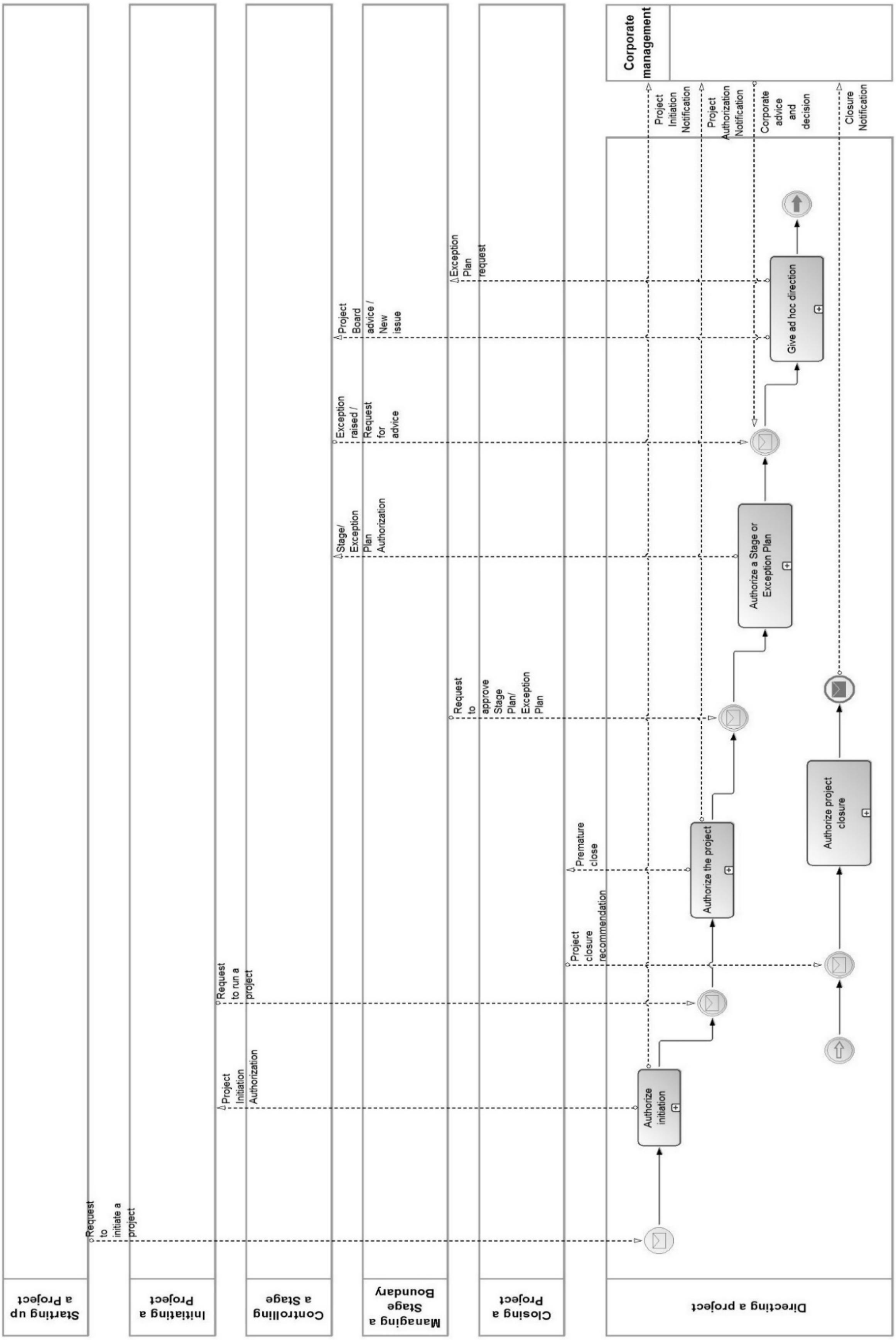

Starting up a Project
Initiating a Project
Controlling a Stage
Managing a Stage Boundary
Closing a Project
Directing a project
Corporate management
Request to initiate a project
Project Initiation Authorization
Request to run a project
Project closure recommendation
Premature close
Request to approve Stage Plan/ Exception Plan
Stage/ Exception Plan Authorization
Exception raised / Request for advice
Project Board advice / New issue
Exception Plan request
Authorize initiation
Authorize the project
Authorize project closure
Authorize a Stage or Exception Plan
Give ad hoc direction
Project Initiation Notification
Project Authorization Notification
Corporate advice and decision
Closure Notification

**Fig. 10. BPD of the process “Directing a Project”**

## V. Conclusions and Further Developments

PRINCE2 is a widely used standard and structured project management methodology that can be applied to various systems engineering application domains. However, the PRINCE2 specifications can be complex to learn and to apply as it defines a large number of interleaved processes and activities that must be orchestrated to implement the project execution. With the objective of easing the learning and the application of the PRINCE2 methodology, we have introduced a model-based specification for the PRINCE2 specification. The specification consists in a process architecture, a set of detailed process diagrams, and a set of indexing tables to directly access the diagrams.

The process architecture has been represented with IDEF0 as the architecture is inherently structured hierarchically in phases, processes, and their interdependencies (inputs/outputs, resources, and enabling conditions). The architecture consists of eight diagrams: one diagram to represent the hierarchical functional breakdown and seven diagrams to represent the PRINCE2 specification from the context to the individual PRINCE2 processes. Each of these processes is mapped onto a detailed BPMN diagram, which is further decomposed to reach the same granularity/detail level to PRINCE2 specification. Overall, about one hundred BPMN diagrams have been developed to cover all the details of the PRINCE2 specification. These diagrams can be immediately and directly accessed by use of a modeling tool, and therefore they offer to the project manager a web-like browsing tool for the entire PRINCE2 specification.

However, only the project management activities related to monitoring & control or to project planning are currently supported by automated tools. Project execution activities are specifically carried out without any tool support, as these activities often require sound judgement and specialist knowledge that are difficult to automate by use of a software tool. Consequently, we have also introduced a set of tables and matrices (overview table, event traceability matrix, product-process matrix, activity summary table) to ease the manual access of the diagrams. These tables and matrices provide synthesized information on the diagrams and a reference to the relevant diagrams. As an example, an overview table summarizes the key features of each process and activity, providing also a reference to the respective diagrams. Similarly, the event traceability matrix provides the project manager with the information on which events can be triggered by which process and which process is triggered by the events.

Asides from the above motivations and advantages, the model-based approach can contribute to provide a wide number of additional benefits, such as communication improvement, quality improvement, risk reduction, etc., which are commonly experienced when applying model-based systems engineering to the analysis and design of complex socio-technical systems.

Moreover, once the PRINCE2 specification is in a machine-readable format, further exploitations can be easily obtained also with current tools and technologies. In line with the current capabilities of model-based systems and simulation engineering [5][14], we envision three further areas of future exploitation for our model:

- simulation of project management processes, for:
  1. the analysis and design of innovative project management methodologies
  2. the optimization of organizational resources
  3. the post-mortem analysis on completed (or cancelled) projects
- specification verification, to ensure the consistency across all the processes and internal documents
- graphical customization of the PRINCE2 specification for specific projects and organizational contexts.

## VI. References

[1] Nokes, Sebastian, *The Definitive Guide to Project Management*, Prentice Hall, 2007.

[2] Harold R.Kerzner, Ph.D., *Project Management: A Systems Approach to Planning, Scheduling and Controlling*, J.Wiley &Sons, Inc.,Hoboken, New Jersey, 2013

[3] INCOSE Systems Engineering Vision 2020 v.2.03, September, 2007, available at: http://www.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf .

[4] G.D. Hernandes, J. de Melo Bezerra, C Massaki Hirata, R. Rizzi Starr, *Towards a workflow to support the integration of aircraft systems' models*, Proceedings of the 32nd IEEE/AIAA Digital Avionics Systems Conference (DASC), Syracuse, NY, USA, 5-10 Oct. 2013.

[5] D. Gianni, A. D'Ambrogio and A. Tolk, *Modeling and Simulation-Based Systems Engineering Handbook*, CRC Press, ISBN 9781466571457, 2014.

[6] Office of Government Commerce, *Managing Successful Projects with PRINCE2 (2009)*, The Stationery Office, UK 2009.

[7] Colin Bentley, *The Essence of PRINCE2*, Hampshire Training Consultants, UK 2009.

[8] Julia Murray, *Model Based System Engineering (MBSE) Media Study*, UK 2012.

[9] *Draft Federal Information Processing Standard Pubblication183: Integration Definition for Function Modellinf (IDEF0),* FIPS, 1993.

[10] Bruce Silver, *BPMN Method & Style*, Cody-Cassidy Press, USA 1994.

[11] M.Dumas, M.La Rosa, J.Mendling, H.A. Reijers, *Fundamentals of Business Process Management*, Springer 2012.

[12] A.C. Correia, *Quality of process modelling using BPMN: a model-driven approach*, Universidade NOVA de Lisboa, January 2014.

[13] D. Coppola, *Applicazione di metodologie model-based per la specifica di PRINCE2 (in Italian)*, M.S. Thesis, Guglielmo Marconi University, Rome (Italy), 2016.

[14] P. Bocciarelli, A. D'Ambrogio, *Performability-Oriented Description and Analysis of Business Processes*. In: Jason A. Beckmann (ed.), Business Process Modeling: Software Engineering, Analysis and Applications, pp. 1-36, Nova Science Publishers, ISBN: 978-1-61209-344-4, 2011.