HAL
open science

Identification and Assessment of Obsolescence in the Early Stages of System Design

Sophia Salas Cordero, Marc Zolghadri, Rob Vingerhoeds, Claude Baron

To cite this version:

Sophia Salas Cordero, Marc Zolghadri, Rob Vingerhoeds, Claude Baron. Identification and Assessment of Obsolescence in the Early Stages of System Design. Journal of Integrated Design and Process Science, 2022, 24 (3-4), pp.15-33. 10.3233/JID-210018 . hal-03837386

HAL Id: hal-03837386

https://hal.science/hal-03837386

Submitted on 14 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Transactions of the SDPS:
Journal of Integrated Design and Process Science
XX (XXXX) XX-XX
DOI 10.3233/jid-201x-xxxx
http://www.sdpsnet.org

Identification and Assessment of Obsolescence in the Early Stages of System Design

Sophia Salas Cordero[a,b*], Marc Zolghadri[b,c] and Rob Vingerhoeds [a,c] Claude Baron[a,c,d]

[a] *ISAE-SUPAERO, Université de Toulouse, Toulouse, France*
[b] *Quartz Lab, ISAE-Supmeca, Paris, France*
[c] *LAAS-CNRS, Toulouse, France*
[d] *INSA, Université de Toulouse, Toulouse, France*

Abstract Obsolescence is the fact that an entity (physical or logical) is becoming outdated or no longer possesses the required level of performance. The objectives of this article are twofold. First, it is intended to contribute to the understanding of obsolescence propagation. Secondly, two supporting approaches for the Identification and Assessment phases are proposed: the House of Obsolescence and the System Obsolescence Criticality Analysis. The former allows the mapping of obsolescence propagation via dependencies, whether imposed changes are desired or imposed, by external actors to the system architecture. Whereas, the objective of the latter is to assign an obsolescence criticality index to the identified risks in order to prioritize them for solution or mitigation determination during the analysis phase. The tools make extensive use of the modeled system knowledge through the application of Systems Engineering. The application of these approaches is presented through an illustrative study.

Keywords: Systems Engineering, Obsolescence management process, Identification, Assessment, House of Obsolescence, Obsolescence Criticality Index.

SUBMITTED

1. Introduction

Progressively obsolescence is becoming a topic that is discussed more and more. Obsolescence can be defined as a normal and natural phenomenon. Often consumers suspect about planned obsolescence, fearing that companies artificially reduce the effective service life of products forcing consumers to replace them, as stated by (Bulow, 1986; Kreziak, Prim-Allaz, & Robinot, 2017). According to the international standard (IEC-62402, 2019), obsolescence is the "transition from availability from the original manufacturer to unavailability". Many technical, financial, legal or technological drivers may be hidden behind this transition process. For example, new computers are not sold with Windows 7, although there might be a specific market niche maybe interested in computers with this operating system. Microsoft announced on January 14th 2020 that the support for Windows 7 was ending (Windows end of support notice), which made this operating system obsolete hence not suitable for sale. Ending support does not mean that computers with

*Corresponding author. Email: sophia.salas@isae-supaero.fr.

1092-0617/$27.50©201X - Society for Design and Process Science. All rights reserved. Published by IOS Press

this operating system will cease to function, but it means that no evolution of the functionalities or protection will be proposed in the future, which could open the door to security breaches. As a result, computers running under Windows 7 may see degraded performance or decreased availability, and as such this ending of support is a driver for obsolescence. Such systems would increasingly have degraded performances, and no longer be able to comply with system requirements more specifically security requirements which means it will be hard to maintain or may soon stop operating, as underlined by (Zheng, Sandborn, Terpenny, & Orfi, 2014).

Understanding the obsolescence phenomenon and its propagation mechanisms is essential for an obsolescence management plan. From a cost avoidance point of view proactive obsolescence management approaches have proven to be very effective in comparison to reactive approaches (SD-22, 2021). Proactive approaches at the same time appear to support one of the main objective of Systems Engineering "to design and develop a system that can be maintained effectively, safely, in the least amount of time, at least cost and with a minimum expenditure of support resources without adversely affecting the mission of that system" (Walden, Roedler, Forsberg, Hamelin, & Shortell, 2015). The objective of this paper is to propose two supporting approaches for the identification and assessment phases of an obsolescence management plan. These approaches are the House of Obsolescence (HOO) and the System Obsolescence Criticality Analysis. HOO allows the study of obsolescence propagation, whether desired or imposed, by external actors to the system architecture. Whereas, the objective of the latter is to assign an obsolescence criticality index to the identified risks in order to prioritize them for solution or mitigation determination during the analysis phase. The correct manner to address obsolescence risks is from the early stages of system design.

The current paper addresses the understanding and modelling of obsolescence propagation. The understanding and modelling of obsolescence and the propagation of its consequences are addressed by linking obsolescence to the fundamental concepts of systems engineering. The stakeholder requirements of a system-of-interest (SOI) and the components are not all at the same level of risk to obsolescence and are not all sensitive at the same degree to the obsolescence occurrence. Two tools are proposed for obsolescence management of a SOI during the identification and assessment phase. They use results in a quantification of this risk for each component and function identified as critical by the experts of the system.

The paper is structured as follows. Section 2 reviews the concepts related to obsolescence and the propagation of the consequences of obsolescence through the system architecture. To this purpose, some fundamental concepts of systems engineering as well as the channels for propagating the consequences of obsolescence will be highlighted through the system models. Section 3 then details the proposal of tools for the identification and assessment phase of the obsolescence management of the SOI. These concepts are illustrated in section 4 through an example of a weather forecasting system taken from (Roques, 2017). The article concludes with a discussion of the results obtained and future work.

SUBMITTED

2. Related Work

In this section, some of the basic elements behind this study are presented. Starting with a description of what obsolescence is about and different manners of classification. Then the attention is turned to systems engineering and model-based systems engineering, so to see where and how obsolescence can have its impact and where and how potential solution approaches could be incorporated. Finally, requirements engineering is discussed, field in which the authors see a good starting point can be found for addressing obsolescence, right from the earliest design phases. This section ends with a conclusion on the identified challenges and how they can be brought to a more holistic approach building on the presented basic elements.

2.1. Obsolescence

Obsolescence is a state in which a piece of equipment becomes no longer useful or/and out-of-date, whether form or function-wise. Obsolescence is situationally dependent, the underlying causes for an item to become obsolescent according to (SD-22, 2021) are: technology (newer technology is preferred), function (no longer functions as intended), regulation (changes in regulation), supportability (an item becomes no longer supportable) or market demand (there is no more demand for the product). (Bartels, Ermel, Sandborn, & Pecht, 2012) provides a categorization for obsolescence which is discussed in the following subsection.

Some principal drivers for obsolescence are:

- technology advancements (Merola, 2006) – new products appear replacing old ones,
- lack of support from vendors (Merola, 2006) – the organization is forced to modify their product to obtain the necessary updates,
- merger and acquisition of a business (Bradley & Dawson, 1998) – the acquired organization may have to change its existing system, if it is not compatible with the other system used in the acquiring organization,
- many authors agree on that the "root cause of obsolescence issues in systems and products is the mismatch of the system and the components or parts lifecycles" (Zolghadri, Addouche, Boissie, & Richard, 2018).
- changes in regulations (SD-22, 2021).

According to (EDSTAR, 2016), the objective of obsolescence management is to ensure that obsolescence is managed as an integral part of design, development, production and in-service support in order to minimize cost and detrimental impact throughout the product life cycle. This is a non-trivial task, as in the design stage many decisions are made that affect the complete lifecycle of a product or service, at which time the designers only have (very) incomplete information at hand and rely mainly on assumptions and models' outputs.

Obsolescence and its effects can be described according to three fundamental characteristics:

(1). The first is that obsolescence has fundamentally delayed effects. Referring back to the Windows 7 example, the end of support will not make the computers stop running; the decreased performance will for instance only start to be "felt" by users when security issues start to arrive. This time delay is a key factor in the design of monitoring techniques in systems obsolescence management, as for example, the failure or non-availability of a system containing an obsolete component may not materialize until well beyond its detection (real and proven). But the obsolete component would not function at the expected performance level.

(2). The second characteristic is that obsolescence can affect elements at any level in the system structure hierarchy. "Issues are not confined to piece parts or devices; obsolescence may occur at the part, module, component, equipment, or system level" (SD-22, 2021).

(3). The third characteristic concerns the propagation of the consequences of the occurrence of obsolescence. Obsolescence is not a confined event because obsolete or near-term obsolescence elements might interact with others which can prevent the system from fulfilling its internal processes. Therefore, the consequences of obsolescence, if not properly solved, may propagate in the same level assembly, to the Next-Higher Assembly (NHA) or the entire system. Obsolescence propagates due to existing dependencies between entities of the system architecture. It is therefore essential to have a precise mapping of all system dependencies. It is exactly to achieve this objective that systems engineering, via a structured model based systems engineering approach (see section 2.4), provides essential help in managing obsolescence.

SUBMITTED

To manage obsolescence, (SD-22, 2021) defines a process that includes the following steps: prepare, identify, assess, analyze and implement. During the preparation phase, it is required to develop an obsolescence management plan to be able to track obsolescence cases. Then the identification phase includes the monitoring and surveillance of emerging obsolescence issues. Afterwards the assessment of obsolescence impact is performed during the assessment phase. Accordingly, during the analysis, a set of potential resolutions for the critical items has to be established under cost-effectiveness constraints. Finally, the solutions have to be implemented.

This paper focuses on the identification and assessment of obsolescence for the early product/service design stages and suggests an approach that allows the prioritization of obsolescence issues.

2.2. Obsolescence Classification

Obsolescence issues may be distinguished based on a voluntary (or planned, see (Bulow, 1986; Kreziak et al., 2017)) or involuntarily action of a company. Some obsolescence classifications refer to this action as the "reason or origin" of the obsolescence. Bartels et al. (2012) defines four classes of drivers:

(1). Logistical – inability to procure,
(2). Functional – the current product function, performance, or reliability becomes obsolete,
(3). Technological – advancement, and
(4). Functionality improvement dominated obsolescence – generated to remain competitive in the market.

Logistical obsolescence is a classification concerned about the availability of parts from their original manufacturer, best known as Diminishing Manufacturing Origins and Material Shortages (DMSMS). The holistic view on obsolescence goes beyond just the availability of a component from its supplier.

Another classification, proposed by (Wilkinson, 2015), distinguishes two origins of obsolescence:

(1). Supply side, and
(2). Demand side and regulation-caused.

Moreover, in his study on obsolescence and lifecycle management for avionics, (Wilkinson, 2015) suggests an obsolescence fishbone diagram that considers four obsolescence drivers:

(1). Software design (airspace requirements, commercial off-the-shelve software),
(2). Systems design (airspace requirements, assurance standards, regulations),
(3). Hardware manufacturing and repair (process, plant, components/sub-assemblies/materials, environmental legislations, component manufacturers), and
(4). Design tools (application, platforms and operating systems).

Finally, (SD-22, 2021) proposes a distinction based on the types of impacted items which are subdivided into:

(1). Software,
(2). Hardware-electronic, and
(3). Hardware – Materials and Structural, Mechanical, and Electrical (MaSME) items.

Hardware-electronic items may become obsolete for example because of low demand, demand for new technologies, or loss of repair support expertise. Software issues are due to newer versions of the software, support termination or because of mergers and acquisitions. Hardware-MaSME obsolescence issues may be due to regulations on hazardous materials, suppliers exiting business, or unavailable tooling. When proactively trying to identify potential obsolescence issues during the early design stages, these issues need to be taken into account.

SUBMITTED

2.3. Systems Engineering

Systems engineering is defined by the INCOSE (the International Council on Systems Engineering) as an "interdisciplinary approach and means to enable the realization of successful systems. It focusses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs" (Walden et al., 2015).

Current systems engineering approaches define how the different lifecycle stages are sequenced, including different models based on linear (e.g. waterfall model, V-model), iterative (e.g. spiral model) or evolutionary approaches (e.g. set-based approaches), while assuming that customer requirements are fixed throughout the system's lifecycle. In reality, this is not the case, neither stated needs nor desires from stakeholders are constant throughout the systems lifecycle (Walden et al., 2015). Several standards trace down the state of the art of systems engineering, of which today the ISO/IEC/IEEE 15288 (ISO/IEC/IEEE 15288, 2015)) is the main standard in use, accompanied by guides of best practices, such as the INCOSE Systems Engineering Handbook (Walden et al., 2015) or the Systems Engineering Body of Knowledge (SEBOK, 2015). In this paper, systems are understood to be man-made, created, and utilized artefacts that need to provide services in defined environments for the benefit of users and other stakeholders –integrated sets of elements, subsystems, or assemblies that accomplish defined objectives.

As defined in the standard (ISO/IEC/IEEE 15288, 2015), six generic lifecycle stages can be distinguished, including a Concept Stage. The concept stage identifies the needs of the stakeholders, explores concepts, identifies enabling technologies, and proposes viable and feasible solutions. Problems identified in this phase, for example for individual hardware parts or software modules, should be addressed early so to minimize the risk that in the end these entities fall short of the required functionality or performance (Walden et al., 2015), (Brazier, van Langen, Lukosch, & Vingerhoeds, 2018), or that they are at risk for obsolescence. As such, the concept stage is an important phase to consider when addressing proactive and strategic management of obsolescence.

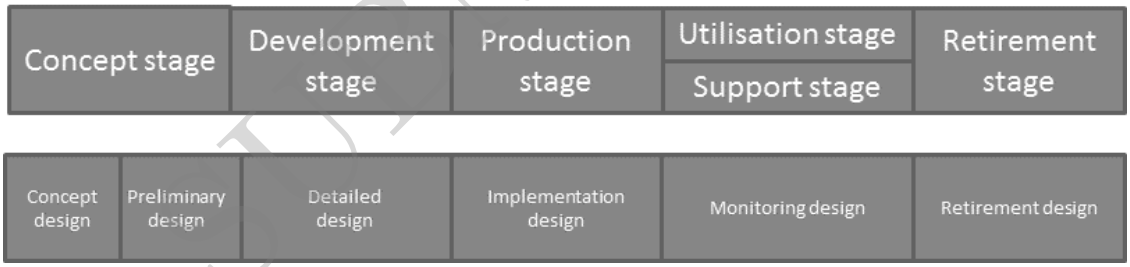


Fig. 1. Design activities per life cycle stage, adopted from (Brazier et al., 2018).

Fig. 1 refines the concept stage by dividing it into concept design and preliminary design. The concept design stage has three main objectives (Brazier et al., 2018):

(1). To interpret/understand a mission statement, supported by a positive (potential) business case,
(2). To produce an initial definition of stakeholder requirements and key performance indicators with respect to the mission,
(3). To produce an initial logical/conceptual description of a design.

The authors believe that this phase is by excellence the phase where proactive management of obsolescence should start.

SUBMITTED

2.4. Model Based Systems Engineering

Model-Based Systems Engineering (MBSE) is a successful approach to support system requirement, design, analysis, verification and validation activities, beginning in the conceptual design phase and continuing throughout development and later life cycle phases (Kaslow, Ayres, Cahill, & Hart, 2018). Models are used to represent the systems and enable to better master the design and the verification of complex systems (Hick, Bajzek, & Faustmann, 2019).

Several languages are used for MBSE. Supported by the Object Management Group since 2006, SysML, the System Modelling Language (ISO/IEC-19514, 2017), is commonly used in systems engineering to analyze, model and design systems. It is a diagrammatic modelling language for systems engineering, widely used in industry and at the moment its second version is being prepared for release over the coming years (SysML-V2, 2020). It is important to note that the SysML standard defines a notation, but not a way of using it. Methods have to be defined to make the use of the diagrams explicit, and to express a dedicated methodology conforming to the approach deployed.

Other approaches include the Object Process Methodology (OPM) (Dori, 2016; ISO/IEC-19514, 2017), with an increasing popularity, and the Architecture Analysis & Design Integrated Approach (ARCADIA) (Roques, 2017), a systems engineering methodology developed by Thales, but now more widely spread in numerous companies. The essential difference between OPM and ARCADIA on one hand and SysML on the other hand, is that the former two have associated a methodology to the syntax. Both try to achieve better structuring systems engineering approaches, while remaining fully compliant with systems engineering standards.

SUBMITTED

2.5. System Requirements

At the beginning of a development, stakeholders express their needs and wishes, that in an iterative process of clarifications, discussions and information exchanges become requirements (Brazier et al., 2018). Gero (1990) suggested that three types of requirements exist, functional, behavioral, and structural. More precisely, functional requirements state functions that a system must provide and are directly related to the mission of a system, its purpose; behavioral requirements specify desired system behavior of a design with respect to its mission, together with key performance indicators with which this behavior can be determined, the way the system acts; and structural requirements define requirements for components/sub-systems of a system. Each of these categories has a unique contribution to the design and development process. For example, the functional requirements correspond in a first step to the system capabilities that the designer needs to address.

An important step in system development is therefore the understanding and translation of stakeholder needs and desires into functional, structural, and behavioral requirements. In the requirements elicitation method, these needs and desires analysis approach helps to integrally address the requirements of the stakeholders. It could be performed iteratively and recursively at the different levels of the system as some components of the system are systems themselves that need a complete design process on which a systems engineering approach should be also applied.

Within the framework of analyzing the impact of obsolescence, functional requirements and structural requirements are first targets. Functional requirements target the different functionalities that a (sub-) system needs to fulfill. Structural requirements may impose (parts of) solutions and therewith certain components. As such, these requirements have a direct impact on obsolescence analysis.

During the early design phases, in particular in the concept stage, the requirements are iteratively refined, reworked and transformed into a functional architecture, leading to system specifications for the system, its sub-systems and eventually components.

Proven or anticipated obsolescence because of changes in requirements and/or component availability,

for example, may lead to non-compliance with specifications of the system, its sub-systems and/or some components, for instance:

- Such obsolescence may prevent the execution of the expected functionalities (e.g. impossibility to predict the weather),
- Such obsolescence may degrade the quality of the execution of these functionalities (e.g. exceeding the response time),
- Such obsolescence may reduce some characteristics of the expected functionalities (e.g. unauthorized access to data),
- Such obsolescence may cause the system to no longer comply with certain constraints (e.g. restriction on the use of certain materials, Freon for instance).

2.6. Challenges

Using the approach proposed in section 2.5, iteratively a set of requirements results, describing the expected system. Developers will then work on the system architecture and will successively make design decisions. The system architecture is the "the embodiment of concept, the allocation of physical/informational function to elements of form, and the definition of relationship among the elements and with the surrounding context" (Crawley, Cameron, & Selva, 2015). Architecting is a creative process in which the architect searches for solutions to a specific problem. The decisions developers may include technologies to be used, best practices, in order to fulfil the functional requirements. Iteratively they will reach a solution that will then be mapped to a physical architecture, comprising sub-systems and components. Part of these may come from the structural requirements, for example if a stakeholder imposes the use of a certain sub-system, technology or component.

A challenge will be to ensure that those sub-systems, components and/or functionalities with a high risk for obsolescence can be identified early enough in the design process. If this can be realized, then different choices open up to the developers: search for alternative solutions, search for double sourcing, accept the risk, etc. But this supposes, first of all, the ability to identify those sub-systems, components and/or functionalities that have a high obsolescence risk.

It is important to understand what obsolescence drivers exist, before we can aim to address them. The authors propose that two main groups of obsolescence drivers:

- Requirements-driven origins: for example, changes in legislation or any change in stakeholder or systems requirements.
- Component-driven origins: for example, the announcement of a stop of production of a component (called PDN or Product Discontinuance Notice) or an update/change.

These drivers can be found in Fig. 2. A requirement that changes obviously may have an impact on the system under development (or already in production), making all or part of developed solutions obsolescent. For example, as anti-pollution legislation for automobiles evolves, some of the cars in production would no longer be allowed to be sold as off a certain date. New solutions for depolluting would be necessary.

A component production change may affect an originally targeted functionality, resulting on it no longer being available for new systems to be produced or to be repaired. For example, when economically it is no longer interesting to produce a certain integrated circuit, a manufacturer may decide to stop its production, usually offering the customers a "last-time buy" of a certain quantity (nevertheless, sometimes due to technological reasons these components cannot be stocked for a long period; last-time buy then in reality is not more than short term mitigation to a still in needed for solution problem). After this quantity will have been used, no more components are available and the systems can no longer be produced.

SUBMITTED

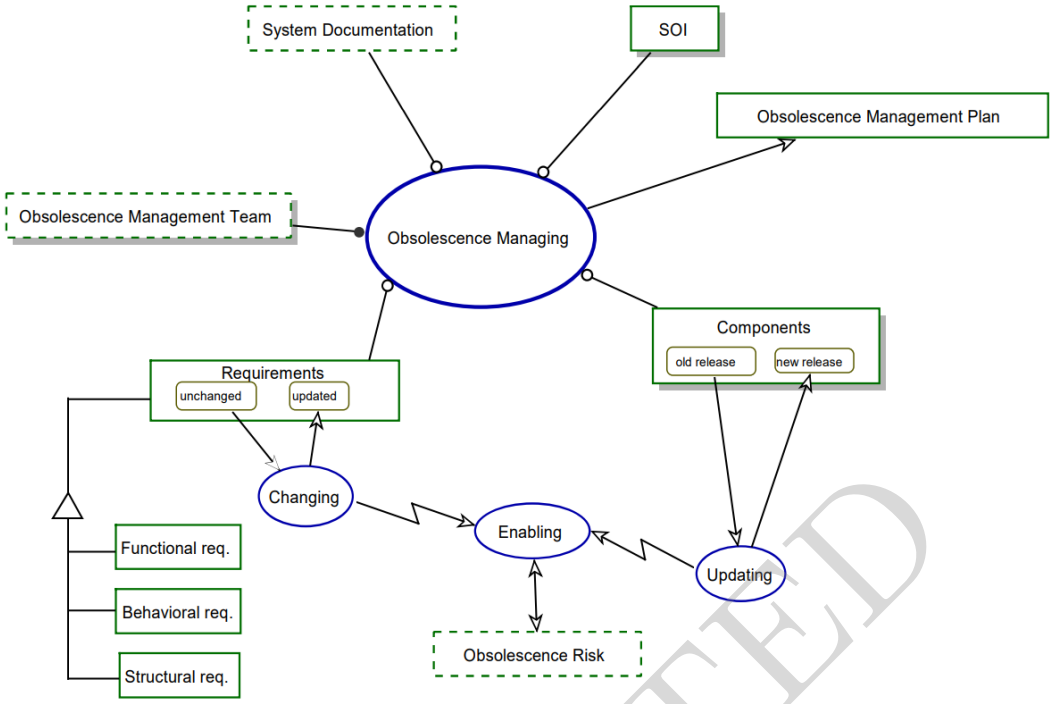


Fig. 2. Obsolescence Management System Object Process Diagram top-level specification.

A challenge is therefore to identify early on the design process which requirements are critical and which components are at a critical path. The use of models resulting from the application of MBSE methodologies, in this case through the use of ARCADIA, allows the chains of dependencies between the entities of the system architecture to be made explicit. These dependencies are specified in different models implemented by ARCADIA but the one that has a sufficient level of detail for the analysis of the propagation of the consequences of obsolescence is the physical architecture insofar as it not only shows the exchanges between components and functions but also specifies the mapping between components and functions (Roques, 2017).

3. Identification and Assessment approaches

This section details the proposal of identification and assessment phase approaches to tackle the obsolescence management early on the stage of design of a SOI. There are two prerequisites that must be met to put these approaches on use. The first is, as in the SD-22 proposal (SD-22, 2021), an Obsolescence Management Team shall exist. In addition, the system documentation should allow the elaboration of the main system architecture models, or they should already exist. How to produce these models is out of the scope of this paper. For the illustrative case of a weather forecasting system (see section 4) the following models are in Capella: Operational, System, Logical and Physical Architecture diagrams, plus available and explained in (Roques, 2017).

Systems are very often composed of a large number of components and modules; it is impractical to put them all under obsolescence monitoring. Therefore, it is necessary to carry out a first screening to identify those most at risk. The screening analysis can be done considering criteria such as those cited by (SD-22, 2021) such as: safety, mission criticality, cost, existing problems, life-cycle phase, sustainment

SUBMITTED

strategy (reflects maintenance possibilities). The screening should be able to identify a short-list of critical components to monitor. Using system architecture models, it is possible to extract the initial short-list of critical functions performed, totally or partially, by these critical components. Critical functions could be also found by analyzing the functional chains in ARCADIA. Since a functional chain represents a sequence of functions whose fulfilment enables the achievement of an operational capability of the system.

Dependencies represent a propagation channel between dependent entities. Therefore, determining the possible propagations of the consequences of obsolescence requires a precise mapping of dependencies within the system. The interdependencies between components (C-C), functions (F-F), requirement-function (R-F), requirement-component (R-C), as well as between function-component (C-F) need to be known. These interdependencies can be partly identified in the first levels of the ARCADIA methodology; however, we rely on the physical architecture (obtained in the last modeling step, cf. Fig. 7) which defines these dependencies precisely.

Fig. 3 presents an in-zoomed Object Process Diagram of the Obsolescence Management System in Fig. 2 as well as its representation in natural language, which is known as a fundamental advantage of OPM (Dori, 2016). As a reminder, the process of Obsolescence Managing as defined by (SD-22, 2021) includes the processes of: preparing, identifying, assessing, analysing, and implementing (in blue, cf. figure 3). This in-zoomed OPD was achieved following the systems modelling paradigm of OPM for representing objects and processes of a system. An object can be seeing as what a system or a product is, and a process as what the system does. This paper focuses on the identifying and assessing process.

SUBMITTED

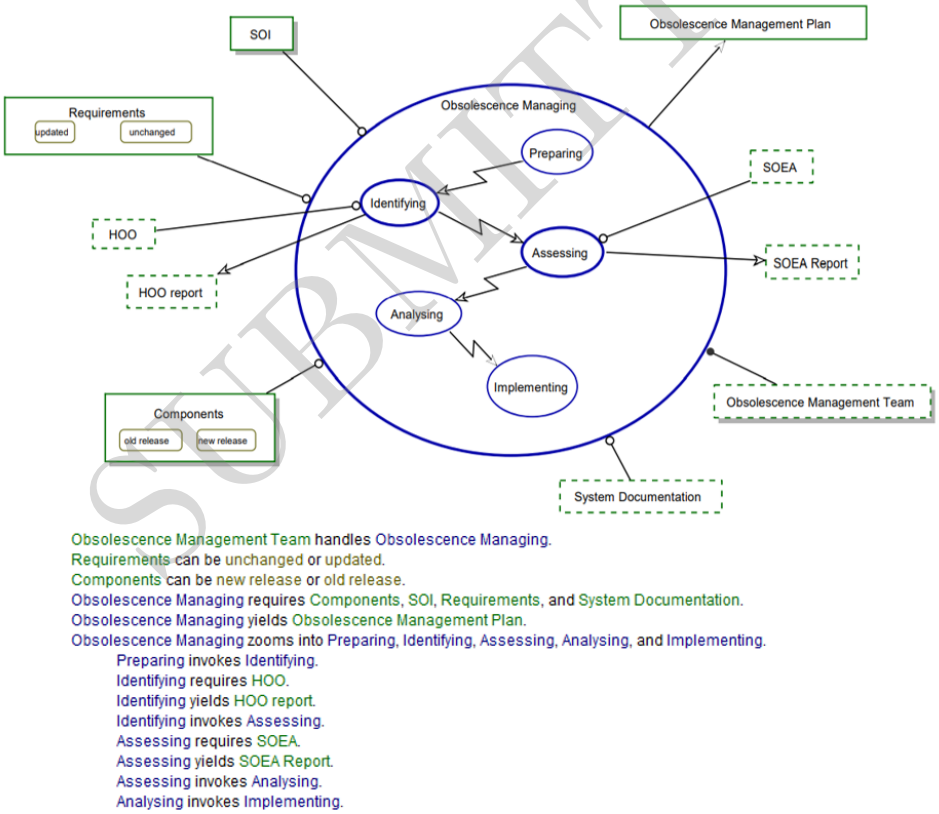


Fig. 3. Obsolescence Management System in-zoomed Object Process Diagram.

The initial critical components and functions are settled in the preparation and the identification phases of Obsolescence Management. Then the Identifying process requires the first approach called House of

Obsolescence (HOO), which is an object (in green as seen on Fig. 3). Section 3.1 defines the HOO, which allows the mapping of obsolescence risk propagation due to critical components. The usage of the HOO may also point out at some other hidden components and functions that could be impacted. The output of the application of HOO is a consolidated list of components and functions to monitor. The second approach, called System Obsolescence Criticality Analysis (SOCA) required by the Assessing process, permits to set a priority within a list of obsolescence risks is presented in section 3.2.

3.1. House of Obsolescence

The House of Obsolescence (Fig. 4) is a concept inspired by the well-known House of Quality (Pyzdek & Keller, 2014). It maps external-driven obsolescence changes to the consolidated short-list of critical functions and components. The approach analyses changes in requirements and components and maps their propagation.

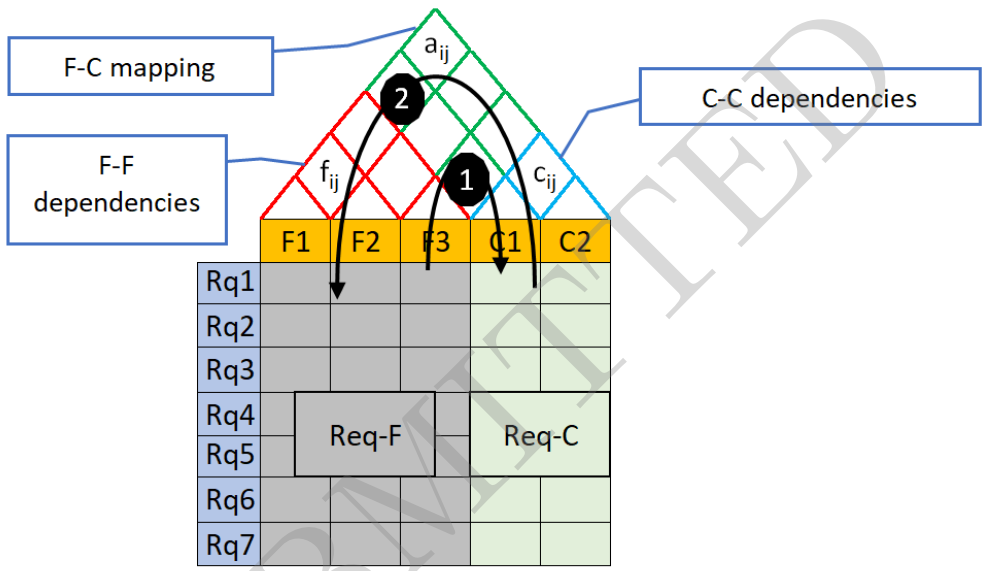


Fig. 4. House of Obsolescence.

SUBMITTED

The roof of the HOO shows the dependencies within the system. It enables to link the consequences of externally sourced obsolescence to system components and functions. Each column corresponds to a critical function or component. The C-C dependencies are mapped on the top of the components columns (colour light blue) and show whether there are any exchanges between every couple of considered components, c_{ij}. The idea is that when two components have a non-directed dependency between them (e.g. data transfer from a microcontroller to a memory), any change in the sender may affect the receiver. From an obsolescence point of view, problems with the sender could lead to modifications of receiver, and vice versa. These modifications are either "first-order hardware changes" or "first-order software changes", see (SD-22, 2021). c_{ij} can be either symbolic value (high, medium, low), or numerical:

- Boolean (1: with or 0: without),
- Natural values (for instance from 0 to 3 using an adopted measure scale) or
- Real values (from 0 to 1).

The F-F dependencies are mapped on the top of the columns that correspond to the functions (the triangle in red colour). These dependencies are gathered through the Boolean f_{ij}. The functions dependencies define

how the outputs of a function are used by others. For instance, the function "Collect Weather Data" supplies data to the function "Elaborate the Current Situation", (see Fig. 7). Therefore, any changes in the first function may impact the second. These are the functional dependencies which are identified since the System Analysis level of ARCADIA methodology (Roques, 2017).

At the very top of the roof of HOO the mapping between functions and the components can be found in colour green rhombus. It answers to the question of "who does what?". The mapping is valuated through the Boolean value of a_{ij}. For instance, in Fig. 7, it can be seen that the function "Acquire Temperature" is performed by the component "Temperature sensor".

It is important to notice the Req-F, Req-C, C-C, F-F and F-C dependencies could be also represented with DSMs (Design Structural Matrices). To be more exact, the HOO could be exchanged with a Multi Domain Matrix (MDM) that includes the Requirement, Component and Function DSM with the pertinent resulting Domain Mapping Matrices (DMM).The dependencies present in the HOO can be extracted from system models. In the case of EOLE (section 4) from the ARCADIA models of the studied SOI present in (Roques, 2017), which contains in-depth knowledge of the system described through the different levels of modelling of the ARCADIA method (cf. section 2.3).

The HOO assists to track the possible functions, components and requirements of the SOI that may be affected by an obsolescence risk. With the aid of the HOO roof it is possible to cascade the possible impacts on the components due to changes in the requirements, when going through the functions affected by a change in the requirements (represented by the arrow numbered "1", Fig. 4). It is also possible to see how changes in the components can impact requirements (represented by the arrow numbered "2" in Fig. 4).

The modeled dependencies are obtained through dependency transitivity; i.e. IF (Y depends on X) and (Z depends on Y) THEN (Z depends on X). This allows to chains the dependencies linking various (directly and indirectly) dependent entities together. Nevertheless, this reasoning process must take into account the following possibilities:

- the obsolescence risk of X may have no impact on Y, or
- that Y has to be modified and the dependency transitivity process is cancelled on Y, or
- that the mitigation resolution implemented on Y do impose modifications on Z. This would constitute the only case where the dependency transitivity is pursued to Z.

In other words, the exploitation of this transitivity can only be done through the inclusion of the listed uncertainties by using probabilistic models such as Bayesian networks for instance (Zolghadri et al., 2018). The key elements in controlling the propagation of obsolescence may be to break the chain of dependency, to reduce the likelihood of propagation through the implementation of remedial solutions throughout the dependency links, or to mitigate its possible impacts. Requirements changes could be due to imposed new environment regulations for instance, or changes imposed by the suppliers. For example, if the "Temperature sensor" has a new operating temperature range it could have an impact on the thermal requirements of the system. As it can be seen in Fig. 7 a change on the temperature sensor could have an impact on function "Acquire Temperature".

3.2. System Obsolescence Criticality Analysis

In the context of obsolescence criticality analysis, the first parameter to set is the time horizon H. H is a parameter that the developers have in mind at the beginning for the project linked with its visibility. It is a parameter taken into account for Obsolescence forecasting (IEC-62402, 2019; Jennings, Wu, & Terpenny, 2016; SD-22, 2021) and in general setting for forecasting for technology planning and road mapping as in (Yuskevich, Smirnova, Vingerhoeds, & Golkar, 2021). The purpose is to define the time frame beyond which the risk factor estimations are too uncertain to be usable but also below which the study loses all its meaning. For example, an analysis of the risk of obsolescence for a smartphone cannot be carried out over

SUBMITTED

a too short period of time (a few days since no changes is expected in such a short period) or over a too long period of time (a few years given the speed at which technological innovations are introduced).

The choice of this horizon H depends on factors such as: the life cycle and the remaining operational time of the system (e.g. the announced end of life of Python 2.0 and the suggestion to change to Python 3), the life cycle and the remaining operational time of the critical component under consideration, and the knowledge available on possible changes in customer needs and requirements (e.g. 5G spreading technology), but also the evolution of competing systems and competing technologies. Once this horizon identified, the rest of the analysis can be performed. Python 2.0 was released in 2000, but after a few years it was discovered that big changes were needed to put in place in order to improve Python. In 2006, Python 3.0 was released, nevertheless many people did not upgrade. To not discomfort the customers and users, for many years the both Python 2 and Python 3 were improved. Eventually the workload became too big, there were improvements Python 2.0 could not handle and it jeopardized the further improvement of Python 3.0. In 2008 the sunset of Python 2 was announced for 2015, and people were asked to upgrade before then. Not everyone upgraded, and in 2014 it was decided to extend that sunset till 2020. If knowing all of this a company would have decided to create a new product based on Python 2.0 in 2010-2015, the company was willingly accepting the future security risks and costs of porting the project to Python 3.0.

In order to prioritize the resources for obsolescence mitigation allocation, it is suggested to assign priority on how to approach obsolescence risk. A risk which can be defined in terms of a combination of impact and likelihood (ISO-31000, 2018). In order to perform this prioritization a System Obsolescence Criticality Analysis (SOCA) is proposed, inspired by the Failure Mode Effects and Criticality Analysis (FMECA) (IEC-60812, 2018), see Fig. 5. The critical system functions and components are listed in the rows. For each of them, it is first determined whether the risk is related to suitability or availability (column "b"). Suitability and availability refer respectively to obsolescence and DMSMS. This is to highlight that the solution to be deployed depends on the nature of the risk whether it is obsolescence (e.g. technological overrun) or DMSMS (the supplier who has stopped manufacturing the component).

Further in Fig. 6, based on the obsolescence classification presented in section 2.2, the obsolescence class is identified in the column "c". Each of the obsolescence risks could have an impact on the system requirements (see section 2.4). The effects in the system are identified and reported in column "d". The following columns,severity (S) and occurrence (O) which correspond to impact and likelihood respectively, define an obsolescence risk index for critical functions and components (Salas Cordero, Vingerhoeds, Zolghadri, & Baron, 2020). While the obsolescence criticality index is defined by the previous mentioned parameters and detectability (D) (see Fig. 5). The values for severity, occurrence and detectability can be attibuted by subject matter experts (SME) with the help of different scale bands or categories as the following:

(1). Occurrence: linked with the likelihood of the appearance of an obsolescence issue. The proposed scale bands for the SME in this case is: Certain, Likely, Unlikely, and Rare.

(2). Severity: relative ranking of potential or actual consequences of the obsolescence issue related to the effects (column d) on the system. The scale indicators for severity are: Catastrophic, Major, Important, Minor, and Negligible.

(3). Detectability: represents the likelihood with which an obsolescence risk is expected to be detected before the actual obsolescence issue arises. For detectability the proposed scale is: Certain, Likely, Unlikely, and Rare. There are various possible ways for a company to discover obsolescence risks. In some cases, there is documented information (published regulations or any discontinuance disclaims).

The Obsolescence Criticality Index, OCI, is then calculated:

$$OCI = S \times O \times D \tag{1}$$

SUBMITTED

Temporal horizon: N years					Obsolescence value				
System obsolescence risk		***Availability/ Suitability***	***Obsolescence Classification***	***Effect***	***Severity***	***Occurrence***	***Detectability***	***Obsolescence criticality index***	***possible mitigation measures***
A		***b***	***c***	***d***	***S***	***O***	***D***	***OCI***	***m***
Components	C1								
	C2								
Functions	F1								
	F2								
	F3								

Fig. 5. System Obsolescence Criticality Analysis form

For the critical analysis it was decided to use the scale as proposed by the French Standardization Association (AFNOR) (Simaillaud, 2017) from 1-10 as it can be seen in Fig. 6. AFNOR defines two situations under which a risk should be considered for future monitoring or treatment (orange boxes in Fig. 6). The practicality of this consideration led authors to use these two situations as potentially important signs of obsolescence risk as well. But clearly, the operation of SOCA can be adapted to the context of the study depending on the nature of the industry, company or product.

- When the severity is 10.
- When the OCI is greater or equal to the criticality threshold (CT). This critically threshold can be reviewed. For the EOLE illustrative case it will be considered that the CT=100.

The resulting obsolescence risks are then ordered in the diminishing direction of the OCI.

Obsolescence Criticality Index																		
Criticality threshold: 100 (or 10 if G=10)		***O***	*Rare*				*Unlikely*				*Likely*				*Certain*			
			1				***4***				***7***				***10***			
		D	*Certain*	*Likely*	*Unlikely*	*Rare*	*Certain*	*Likely*	*Unlikely*	*Rare*	*Certain*	*Likely*	*Unlikely*	*Rare*	*Certain*	*Likely*	*Unlikely*	*Rare*
			1	***4***	***7***	***10***	***1***	***4***	***7***	***10***	***1***	***4***	***7***	***10***	***1***	***4***	***7***	***10***
G	*Negligible*	***1***	1	4	7	10	4	16	28	40	7	28	49	70	10	40	70	100
	Minor	***3***	3	12	21	30	12	48	84	120	21	84	147	210	30	120	210	300
	Important	***6***	6	24	42	60	24	96	168	240	42	168	294	420	60	240	420	600
	Major	***9***	9	36	63	90	36	144	252	360	63	252	441	630	90	360	630	900
	Catastrophic	***10***	10	40	70	100	40	160	280	400	70	280	490	700	100	400	700	1000

Fig. 6. Obsolescence Criticality Index and values requiring obsolescence monitoring.

4. Illustrative Case: EOLE

The Environment Observation Link to Earth (EOLE) case, developed by (Roques, 2017), is used to illustrate partially the proposed approaches. EOLE is composed of an acquisition and a ground system. The acquisition system is a sounding balloon launched into the atmosphere in charge of data collection using

SUBMITTED

sensors and data transmission to the ground system. The pressure and temperature are the two main sensors of the payload. The ground system is in charge of data acquisition planning, collecting the data from the acquisition system. The system is modelled using ARCADIA methodology, supported by Capella. All details of the use case can be found in (Roques, 2017).

The ARCADIA methodology allowed to identify the Sounding balloon and Ground station; the yellow boxes in the middle of the physical architecture, see Fig. 7. The Sounding balloon is composed of two modules: Sensor-Holder and Nano-Computer. The Ground station is made of Publication and Processing Servers. The deep blue boxes define the mappings between the functions and the components of the system. For instance, the Temperature Sensor performs the function Acquire Temperature, i.e., defining "Who does What". The external entities and their exchanges with EOLE are represented in clear blue: Earth atmosphere, Weather operator, etc. They contain their respective functions. The exchanges between the internal and external entities are represented by oriented arcs.

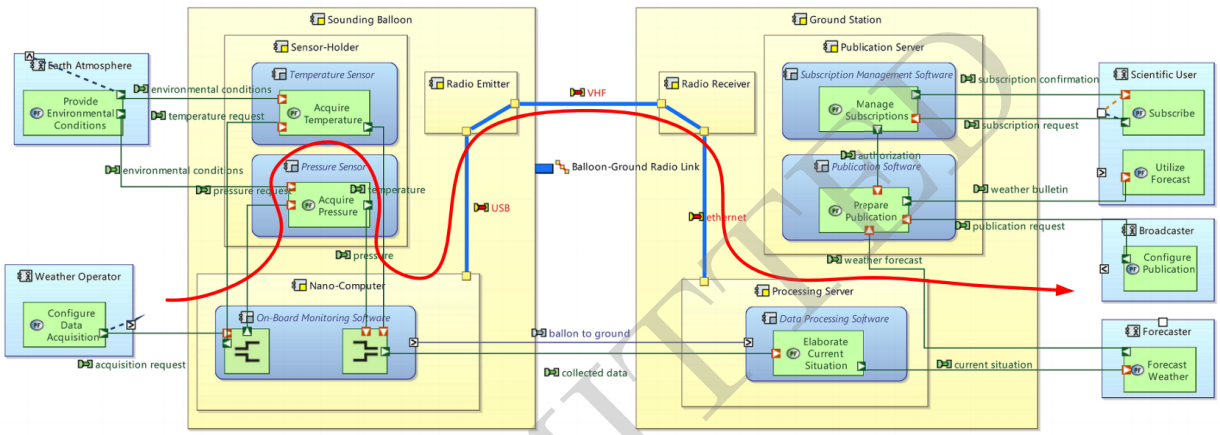


Fig. 7. Risk analysis by System Obsolescence Criticality Analysis, and OCI; Capella Physical Architecture model. (Adapted from (Roques, 2017).

SUBMITTED

Obsolescence risk rationale. For this illustrative case it was considered that the VHF technology has shortcomings comparing to UHF (Ultra High Frequency, 430MHz). The advantage of UHF is the reduction of interference due to a more accessible frequency spectrum. This contributes to the reliability of data transmission between the two sub-systems: the Sounding balloon and the Ground station.

1) **HOO**. Remind that the goal is to map requirements (Req.) to system functions and components. For this illustrative case it is considered that the Req. 01W is linked with the following functions:

- Emission and Reception: transmissions of the Sounding Balloon with the Ground station (by Radio Emitter and Receiver).
- Elaborate Current Situation (by Ground Station).

This requirement can be found on Fig. 8 at the beginning of the red dotted line in the HOO roof. The red, blue and black dotted line represent the entire dependency chains. The blue line shows how a change in the emission function might affect the radio emitter. Whether if a change in Req. 01W would affect the emission function but would not propagate to the reception function then the elements through which the black dotted line goes would not be affected. The two components, radio emitter and radio receiver, are central to this functional chain; the obsolescence mitigation/resolution may have a direct impact on them. While the processing server may experience or not an indirect impact in case any measure is taken. Therefore, the radio receiver and emitter are critical and should be monitored. Further it is necessary to estimate its OCI through the use of the SOCA matrix.

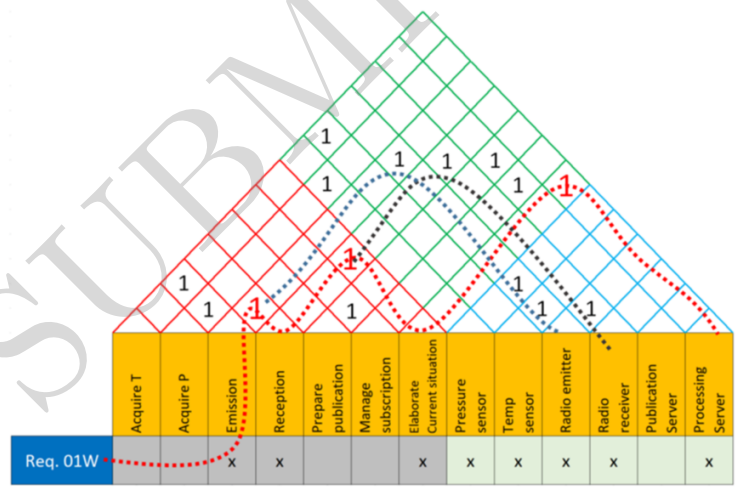
2) **SOCA application**. Its goal is to assess the criticality of obsolescence risk, and to brainstorm mitigation strategies. This obsolescence risk is related to the suitability, column b of the SOCA matrix. It corresponds to a technological obsolescence according to (Bartels et al., 2012); column c. The main effect (column d) is the possible message errors due to the interferences. This means the system provided service may be degraded, or in fact out of service (OOS). Suppose that the analysis is performed in 2019 and the horizon of the study is 5 years due to the newer version of EOLE that is predicted to be sold. The computation of the OCI is obtained based on the expert assessment of severity, detectability and occurrence:

- S = 9: the obsolescence will degrade the service usability of the whole system.
- O = 7: the UHF technology is already available; the obsolescence of VHF is highly probable.
- D = 1: the state-of-the-art of radio transmission is easily available; there is no need for any specific effort to detect it.

The OCI in this case is equal to 63.

Since the severity is less than 10 and the overall index lower than 100, according to the previously defined critical levels (section 3.2) there are no measures to be taken at the moment during the obsolescence management process, but to continue monitoring. However, the significant value of the severity level defined by the subject matter experts (S=9) may lead to a more in-depth analysis determining the obsolescence mitigation solution that could be considered. If this is the case, the comparison of the available solutions (see Appendix) should lead to the least expensive solutions, i.e. the ”Simple Substitute” or ”Complex Substitute”, as the implementation of the UHF solution will require adaptations, for example in the installed programs of the Nano-computer. The final choice of resolution to implement requires more technical definition which is out of scope of this example.

SUBMITTED



Temporal horizon: 5 years				Obsolescence value				
System obsolescence risk	*Availability/ Suitability*	*Obsolescence Classification*	*Effect*	*Severity*	*Occurrence*	*Detectability*	*Obsolescence criticality index*	*possible mitigation measures*
A	*b*	*c*	*d*	*S*	*O*	*D*	*OCI*	*m*
Radio Emitter, Receiver.	S	Tech	OOS	9	7	1	63	2,6 or 7

Fig. 8. HOO and SOCA of EOLE.

5. Conclusions and perspectives

This paper addressed the fundamentals of obsolescence and proposed two approaches that aim to support the key mechanisms of obsolescence understanding, propagation and mitigation for a given application.

Obsolescence issues are one of the main costs in the life-cycle of sustainment-dominated systems, those that require support for many decades. This paper has identified the main drivers for obsolescence as being related to the system requirements, and underlying sources such as legislation, and the actual components used for the realisation of the system.

Starting from the observation that a very significant part of the system is defined in the early design stages, this paper advocates for addressing obsolescence assessment and management from the earliest design stages. This paper therefore proposed two tools building on a model-based systems engineering approach to proactively assess obsolescence risks.

The application of the proposed tools allows leading obsolescence management teams towards a targeted consideration of the system components and functions likely to be impacted by obsolescence. The House of Obsolescence seeks to guide towards a mapping of how the modifications of requirements or components affect the system of interest. Once the main components and functions that could be affected have been identified, the use of SOCA should lead the team to estimate the priority within the obsolescence risks and the techniques that can be put in place to mitigate them. The use of systems engineering models is fundamental to the operation of these tools.

For future perspectives, different approaches can be investigated and are currently being explored by the authors. An option could be to build upon the predictive models that can be obtained from the proposed tools (probabilistic graphs and mainly the Bayesian networks) and to use them in the determination of components and functions impacted by obsolescence. Another option could be to build upon Design Structure Matrices (DSM) as modelling tool (Salas Cordero et al., 2020) and using this approach to assess the critical components. During early stages of design, different architectures could be analysed and the results from such approaches, taking into account technology and/or component maturity for the given application, may then lead to a complementary view on obsolescence risk.

Further perspectives include evaluating the impact of early obsolescence risk analysis of the design on the life cycle of a system, as well as observing how design changes propagate when utilizing these tools method during the conceptual stage on the rest of the system development whilst actively tracking the risks for each design decision made.

SUBMITTED

Acknowledgments

This paper is an extended and revised version of the paper: Analysing the Impacts of System Obsolescence based on System Architecture Models by Zolghadri M., Vingerhoeds R., Baron C., Salas S., presented at the 13th International Tools and Methods of Competitive Engineering Symposium (TMCE), 11-15 May 2020, Dublin. (Marc, Vingerhoeds, Baron, & Salas Cordero, 2020)

References

Bartels, B., Ermel, U., Sandborn, P., & Pecht, M. G. (2012). *Strategies to the prediction, mitigation and management of product obsolescence* (Vol. 87). John Wiley & Sons.

Bradley, M., & Dawson, R. J. (1998). An analysis of obsolescence risk in it systems. *Software Quality Journal*, *7*(2), 123–130.

Brazier, F., van Langen, P., Lukosch, S., & Vingerhoeds, R. (2018). Design, engineering and governance of complex systems. In *Projects and people – mastering success* (p. 34-59). NAP Foundation Press.

Bulow, J. (1986). An economic theory of planned obsolescence. *Quaterly J. Econ.*, *101*(4), 729–750.

Crawley, E., Cameron, B., & Selva, D. (2015). *System architecture: Strategy and product development for complex systems*. Prentice Hall Press.

Dori, D. (2016). *Model-based systems engineering with opm and sysml*. Springer-Verlag.

EDSTAR. (2016). *Expert group 21 : Obsolescence management final report* (Report).

Gero, J. (1990). Design prototypes. a knowledge representation schema for design. *AI Magazine*, *11*(4).

Hick, H., Bajzek, M., & Faustmann, C. (2019). Definition of a system model for model-based development. *Applied Sciences*, *1*(9), 1074.

IEC-60812. (2018). *Failure modes and effects analysis (fmea and fmeca)* (Standard).

IEC-62402. (2019). *Obsolescence management* (Standard).

ISO-31000. (2018). *Risk management* (Standard).

ISO/IEC-19514. (2017). *Information technology — object management group systems modeling language (omg sysml)* (Standard).

ISO/IEC/IEEE-15228. (2015). *Systems and software engineering — system life cycle processes* (Standard).

Jennings, C., Wu, D., & Terpenny, J. (2016). Forecasting obsolescence risk using machine learning. In *11th international manufacturing science and engineering conference.* ASME. doi: 10.1115/MSEC2016-8625

Kaslow, D., Ayres, B., Cahill, P., & Hart, L. (2018). A model-based systems engineering approach for technical measurement with application to a cubesat. In *2018 ieee aerospace conference* (pp. 1–10). IEEE.

Kreziak, D., Prim-Allaz, L., & Robinot, E. (2017). *Rapport de recherche du projet coop, consommateurs et objets à obsolescence programmée.* (Report).

Marc, Z., Vingerhoeds, R., Baron, C., & Salas Cordero, S. (2020). Analysing the impact of system obsolescence based on system architecture models. In *International symposium on tools and methods of competitive engineering (tmce).*

Merola, L. (2006). The cots software obsolescence threat. In *Fifth international conference on commercial-off-the-shelf (cots)-based software systems (iccbss'05)* (pp. 7–pp).

Pyzdek, T., & Keller, P. (2014). *The six sigma handbook: a complete guide for green belts, black belts, and managers at all levels*. McGraw-Hill Companies.

Roques, P. (2017). *Systems architecture modeling with the arcadia method: A practical guide to capella*. Elsevier.

Salas Cordero, S., Vingerhoeds, R., Zolghadri, M., & Baron, C. (2020). Addressing obsolescence from day one in the conceptual phase of complex systems as a design constraint. In *Product lifecycle management enabling smart x* (pp. 369–383). Cham: Springer.

SD-22. (2021). *Diminishing manufacturing sources and material shortages (dmsms) a guidebook of best practices for implementing a robust dmsms*.

SEBOK. (2015). *Systems engineering body of knowledge.* Retrieved from https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)

Simaillaud, B. (2017). Phase analyser. *AFNOR Training 216, version 7*.

SysML-V2. (2020). *Sysml v2 (2020), introduction to the sysml v2 language.* Retrieved from http://openmbee.org/sysml-v2-release/2020

Walden, D., Roedler, G., Forsberg, K., Hamelin, R., & Shortell, T. (2015). *Systems engineering handbook: A guide for system life cycle processes and activities*. Wiley.

Wilkinson, C. (2015). *Dot/faa/tc-15/33: Obsolescence and life cycle management for avionics* (Report).

Yuskevich, I., Smirnova, K., Vingerhoeds, R., & Golkar, A. (2021). Model-based approaches for technology planning and roadmapping: Technology forecasting and game-theoretic modeling. *Int. Journal on Technological Forecasting and Social Change*, *168*, 120761.

Zheng, L., Sandborn, P., Terpenny, J., & Orfi, N. (2014). An ontological model and method for obsolescence resolution and management. In *24th international conference on flexible automation and intelligent*

SUBMITTED

manufacturing, faim 2014 (pp. 185–192).
Zolghadri, M., Addouche, S.-A., Boissie, K., & Richard, D. (2018). Obsolescence prediction: a bayesian model. *Procedia CIRP*, *70*, 392–397.

Author Biographies

Sophia Salas Cordero is a Ph.D. Candidate in the Department of Complex Systems Engineering at ISAE-SUPAERO, France. Sophia obtained her BSc and MSc in Mechatronics and Robotics at Bauman Moscow State University; where she specialized in Control Systems. Later she obtained a MSc in Space and Engineering Systems at Skolkovo Institute of Science and Technology. Experienced in working with Systems Engineering Methodologies, previous research on Conceptual Concurrent Design and MBSE applied to Space Systems. Her current research interests include Systems Engineering, Design Engineering, Concurrent Engineering, Risk and Obsolescence Management.

Marc Zolghadri is Professor of manufacturing operations management at ISAE-Supmeca (Paris). He is a Doctor of Operations Management from Bordeaux University, France, where he studied Supply Chain Planning and Scheduling issues. Marc has been involved in numerous national and international contracts and projects. He has collaborated with many companies in different industrial sectors such as Railway industry, Fragrance and Cosmetic industry, Diamond Industry, Automobile OEM, Agricultural machinery. His research covers two areas. The first concerns the re-design and re-configuration of manufacturing production systems and control of digital transformation of companies. The second series of his work concerns change engineering. In particular, the scope of application is that of obsolescence and its propagation mechanisms. In this sense, the research work is focused on determining the tools to help in designing and choosing obsolescence mitigation solutions. Based on system engineering, his work is being carried out to propose techniques for designing systems that are resilient to obsolescence. He is a member of the Interest Force "Obsolescence" of the French National Agency for Standardization. He is also an expert on the French Committee for Standardization in Obsolescence, and IEC. He is co-founder and president of the French Institute of Obsolescence.

Rob Vingerhoeds is Full Professor of Systems Engineering and Head of the Department of Complex Systems Engineering at ISAE-SUPAERO, Université de Toulouse, France. Rob started his career as academic researcher, first in the aerospace field at Delft University of Technology, then in the field of automation and intelligent real-time systems at the universities of Ghent, Delft and Swansea, which period allowed him to get into systems engineering, a field that would become a key topic in his career. Aiming to acquire industrial experience, Rob integrated a multinational automotive supplier, during which time, he contributed to a wide range of activities at all industrial levels with systems engineering as red line. He is Deputy Editor of the International Scientific Journal "Systems Engineering". Rob's today research interests include systems engineering and architecture, model-based systems engineering, concept design, the integration of project management and systems engineering, and artificial intelligence techniques.

Claude Baron is Full Professor in Computer Sciences at INSA, Université de Toulouse, France. She conducts research in systems and software engineering at the LAAS-CNRS laboratory where she leads the Systems Engineering and Integration team. She is interested in modelling, optimizing and smoothing product development processes. Her research work is based on a multidisciplinary and collaborative vision of the design of complex systems. She addresses system modelling and process monitoring, considering embedded and critical systems and applications in avionics and automotive.

SUBMITTED

A. Appendix

The obsolescence mitigation/resolution methods (SD-22, 2021).

	Mitigation/ Resolution	***Definition***
1	No solution required	Existing stock will satisfy future demand.
2	Approved item	The issue is resolved by the use of items already approved on and still in production.
3	End-of-need buy	A sufficient quantity is purchased to sustain the product until its next technology refreshment or the discontinuance of the host assembly.
4	Repair	The issue is resolved by: Repair, Reclamation of items from marginal, out-of-service, or surplus materials, … to ensure continued support.
5	Extension of product or support	The supplier is incentivized to continue providing the obsolete items.
6	Simple substitute	The item is replaced with an existing item that meets all requirements without modification to either the item or its Next-Higher Assembly and requires only minimal qualification.
7	Complex substitute	A replacement item that has different specifications but requires no modification of the origin product or the NHA, is researched and validated.
8	Development of a new item or origin	A replacement product is developed that meets the requirements of the original product without affecting the NHA.
9	Redesign-NHA	The affected item's NHA must be modified. Only the NHA is affected, and the new design will not affect anything at a higher level.
10	Redesign–complex/ system replacement	A major assembly redesign affects assemblies beyond the obsolete item's NHA and may require that higher level assemblies, software, and interfaces be changed.

Fig. A.1. The obsolescence mitigation/resolution methods (SD-22, 2021)

SUBMITTED