

# Evaluating Emulation and Migration: Birds of a Feather?

Mark Guttenbrunner and Andreas Rauber

Secure Business Austria, Vienna, Austria  
Vienna University of Technology, Vienna, Austria  
`{mguttenbrunner, araubert}@sba-research.org`

**Abstract.** Evaluating the results of a digital preservation action, be it migration or emulation, is a complex task. The usual approach for migration is to evaluate object properties after the migration. For emulation strategies the result of the rendering of the object is evaluated. In this paper we argue that the change of object properties when migrating is not sufficient evidence if a digital preservation action is successful or not. Even for migration the rendering process of the digital object is crucial, and as such evaluating object properties is not enough. The difference in evaluation between emulation and migration as a strategy for digital preservation becomes blurred as migration results have to be compared based on the rendering of the target format and the environment used to render the migrated digital object. Evaluation of object properties when migrating will only validate a necessary condition for preserving significant properties, i.e. whether the information underlying a specific property is still present in an object after migration. It cannot guarantee that the rendering based upon the migrated object will exhibit a specific significant property. In this paper we show the view-path of digital objects and explain how emulation and migration actions affect it. We then compare the changes that occur in the view-path and show that these are at least as severe when migrating a digital object as when emulating its rendering environment.

## 1 Introduction

Emulation and migration as the main strategies in digital preservation are usually treated as entirely different strategies. While for evaluating the success of a migration action the object properties are compared to check if significant properties of the object change, the rendering environments of the digital object (i.e., the environment the object was originally rendered in and the environment it will be rendered in after the migration) are quite frequently not taken into account. With emulation, on the other hand, the digital object does not change, so only the rendering of the digital object in the original environment and the rendering in the emulated environment are compared to see if the rendering is identical.

What is usually not considered in the evaluation of a migration action is that every extraction of significant properties of an object is already a form of

rendering i.e. interpretation of this object. Even though not necessarily directly visible to the user, the object is rendered by the routines used to extract the properties. The problem with this approach is that the program "rendering" the object is neither necessarily the program originally used to render it nor the one that will be used to render it and thus the results are not necessarily authentic to the original rendering once the object is rendered in a different environment. (Note that "rendering" in this context is not restricted to the visual display of an object. It refers to all kind of interpretations of an object and the resulting effect on an environment, be it visual, acoustic or effects on a system state, files stored on media, or communication/voltage levels on I/O ports, etc.)

In this paper we will show how migration and emulation strategies affect the view-path of a digital object on different levels. By applying a digital preservation action the view-path is changed, both for emulation and migration actions. The rendering using the new view-path has to be compared to the object rendered in the original view-path to evaluate changes of significant properties of the object when rendered in the new view-path. While this is common knowledge when evaluating emulators this paper will emphasize the similarities in rendering of a migrated object and the changes in the view-path. Both the evaluation of a migration action and an emulation action should thus always include the combination between object and rendering environment.

This paper is structured as follows. First, we provide an overview of work relevant for this paper. In Section 3 we examine the generic view-path of rendering any digital object and describe different layers that appear in the view-path and on what levels emulation can take place. Next, we show how the view-path changes with emulation on the different levels in Section 4. In Section 5 we then explain how the view-path changes when migrating a digital object. Next, we compare the resulting view-paths after migration and emulation and argue why there should be no difference in evaluation between migration and emulation. Finally, in Section 7 we present our conclusions and give an outlook to future work.

## 2 Related Work

Rothenberg argues, that „Digital Informational Entities are Executable Programs” [7], i.e. that every digital object is a program that has to be interpreted by a process that knows how to perform the commands in the formal language (the format) the program is written in. This can be as simple as interpreting a string of ASCII character codes to make it human readable. Most of these interpreters are software, but on the lowest level the machine-code is interpreted by hardware (i.e., the CPU of a system).

This stack of interpreters used to decode a digital object is called the view-path. Van Diessen describes it as „a full set of functionality for rendering the information contained in a digital object” [10]. The view-path contains the hardware and all the secondary digital objects needed to render an object along with their configuration. It is possible to use different view-paths to display the same object (e.g., different viewer applications that can render the same digital object).