

The Branding Analysis Pattern

Huascar A. Sanchez¹ and Mohamed E. Fayad²
Computer Engineering Dept., College of Engineering, San Jose State University
One Washington Square, San Jose, CA 95192-0180
¹hsanchez@email.sjsu.edu, ²m.fayad@sjsu.edu

Abstract.

Branding is a concept that spans multiple areas of application. It is used to create identifiable associations between any type of entity and its brands, with the sole purpose of differentiating it from its peers. This paper introduces the Branding analysis pattern. This pattern focuses on providing a conceptual model that represents any type of branding need in any domain. To achieve this, we used Software Stability Models "SSMs" [1, 2, 3, 4] as the main recipe. This paper also provides full pattern documentation. Additionally, in order to demonstrate the application of this pattern, we provide two examples: one related to HCI, and the other one related to Marketing.

Keywords: Branding, Branding Analysis Patterns, Software Stability Models.

I. Introduction

Branding is a mental construct that establishes a connection between your desires and a particular entity. This connection is achieved through the use of a combination of logos, names, trademarks, tag lines, color palettes, sounds and tones, and physical and behavioral features in a manner that reflects the emotional make-up of any entity. Branding's origin references to the earliest shepherds that branded their cattle to identify them and differentiate them from other shepherds' cattle. Nowadays, this activity has found great acceptance in areas as diverse as Marketing and Forensics. In Marketing, branding has emerged as the means for both enhancing customers' perception with respect to an identifiable product, establishing a pervasive means to communicate information and mark differences among this product and its peers, i.e. Nike,

Starbucks, Apple, Microsoft, etc. In forensics, for example, branding is used as the means for identifying and differentiating suspects from other suspects in a determined case, i.e. Biometric systems.

Branding, as an activity, is explicitly and implicitly used in different application areas that look for a distinctive identity for a particular thing, either real or conceptual, about which the system needs to hold certain information (Entity). These application areas are different in nature and requirements, especially when covering different entities, which physical or behavioral differences are hard to recognize. Unfortunately, these software products are bound to their context of application; providing then, not enough flexibility for them to be scaled and reused in other areas. As a result, these products will be constantly involved in multiple adaptations to satisfy their specific needs from a determined area. This brings as a consequence two things: 1 - it is overwhelming for software developers to cope with vast segment of changes in their software in a cost effective manner, especially when trying to use the branding core in different application areas, and 2 – software products are more opened to bugs and serious ripple effects that may jeopardize their use [8]. These consequences demonstrated that: 1 – there is wide variance in knowledge and resources used to brand any entity; 2 – it is difficult to assure a true scalability, and reusability in software solutions, especially when dealing with branding in multiple domains.

The question of how to push for a standard solution that covers different branding mechanisms or processes, and a myriad of entities and identity types to address different needs, from identity creation to branding assessment is a complicated task. We cut through this complexity by providing the core knowledge of branding, which is a domain neutral and fully scalable core, using Software Stability Models “SSM” [1, 2, 3, 4]. By doing this, we will help branding providers and practitioners to use an insightful solution to solve a set of recurrent problems that require branding.

Software Stability Models "SSMs" are the visual representation of Software Stability concepts. They categorize the aspects that exist in a determined domain based on their rationale and significant nature. In other words, they separate the domain enduring knowledge from transient aspects. The domain enduring knowledge is represented by: Enduring Business Themes (EBTs) or domain specification classes, and Business Objects (BOs) or the significant capabilities of EBTs. The transient aspects are represented by Industrial Objects (IOs) or context-specific classes. These transient aspects will form the specific context where the branding core will be used. The adaptation to these specific contexts is done by attaching the IOs to the branding core by means of using Hooks or extension points [5, 8].

The rest of the paper is organized as follows: Section II describes the pattern using a pattern template. We conclude the paper in Section III. Finally, a list of references is provided in Section IV.

II. Pattern Structure

Pattern Name: The Branding Analysis Pattern.

Pattern Intent: To provide a standard solution that can be utilized to handle any branding need in different domains.

Context: Branding is explicitly and implicitly used in different areas, such as Security systems, HCI, Marketing, Game industry, Virtual Reality, etc. For instance, branding is used in security systems, especially in clearance systems, to assure a unique individual's identity. This identity will be validated by means of using certain physical and behavioral features, such as retinal patterns, fingerprints, iris patterns, voice, etc. In Virtual Reality, especially in virtual collaborative

environments, branding is used to identify an immersed user in a virtual world and differentiate this user from his/her peers. This identification and differentiation is achieved by means of using avatars, and other types of physical and behavioral features, such as voice, body shape, walking mode, etc.

We have just seen that branding is used in multiple domains, different in nature and with specific needs, to develop an entity's identity. The value of this entity's identity can be recognized locally or globally by its observers. For example, commonly known brands reflect such value recognition: signs, like Apple and AlienWare; shapes, like Apple's Ipad, Mercedes Benz, and Ferrari; Typography, like those from IBM, Microsoft, or Costco; or Words, like Pepsi, McDonald's, Starbucks, Nike, Oracle, and much more. These brands build emotional and cultural responses that seem to shape a well-formed and identifiable image in observers' minds.

Problem:

Branding is a subject that exists in multiple domains covering any entity's identification and differentiation necessity. However, it has happened that the addressing of this subject in one particular domain is different than in other domains due to different requirements or needs i.e. in HCI, branding supports the creation of intuitive and user-centered user interfaces, and in the Game Industry, where games are developed with certain capabilities that allow players to changed the physical and behavioral features of their selected game characters at will. As you have seen, each of the examples targets a different context with different needs, but they all address, implicitly or explicitly, directly or indirectly, the same necessity, which is branding. The question is: do we have to implement this solution all over again every time we have to cope with new requirements and new contexts or applicability? The answer is, "No, we don't have to!" because we can have a standard solution that adapts to changes with ease, and applies to different contexts with not side effects whatsoever, such as constantly patching the software

product (upgrades) to satisfy transient needs, or the serious ripple effects product of scaling up/down/in/out our software solution [8] to work in constrained or unconstrained environments.

So, where is that standard solution that will be used in different domains to handle any branding need? Before defining this solution, we will provide a full list of challenges that we may face when defining this standard solution.

Pattern Challenges:

The following are the challenges we overcome in this paper.

- Branding possesses a domain neutral nature, which main purpose is to create any type of association between entities and their brands regardless of the context of application. Therefore, branding requires an enduring structure applicable to any domain, without losing the branding's main purpose.
- Branding can take a vast number of entities, real or conceptual. As such, it is a great challenge to define a standard structure applicable to any entity.
- Branding's identifiable image is achieved by different types of identity, such as logos, names, trademarks, tag lines, color palettes, palm prints, facial images, sounds and tones in a manner that reflects the uniqueness of any entity. Therefore, we need an aspect that handles the wide segment of identity forms.
- Branding is performed and/or recognized by a vast number of practitioners in a given geography. It is a challenge to handle distinct practitioners' roles, within branding, by means of using one significant aspect.
- Branding requires an ultimate workhorse: the actual brand. This workhorse represents the mental construct that identifies the origin of any entity. This mental construct is driven by certain qualities, such as brand distinctiveness, disruptiveness, effectiveness,

appeal, etc. and much more [6]. The act of handling these qualities becomes a great challenge that must be covered when creating any brand.

- An entity's association with its brands is created by invoking a determined branding mechanism. However, the number of mechanisms available for branding can be overwhelming. Therefore, it is necessary to have a significant concern that handles this vast number of mechanisms.

Pattern Constraints

The following are some of the perceived constraints applied to the analysis pattern.

- We must select a process or mechanism before branding an entity. This selection is performed by a particular party.
- Branding analysis pattern per se does not provide the means for deploying its use across different media. This necessity can be fulfilled by extending the pattern and adding the AnyMedia design Pattern to it.
- A brand requires at least a type of identity to be used in the association of a brand to an entity, i.e. logo, color palette, fingerprints, palm prints, or name, etc.
- Branding analysis pattern must allow the branding of any entity as a whole, and the branding of the entity's sub-entities or elements.

Pattern Solution:

Our solution focuses on the branding concept, and tries to extract its main components, leaving domain-specific aspects away from the model. These components are general enough, so they can be used by developers to satisfy certain needs. See figure 1.

- AnyIdentity: Represents the sum of all characteristics, tangible and intangible, by which an entity is recognized or known.
- AnyMechanism: Represents any particular mechanism(s) that creates an identifiable brand.
- AnyParty: Defines the different roles that a specific party, in the branding process, can take.

Pattern Benefits:

The use of the Branding Analysis Pattern provides the following benefits:

- Branding solution will not change whenever it appears regardless the domain where is being used.
- Branding solution will cope with a myriad of heterogeneous Entities by means of using the AnyEntity pattern-BO.
- It allows the communication of any brand across a wide segment of parties. The different types of parties are represented by the AnyParty pattern-BO.
- Branding solution can handle multiple types of identity at the same time for a single entity, or group of identities.
- Branding solution provides a core abstraction for any mechanism which captures the common behavior and structure of all types of mechanisms.
- Captures the elements and behaviors of an identifiable aspect associated with any entity. This is represented by the AnyBrand aspect.

Examples

In order to illustrate the use of the Branding Analysis Pattern, we provide two examples. Each one of them is different in nature and in the number of requirements involved. The first scenario concentrates on showing the use of dynamic branding for user interfaces. The second scenario describes the process of branding in DNA Fingerprinting detection system.

Scenario 1, Dynamic Branding for GUIs [7]: A skin is a collection of images and a definition file, which together describe an application interface. On Windows, Linux, and other operating systems, an increasing number of applications (and games) allow users to completely change the look and feel of the user interface. Why should we use skins? There are some technical and marketing considerations to answer this question. The first consideration allows handing over of full control of your GUI layout, and second represents a big selling point for branding practitioners, since after branding any application's GUI for any company, it is possible to re-brand the same application for another company by easily changing the skins. Figure 2 shows the model for scenario 1.

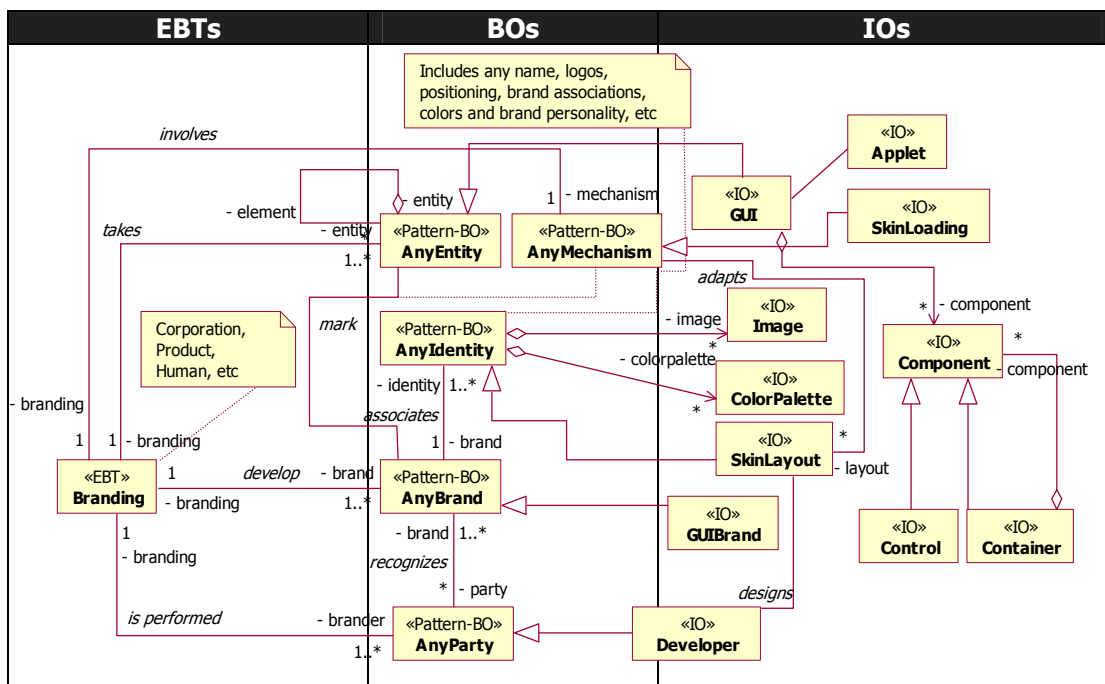


Figure 2: Scenario 1 model

Scenario 2, DNA Fingerprinting Detection Systems: DNA fingerprinting is a technique for analyzing and comparing DNA from different sources. It is a technique used in law enforcement to identify suspects from hair, blood, and other type of biological material found on a crime scene. The DNA Fingerprinting Detection system will try to compare previously stored (Oracle Database) information or brands of a list of suspects with the biological material found on the

The branding analysis pattern has been used to model two different applications: one related to the HCI domain, and the other one related to Law enforcement and genetics. We found that this pattern is flexible and general enough to be incorporated into these two different applications.

IV. References:

Bibliography:

1. M.E. Fayad, and A. Altman. "Introduction to Software Stability." *Communications of the ACM*, Vol. 44, No. 9, September 2001.
2. M.E. Fayad. "Accomplishing Software Stability." *Communications of the ACM*, Vol. 45, No. 1, January 2002
3. M.E. Fayad. "How to Deal with Software Stability." *Communications of the ACM*, Vol. 45, No. 4, April 2002
4. H. Hamza and M.E. Fayad. "A Pattern Language for Building Stable Analysis Patterns", *9th Conference on Pattern Language of Programs (PLOP 02)*, Illinois, USA, September 2002.
5. Froehlich, G., Hoover, H.J., Liu, L. and Sorenson, P.G. "Reusing application frameworks through hooks", in *Object-Oriented Application Frameworks*, M. Fayad, D. C. Schmidt, and R. Johnson eds., John Wiley, 1998, in press.
6. Guy Kawasaki, "The Art of the Start; The Time-Tested, Battle-Hardened Guide for Anyone Starting Anything," Portfolio 2004.
7. Jason Briggs, Dynamic User Interface is only Skin Deep, Javaworld, May 2000, www.javaworld.com
8. Mohamed E. Fayad, Haitham S. Hamza, and Huascar A. Sanchez, "Towards Scalable and Adaptable Software Architectures", *The 2005 IEEE International Conference on Information Reuse and Integration*, Las Vegas, NV.