VAP Quick Reference

The Values at Play Framework: A (Semi-) Quick Reference

The Values at Play (VAP) methodology for incorporating values in the context of system design is characterized by three analytically distinct activities: *Discovery, Translation,* and *Verification.* These are pursued in tandem, the results of each iteratively affecting successive versions of the system.

Discovery

The question that drives Discovery is what values are relevant to, inspire, or inform a particular design endeavor? The outcome is a list of values, explicit recognition of design aspirations that often remain implicit, or sometimes entirely unrecognized. These lists may include values abstractly construed, such as privacy, autonomy, tolerance, security, cooperation, sociality, equality, trust, and creativity, or values more specifically construed, such as freedom of expression, gender equity, environmental conservation, and racial diversity. Although the VAP methodology recommends a stable heuristic to guide the process of Discovery, the process is likely to yield lists that are highly variable from one project to the next.

To guide the process of Discovery, a useful heuristic is to consider likely sources of values in relation to a system under construction. The three below do not necessarily exhaust all possibilities:

Values expressed in the functional definition of a system

What phrase or sentence answers the question: What are you designing/building? In some cases, values form an essential part of the answer, on par with or even more important than other functional dimensions or features. Thus, designers might declare, "This is a privacy-preserving database," or "This is a game to promote environmental conservation."

Values emerge through the specification of design features

Even when values are not inherent to functional design, they may emerge as important factors when the myriad characteristics of a system underdetermined by core functional requirements are settled. In deciding, for example, how users are to gain access to a system (a website, database, information repository), designers might discover that one design option promotes security, another user-autonomy, a third ownership rights. In designing the reward structure for a multi-player game, designers might find that one approach ("zero sum") promotes competition, another encourages interactivity and cooperation, a third supports independence. The important thing for designers to notice is that seemingly routine and gritty decisions may be a source of values in design.

Stakeholders

A variety of individuals or social groups, whose values directly or indirectly inform system design, have a stake in a project's outcome. The VAP methodology recommends that designers give as full and explicit an account of potential stakeholders whose preferences, expectations, and interests my serve as sources of values in design. Several key parties emerge across many projects:

> *Designers:* Designers and members of design teams bring to the table understandings, preferences, and expectations, shaped by factors such as education, culture, and socio-economic origins. "Where is the team coming from?" is a difficult but necessary part of reflection in the creative design environment. This reflection may reveal values commitments of differing strengths, from absolute to negotiable, that ultimately shape design outcomes.
>
> *Users or consumers:* Systems may be shaped by values assumed by designers to be important to potential users or buyers. Alternatively they may be directly influenced by requirements laid down explicitly by users or indirectly expressed through marketplace dynamics.
>
> *Enterprises:* Designers might be swayed by enterprises —institutions, companies, and governmental agencies, for example—mediating the successful uptake of systems in design. In the case of educational games, for example, designers might be influenced not only by preferences, expectations, and interests of players themselves, but by entities such as schools or school districts, which are the likely intermediaries.

Translation

In Discovery, the charge is to uncover values relevant to a system; in translation, the charge is selectively to "embed" these values in its design. According to the VAP approach, Translation comprises three sub-activities: Operationalization, Implementation, and Resolving Value-Conflicts.

Operationalization

In order to make values practically accessible in the context of design, it is necessary to render them in concrete and specific terms. Members of design teams might, for example, be firmly committed to justice and agree that justice is relevant to a particular design project. But in order to move from this point to the design table, they still have work to do articulating what justice means in the context of their project. Does it mean that the system should be equally accessible to all – old and young, male and female, rich and poor, skilled and unskilled? If not, is discrimination is necessary, then what is a just basis for discrimination? Fortunately, designers need not address all these issues from scratch but may find help in prior work on the conceptual analysis and application of values to concrete problems.

Implementation

The heart of design, implementation involves the transformation of ideas, intentions, requirements, and concepts into concrete specifications through a variety of formal and informal techniques, such as brainstorming, “body storming,” paper prototyping, reference to past work, etc. Implementation, here, is the move from operationalized values concepts to system features. Our designers (above) having, say, committed to justice operationalized as equal access by both young and old, might take particular care to render text and images clearly enough to accommodate a range of visual acuity and physical dexterity. And even so, this value need not fully determine design: one team might insist on large print for all, while another might offer configurable screen-appearance with easy-to-use controls. Importantly, designers should maintain vigilance for the influence of values on design at all levels, from overarching design themes down to gritty details..

Resolution of Conflict

In conscientious design, as in life, we encounter values’ conflicts. Intent on promoting two or more values, designers may find that all cannot be simultaneously satisfied but one or more, only at the expense of others. Examples of particularly intractable conflicts that we have witnessed in the area of software design are those between security and ease-of-use, access to information and private property, privacy and transparency.

There are no across-the-board answers to resolving values’ conflicts, in design, as in life, but the VAP approach offers three strategies designers might consider when facing such conflicts.

> *Dissolving conflict through redesign:* Where material constraints imposed by a particular design idea make it impossible to realize two (or more) values, a redesign might alleviate the problem. For example, security might conflict with ease-of-use when it calls on users to master complex entry requirements, such as, hard-to-remember passwords. Redesign that utilizes a reliable biometric might dissolve this conflict, offering security and ease-of-use.
>
> *Resolving conflict via compromise:* When dissolving conflict is impossible, designers might accept that a degree of deference to competing values is better than nothing. In trying to resolve security with ease-of-use, creating the possibility of a system’s “remembering” login attributes satisfies both values, though at a compromised level.
>
> *Resolving conflict through trade-offs:* After careful consideration, designers prioritize conflicting values and adopt a design that promotes one (or more) over another (or others.) Thus, in certain circumstances designers conclude that security is the pre-eminent value and develop access controls in which ease-of-use is traded off in favor of security.

Verification

The function of the verification cycle is to ensure—to the greatest degree possible--that the design team has successfully implemented the values identified throughout the discovery process. Significant questions in this process might include: Do system features afford activities that support identified values? Does the overall system design adequately represent the values in question?

Similar to techniques employed in software usability testing, verification activities for values are intended to provide a method for confirming that individual features and the overall system design maps to the values. These techniques may include (but are not limited to): internal testing among the design team, user-testing in controlled environments, formal and informal interviews and surveys, the use of prototypes, and traditional quality assurance measures such as automated and regression-oriented testing. Further, end-user surveys, field observations, case studies, and ethnographic techniques may be employed in order to conduct empirical investigations into users' perceptions of system values. Although these may be conducted using whole systems, the VAP approach recommends continuous engagement with verification on focused prototypes.

Summary

Three analytically distinct activities comprise the VAP method: *Discovery,* in which a list is compiled of values relevant to a project; *Translation,* in which values are operationalized and implemented in material features; and *Verification,* in which designs are assessed for successful inclusion of values

VAP and the iterative cycle

Although analytically distinct, the VAP activities of discovery, translation, and verification can co-occur for the duration of a system's design; results from each are iteratively fed back into successive versions. Discovery is not restricted to the early phases of a project, but, according to this conception, is likely to crop up continuously throughout, revealing new values as the system evolves through translation and verification activities. Similarly, verification, is not reserved for the capstone, but, according to this conception, is recommended as a check from early efforts onward, feeding continuously and dynamically back into discovery and translation. And so on.