Domain-Driven Design Reference

Definitions and Pattern Summaries

Eric Evans

Domain Language, Inc.

IV. Context Mapping for Strategic Design

bounded context

A description of a boundary (typically a subsystem, or the work of a particular team) within which a particular model is defined and applicable.

upstream-downstream

A relationship between two groups in which the "upstream" group's actions affect project success of the "downstream" group, but the actions of the downstream do not significantly affect projects upstream. (e.g. If two cities are along the same river, the upstream city's pollution primarily affects the downstream city.)

The upstream team may succeed independently of the fate of the downstream team.

mutually dependent

A situation in which two software development projects in separate contexts must both be delivered in order for either to be considered a success. (When two systems each rely on information or functionality of the other - something we would generally try to avoid - naturally we see the projects that build them as interdependent. Yet there are also mutually dependent projects where system dependencies run only one direction. When the depended-on system has little value without the dependent system and the integration with that system -perhaps because this is the only place it is used - then a failure to deliver the dependent system would be a failure of both projects.)

free

A software development context in which the direction, success or failure of development work in other contexts has little effect on delivery.

Context Map

To plot strategy, we need a realistic, large-scale view of model development extending across our project and others we integrate with.

An individual bounded context leaves some problems in the absence of a global view. The context of other models may still be vague and in flux.

People on other teams won't be very aware of the context boundaries and will unknowingly make changes that blur the edges or complicate the interconnections. When connections must be made between different contexts, they tend to bleed into each other.

Even when boundaries are clear, relationships with other contexts place constraints on the nature of model or pace of change that is feasible. These constraints manifest themselves primarily through non-technical channels that are sometimes hard to relate to the design decisions they are affecting.

Therefore:

Identify each model in play on the project and define its bounded context. This includes the implicit models of non-object-oriented subsystems. Name each bounded context, and make the names part of the ubiquitous language.

Describe the points of contact between the models, outlining explicit translation for any communication, highlighting any sharing, isolation mechanisms, and levels of influence.

Map the existing terrain. Take up transformations later.

This map can be a basis for realistic design strategy.

The characterization of relationships is made more concrete in the following pages, with a set of common patterns of relationships between bounded contexts.