

リアクティブ宣言

2014年9月16日版 (v2.0)

異なる分野で活動する組織が、同じようなソフトウェア構築のパターンを独立に発見している。このようなシステムはより堅牢で、より耐障害性があり、より柔軟で、より最新の要求を反映しやすくなっている。

こうした変化が起きているのは、近年、アプリケーションの要求が著しく変化してきているからだ。ほんの数年前、巨大アプリケーションは数十のサーバから構成され、数秒の応答時間と数時間のオフラインメンテナンスを許容し、データは数ギガバイトだった。今日のアプリケーションは、モバイル機器から数千のマルチコアプロセッサによって動作するクラウドベースのクラスタまで、あらゆる機器上に配備される。ユーザはミリ秒の応答時間と100%の稼働率を期待する。データはペタバイト単位で測定される。昨日のソフトウェアアーキテクチャは、今日の要求を全く満たしていない。

求められているのは、システムアーキテクチャに対する明快なアプローチであると我々は考える。そして、必要な側面の全ては既に独立に認識されている: 求めるものは、即応性と、耐障害性と、弾力性と、メッセージ駆動とを備えたシステムだ。我々はこれをリアクティブシステム (Reactive Systems) と呼ぶ。

リアクティブシステムとして構築されたシステムはより柔軟で、疎結合で、[スケーラビリティ](#)がある。これによって開発が容易になるだけでなく、変更を受け入れやすくなる。これらは[障害](#)に対してより著しい耐性を持ち、たとえ障害が起きても災害を起こすことなく優雅に対処する。リアクティブシステムは高い即応性を持ち、[ユーザ](#)に対して効果的な対話型フィードバックを与える。

リアクティブシステムとは:

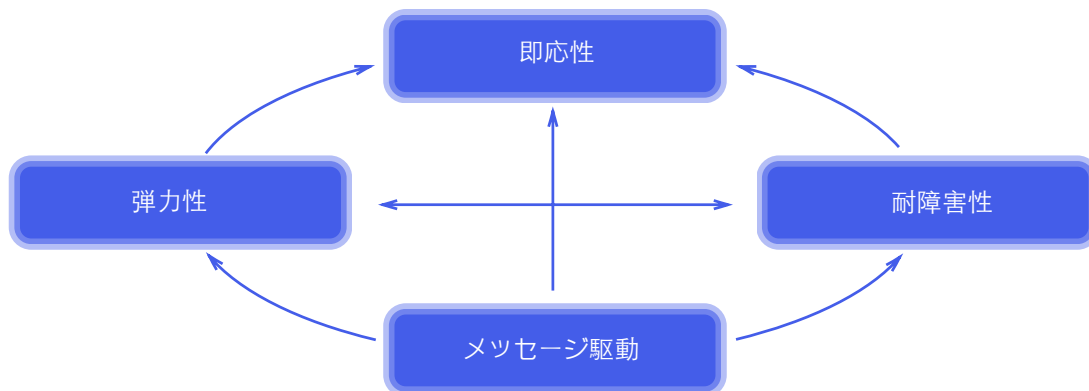
即応性 (Responsive): [システム](#)は可能な限りすみやかに応答する。即応性とは使い勝手と実用性の基盤だが、しかしそれだけではなく、問題が素早く検出され効果的に対処できることを意味する。即応性のあるシステムは、迅速で、かつ一貫した応答時間を提供することに主眼を置く。システムは応答時間に信頼性のある上限を確立し、一貫した品質のサービスを供給する。この一貫した挙動によってエラー処理が単純化され、エンドユーザの信頼を醸成し、さらなる相互作用を促す。

耐障害性 (Resilient): システムは[障害](#)に直面しても即応性を保ち続ける。これが当てはまるのは高可用性のミッションクリティカルシステムだけではない—耐障害性を持たないシステムは障害が起きると即応性を失う。耐障害性は、[レプリケーション](#)、封じ込め、[\[隔離\]\(/glossary.ja#Isolation\)](#)、そして[委譲](#)によって実現される。障害はそれぞれの[コンポーネント](#)に封じ込められ、コンポーネントは互いに隔離さ

れるので、システムが部分的に故障してもシステム全体を危険に晒すことなしに回復することが保証される。各々のコンポーネントの回復処理は（外部の）他のコンポーネントに委譲され、また必要な場合はレプリケーションによって高可用性を保証する。コンポーネントのクライアントはコンポーネントの障害への対処に苦しめられることがなくなる

弾力性 (Elastic): システムはワークロードが変動しても即応性を保ち続ける。リアクティブシステムは、入力の実現に割り当てるリソースを増加あるいは減少させることで入力量の変化に反応する。これは、システムの中に競合する場所や中心的なボトルネックが存在しないように設計し、シャーディングしたりレプリケーションしたコンポーネント間に入力を分散させることを意味する。リアクティブシステムは関連するライブな性能測定を提供することで、予測的かつリアクティブなスケールリングアルゴリズムをサポートする。これらは、コモディティなハードウェアとソフトウェアプラットフォーム上で費用対効果の高い弾力性を実現する。

メッセージ駆動 (Message Driven): リアクティブシステムは非同期なメッセージパッシングに依ってコンポーネント間の境界を確立する。これによって、疎結合性、隔離性、位置透過性を保証すると共に、エラーをメッセージとして委譲する手段を確保する。明示的なメッセージパッシングは負荷の管理と弾力性を可能とする。また、システム内にメッセージキューを作成して監視し、必要ならバックプレッシャーを適用することでフロー制御が可能になる。通信の手段として位置透過なメッセージングを使うことで、通信がクラスタを跨ぐ場合も単一のホスト内の場合も、同じ構成とセマンティクスで障害を管理できる。ノンブロッキング通信により、受信側はアクティブ時のみリソースを消費できるのでシステムのオーバヘッドを抑制できる。



大きなシステムはより小さなシステムからできているので、故にそれらの構成要素のリアクティブな性質に依存する。リアクティブシステムは設計原則を適用してリアクティブな性質をあらゆる規模で適用し、それら構成要素を合成できるようにする。世界最大級のシステムは、これらの性質に基づくアーキテクチャに依存することで数十億の人々のニーズを日々満たしている。こうした設計原則をその都度再発見するのをやめて、最初から自覚的に適用する時だ。