

派生開発・保守開発におけるソフトウェア構造解析ツールの活用

ソフトウェア構造の可視化と影響分析

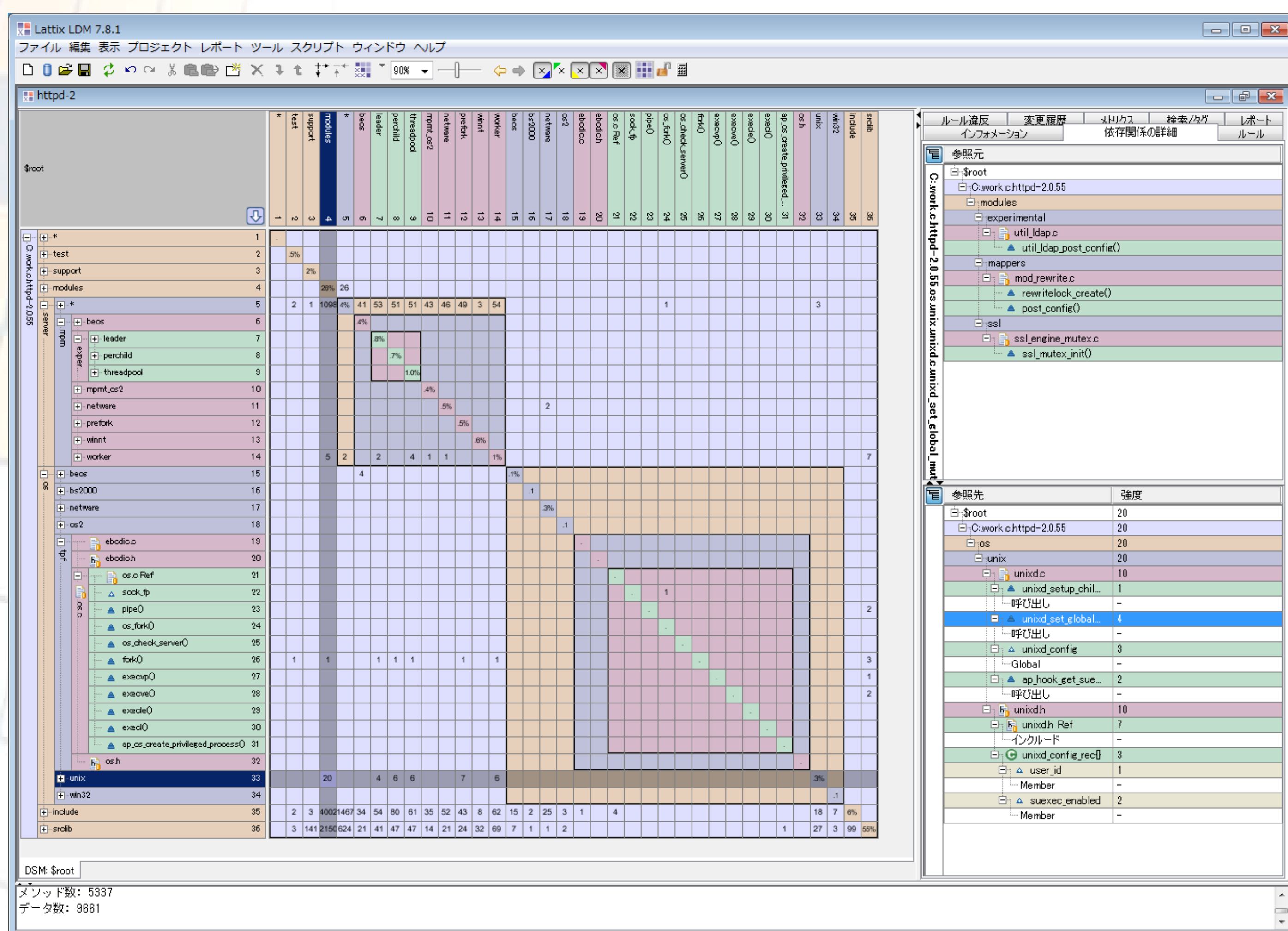
DSMのアーキテクチャ分析への応用

DSMとは、Dependency Structure Matrixの略であり、繰り返しやフィードバックが多い開発プロセスをわかりやすく表現するために開発された、プロセス改善のための分析技法です。DSMを利用することで、これまでは分析が困難だったシステムの依存関係を、より直感的に可視化・分析することが可能になります。

	Module A	Module B	Module C	Module D
Module A	1			X
Module B	2	X	X	
Module C	3	X	X	
Module D	4		X	

ModuleCは
ModuleBとModuleD
に『依存している』

ModuleBは
ModuleAとModuleC
に『依存されている』



変更時の影響度分析

- ソースコード内部の依存関係情報を自動解析することで、ソースコード変更時の影響範囲を正確に把握。
- 変更する要素をマークするだけで影響リスクを素早く分析。
- 変更によるリスク・コスト見積もりを強力にサポート。

17種類のアーキテクチャメトリクス

- アーキテクチャや設計の観点から品質を定量的に評価する17の資料を搭載。
- 過去のデータを蓄積して傾向を分析することも可能。

構造解析・分析・リファクタリング

- ソースコードやモデルの情報を解析し、DSM化することで直感的なアーキテクチャ分析とリファクタリング(改善)を支援。

アーキテクチャルール管理

- アーキテクチャの乱れをいち早く発見。
- 派生開発において、保守性の高い状態を維持することが可能。

派生開発・保守開発におけるソフトウェア構造解析ツールの活用

● [要求]-[設計]-[ソースコード]のトレーサビリティの実現

トレーサビリティとは、要求（不具合修正や機能追加も含む）をスタートとした[要求]-[設計]-[ソースコード]などの対応関係のことです。その依存関係をDSMを用いて表現することにより、全体構造を可視化したトレーサビリティマトリックスを作成できます。

- XDDPの3点セットにおける「トレーサビリティマトリックス」は、変更要求仕様書にしたがい、要求に対応したソースコードの修正箇所を示しているものです。しかしながら、要求の影響範囲を正確に把握するためには、その要求に対応するソースコードを確認するだけでなく、そのソースコードの依存関係も抜け漏れなく確認する必要があります。
- ソースコード間の依存関係については、ツール(Lattix)を用いることで自動的にトレーサビリティマトリックスを生成することができ、ソフトウェア全体の構造を可視化できます。ソースコード間のトレーサビリティマトリックスを把握することで、XDDPの3点セットの3つ目である「変更設計書」においてソースコードの修正箇所すべてを記述できるようになります。

\$root	システム要求 ... 1	システム要求 ... 2	システム要求 ... 3	リスクコントロール... 4	ソフトウェア要... 5	ソフトウェア要... 6	アーキテクチャ... 7	アーキテクチャ... 8	システムバリエ... 9	システムバリエ... 10	結合テスト A... 11	結合テスト B... 12	ユニットテスト... 13	ユニットテスト... 14	詳細設計 S... 15	詳細設計 S... 16	詳細設計 S... 17	ソース i 18	ソース ii 19	ソース iii 20	ソース iv 21	Module B 22	
システム要求 SyRS A	1																						
システム要求 SyRS B	2	1																					
システム要求 SyRS C	3																						
リスクコントロール要求 ROR A	4	1																					
ソフトウェア要求 SRS A	5	1																					
ソフトウェア要求 SRS B	6		1																				
アーキテクチャ設計 Function 1	7																						
アーキテクチャ設計 Function 2	8																						
システムバリエーションテスト...	9																						
システムバリエーションテスト...	10																						
結合テスト Alpha	11																						
結合テスト Beta	12																						
ユニットテスト Alpha	13																						
ユニットテスト Beta	14																						
詳細設計 SDS 1	15																						
詳細設計 SDS 2	16																						
詳細設計 SDS 3	17																						
ソース i	18																						
ソース ii	19																						
ソース iii	20																						
ソース iv	21																						
Module B	22																						

異なるドメインの依存関係を可視化

- [システム要求]-[ソフトウェア要求]-[設計]-[ソースコード]-[ユニットテスト/結合テスト/システムテスト]の成果物と結果に対して、全体構造を可視化した状態でトレーサビリティを確保。

膨大な情報をコンパクトに管理

- 全ファイルに対してのトレーサビリティをまとめて表示するだけでなく、ディレクトリ単位等を任意に設定することで膨大となっている情報を俯瞰。