ソフトウェアメトリクスアプローチに基づく コンピュータシステムのインフラストラクチャ品質の検証

尾花将輝[†1]　花川典子[†2]

コンピュータシステムおけるインフラストラクチャの品質を改善するために，デザインシートを用いたインフラストラクチャの特徴を計測するメトリクスを提案する．提案するメトリクスはソフトウェアプロダクトメトリクスのソースコードの規模，結合度と凝集度の概念をインフラストラクチャのデザインシートに適用し，インフラストラクチャの規模を示す LOI(Line of Item)，結合度を示す CBD(Coupling Between Design sheets)と凝集度を示す LCOD(Lack of Cohesion in Design Sheet)を提案する．LOI はデザインシートの設定項目数とし，CBD はデザインシート間で相互参照されている項目値数，LCOD はデザインシート内のサブシートの同一設定項目値の出現割合とする．大規模プロジェクトに適用して 52 デザインシートのメトリクスを計測した．計測結果，コピー作成したデザインシートがある場合，LCOD や CBD の値が極端に高くなるデザインシートグループが作成され，単純な比較が難しいことがわかった．また，LCOD におけるサブシートをメソッドと仮定するときの問題点もあきらかになった．問題点を解決したのち，本メトリクスを他のプロジェクトで検証する予定である．

Infrastructure quality of computer system based on software metrics approach

MASAKI OBANA[†1]　NORIKO HANAKAWA[†2]

We propose infrastructure metrics of the metrics to improve infrastructure quality based on software product metrics approach. The metrics are LOI(line of item), CBD(Coupling Between Design sheet), LCOD(Lack of Cohesion in Design sheet). LOI means the number of items that are set in design sheets, CBD is the number of items that are referred from the other design sheets, LCOD means ratio of sub-sheets that have same items. The metrics were applied an industrial large-scale computer system including infrastructure. As a result, several problems of the metrics were clarified. One is too high values of CBD and LCOD because several design sheets are created by copy and paste from one design sheet. In addition, assumption that sub-design sheet is equal to “method” is problematic because sub-sheet definition is not clear. In future, after these problems will be solved, the metrics will be evaluated in other project.

1. はじめに

近年，公共交通システム，金融システム，証券取引システム等の重要性は増加している．特に東京証券取引所システム等の金融関係のシステムダウンは国内のみならず，世界規模での混乱を招く可能性がある．また，大規模システムはモバイル端末や通信技術を含む高機能化や高性能化が要求され，ソフトウェアのみならず，インフラストラクチャも大規模化，複雑化の一途をたどっている．インフラストラクチャ（以下インフラ）とは，コンピュータシステムのアプリケーションソフトウェア以外の部分を示し，サーバのハードウェア構築や，その OS，ミドルウェアの設定，ネットワーク関係の機器の設定，仮想化技術で構築された仮想マシンの設定等で構成される．インフラ構築の複雑さが影響し，近年では社会的に影響を与えるシステムダウンに直接結びつくケースも多数存在する．

例えば，2013 年 1 月から 6 月の日本の新聞紙上で取り上げられた社会に影響を与えたコンピュータシステム障害は 21 件であった．純粋なソフトウェア障害が起因したシステムダウンは 6 件であり，インフラ構築の設定ミスによるシステムダウンが 11 件，物理的機器故障等によるシステムダウンは 4 件であった[1]．このように，大規模化，複雑化するインフラ構築における機器の設定ミスが，コンピュータシステム全体のシステムダウンを引き起こす重要な要因のひとつとなっている．

大規模システムのインフラ設定の具体例を述べる．本稿で検証対象としたプロジェクトは，サーバ，クライアント，ネットワーク機器等，総数 1471 台のインフラ構築をおこなった． 52 個の MS-Excel ファイルで管理され，そのワークシート数は 929 個，設定すべき項目数は 224,907 であった．同じ値を複数ワークシートに記入された箇所も多く，1 つの設定項目値の変更が発生すると，全てのワークシートを精査する必要があり，その作業は非常に煩雑であり，修正抜け等のミスを誘発する可能性が高い．

†1 大阪工業大学
Osaka Institute of Technology..
†2 阪南大学
Hannan University

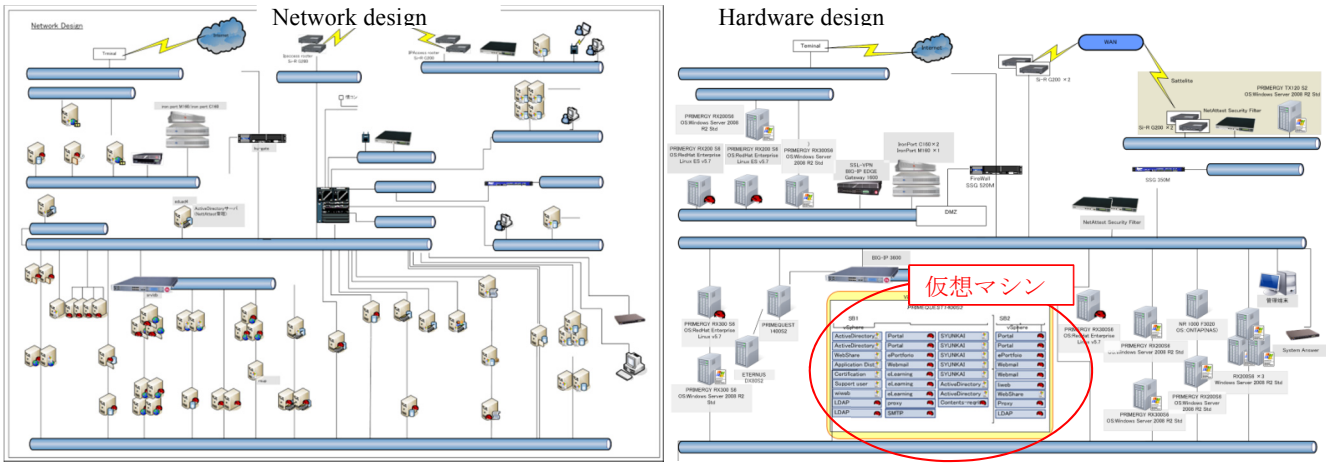

図 1 インフラストラクチャの設計概要の例

Figure 1 Outline of infrastructure design

そこで，本稿ではコンピュータシステムにおけるインフラの規模，結合度，凝集度を計測するメトリクスを提案する．ソフトウェアと同様に「計測できないものはコントロールできない」の概念を基に，インフラの規模や凝集度等を計測し，その値に基づいてインフラの品質を管理，改善することを目指す．従来のインフラの品質の研究はハードウェア故障時のリスク等が主であった[10,11]. 本研究では，ハードウェアの物理的故障は範囲外とし，インフラ構築時の設定ミス等に着目する．設定ミスは設計上の不良やヒューマンエラーが要因と考えられ，ソフトウェアのエラーと酷似する. したがって, 提案するインフラの規模や結合度，凝集度のメトリクスはソフトウェアプロダクトメトリクスの概念を適用して提案する．これによって，大規模なインフラの規模や構造を明らかにし，品質向上に役立てることを目指す.

２章では関連研究を示し，３章でインフラの規模，結合度と凝集度のメトリクスを提案する．４章で実際のコンピュータシステムのインフラで計測する．その結果とリリース後の障害との関係について５章で述べる．６章でまとめと今後の課題について述べる.

2. 関連研究

多くのインフラやシステムアーキテクチャの設計技術が提案されている. Clements らは ATAM (Architecture Tradeoff Analysis Method)というインフラの設計手法を提案した[3]. ATAM は非機能要求を技術的要求へ変換するためのテンプレートとインフラは非機能要求と技術要求のトレードオフで設計する方法を提供した．また，長谷川らは早い技術革新に対応したインフラ設計方法を提案している[4]．特徴は，ビジネス分析と同時に実施される反復されるインフラ設計プロセスである．これらの設計方法はインフラの設計に有効であるが，品質を管理するための指標等は提案されていない．我々の提案するメトリクスはインフラの規模と結合度等を定量的に計測する．計測された値と品質との関係を明確にすることで，品質予測への貢献が期待できる.

また，クラウンドコンピューティングサービスのためのインフラ設計方法も提案されている[5]．インフラのコストと，サービスレベルの合意違反を発生させるビジネスロスの両方をトレードオフして，サーバ数やルータ数，通信帯域幅を決定する方法である．Touzi らは異なる組織での協調サービスのための Model-Driven Architecture(MDA)に基づくアーキテクチャ設計方法を提案した[6]．これらのインフラ設計方法は様々なサービスとのトレードオフが含まれており，サービスはソフトウェアだけではなく，ハードウェアを含むインフラも対象となる．したがって，ソフトウェアとハードウェアのパッケージをサービスとみなしている．我々の研究では，ソフトウェア開発とインフラ構築は異なる作業プロセスで，異なるスケジュール上で進むと考える．ソフトウェア技術者はソフトウェアを開発し，インフラ技術者はインフラを構築する．我々のメトリクスは実際の産業界の開発プロセスに近い考え方を取り入れた概念である.

さらに，インフラを構築する時のヒューマンエラーによる設定ミスに関する研究もおこなわれている．Pappas らは DNS のヒューマンエラーによるコンフィグファイル設定ミスについて調査した[7]. DNS コンフィグファイルの設定ミスは DNS ゾーンの 15%エリアに影響を及ぼし，DNS ゾーンの 21%に設定の矛盾が生じていることを明らかにした．大規模システムのインフラのコンフィグファイルの単純ミスを防ぐための自動チェックシステムの必要性を主張した．また，Wool は Firewall コンフィグファイルの設定ミスによるエラーを定量的に計測した[8]．システムが大規模になると，コンフィグファイル設定ミスは劇的に増加すると示唆する．これらの研究のコンフィグファイルのエラーを量的に計測する点で我々の研究に類似する．しかし，これらの研究では DNS や Fire Wall という特定の機器に依存しているが，我々の研究ターゲットはインフラすべてを含む点で異なる.

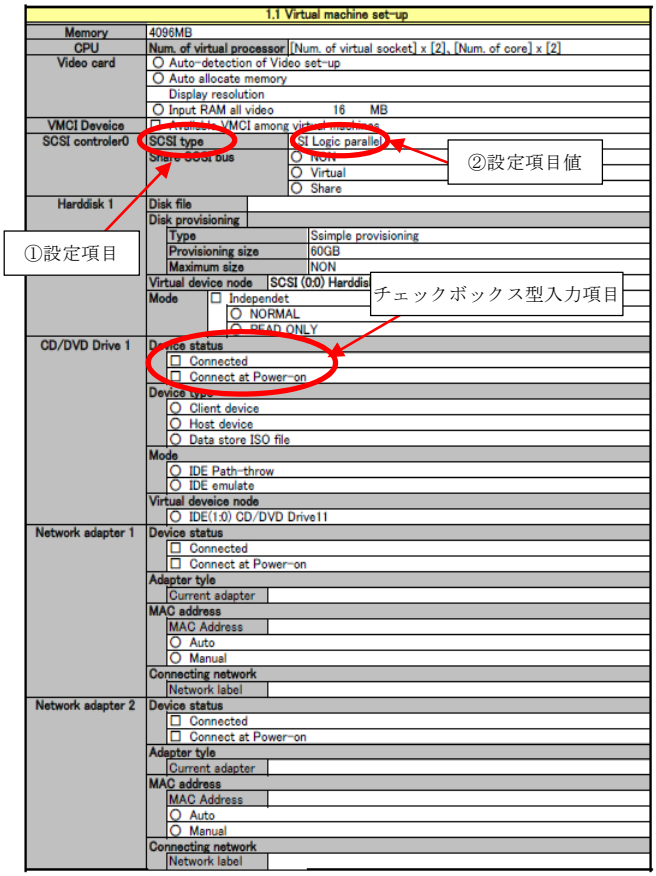

図 2　仮想マシンのデザインシート例

Figure A sample of design sheet

3. インフラ構築とデザインシート

3.1 インフラ構築とは

本稿で利用する「インフラ構築」について説明する．インフラ構築とはコンピュータシステム上の機器，OS，ミドルウェアであり，それらを構築する作業をインフラ構築とする．機器はコンピュータ機器，ネットワーク機器，周辺機器に分類される．コンピュータ機器はサーバコンピュータ，クライアントコンピュータ，モバイルターミナルを含み，ネットワーク機器はルータ，アクセスポイント，負荷分散機器，スイッチングハブを含む，周辺機器はプリンター, スキャナー, UPS やそのほかの様々な最新機器も含む．

また，ミドルウェアとはデータベースサーバ，Web サーバ，ファイルサーバ，メールサーバ，プロキシ―サーバ，DNS サーバ，認証サーバ，VPN サーバ，ログ監視サーバ，バックアップサーバ，セキュリティサーバ，Fire wall サーバ，アプリケーションサーバ等である．サーバコンピュータとクライアントコンピュータ，モバイルターミナルは Linux や Windows，Android 等の OS が装備される．サーバソフトウェアや OS はそれぞれのバージョンやディストリビューションを持ち，それぞれのバージョンで機能や設定項目が少しずつ異なる．

インフラ構築作業は，機器搬入，機器設置，機器セットアップ，OS とミドルウェアインストールとセットアップと環境整備作業，最終動作確認作業等である．近年はインフラの高性能化，高機能化に伴い，物理機器のみならず仮想機器の種類や数が増加して，ハードウェアとサーバソフトウェアのセットアップ項目数は劇的に増加している．さらに，モバイルターミナルの導入はクライアント数を劇的に増加させ，端末ごとに異なる OS はセットアップ作業量の大幅な増加につながる．

また，仮想マシンは物理的機器ではないが，現在のインフラ構築では重要な要素の一つである．デザインシートはメモリサイズ，ビデオカード，VMCI デバイスの利用，ハードディスク，ネットワーク等，仮想マシンに設定すべき項目がすべて記載される．また, 物理的設定値のみならず, OS やミドルウェア等のインストール時のオプション値も記載される．

インフラ構築のドキュメントは，概要設計書，詳細設計書，操作設計書，セキュリティ設計書，ネットワーク構成設計書，運用設計書等がある．図 1 にネットワークとハードウェアの概要設計のサンプルを示す．概要設計を基に詳細設計を行い，詳細設計には図 2 のような各機器の具体的な設定値を設計するためのデザインシートを作成する．本研究ではこのデザインシートの複雑さを計測することで，インフラ構築の複雑さを定量的に計測する．

3.2 デザインシート

図 2 は図 1 の仮想マシンのデザインシートの一部を示す．本研究で述べるデザインシートには IP アドレス，ポート番号等の機器を稼働させるために必要な設定項目が記述された設計書を指す．また，本研究でのデザインシートは MS-Excel のようなスプレットシートソフトで作成されており，１種類の機器（または仮想マシン）に対し，１つの Excel ファイルが作成されている．１種類の機器とは，例えば同一機種を複数台設置する場合も１つの Excel ファイルで管理されており，その中で，複数のワークシートで各機器を管理している．ただし，複数機器がほぼ同じ設定値の場合は１つのワークシートとなっている．

また，機器ごとでは無くてもサブシートに分かれていることもある．これは，ひとつのワークシートが大きくなりすぎて，意味ある塊にわけたサブシートを作って管理をしやすくするためである．例えば，図 2 で示す仮想マシンのデザインシートでは 14 のワークシートを含む. サブシートは（1）OS 設定，（2）kernel パラメータ，（3）インストールオプション，（4）ネットワーク設定等，意味ある項目がワークシートごとに分類されている．

本稿では１つの Excel ファイルをデザインシートと呼び, その中のワークシートをサブシートと呼ぶ．また，本稿で計測対象のデザインシートでは他のプロジェクトでも利用できる汎用性を高める工夫はしている．ただし，ターゲットプロジェクト専用に作成したデザインシートも含まれる．

また，本研究ではデザインシートのフォーマットや規格に関し，研究対象外とする．しかし，フォーマットの記述方法や，記述するソフトウェアが異なるとしても，本研究のアプローチや計測結果が大きく変わることが無いと考える．この理由として，記述される内容が IP アドレスや，ポート名，DNS 名，BIOS の設定等の機器やサーバソフトウェアを設定するための実際のパラメータ値をデザインシートに記述するからである．そのため，インフラ構築では，ソフトウェア設計書のように，自然言語な文言を記述するのではなく，ソースコードレベルの内容をデザインシートに記述するため，フォーマット等が異なるとしても，デザインシートに記述される内容が大きく異なることは無いと考える．

続いて，設計者がデザインシートへの記述方法について述べる．設計担当者はデザインシートの設定項目に設定項目値を記入する作業となる．チェックボックスにチェックを入れる場合や，IP アドレスやポート番号を数値で入力，命令とパラメータで構成されるコマンドラインをテキストで入力する場合もある．また，サービスリストや IP アドレスリスト等のリスト関係も手作業で作成する必要がある．

図２の仮想マシンデザインシートの設定される項目の総数が例えば 2642 個である場合，システム全体で 10 個の仮想マシンが構築された場合，総設定項目数は 26420 個となる．このように，インフラ構築の詳細設計に作成されるデザインシートの項目数はインフラの規模の拡大に伴い，劇的に増加する．

本稿では，デザインシートの設定すべき項目のことを「設定項目」（図２の①参照）とし，その設定項目に入力された値を「設定項目値」（図２の②参照）と呼び，それぞれを区別する．

4. インフラストラクチャメトリクスの提案

4.1 インフラメトリクスのコンセプト

インフラ構築における要求分析フェーズでは，ソフトウェア開発技術者とインフラ構築技術者と顧客がともに要件を決定する．その後，ソフトウェア開発技術者はソフトウェア開発，インフラ構築技術者はインフラ構築をそれぞれ独立したプロセスとスケジュールで平行に実施する．２つのプロセスの外観は類似しており，概要設計，詳細設計，実装，テストとなる，その後，システム全体を統合して総合テストが行われる．インフラ構築の詳細設計フェーズでは図２で示すデザインシートを作成し，実装フェーズではデザインシートを基にコンフィグファイル等のバッチファイルを作成し，その実行を行う．したがって，ソフトウェア開発とインフラ構築を比較した場合，インフラのデザインシートはソフトウェアのフローチャートや UML 等の詳細設計に対応し，コンフィグファイルはソフトウェアのソースコードに対応することがわかる．

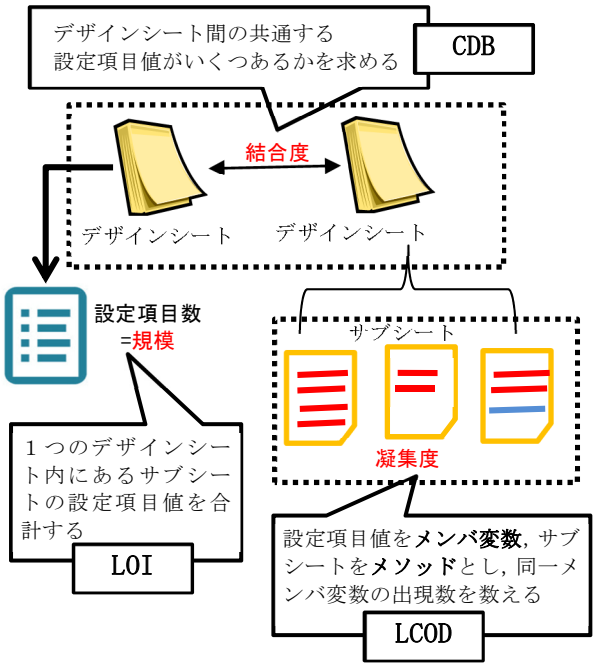

図 3　規模，結合度，凝集度の概念
Figure 3 Outline of LOI, CDB, LCOD

そこで，ソフトウェアプロダクトメトリクスの規模 LOC（Line Of Code），結合度 CBO(Coupling Between Objects)，凝集度 LCOM(Lack Of Cohesion in Methods)[12,13,14]をインフラのプロダクトであるデザインシートに当てはめ，インフラの規模 LOI(Line Of Items)，結合度 CBD(Coupling Between Design sheet)，凝集度 LCOD(Lack Of Cohesion in Design sheet)を計測する．各メトリクスのコンセプトはソースコードのメトリクスとほぼ同じ意味である．

本来はソースコードと同フェーズで作成されるコンフィグファイルを対象とすべきであるが，コンフィグファイルは機器に依存し，異なる性質であるコンフィグ間でメトリクスを計測し，比較することは望ましくない．つまり，メーカや機器に依存するコンフィグファイルとなり，汎用性が下がるので，デザインシートを計測の対象とした．

4.2 計測するメトリクス

以下の３つのメトリクスを提案する．

- LOI：デザインシートの規模（設定項目数の総数）
- CBD ：デザインシートの結合度
- LCOD ：デザインシートの凝集度

(1) LOI(Line of Item)

インフラの規模はデザインシートの LOI で計測する．図２の表形式のデザインシートの実際に入力する項目の数をカウントする．ソフトウェアメトリクスの LOC と同様にデザインシートの規模を示す．入力する項目は複数選択のチェックボックス（図２のチェックボックス型入力項目）

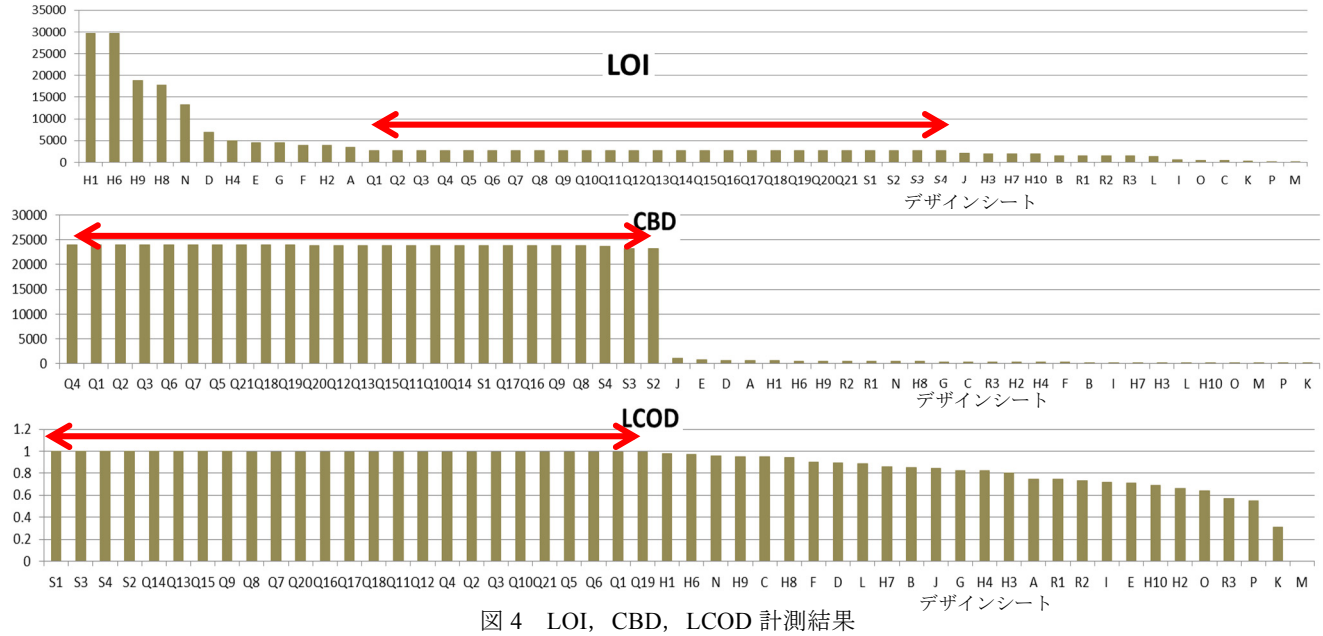

図 4　LOI，CBD，LCOD 計測結果
Figure 4 Measurement of LOI, CBD, LCOD

や IP アドレスのようにテキスト入力項目もあるが，すべての入力形式を含めてひとつの入力項目と数える(図 3 の LOI 参照).

(2)　CBD(Coupling Between Design sheet)

デザインシート間の結合の強さを CBD で計測する．基本的概念を図 3 の CBD に示す. 例えば, IP アドレス 150.168. 1. 1 がデザインシート A とデザインシート B にそれぞれ記入されていると, デザインシート A とデザインシート B に結合関係があると考え，結合度は 1 となる．さらに，別の IP アドレスが A と B のデザインシートに記入されていると，その結合度の値は 2 となる．結合度に利用する入力項目は IP アドレスのほかに，DNS などのサーバ名であり，いずれもテキスト入力する項目である．ソフトウェアメトリクスのモジュール間結合度と同様に, CBD の値が大きければ 2 つのデザインシート間の結合度が高く，値が小さければ 2 つのデザインシート間の結合度が低い．ソフトウェアモジュールと同様にデザインシート間の結合度が高いと保守性が劣る．たとえば，ひとつの IP アドレスが変更された場合, 関係あるすべてのデザインシートの IP アドレスを変更する必要があり，変更忘れなどの不具合に発展する可能性が高くなる．したがって，結合度の低い，すなわち独立性の高いデザインシートが良いデザインシートと考える.

表 1　検証対象のプロジェクトの導入概要
Table 1 Target project

開発期間	クライアント台数	サーバ台数	仮想マシン数	サーバソフト（パッケージ）数	ネットワーク機器数	クライアントソフト数
2011 年 11 月 ～ 2012 年 3 月	1184	19	35	7 (6)	258	36

(3)　LCOD(Lack Of Cohesion in Design sheet)

ひとつのデザインシート内の凝集度は LCOD で計測する．ソフトウェアプロダクトの凝集度 LCOM では，一つのクラス内のメンバ変数にアクセスするメソッド数を計測した．これをデザインシートに当てはめるために，まず，ソフトウェアのメンバ変数をデザインシートの全ての設定項目値とする．つまり，入力された IP アドレス 150.168. 1. 1 を 1 つのメンバ変数とする．また，デザインシートの中のサブシートをメソッドと考える．つまり，LCOM の考え方を適用すると，デザインシート内の複数のサブシートに同一設定項目値が多く参照されているほうが高い凝集度となる．

図 3 の LCOD に概念を示す．設定項目値をメンバ変数，サブシートをメソッドとすると，ひとつのデザインシートに同じ IP アドレスが何度も出現すると, そのデザインシートには当該 IP アドレスが集中して必要情報であると考えられる．当該デザインシートをひとまとめにすることに意味があると発想する概念である．反対に，ひとつのデザインシートに当該 IP アドレスの出現頻度が少ない場合, 当該デザインシートの設定方法に問題があるのではないかという考えである.

5.　メトリクスの計測

5.1　ターゲットプロジェクト

提案するメトリクスを産業界のプロジェクトに適用する．ターゲットプロジェクトは，様々なアプリケーションソフト，ミドルウェア，大規模インフラを含む大学の教育システムである（表 1 参照）．開発期間は 2011 年 10 月から 2012 年 3 月，クライアント PC 数は 1184，物理サーバ数は 19，仮想マシン数は 35，パッケージソフトを含むサーバソ

フトウェア数は7，ネットワーク機器数は 258，クライアント側の主なアプリケーション数は 36 のコンピュータシステムである．

5.2 ターゲットプロジェクトのデザインシート

ターゲットプロジェクトで 52 個のデザインシートが作成された．デザインシートは MS-Excel のようなスプレッドシートソフトで作成されており，全てのデザインシートにサブシートが存在する．例えば，図２で示す仮想マシンのデザインシートでは 14 のサブシートを含む.サブシートは（1）OS 設定，（2）kernel パラメータ，（3）インストールオプション，（4）ネットワーク設定等，意味ある項目のサブシートが存在する． また，デザインシートは主に物理機器や仮想機器ごとのように機器ごとに作成されており，例えば，ファイルサーバのようなストレージ機器はひとつのデザインシートを作成する．

ターゲットプロジェクトの詳細設計では，929 個のサブシートを含む 52 個のデザインシートを作成されていた．

5.3 計測結果

図 4 に計測した結果を示す．LOI の最大値は 29,700 であり，1 つのデザインシートに 29,700 個の設定項目があるという意味になる．最小値の LOI は 40 であった．本システムのインフラの総 LOI は 224,907 となる．CBD は式（2）に基づき計算し，52 デザインシートを CBD 値の降順に並べた結果である．最大の CBD 値は 24,018 であり，最小値は 122 である．LCOD 値の最大は 0.998 であり，最小値は０でとなった．

5.4 提案するメトリクスの問題点

本提案における計測結果の問題点について述べる．LOI ではデザインシートの規模の目安となったが，CBD 値では極端に大きい値のグループとそれ以外に分かれ，単純に比較するには難しい．LCOD も同様に値が大きいものが多く，ほぼ同等の値であり，比較することは困難であると考える．

本計測は，ソフトウェアプロダクトの LOC と CBO と LCOM をインフラのデザインシートに，ほぼ同様に適用したため，インフラとソフトウェアの根本的な差異により，計測値を単純に障害予測等に利用することができない．今後更なるメトリクス改善が必要であると考える．

また，本稿の計測結果では，ほぼ同じ値の結果となった．この理由として，ひとつのデザインシートをコピーして作成しており，僅かな部分だけの修正を行ったため，同等の値となった．具体的には仮想マシンのデザインシートであり，内容はほぼ同じであり，わずかだけ，修正を行って作成されているためである．そのため，同等の値となっている．

このようなコピーして作成するデザインシートではほぼ同じ値となるため，本提案であるメトリクスから大小を比較し，分析することは困難である．今後，メトリクスを更に改良，分析し，新たなメトリクスとインフラ構築の障害との関係についての調査を行う．

6. おわりに

インフラの品質を計測するメトリクス LOI, CBD, LCOD を提案した．各メトリクスはソースコードのメトリクスである LOC，CBO，LCOM とほぼ同じコンセプトを元に提案した．これらを，インフラ構築時に作成されるデザインシートで計測することで求める．提案するメトリクスを産業界のプロジェクトに適用した．今後はメトリクスの更なる改善とともに，インフラ構築に特化した新たなメトリクスの提案と，メトリクスとインフラ構築における障害の関係の検証を行う予定である．

謝辞　本研究の一部は科研費 26330093 の助成を受けた．

参考文献

1) 松田晃一，鈴木三紀夫，大髙浩:情報システムの障害状況 2013 年前半データ, SEC Journal，Vol.9，No.3，pp.142-146(2013)
2) ISO 9000 Quality Systems Handbook - updated for the ISO 9001: standard(2008)
3) P. Clements, R. Kazman, M. Klein.: Evaluating a Software Architecture, Addison Wesley, United States(2002).
4) M. Hasegawa, E. Ishida, H. Ogawa,: The establishment of the method for designing infrastructure to cope with changes quickly, IBM Provision (50), pp.68-pp.75(2006)
5) H. Shi, Z. Zhan,: An optimal infrastructure design method of cloud computing services from the BDIM perspective, Computational Intelligence and Industrial Applications Asia-Pacific Conference on, pp.393-pp.396 (2009)
6) J. Touzi, F. Benaben, H. Pingaud, J. Pierre Lorré,: A model-driven approach for collaborative service-oriented architecture design, International Journal of Production Economics, Volume 121, Issue 1, pp.5-pp.20(2009).
7) V. Pappas, D.Wessels, D. Massey, L. Songwu, A. Terzis, Z. Lixia,: Impact of configuration errors on DNS robustness, Selected Areas in Communications, IEEE Journal on , Vol.27, No.3, pp.275-pp.290(2009)
8) A. Wool.: A quantitative study of firewall configuration errors, IEEE Computer , Vol.37, No.6, pp.62-67(2004)
9) T. Kamiya, S. Kusumoto, K. Inoue: CCFinder: a multilinguistic token-based code clone detection system for large scale source code, IEEE Transactions on Software Engineering, Vol.28, No.7, pp. 654-pp.670(2002)
10) 船越裕介, 松川達哉, 吉野秀明, 後藤滋樹: 通信ネットワークの保全度向上のための故障修理時間分布の特性分析(ネットワーク), 電子情報通信学会論文誌, Vol.92, No.7, pp.1153-1163(2009)
11) 鳥山裕史, 町澤朗彦, 岩間司: ハードウェア SNTP サーバの開発, 電子情報通信学会論文誌, Vol.89, No.10, pp.1867-1873(2006)
12) S. Chidamber, C. Kemerer: A metrics suite for object oriented design, IEEE Transactions on Software Engineering, , Vol.20, No.6, pp.476-493(1994)
13) A.J. Offutt, M.J. Harrold, P. Kolte: A software metric system for module coupling, Journal of Systems and Software, Vol.20, No.3, pp.295-308(1993)
14) A.J. Albrecht, J.E. Gaffney: Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation, IEEE Transactions on Software Engineering, Vol.9, No.6, pp.639-648(1983)