デザインツールの一元化と自動化
-FLUX とプログラミング言語の併用システム-

○竹内　篤史*1　城所　竜太*1

キーワード：FLUX　一元化　自動化　可視化

1.　背景

建築設計は一言で言えば、限られた予算の中でクライアントの要求する空間、機能を満たす建築物の設計図を作成するということであるが、実際の業務内容は多岐に及ぶとともに建築主、意匠、構造、設備、環境、積算、防災など大勢の関係者が複雑に絡み合いながら進められていくものである。

構造設計ひとつを取っても、重力や地震、台風などの自然界の外力に対して安全となるように計算に基づき、意匠設計者と調整していくということは単純ではない。意匠設計者との調整の中では、建物形状、スパン、階数、階高、部材（柱・梁・壁・ブレース・免制振ディバイスなど）サイズやレイアウトなど無数にあるパラメータを双方の理想に近づけるためにコストも睨みながら、変更を繰り返す。また、パラメータを変えることによって、生じる地震等の外力も都度変わることになるため、より複雑である。

上記のようなプロセスの中で、構造設計者が使用するツールもまた図面作成用の CAD（伏、軸組、部材リスト、詳細図）、静的な外力に対して検証する解析ソフト、地震のような動的な外力をシミュレーションする解析ソフト、荷重条件を整理するソフト、断面設計をするソフトと多種多様である。そのプロセスの中に一つの変更が生じた場合、複数のデータ更新をして再設計をしなければならず、多大な労力を要することになる。また、そこに秘めている問題点として、複数人の関与や繰り返し作業に伴うヒューマンエラーということもある。中には、上記に挙げた複数の機能を統合した一貫系解析プログラムや3次元で図面を作成することにより伏、軸組、部材リスト間の不整合を無くすことが出来る BIM の存在はあるが、全てが一元化されている訳ではない。

本報告では多数存在する情報を FLUX というプラットフォームを利用することにより一元化し、プログラミング言語によって関連する複数のソフトウェアを起動させ、整合性の取れたインプットとアウトプットを常にプラットフォーム内にアップロードさせるシステムを提案する。

2.　FLUX という仕組み

FLUX とは、サスティナブル住宅供給プロジェクトをきっかけに、X (旧: Google X) から独立した、FLUX (組織名) が開発を進めているクラウドベースのデータ管理に特化したプラットフォームである。一般に、設計業務に必要な専門ツールは多岐にわたるため、それらを特定のマスターソフトウェアに集約することは現実的ではない。専門ソフトウェアによる検討は従来通り行い、そこに出入力するデータのみを共有することが、FLUX の試みである。

FLUX は、API を公開している複数ソフトウェアとの連携を確立しており、今後も対応ソフトの種類は増える見込みである。その一方で、FLUX が具体的に有しているのは、極めて基本的な機能に限られている。中心となるのは、(1) 特定ソフトウェアに入力するデータを、クラウドから直接

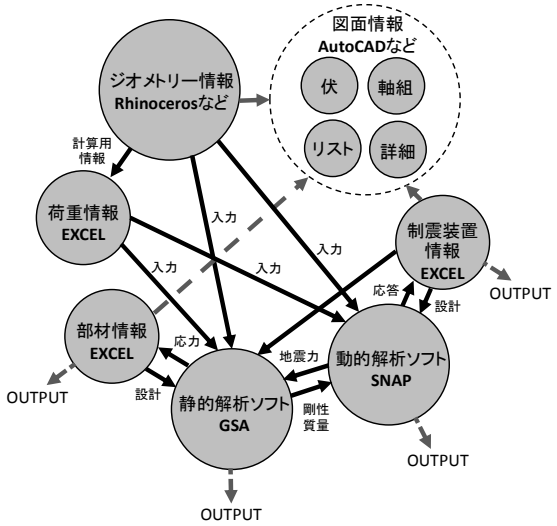


Fig. 1　一般的な情報相関図

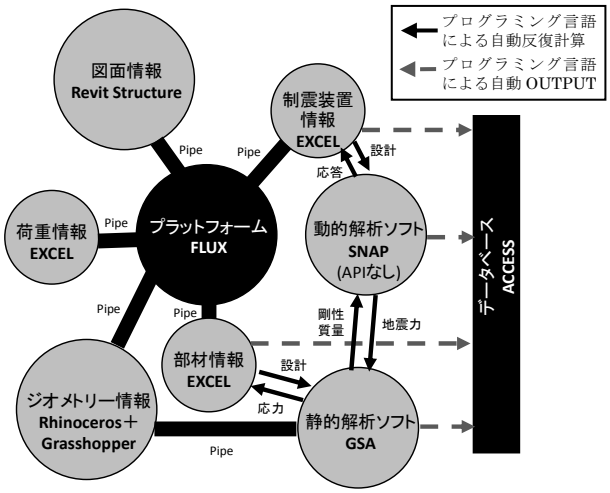


Fig. 2　FLUX 導入時の情報相関図

取り込むダウンロード機能、(2)検討の結果得られたデータを、クラウドに直接保存するアップロード機能、そして (3) ダウンロードとアップロードの更新頻度を指定する機能、以上の３点である。これらの機能により、従来は各専門分野内で完結していたデータの流れを、共有プラットフォームを介して一元化することが可能となる。

3.　適用対象

3.1　対象建物

本報告でのデザインツールの一元化・自動化の対象とする建物は平面形状約 90m、高さ 230m の大型超高層ビルとした。この建物の構造設計に係わる情報を対象とした。

3.2　関連ソフトウェア

本システムにより一元化・自動化を図ったソフトウェアは下記に示す通りである。

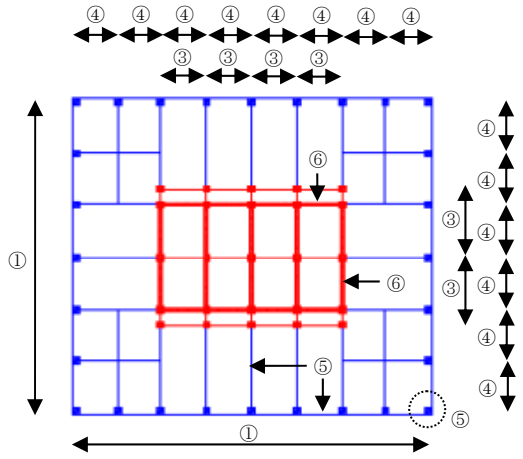
A. ジオメトリー情報：Rhinoceros＋Grasshopper
B. 荷重情報：EXCEL
C. 静的解析ソフト：GSA
D. 動的解析ソフト：SNAP
E. 部材情報・断面検討・積算：EXCEL
F. 図面情報：Revit Structure
G. 解析結果データベース：ACCESS

また、情報をつなぐパイプとしての役割を果たすプラットフォームには FLUX、複数のソフトウェアを起動させるプログラミング言語には Visual Basic を採用した。

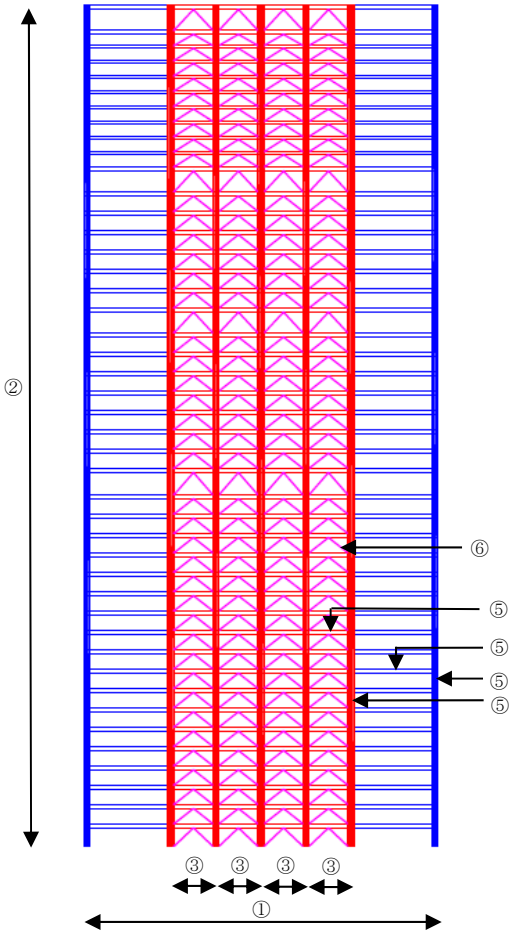
3.3　設定パラメータ

建築空間的な理想に近付けながら、コストを抑えた効率的な構造計画とするためにはより多くのパラメータを容易に変更可能な設定とすることが好ましい。3.2 節に示すように構造設計で必要となる全情報が一元化されているため、その中でのパラメータ設定は自由となる。しかし、日本の超高層ビルの構造設計で最もクリティカルとなるのは耐震設計であり、重要なのは建物のコア計画（EV シャフトや設備の DS・PS なども集約される）、外周ペリメーターゾーンの柱配置、制振装置の計画である。一般的な設計による場合には試行錯誤しながら、計算を繰り返す。そのため、ネットワークを簡素化させるために下記の項目にパラメータとして絞り込むことで、構造設計関係者が皆容易に操作でき、結果を確認し、協働しやすいようにした。

①建物外形（平面形状、立面形状）
②階数・階高
③コア内グリッド
④外周ペリメーターゾーンの柱ピッチ
⑤部材断面
⑥制振装置の種類と配置



(a) 伏図



(b) 軸組図

Fig.3　対象建物・設定パラメータ

ここで、①～⑤までのジオメトリー的な自動化・最適化の技術はコンピューターの進化とともに発展しており、限られた制約条件の中で最適な断面を割り当てる木下らの研究 [1]、久保らの最適ダンパー配置の研究 [2]などが例として挙げられる。また、国外の多くのプロジェクトでも３次元モデリングソフトと静的解析ソフトを連携させて、パラ

メトリックスタディし、その結果をデータベースによってわかりやすく可視化することが行われており、Chok らの報告[3]などがその例である。しかし、⑥で性能検証が必要となる動的解析も含めた建物全体のデザインツール自動化・最適化といった試みはなされておらず、その実現は非常に有用である。より多くのパラメータが存在し、設計条件が複雑に絡み合っているケースの場合に本報告の仕組みはより良い効果を発揮すると考えられる。

4. インプットからアウトプットまでの流れ

4.1 インプット

FLUX というプラットフォームを使ったときの一元化や自動化の方法は無数に存在すると言える。本報告における超高層ビルの一元化では、容易な変更対応が可能とするため、EXCEL での入力を基本とし、3.3 節にパラメータとして掲げる項目を数値入力出来るようにした。下記に一例を示すが、外周柱のピッチを EXCEL 上で変えた場合には FLUX と Grasshopper を介して、解析モデルや図面情報が自動アップデートされるという機構である。

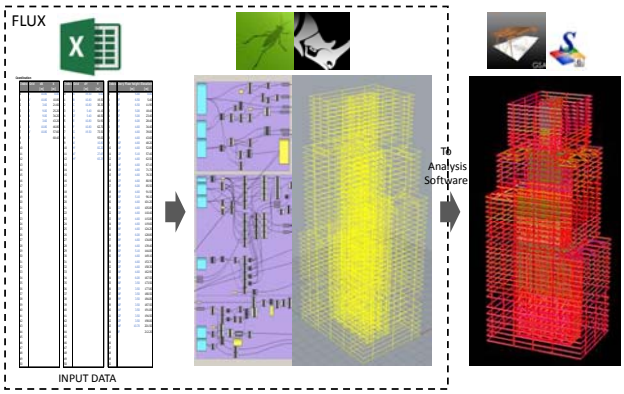


Fig.4 インプットイメージ

4.2 プログラミング言語の実行

FLUX はあくまで情報を一元化するためのプラットフォームであり、プログラムを起動・実行させるエンジンは保有していない。つまり、解析ソフト内のインプットデータを自動アップデートさせることは出来るが、静的解析、動的解析を行った上でアップデート後の結果を出し、部材の断面検討をするためには別の仕組みが必要となる。そのため、本システムでは Visual Basic をプログラミング言語として採用し、解析実行させるようにした。

構造設計をする際に、C.静的応力解析、D.動的応答解析、E.断面検討は従属関係にあり、解くためには反復計算が必要となる。本システムでは Visual Basic がエンジンとなり、C→D→E→C→D→E→・・・と反復させることにより、解析結果も含めた整合性を取ることが可能となる。

4.3 アウトプット

4.2 節のプログラム実行により、C.静的応力解析からは常時や地震時などの個材の応力状態・力の流れや損傷度、D.動的応答解析からは地震時や耐風時の揺れや変形、E.断面検討からは部材の断面余裕度や積算数量（＝コスト）などの情報を得られることとなる。通常の設計においては、C、D、E から得られた結果を個別に分析することが一般的であるが、大きなプロジェクトになればなるほど、データ量も大きく、より複雑化するため、分析することも容易ではなくなる。そのため、本システムでは得られた結果をプログラミング言語により自動的にデータベースにアウトプットさせる機能を付けることとした。それにより、膨大なデータは可視化され、大まかな傾向から細かな部位に至るまでスピーディーに表示・確認・分析し、エンジニアリング判断、設計上の調整、資料作成、プレゼンテーションもスムーズにすることが可能となる。

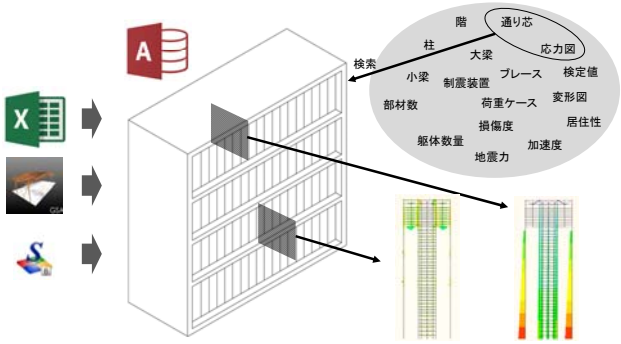


Fig.5 データベースイメージ

5. ワークフロー

大型超高層ビルの構造設計を行う際には通常、A 意匠設計者との調整担当者、B 静的解析によって構造部材の断面設計を行う担当者（通常はパーツで分けて数名）、C 動的解析を繰り返しながら制振装置の設計担当者、D 図面作成担当者のように複数人によってパラレルに設計が進められる。ここでは、従来の方法と本システムを導入したときの方法について、ワークフローを比較する。

5.1 従来のワークフロー

仮に、A 者が調整の中で梁のスパンを変更した場合、設計用応力は変わることから B 者の設計条件は変わり、剛性の変動に伴って C 者の条件も変わり、当然図面のアップデートが必要な D 者にも影響を及ぶ。同様に B 者や C 者が設計の中で変更をした場合、双方に影響を与えることから、A～D のどこかで変更が出た場合には皆が別々に条件をアップデートし、再設計を要することになり、多大な労力を要することになる。

小さな変更は設計プロセスの中で頻繁に発生することであり、都度膨大な時間をかけて皆がアップデートすることを避けるため、異なる条件でパラレルに検討を進め、定

期的に整合性を取るというケースが少なくない。この整合性を取るタイミングは設計者の経験値に左右され、通常は影響度の多い少ないを都度判断しながら、進められていく。しかし、未知な変更を想定した「安全率」を見込んでおいたり、想定を超える影響度があった場合に大きな問題が生じたり、アップデート時にヒューマンエラーによって整合性が取り切れていないことが現実として発生するリスクを抱えている。

5.2　本システムのワークフロー

本システムによって一元化が図られている場合、A～D者が個別に変更したときにその変更内容が他の担当者が管理しているモデルや図面に自動的にアップデートされることになる。また、プラットフォームである FLUX とエンジンであるプログラミング言語との併用によってエンジン起動後はデータベースに一連の整合性の取れた結果、常に最新状態の図面が出されることになる。

これにより、大幅な作業短縮が可能となるだけでなく、確実にヒューマンエラーを無くすことが出来る。また、変更の影響度に応じた可否を判断することが容易となるために設計自体もスムーズとなる。一つの変更に対して、構造設計者間での打ち合わせを設けて、説明をするという行為も減らすことが可能になることから、例えばワークタイムが異なる海外担当者と協働することにも有効である。

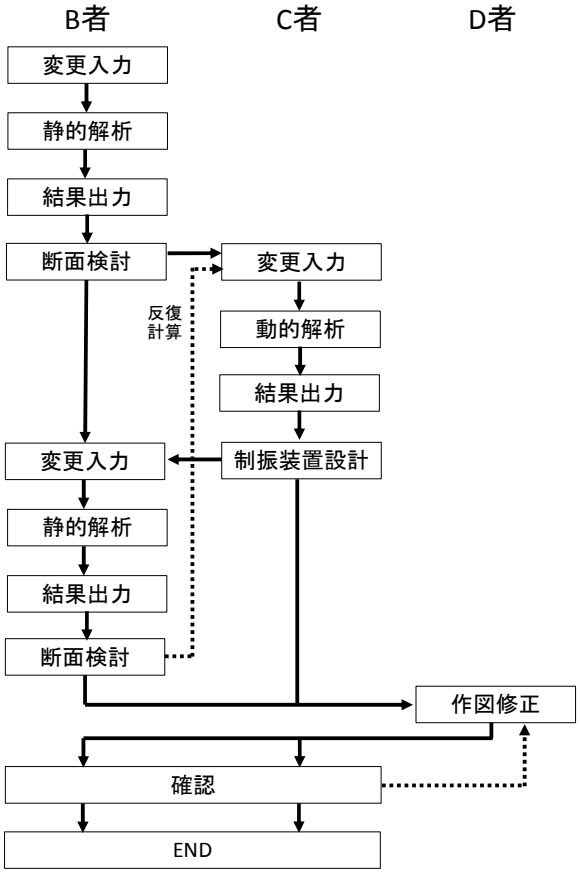


Fig.6　従来の変更時ワークフロー

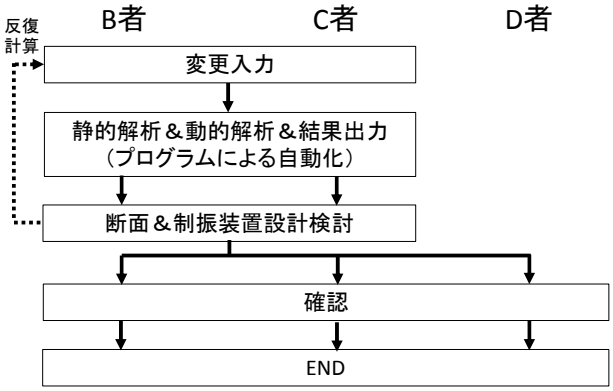


Fig.7　本システムの変更時ワークフロー

6.　今後の展望

本報告では、FLUX というプラットフォーム、プログラミング言語（Visual Basic）の併用により、構造設計に係わる多くの情報を一元化、計算の自動化、結果の可視化が可能となるシステムを提案した。

このシステムは将来的に発展させれば、構造だけでなく、意匠、設備、環境、防災、施工と建物に係わる全情報を共有化することも可能と思われる。また、パラメータをも自動で割り振り、検討機能の自動化 [4),5)]が実現できれば、幅広い意味での自動最適化が可能になるかもしれない。

一方で、このようなシステムは時間的な効率化の上で大変有効であるものの複雑化した情報の正誤を見極める能力は確実に必要な職能と言えよう。また、関係者の非常に多い建築設計において、終着点は様々な設計プロセスによって大きく左右される中、予期せぬ事象や不具合への対応も考えて重要パーツには適切な安全性・冗長性を見ながら設計を進めていくべき点などで、まだまだエンジニアの経験値や判断は重要に思う。

[参考文献]

1) 木下拓也,大崎純,発見的手法を用いた鋼構造平面骨組の多目的最適化,日本建築学会近畿支部研究報告集,2006
2) 久保和毅,幡司祐弥,藤井大地,太田和彦,骨組の位相最適化手法を用いた制震ダンパーの最適配置,日本建築学会中国支部研究報告集 第34巻,2011
3) Kermin Chok, Pavel Tomek, IMPROVING EFFICIENCY AND COLABORAION IN EARTHQUAKE TIME HISTORY ANALYSIS OF TALL BUILDINGS, NAFEMS AMERICAS CONFERENCE,2016
4) 澤田樹一郎,松尾彰,清水斉,骨組構造物の離散断面最適化のための GA の計算性能改善に関する研究, 日本建築学会中国支部研究報告集 第32巻,2009
5) 大崎純,最適化と冗長性,建築雑誌,2010

＊1　ｵｰｳﾞ・ｱﾗｯﾌﾟ・ｱﾝﾄﾞ・ﾊﾟｰﾄﾅｰｽﾞ・ｼﾞｬﾊﾟﾝ・ﾘﾐﾃｯﾄﾞ