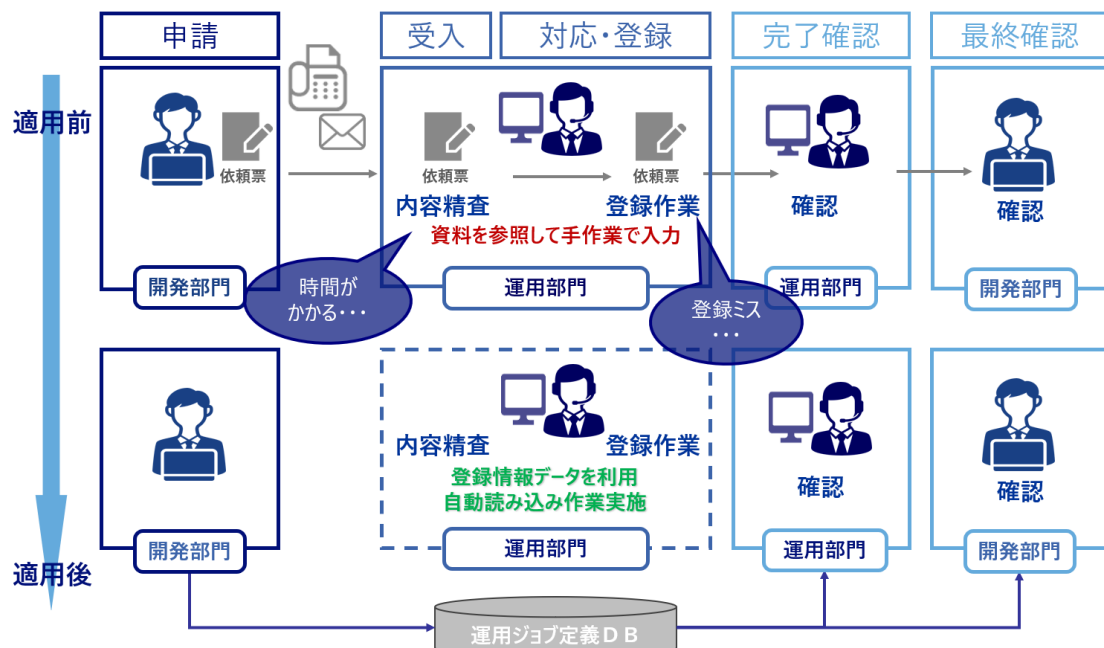


### 現状の課題

ジョブスケジュールは、本番環境のジョブ管理サーバーにある「ジョブ定義データ」に基づいて運用される。ジョブを追加したりジョブの構成を変更したりするには定義データの作成・変更が必要だが、この作業が現場でさまざまな問題を生んでいる。近年、システム運用における開発と運用の職務分離が進み、開発部門が本番環境の運用管理に関わる定義を直接変更できなくなっている。ジョブ定義データの維持管理は開発部門主導で行うが、開発部門は直接、本番環境の定義データに触れることができない。そのため、開発部門は申請書を表計算ソフトなどで作成し、運用部門はその申請書に従って本番環境の定義データを作成・変更する。運用部門は代行作業をしていることになる。

開発部門は軽微な変更でも一つひとつ詳細な変更申請書を作成し、運用部門に依頼しなければならない。ジョブ追加・変更の申請数が増加するのに伴い、開発部門も運用部門も、その作業負担は重くなり、運用コスト増となってしまう。作業負担が重くなれば、ヒューマンエラー（登録ミス・作業漏れ）の発生頻度も高まる。また、ジョブの構成が複雑化するにつれ、運用部門では申請の目的・意図を理解することが難しくなっている。その結果、開発部門と運用部門の間で誤解が発生し、開発部門が意図してない変更が行われることがある。



## **解決策**

開発部門によるジョブ定義データの作成・変更依頼に対し、運用部門が代行作業しなくてもよいように自動化する。具体的には、開発部門が自席の PC でジョブ定義データの作成・変更作業ができ、その定義データを電子化し、本番環境のジョブ管理サーバーに反映できる仕組みを作る。

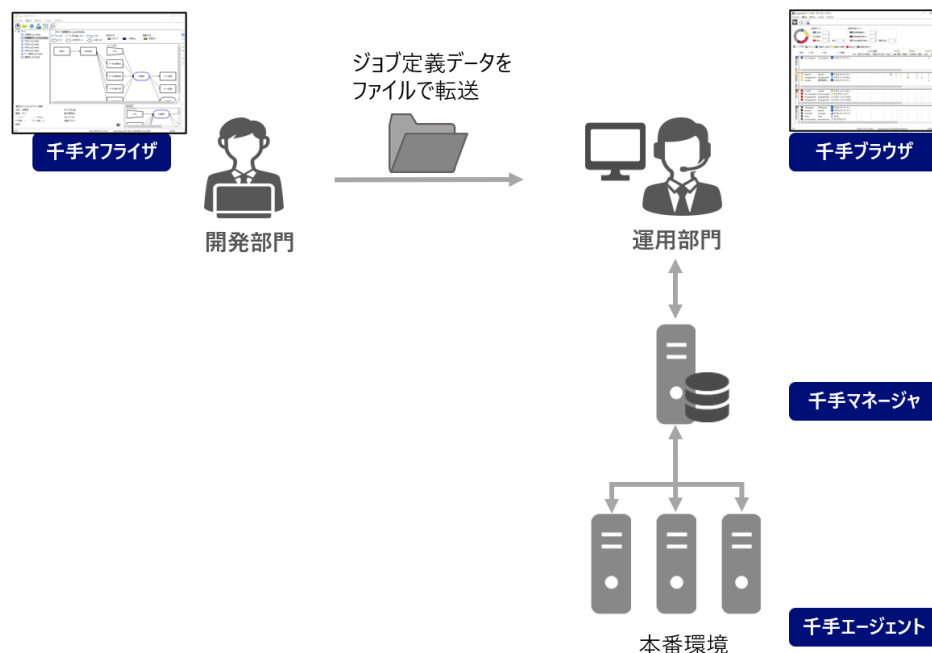
開発部門がジョブ定義データを効率よく作成・変更するには、視覚的なフロー図を使用できたり、既存の定義データをコピーして再利用できたりする必要がある。また、定義データに不備があると、本番環境の反映時にエラーとなり手戻りが発生する。そうならないように、自席の PC でジョブ定義の上限値や入力制限などのチェックが行えることが必要である。定義データを本番環境に反映する際は、変更した部分だけを反映するようになっている他、既存ジョブへの影響の有無を見極めなければならない。そのために、変更前の状態と変更後の状態の差分をチェックできる機能が重要となる。

ここまでの仕組みを作れば運用部門は定義データの作成・変更を代行作業しなくなるものの、定義データそのものについては内容を把握しておく必要がある。

そこで、誤解を生じさせないようにするため、運用部門と開発部門が同一インタフェースの GUI で、同一のジョブフロー図を共有できることが必要である。

## **Senju Family での実践方法**

Senju DevOperation Conductor では「千手オフライザ」というツールを利用する。開発部門の PC にこのツールをインストールすることで、ジョブ定義データ(「千手オフライザファイル」と呼ぶ)の作成や変更が可能になる。手順は次のようになる。運用部門は、本番環境の千手マネージャ(ジョブ管理サーバー)に接続した千手ブラウザ(コンソール)から、最新の千手オフライザファイルを書き出し、開発部門に渡す。開発部門は、千手オフライザで千手オフライザファイルを参照・変更し、修正後の千手オフライザファイルを運用部門に戻す。運用部門は開発部門が修正した千手オフライザファイルを、千手ブラウザの「読み込み」操作で本番環境に反映する。本番環境に反映する際は、全てのジョブ定義を反映するか、追加・変更した部分のみを反映するかを選択できる。



各種定義はテキストファイルで書き出し、読み込みも可能であり、大量のジョブ定義を作成する際などはテキストデータで複製、修正も可能である。

千手オフライザで定義を読み込んだ際に、各オブジェクトの読み込み結果や差分の詳細情報をコピーすることも可能なため、適用作業時の記録として情報を保存する上での利便性も向上している。

また、差分表示モードにてツリービューに追加・変更・削除のオーバーレイアイコンを表示する機能により各エンティティのリストビューを個別に参照する必要がなくなり、定義変更作業時の差分の把握が容易となっている。

