３．４　システム環境の変化への対応に関する教訓（Ｔ４）

[教訓 T4]
システム全体に影響する変化点を明確にし、その管理ルールを策定せよ！

問題

制御系システムでは、監視・制御を行う上位システム（以下、上位）とそれぞれが独立した制御装置・移動装置を持つ下位システム（以下、下位）とで役割分担をしているシステムがある。

列車制御システムも同様な構成で、列車の運転は上位の指令センターと呼ばれる所で集中監視を行っている。さらに、事故などがあって列車が乱れた場合に対処するために、数時間先までの運転がどのようになるかの予測をシステムで行い、その結果を予測ダイヤとして画面に表示している。

ある日、雪によるポイント不具合が早朝に複数の駅で発生した。駅間停車防止のため複数列車に抑止（駅に留まっている指示）を入力したところ、画面表示が全て消えてしまい、予測ダイヤを見ることができなくなった（図３．４－１）。

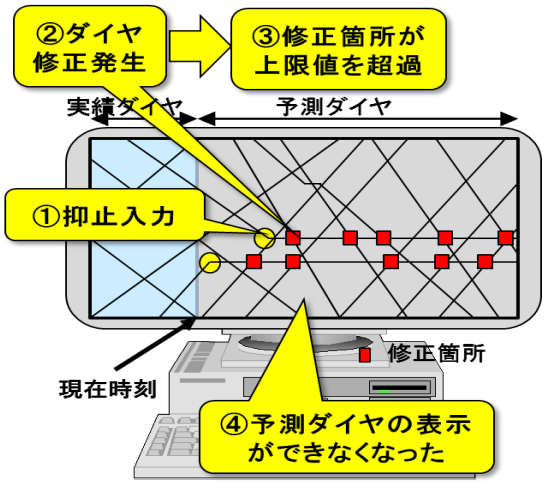

図３．４－１　障害発生状況とモニター画面

原因

直接原因は。駅間に列車が停車するのを防止するため、複数列車に対し駅で抑止入力を行った。この入力が早朝であったため、変更入力に対する予測ダイヤ上の「修正箇所」がほぼ 1 日分発生したので、システムの上限値（修正箇所：600 件）を超えてしまい、予測ダイヤを表示できなくなった。（上限値を超えた場合には画面を消すという仕様になっていた）

この上限値はシステム構築当初から決まっていたものだった。

根本原因は、システムに大きな変化点があったにも関わらず、それを見逃していたことである。以下、変化点の見逃しが、２つ存在していた。

【原因１】予測時間を 4H⇒24H に変更した際、そのような要件変更があったにも関わらず、「修正箇所数」の上限値の増加などシステム全体の機能要件変更を行わなかった。

【原因２】列車の本数が年々増加しており、本来ならば（運転本数の増加の都度、）上限値を超えた際のシステムの挙動を見直す必要があったにも関わらず、行わなかった。

このことから、根本原因は、全体に影響する変化点（この場合、予測時間、列車運転本数）が明確になっていなかったことである。

対策

予測ダイヤの処理（1 分おき）のたびに修正箇所のクリアを行うとともに、修正箇所 600 件までは予測ダイヤの演算処理を継続し、予測ダイヤを描画するようにプログラムを改修した。

また、システム全体で変更管理を行うルールを設け、要件が変更になるような案件に対しては、システムの見直しを行うようにした。

制御系システムは、上位と下位とで役割分担をしているため、上位が変更されても、下位（それぞれ独立した制御装置・移動装置）が変更されても、お互い無関係と思われがちである。そのために、システム全体での変化点の管理が必要になる。

今回の問題については、以下の 2 つの変化点の管理を行った。

・上位における仕様変更（予測時間の変更等）　→　【原因１】の対応

・下位における列車制御装置の変更（列車本数の増加等）　→　【原因２】の対応

変化点とは、全てのシステム環境に変化があった時点を指す。制御系システムでの変化点は、一般的なシステムの仕様変更の他に、対象時間、対象機器の動き、機器数の変化なども含む。障害を事前に防止するためには、この変化を見逃さない仕組みをつくることがポイントである。

特に、機能の拡張（制限値に影響する場合）時は、変化点を見逃してはならない。この事例のようにアプリケーションプログラムの中でテーブル内のデータ件数の制限を持っているパターンは非常に多い。このような制限値が変化点の管理指標となる。

制御系システムの変化点の管理ルールを明確にし、そのルールを守る仕組みを構築するために、次の 3 点を行う。

・システムが監視・制御する対象と仕様の変化点を網羅する。

・変化点管理のルールとそれを守る仕組みを構築する。

・変化点管理で使用する管理指標を関係部門で共有し、「変化点の見落とし」を防ぐ。

これらの対策により、現場環境の変化、システム要件に変化があった時点を可視化することができる。

効果

制御系システムの変化点の管理を行うことにより、社会インフラの混乱を防ぐことができる。

このような制御系システムは、列車運転に限らず、工場における生産ライン制御、電力の供給ライン管理制御、通信の交換機制御など、世の中に数多く存在している。これらのシステムにおいても変化点の管理を応用し、社会インフラの混乱を防ぐことができる。

教訓

変化点を見逃すと、後日重大な障害を起こすことになる。システム全体に影響する変化点を明確にし、その管理ルールを策定することが重要である。