3.29 システム環境の変化への対応に関する教訓（その 2）（T29）

教訓 T29　単位などの定義が異なる制限値、連携するシステム間で使っていませんか？

問題

A 社の物流センタでは、全国に拠点ターミナルを設置し、拠点ターミナルでは、配送担当の運転手にハンディ端末を持たせ、配送先への荷物の優先度に応じた配送を管理している。この管理は、連携する 3 つのシステムを用いて行われている。中央管理センタの全社配送管理システムは、全社の配送計画を作成し、拠点配送計画システムへ全体配送計画を送る。拠点配送計画システムは、全体配送計画を受け取り、拠点ターミナルから入力される「配送ルート」データと「配送車両」データを受け取り、配送車両に応じた個別の配送ルート計画を策定し、その情報を個別配送監視システムに送る。個別配送監視システムは、個々の配送車両と配送ルートの配送状況をリアルタイムで管理する（図 3.29－1）。

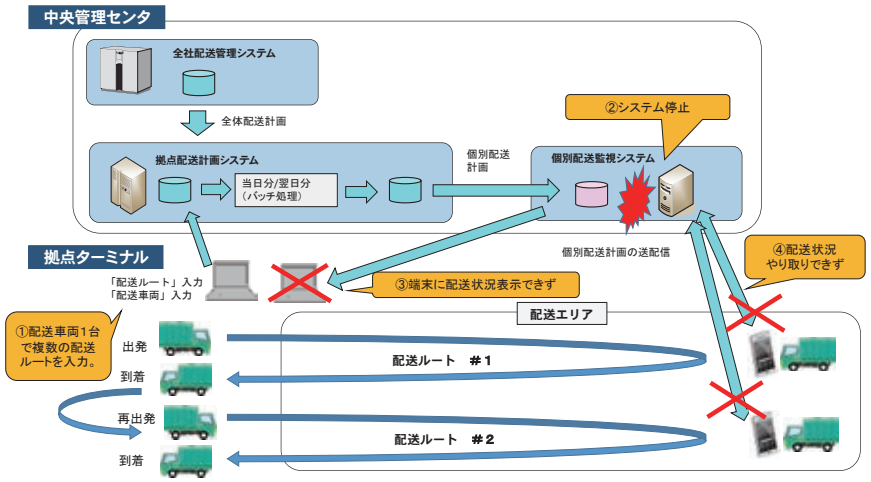

図 3.29－1　システム構成と障害状況

障害のあった日は、大型連休明けの初日でもあり、大量の荷物を扱うことになったため、前日に登録する配送車両1台当たりの配送ルート数が増えていた（図 3.29－1①）。障害のあった日の早朝、個別配送監視システムが停止してしまい（図 3.29－1②）、拠点ターミナルでは配送状況が分からなくなり（図 3.29－1③）、運転手も配送状況のやり取りができなくなった（図 3.29－1④）。原因はすぐにつかめなかったが、配送そのものについては順調に行われていたので、拠点ターミナルと運転手の間で、電話でのやり取りで対応し、その日の業務を終えることができた。

業務終了後に原因が判明し、緊急システム改修（暫定対応）を実施し、翌日早朝にシステムは復旧した。

原因

A 社は、過去にシステムの配送ルート数の制限値の管理をしていなかったために、システムが停止する障害を発生させたことがあった。その障害を踏まえ、配送ルート数の制限値を拠点配送管理システムと個別配送監視システムを通し、一律管理する運用を行っており、1 日当たりの配送ルート数の制限値を、「1,000 ルート」としていた。ここでいう配送ルート数は、例えば配送車両 1 台が拠点ターミナルから出発して戻ってくるまでを 1 ルートと数える。繁忙時に、1 台の車両で拠点ターミナルを 2 回出入りすれば、2 ルートと数えるので、物理的な車両台数ではない。

A 社のシステム部門は、拠点配送管理システムから個別配送監視システムに転送されるデータは、拠点配送管理システムで上限を超えないことについて担保されるため、下流側の個別配送監視システムでは制限値を超える可能性についての検討は重要視していなかった。そのため、システム部門は、拠点配送計画システムの開発ベンダとは別のベンダである個別配送監視システムの開発ベンダに、制限値「1,000 ルート」と伝えただけであった。

しかし、このシステムでは、配送ルート数として持つ制限値としては、当日分の配送ルート数と、翌日分の配送ルート数が存在していた。それは、拠点ターミナルで当日に翌日分の配送計画を事前に立案できるようにし、拠点ターミナルでの負担を減らすためであった。その制限値が 2 つあるにも関わらず、個別配送監視システムの開発ベンダは、当日「1,000 ルート」、翌日「1,000 ルート」とあるべきところ、当日、翌日あわせて「1,000 ルート」と理解して、システムの設計、製造を行ってしまった（図 3.29－2）。

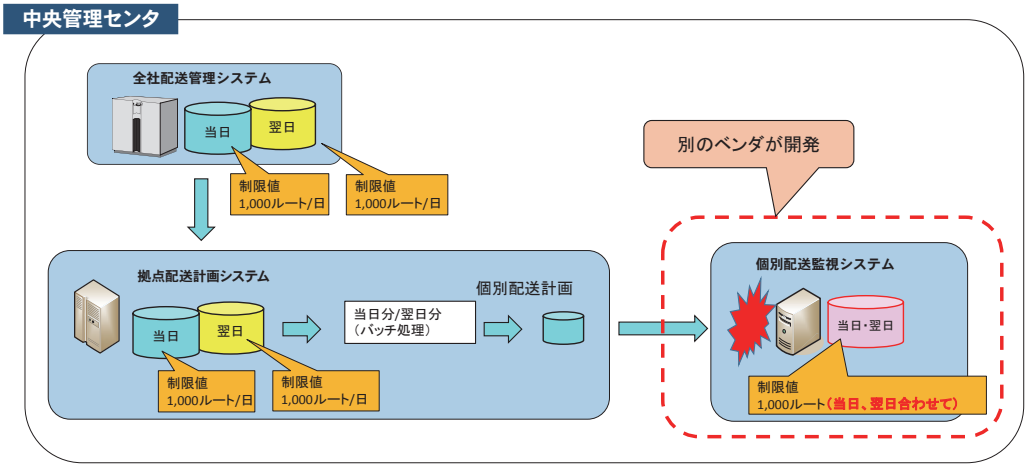

図 3.29－2　制限値の意味の違い

今回の障害は、当日「900 ルート」、翌日「400 ルート」の 2 日分のデータが、個別配送監視システムの制限値「1,000 ルート」を超えた「1,300 ルート」であったために起きた。直接原因は、上位システムと下位システムとで制限値の定義が異なっていることに気づかなかったことであった（図 3.29－3）。

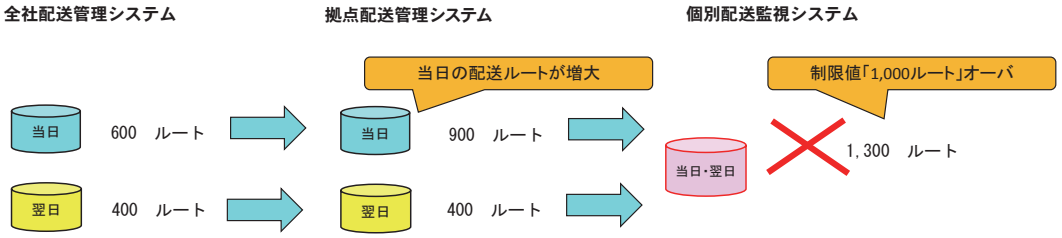

図 3.29－3　制限値オーバになるケース

根本原因は、機能仕様書を作成するとき、制限値の単位（本事例では、1 日ごと）や、システム連携範囲（本事例では、システム全体で一元管理）を明確にすることまで考慮が及ばず、記載内容を曖昧なままにしていたことであった。そのため、設計レビューでもその不整合を確認することができなかった。また、このような制限値を確認する総合テストも行っておらず、そのため、システム部門、開発ベンダともに十分な確認が行われていなかった。

対策

緊急対策として、個別配送監視システムの制限値を、「1,000 ルート」→「2,000 ルート」に拡張した。

再発防止策としては、以下の手順を実施した。

⑴ 個別配送監視システムの機能仕様書の制限値の定義見直し

- 機能仕様書とプログラム実装の制限値の定義に差異がないよう比較調査し、記載内容の見直しを行った。
- 文章だけではなく、シーケンス図等を使い処理の流れを明確化にした。

⑵ 個別配送監視システムの制限値超過時の振る舞い調査

- 個別配送監視システムの全制限値において、制限値超過時の振る舞いを設計書から調査した。誤りがあれば修正し、また説明が足りない部分は、それを設計書に明文化した。

⑶ 各システム間の制限値再点検と一元管理

連携するシステム間で持つ制限値は、その定義を明確にし、複数のシステムの制限値が同じ意味を持つことを確認する。さらに、そのような制限値を一元管理する。

- システム間の制限値の整合性を確認した。
- 運用時に、実際のデータが制限値にどこまで迫っているのかのデータ使用率調査を行うこととした。
- 上記の調査結果を常に監視できるよう、データ管理方法の見える化（システムごとの制限値一覧の作成、その制限値の変更履歴管理、性能調査状況履歴管理、等）を行い、システム間での制限値がどのように使われているかを明らかにし、すべての制限値を共通パラメータ化して定義する一元管理の運用を開始した。将来的には、共通パラメータ値の変更により自動的にすべての制限値定義の変更ができるよう、システム化することを検討する。

制限値には、機能要件と言われる業務要件から決めるものと、非機能要件と言われるシステムリソースの制限から決められるものがある。

また、システム間で管理すべきものもあるが、あるシステム内の閉じた中で管理するものもある。さらに個々のプログラムのロジックの関係から制限値を決めている場合もあり、その管理は複雑であり、すべてを管理することは至難である。

したがって、制限値の確認が必要な事象をとらえ、実際の運用に入る前に、本番環境と同様な環境で、事前テストを行うことが有効である。

個別配送監視システムの恒久的な対策として、制限値再調査の結果を受け、システム間で整合性を保つよう制限値を拡張した上で、制限値、およびピーク時の実データでの確認テストを実施することにした。

事例では制限値の中の上限値オーバが障害となったが、下限値についても同様に管理する必要があり、その他の注意点も、当然のことではあるが、以下になる。

- 制限値には、上限値、下限値がある。また、等号（=）が含まれるかも確認する。
- 上限値、下限値は、プラス、ゼロ、マイナス　なのかを確認する。
- 上限値、下限値は、一般に、「△△当たり○○」（または、○○ / △△）といった形で表され、△△と○○は、単位を明確にする。例えば、「1 日当たり 1,000 個」（または、1000 個 / 日）、「1 時間当たり 100 円」（または、100 円 / 時間）などのように確定する。

  「△△当たり」は、「時間当たり」、「面積当たり」、「個別当たり」などが当てはまる。「時間当たり」の単位は、1 秒、1 分、1 時、1 日、1 月、1 年、・・・などであり、「面積当たり」の単位は、1㎢、1㎡、・・・など、「個別当たり」は、1 プログラム当たり、1 人当たりなど、様々な単位が考えられる。○○は、「数量」の単位（ルート、台、回、個、トン、・・・など）であるが、「%」などの率もある。

図にまとめると、以下のようになる（図 3.29 – 4）。

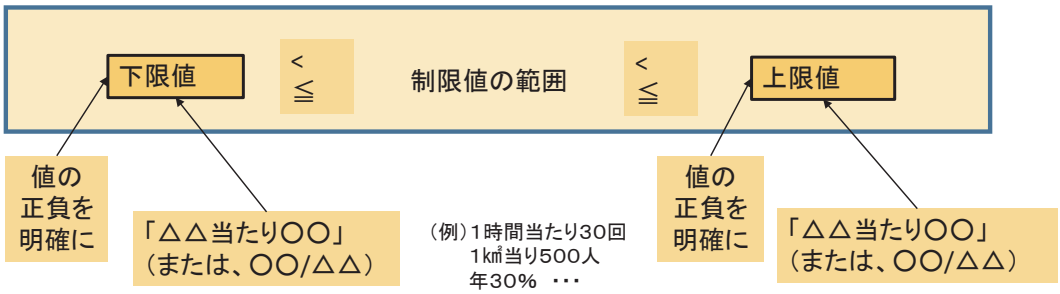

図 3.29 – 4　制限値の明確化ポイント

効果

制限値の管理を厳密に定義する、つまり、システムで持つ制限値は、数値だけでなく、「単位」、「単位当たり」、サイン（符号）、上限値、下限値など、正確な定義をした管理を行うことにより、連携システム間の不整合によって起こるシステム障害の発生を防ぐことができる。具体的には、以下の対策を行うことにより実現される。

- 設計時のレビュー時に制限値の定義を明確にする。
- 制限値付近の境界テストや、制限値の定義に合った動作ができるかのテストを行う。

また、これらの対策を実施するにあたり、マルチベンダでの開発時におけるコミュニケーションギャップによる不具合にも対応することができる。

教訓

連携システム間で持つ制限値は、その定義を明確にすることにより、複数のシステムの制限値が同じ意味を持つことが確認でき、さらに、そのような制限値は、一元管理することが重要である。

本事例では、1 日の制限値「1,000 ルート／日」であったが、システムによっては、1 日の制限値に加えて、1 時間の制限値、あるいは 1 秒の制限値など、意味の異なる複数の制限値を個別に管理する場合もあり得る。また、本事例では上限値オーバであったが、下限値の制限も忘れてはならない。特に「－（マイナス）」がチェックできなかったため、誤った結果をもたらすこともある。そのような点も踏まえ、この教訓を活用していただきたい。

この教訓では、制限値の定義を主題にしたのだが、他に、制限値（上限値）オーバに関する教訓があるので、参考にしていただきたい。制限値の管理に関しては、その定義を明確化することと、それが変わる変化点を見逃さないこととを合わせて検討していただきたい。

【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！

また、システム間での連携の在り方を述べた教訓がある。こちらも参考になると考える。

【教訓 T5】 サービスの視点で、「変更管理」の仕組み作りと「品質管理責任」の明確化を！

さらに、このような制限値を一元管理する教訓がある。こちらも参考になると考える。

【教訓 G13】 キャパシティ管理は関連システムとの整合性の確保が大切

【教訓 T18】 新たなサブシステムと老朽化した既存システムとを連携する場合は両者の仕様整合性を十分確認すべし