

ガス開閉栓システムのモバイル化

HTML5を使ったオフラインWebアプリケーションの 業務システムでの適用

0. はじめに（会社概要）

当社は、東京ガスのシステム部門から1987年に分離独立して設立された東京ガスの100%子会社であり、前身の東京ガス時代を含め、およそ半世紀に渡って東京ガス関連のシステム開発・インフラ構築・運用・維持管理を行っている会社です。

エネルギー業界においてトップランナーとして走り続ける東京ガスを、ITの側面からサポートしリードしていくことを使命としています。

このたび、長年に渡って利用されてきた既存の開閉栓のシステムを再構築することになり、その中で構築したモバイルシステムにて次世代WEB標準技術「HTML5」を採用しました。

基幹系業務システムでのHTML5の採用事例はまだ少なく、本書では、その事例として当社での開発事例をご紹介します。

1. ガスの開閉栓業務の概要



開栓

お客さまが入居時にガスをご利用できるようガスメーターの栓を開くとともに、点火試験、ガス機器の安全確認を行う作業

①ガスメーターの栓を開く



②お客さま宅内でガス点火試験



③お客さま所有ガス機器安全確認



④安全に利用いただく為の各種ご案内



閉栓

お客さまが転居時にガスのご使用を中止するために、ガスメーターの栓を閉じる作業

①ガスメーターの栓を閉じる



②必要に応じ、その場で料金精算



東京ガスから委託を受けた、東京ガスライフバルの作業員にて作業を実施します。

2. 新ガス開閉栓システム開発の目的

①お客さま対応向上

お客さまからの当日の急な開閉栓のご依頼があっても、速やかに現場作業員に作業情報を連携することで、迅速なお客さま対応が可能に。

②事務所管理業務効率化

現場での作業結果入力により、管理者による内容チェックの軽減や事務担当者によるシステム入力作業が不要に。

従来、作業票（紙）で実施していた作業報告をモバイル化することで刷新！

③現場業務効率化

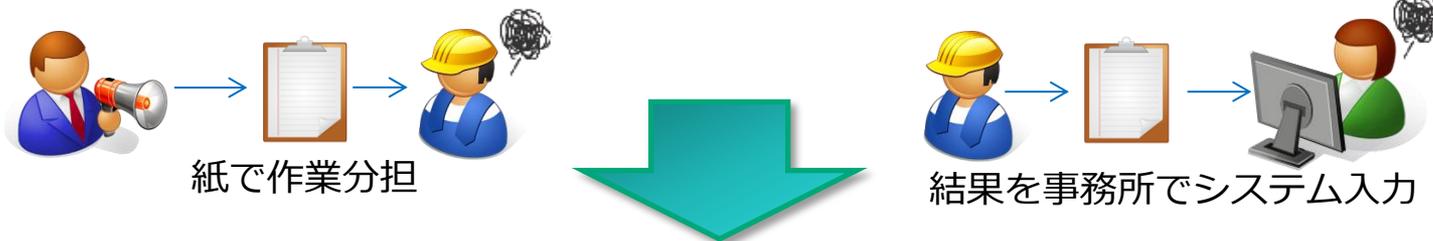
モバイルによるリアルタイムでの作業結果報告により、作業後の手書き報告の負担や、帰社後の作業票チェックが軽減。また追加分担の情報を出先にて受取ることができ、作業票を事務所へ取りに戻る手間が不要に。

④作業品質向上

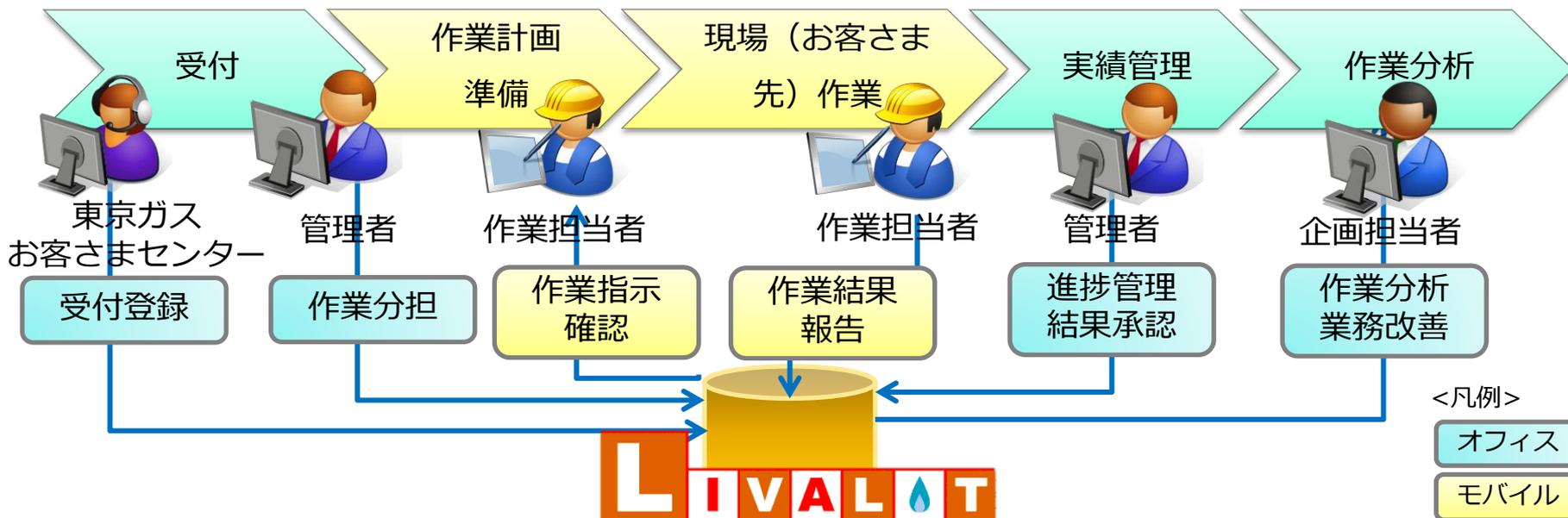
アプリケーションのチェック機能で、現場で入力した報告がその場でチェックされ、報告漏れや間違い報告を防ぎ、作業品質の向上に。

3. 業務フロー

✓ 今までは作業票（紙）で作業、帰社後に報告で非効率。



✓ モバイル端末導入で作業毎にリアルタイム報告が可能に。
⇒作業効率化、作業品質向上の実現！



4. システム規模

✓ 開閉栓業務は繁閑差が激しい業務。

	通常時	ピーク時（3月末）
利用者数	500～800人	2,000人
作業件数	6,000～10,000件/日	40,000件/日

⇒年度末の引越しピークには、通常期は他の業務を行っている人も応援に入り、現場総動員。



5. HTML5の採用

- ✓ 地下や高層建物など通信圏外エリアでの作業もある開閉栓の現場において、求められる要件は通信圏外であっても業務継続を可とすること。
- ✓ 開発当時、スマートフォン・タブレットが台頭し、将来の主流OS・デバイスが不透明。



- ✓ オフライン機能を具備し、マルチデバイス・マルチプラットフォームで動作することを実現する技術「HTML5」を採用！



6. システム構成

<端末>



LenovoX220、X230Tablet
(Windows7搭載コンバーチブル型タブレット)

<利用ブラウザ>



Google Chrome

※Chromeの採用理由
開発当初（2010年）、オフラインDBとしてWebSQLDatabaseを前提としていた為、実装が進んでいたChromeを採用。現在HTML5のオフラインDBとして標準として進められているIndexedDBは、2010年頃は搭載ブラウザはほぼ無かった。

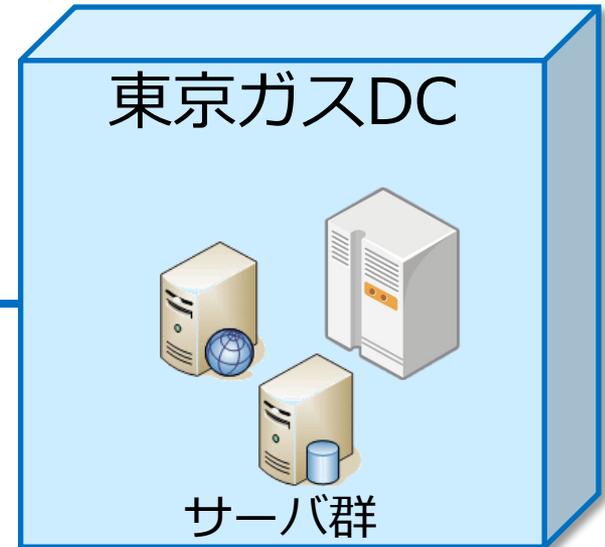
データ通信回線



認証設備



閉域ネットワーク



東京ガスDC



サーバ群

7. HTML5の利用機能

✓ ApplicationCache

- ✓ 通信圏外でもオフラインで利用でき、高速なレスポンスを実現。

✓ LocalStorage

- ✓ ユーザーセッション情報やアプリで必要となる各種設定値の保存に利用。

✓ WebSQLDatabase (※現在はHTML5仕様外)

- ✓ 業務に必要なデータを端末上に保持し、レスポンスを向上させ、オフライン時でも利用できるように。

✓ Canvas

- ✓ 作業完了時にいただくお客さま手書きサイン等に利用。

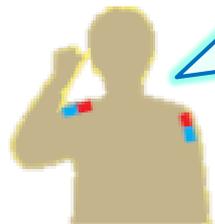
8. HTML5開発における工夫した点

- ✓ 勧告されていないHTML5の仕様変更リスクに備え、アプリケーション影響を最小化するために、HTML5のAPIをラップする部品を開発・標準化。
- ✓ HTML5採用にあたり、モバイル基盤のJavaScriptフレームワークを社内で整備。
- ✓ HTML5とJavaScriptを利用してリッチなアプリケーションを構築したが、一部ブラウザの機能だけでは完全に要件を満たせなかった。（例.モバイル帳票等）
⇒JavaAppletを利用して、OS依存のローカルアプリと連携できるようにした。

9. 画面イメージ

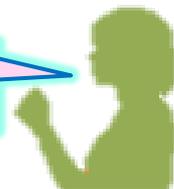


10. 導入効果（現場の声）

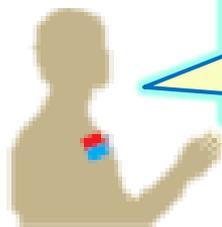


以前は、作業票に記載した結果を帰社してからシステムに入力していましたが、モバイルになりその場で報告が完結するので、残業が減って家に早く帰れるようになりました。

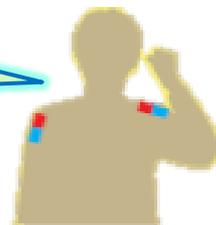
事務の負担もだいぶ減りました。



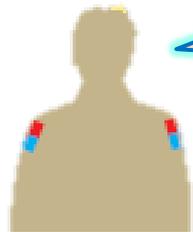
私は管理者ですが、進捗状況がすぐ分かるので追加の作業分担などの管理が行いやすくなりました。



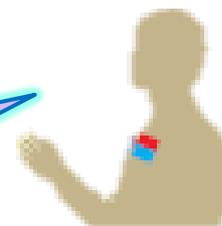
事務所に戻ることなく追加の分担が受け取れて、効率的になりました。



アプリケーションのチェックによって入力ミスや報告漏れが減りました。



作業票の紛失リスクが減って安心して作業できます。



1 1. 今後の展開

- ✓ 今回開発したモバイル基盤にて、定期保安点検業務のシステムも再構築を行っており、2013年7月に稼動する予定。

※定期保安点検業務とは、ガス事業法に基づいて、3年に1度各家庭のガス配管や給排気設備の調査を行う業務。



- ✓ 開閉栓や定期保安点検以外のフィールド業務のシステムについても、再構築時に当モバイル基盤での開発を検討。
- ✓ 今回の開発ではセキュリティや既存アプリ動作担保等、様々な要件によりWindows7のモバイル端末を採用したが、ユーザーからはより軽量・薄型モバイルが望まれており、今回構築したモバイル基盤が将来的に新たなデバイスで動作するよう継続して挑戦していく。



End