

UHF 帯 RF タグに適したデータフォーマット変換手法の評価

米澤 祐紀[†] 陳 卓^{††} 江原 正規[†] 井上 亮文[†] 星 徹[†]

[†]東京工科大学コンピュータサイエンス学部 ^{††}東京工業大学大学院社会理工学研究科経営工学専攻

1 はじめに

RF タグ内のデータフォーマットの国際標準である ISO/IEC 15962:2004[1] では、RF タグに格納するデータは Object と呼び、Object の識別番号である Object-Identifier を付加し、圧縮などの操作を経たデータ (以下、Data-Element) を RF タグに格納する。RF タグには、Data-Element をユーザが自由に追加、変更可能なユーザメモリ領域が存在する。しかし、UHF 帯 RF タグのユーザメモリ (以下、ユーザメモリ) は、word 単位 (1word=16bit) でのアクセスしか出来ない。そのため、Data-Element が word を使い切らない場合があるため、Data-Element の追加、変更の際に、Data-Element 間の境界処理が冗長で複雑になる。

RF タグの書き込み手法は、一部分の変更、追加の際に全ての Data-Element を書き直すという手法 [1] が一般的である。また、ユーザメモリ内の構造を事前にデータベースへ登録することにより、読み込みを行わずにユーザメモリ内部を把握する手法 [2] が存在する。しかし、これらの手法は、Data-Element 追加の際に生じる境界処理の問題を解決してはいない。

本研究では、前述した Data-Element 間の境界処理の冗長性を簡易化することを目的とし、ISO/IEC 15962 にて Data-Element をロックするためのパディング機能である Offset を用いて問題の解決を図る。

2 提案

Offset は Data-Element に組み込むことが可能なパラメータであり、パディングの役割を果たす。Offset を利用し、Data-Element を word 単位に配置する。Offset を利用したユーザメモリのデータ構造を図 1 に示す。データ構造は、データのエンコードルールとマッピングルールを示す DSFID と複数の Data-Element の並び順からなる。1つの Data-Element は、複数の要素から成り立っている。

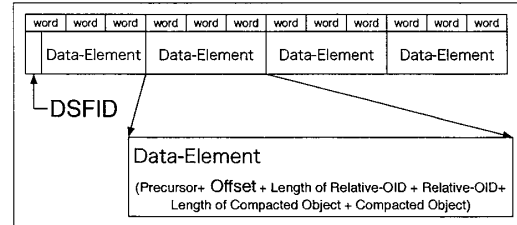


図 1: ユーザメモリ内のデータマッピング

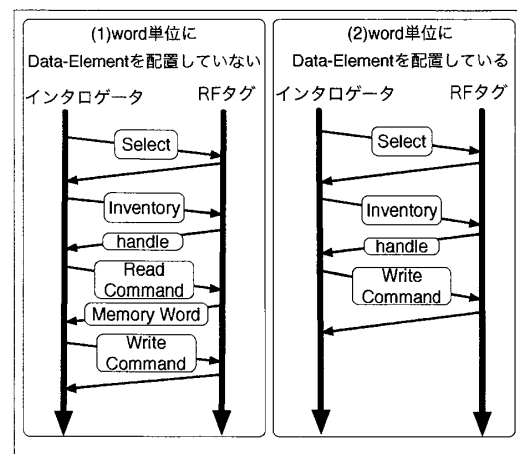


図 2: Data-Element を格納する際のエアコマンドのシーケンスの比較

図 1 のように、Data-Element が word 単位に配置されることにより、特定の Data-Element の変更や追加を行う際に、前後の Data-Element を考慮せずに書き込むことが可能になる。

word 単位に Data-Element を配置している場合と格納していない場合のエアコマンドの違いを図 2 に示す。前提条件として、ユーザメモリ内のデータ構造は事前に把握しているものとする。操作は、新規 Data-Element を追加する場面を想定する。(1) では、既にユーザメモリに書かれている Data-Element が word の途中で終了している可能性があるため、一度全ての Data-Element を読み込む。そして、word の途中で Data-Element が終わっている場合は、最後の word の 8bit を含めた形にして Data-Element を書き込みを行う。(2) は、指定された場所に Data-Element を書き込む。このように、(2) では、Read Commandを行う必要がなくなり、処理の高速化につながると考えられる。

Data Format Conversion Method for UHF RFID

[†] Yuki YONEZAWA(yyonezawa@star.cs.teu.ac.jp)

^{††} Zhuo CHEN(chen.z.ab@m.titech.ac.jp)

[†] Masaki EHARA(mehara@star.cs.teu.ac.jp)

[†] Akifumi INOUE(akifumi@cs.teu.ac.jp)

[†] Tohru HOSHI(hoshi@cs.teu.ac.jp)

School of Computer Science, Tokyo University of Technology,
1404-1 Katakura, Hachioji, Tokyo 192-0982, Japan ([†])

Department of Industrial Engineering and Management,
Graduate School of Decision Science and Technology, Tokyo
Institute of Technology (^{††})

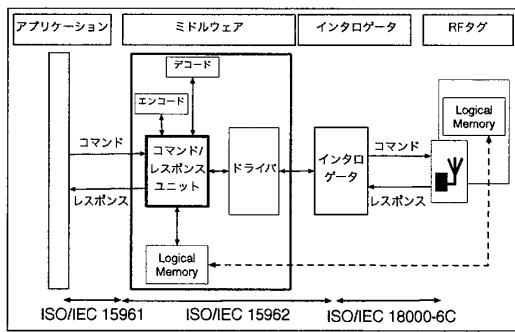


図 3: 実装の概要図

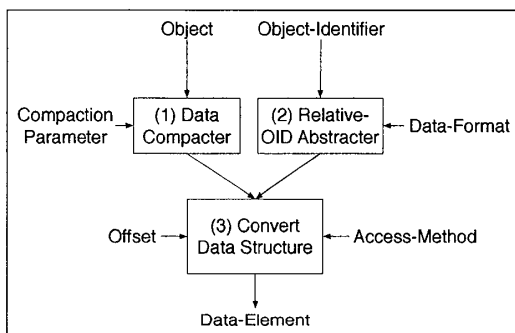


図 4: Data-Element を出力するためのフローチャート

3 実装

RF タグを利用するシステムを図 3 に示す。システムは、アプリケーション、ミドルウェア、インタロゲータ、RF タグの 4 つに分けられる。本研究では、ミドルウェア部分の実装を行った。そして、Offset を付加する実装は ISO/IEC 15962 部分のコマンド/レスポンスユニット部分に実装を行った。

ユーザメモリに書き込む Data-Element を出力するためのフローチャートを図 4 に示す。(1) 指定された Compaction Parameter により Object の圧縮を行う。(2) 指定された Data-Format から Object-Identifier の Relative-OID を抽出する。(3)(1), (2) から得られた情報と、指定された Access-Method から、Data-Element を形成する。この時に Data-Element が word 単位を使い切らない場合のみ Offset を付加する。

実装に使用した機器を表 1 に示す。

4 評価

UHF 帯 RF タグは物流に使用される。そこで利用が想定される、GLN[†]とタイムスタンプを 1 組として、RF タグへ追加書き込みを行った。書き込む 2 つの Data-Element は 184bit であった。結果を図 5 に示す。(1) は、Offset を用いず追加書き込みを行う手法である。この手法では、Data-Element が word の途中で終わっ

表 1: 使用機器

開発言語	python 2.5
RFID リーダライタ	日立情報通信エンジニアリング UHF 帯 RFID 対応 RW モジュール
RF タグ	μ-Chip Hibiki シールラベル (ユーザメモリ領域 1532bit)

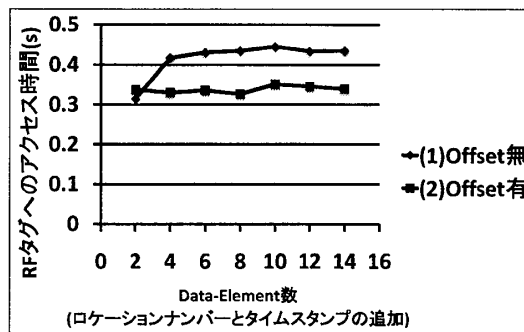


図 5: Offset の有無による RF タグへのアクセス時間の比較

ている可能性があるため、一度全ての Data-Element を読み込む。(2) は本稿が提案する Offset を利用して Data-Element を word 単位に配置する手法である。結果より (2) は (1) と比べて 22 パーセントの時間短縮が可能となった。これより、大量の荷物を扱う場合には流入量を増やすことが可能となる。

5 まとめ

本稿では Offset を用いて Data-Element を word 単位に配置することを提案し、その評価を行った。word 単位に配置することで、Data-Element 間の境界処理の冗長性を簡易化することができ、結果として RF タグへのアクセス時間の短縮を図ることができた。

今後は、ISO/IEC 15962:2008 で採用が決まっている Profile 方式との比較、また、複数の RF リーダライタを用いた Data-Element の書き込み手法などを検討していきたい。

参考文献

- [1] ISO/IEC 15962: Information technology - Radio frequency identification (RFID) for item management - Data protocol: data encoding rules and logical memory functions (2004).
- [2] 荻原浩輔, 三次 仁, 中村 修, 村井 純: EPC-global network におけるユーザデータの効率的な処理のためのフィルタ&コレクションの拡張, 電子情報通信学会技術研究報告. SIS, スマートインフォメディアシステム, Vol. 107, No. 373, pp. 63-68 (2007).

[†] Global Location Number : 国際標準事業者コード