

配信型情報源統合環境における 統合要求記述の正当性の検証

筑波大学

渡辺陽介 石川佳治 北川博之

DEWS2002

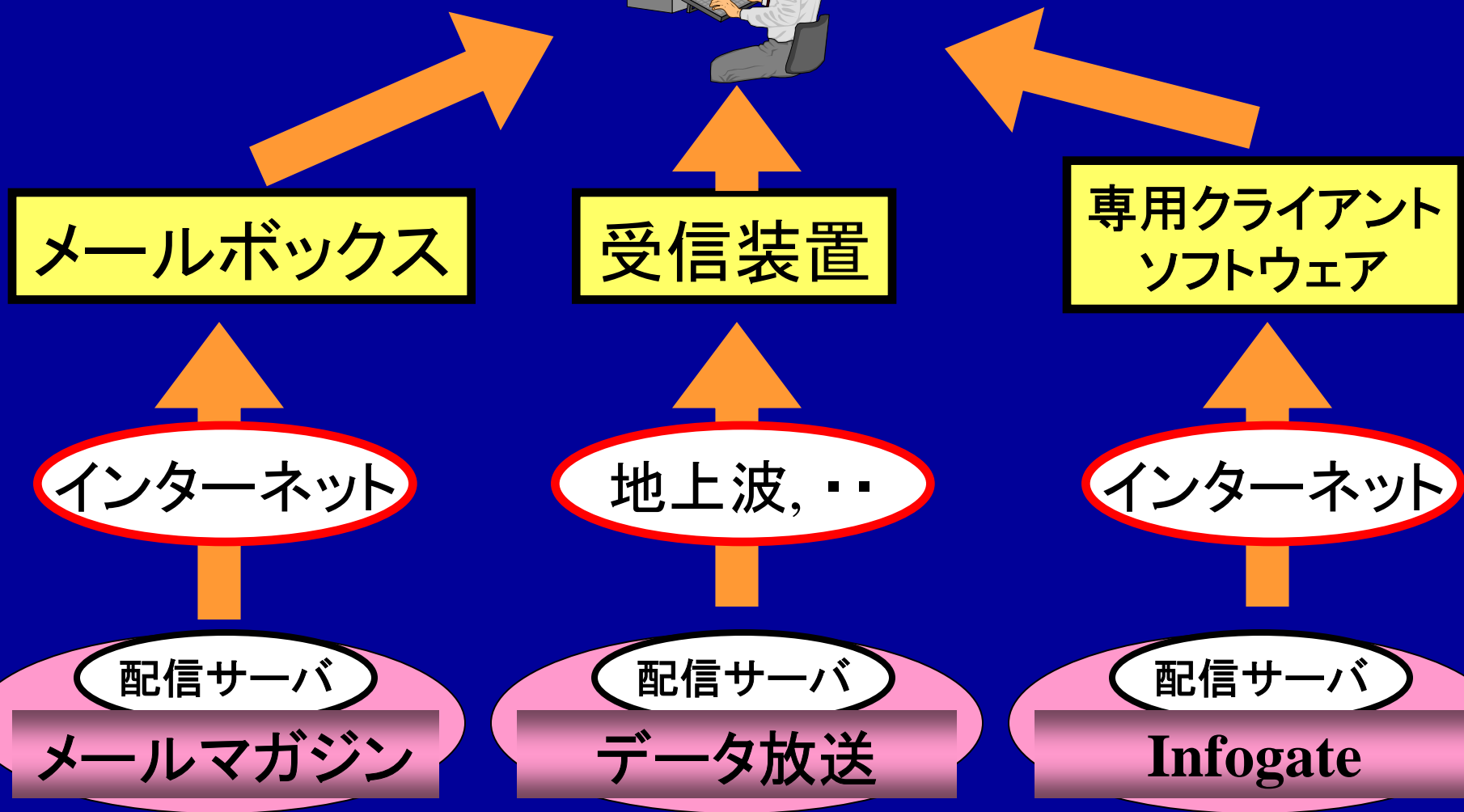
発表手順

1. 背景
2. 配信型情報源統合環境
3. 目的
4. 統合要求記述の検証
 - 情報源の性質記述
 - 整合性と充足可能性
 - 検証手法
5. まとめと今後の課題

背景

- ネットワークの発達により, 大量の情報源へ容易にアクセス可能
 - 情報源の統合利用の重要性が増大
- さまざまな種類の情報源が出現
 - データ放送, メールマガジン等, 配信型情報源と呼ばれる新しい情報源

配信型情報源



背景

- ネットワークの発達により, 大量の情報源へ容易にアクセス可能
 - 情報源の統合利用の重要性が増大
- さまざまな種類の情報源が出現
 - データ放送, メールマガジン等, 配信型情報源と呼ばれる新しい情報源

配信型情報源統合環境の構築

発表手順

1. 背景
2. 配信型情報源統合環境
3. 目的
4. 統合要求記述の検証
 - 情報源の性質記述
 - 整合性と充足可能性
 - 検証手法
5. まとめと今後の課題

配信型情報源統合環境

代数式による統合要求記述を
与えることで、新たな情報配信
サービスを定義できる

代数式による
統合要求記述



新たな配信
サービス

統合結果

配信型情報源統合環境

配信単位

配信単位の
選択・蓄積

Server

メールマガジン

配信単位の
選択・蓄積

配信単位

Server

データ放送

RDB



統合利用例(1)

◆ サッカー情報配信サービス (Soccer)

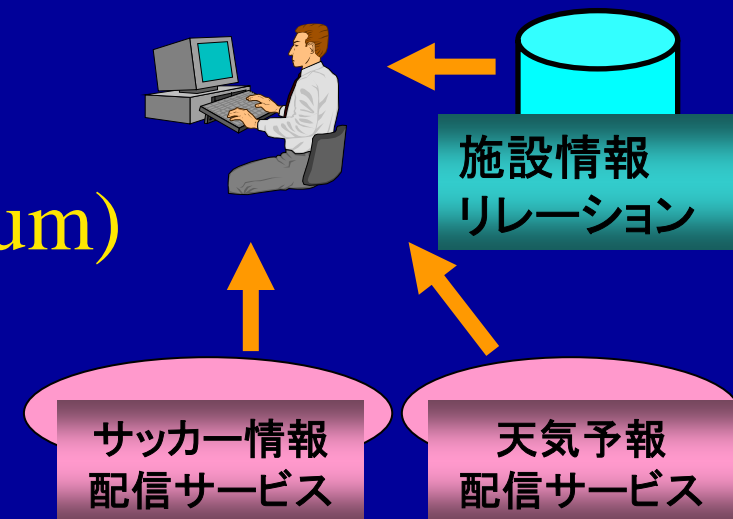
- サッカーの試合に関する情報源
- 毎週水曜日に情報が届く

◆ 天気予報配信サービス (Weather)

- 7日分の天気予報を地域ごとに提供する情報源
- 毎日午前6時に情報が届く

◆ 施設情報リレーション (Stadium)

- 試合会場に関する情報源



統合利用例(2)



サッカーの試合に関する情報が到着したら同じ日に届いた天気予報と試合会場の地図をつけて翌日の午前0時に配信して欲しい

```
<result>
  <event> game </event>
  <weather> fair </weather>
  <map> [map icon] </map> ....
</result>
```

サッカー情報

date	location	weather	type
3/06	Tokyo	fair	
3/07	Tokyo	rain	7days
3/08	Tokyo	fair	7days
3/09	Tokyo	fair	7days

天気予報

<soccer>	<event>	game	</event>
date>	3/09</date>	<place>	E...

試合会場の地図

name	location	map
E	Tokyo	[map icon]
F		
⋮		

施設情報リレーション

配信サーバ
天気予報配信サービス

配信サーバ
サッカー情報配信サービス

統合要求記述(1) 統合結果

配信予定時刻を
サッカー情報の
到着時刻の次の日の
午前0時に設定

OTS	place	date	weather	map
3月07日0時	A	3/09	fair	
3月14日0時	B	3/16	rain	

タイムスタンプ関数

Ω next 0:0:0(Soccer.ITS)

ユーザが要求する
統合結果の構造・
条件を指定する

リレーショナル代数式 $E(I_1, \dots, I_n)$

ITS	place	...
3月06日5時	A	...
3月13日5時	B	...

サッカー情報
配信サービス Soccer

ITS	date
3月06日6時	3/09
3月13日6時	3/16

天気予報
配信サービス Weather

name	map
A		
B		

施設情報
リレーション Stadium

統合要求記述(2)

$$O_{\text{new}} = \Omega_{f(I_k.\text{ITS})}(E(I_1, \dots, I_n))$$

O_{new} : 統合結果のリレーション

Ω : 配信時刻を求める演算子

$f(I_k.\text{ITS})$: タイムスタンプ関数

$E(I_1, \dots, I_n)$: リレーショナル代数式

I_1, \dots, I_n : 統合対象の情報源

統合例における要求記述

$O_{new} = \Omega_{\text{next(Soccer.ITS)} ($
 $0:0:0$
 Soccer
 $\boxtimes \text{Soccer.date} = \text{Weather.date}$
 $\wedge \text{Weather.ITS} < \text{next(Soccer.ITS)}$
 $\wedge \text{previous(next(Soccer.ITS))} \leq \text{Weather.ITS}$
 $\sigma_{\text{Weather.type} = '7\text{days}'} ($
 $0:0:0, 0:0:0$
 $\text{Weather})$
 $\boxtimes \text{Weather.location} = \text{Stadium.location}$
 $\wedge \text{Soccer.place} = \text{Stadium.name}$
 Stadium
 $)$

タイムスタンプ関数

`immediate(t)` : t そのものを返す

`nextp(t)` : 最も近い未来に条件 p を満たす時刻を返す
「次の日の午前0時」

`previousp(t)` : 最も近い過去に条件 p を満たす時刻を返す
「前の日の午前0時」

`after δ_t (t)` : δ_t だけ経過した時刻を返す
「一日後」

`before δ_t (t)` : δ_t だけ遡った時刻を返す
「一日前」

配信・統合処理



統合要求
記述

ルール生成
モジュール

ルール

ルール処理
モジュール

蓄積要求
統合・配信要求

統合結果

イベント
通知

メディエータ

ラッパー

ラッパー

ラッパー



タイマー

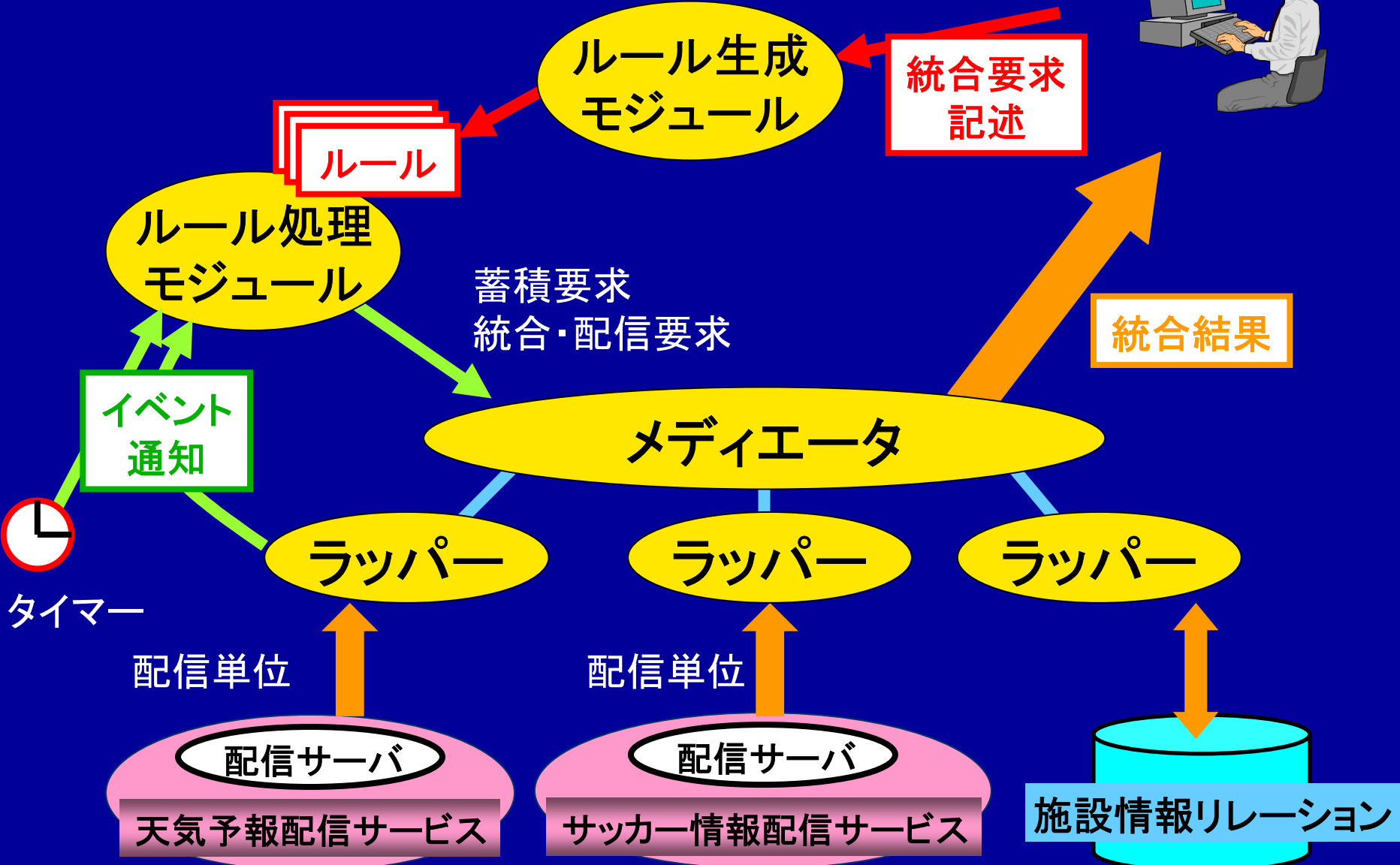
配信単位

配信単位

配信サーバ
天気予報配信サービス

配信サーバ
サッカー情報配信サービス

施設情報リレーション



発表手順

1. 背景
2. 配信型情報源統合環境
3. 目的
4. 統合要求記述の検証
 - 情報源の性質記述
 - 整合性と充足可能性
 - 検証手法
5. まとめと今後の課題

統合要求記述検証の必要性

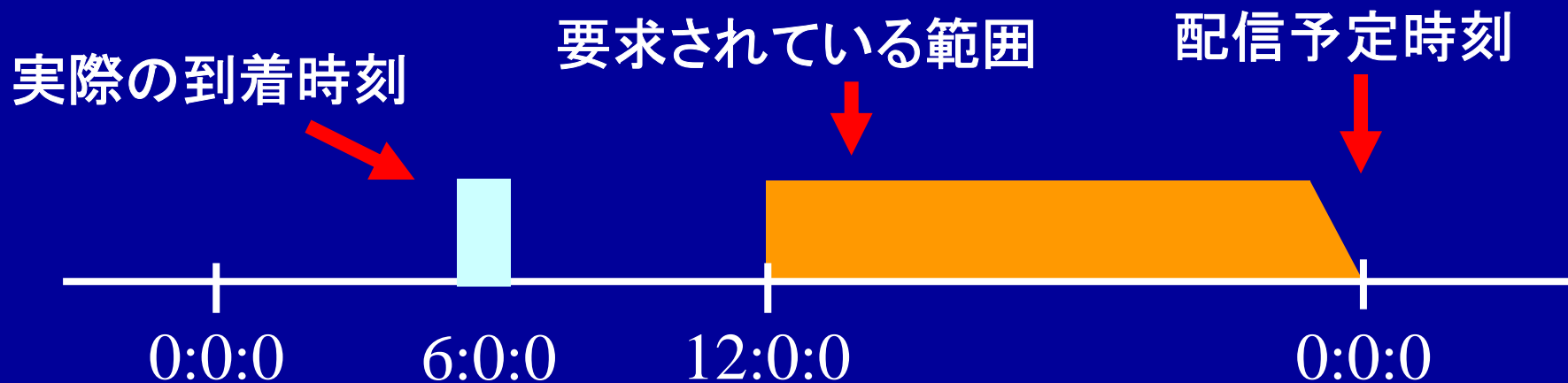
- 配信型情報源では、配信単位が届くまでに時間がかかる
 - 要求を記述した次の日の午前0時に結果が届く
- 統合結果を見てから、統合要求記述のエラーを修正することは効率的でない
 - 要求を修正し、さらに次の日の午前0時まで待つ

統合要求記述を検証し、事前にエラーを検出することが重要

情報源の性質を考慮した検証

統合要求記述のみを検証したのでは不十分

(例) 「毎日午前0時に、前の日の午後が届いた
天気予報を配信して欲しい」



情報源の性質から、要求された範囲に
配信単位が来ないことが検出可能

本研究の目的

配信型情報源の性質を考慮した 要求記述の妥当性検証

- 配信型情報源の性質記述
- 検証のための基準
整合性, 充足可能性
- 実際の検証方式

発表手順

1. 背景
2. 配信型情報源統合環境
3. 目的
4. 統合要求記述の検証
 - 情報源の性質記述
 - 整合性と充足可能性
 - 検証手法
5. まとめと今後の課題

配信型情報源の性質記述

$$I \equiv \sigma_{\text{PROP}}(I)$$

σ : 選択演算

左辺の式と右辺の式が
同じ結果になることを保証する

 情報源 I に含まれるすべてのタプルは
必ず条件PROPを満たす

性質記述の例(1)

◆ サッカー情報配信サービス(Soccer)の性質

「情報が毎週水曜日に到着する」



ITS属性が水曜日の午前0時から
木曜日の午前0時の間にある

$$\text{Soccer} \equiv \sigma_{\substack{\text{previous(ITS)} \leq \text{ITS} \\ \text{Wed:0:0:0}} \wedge \text{ITS} < \text{after}(\text{previous(ITS)}) \\ \text{1:0:0:0} \quad \text{Wed:0:0:0}} (\text{Soccer})$$

性質記述の例(2)

- ◆ 天気予報配信サービス (Weather) の性質

「情報は毎日午前6時に届く」

Weather $\equiv \sigma$ (Weather)

ITS = after(previous(ITS))
6:0:0 0:0:0

性質記述の適用

統合要求記述中で参照している情報源の性質を
統合要求記述へ反映させる

統合要求記述

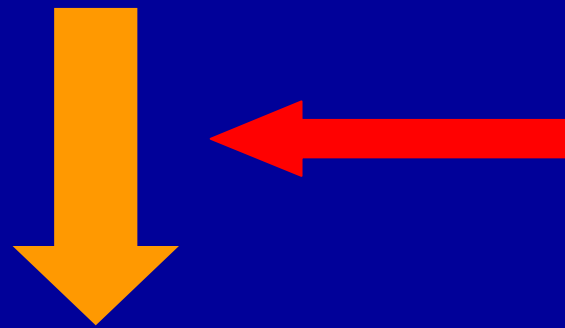
$$\Omega_{f(I_k, ITS)} E(I_1, \dots, I_n)$$

情報源の性質記述

$$I_1 \equiv \sigma_{PROP1}(I_1)$$

⋮

$$I_n \equiv \sigma_{PROPn}(I_n)$$



$$\Omega_{f(I_k, ITS)} E(\underline{\sigma_{PROP1}(I_1)}, \dots, \underline{\sigma_{PROPn}(I_n)})$$

発表手順

1. 背景
2. 配信型情報源統合環境
3. 目的
4. 統合要求記述の検証
 - 情報源の性質記述
 - 整合性と充足可能性
 - 検証手法
5. まとめと今後の課題

整合性

意味

ユーザに配信する統合結果が、配信時刻までに到着した情報のみを使って生成できることを保証する

定義

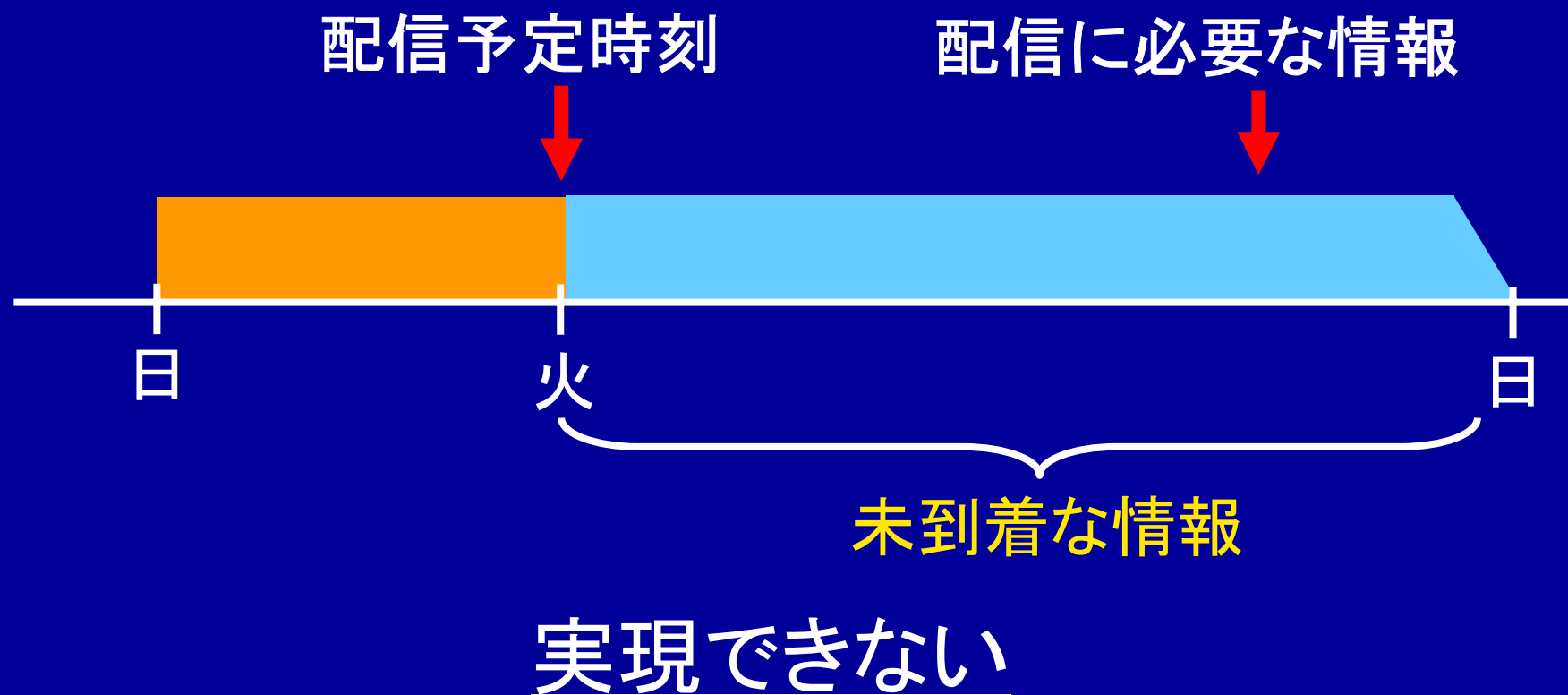
すべての配信時刻 $f(u)$ において以下の等式が成り立つ

$$E'(I_1, \dots, I_n, f(u)) = E'(\sigma_{ITS \leq f(u)}(I_1), \dots, \sigma_{ITS \leq f(u)}(I_n))$$

ただし $E'(I_1, \dots, I_n) = \sigma_{IK.ITS \in f^{-1}(t)}(E(\sigma_{PROP}(I_1), \dots, \sigma_{PROP}(I_n)))$

整合性を満たさない配信要求例

(例) 「毎週火曜日の午前0時に、その週に届いたサッカー情報全てを配信して欲しい」



充足可能性

意味

ユーザが配信を要求する情報が実際に存在することを保証する

定義

ある未来の $f(u) \geq \text{now}$ において、以下の式が成り立つ

$$E'(I_1, \dots, I_n, f(u)) \neq \phi$$

ただし $E'(I_1, \dots, I_n) = \sigma_{IK.ITSE \in f^{-1}(t)}(E(\sigma_{PROP}(I_1), \dots, \sigma_{PROP}(I_n)))$

発表手順

1. 背景
2. 配信型情報源統合環境
3. 目的
4. 統合要求記述の検証
 - 情報源の性質記述
 - 整合性と充足可能性
 - 検証手法
5. まとめと今後の課題

統合要求記述の検証手法(1)

- 整合性の判定

統合結果が配信時刻 $f(u)$ までに到着した情報から生成できる

→ 整合性を満たす統合要求には $ITS > f(u)$ という条件を満たす統合結果は存在しない

- 充足可能性の判定

ユーザの要求する情報が実際に存在する

→ 充足可能性を満たす統合要求には、統合結果が1つ以上存在する

統合要求記述の検証手法(2)

条件を満たす統合結果が存在するか否かを調べる

問題点: タイムスタンプ関数に依存した検証機構が必要

$$\begin{aligned} \text{previous}(\text{after}(\text{Soccer.ITS})) &\leq \text{Soccer.ITS} \\ \text{Soccer.ITS} &< \text{after}(\text{previous}(\text{Soccer.ITS})) \\ \text{previous}(\text{next}(\text{Soccer.ITS})) &\leq \text{Weather.ITS} \\ &\vdots \end{aligned}$$

→ タイムスタンプ関数を線形な関数に変換した上で条件を満たす解が存在するかを検証する

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &\leq b_1 \\ &\vdots \\ a_{m1}x_1 + \dots + a_{mn}x_n &\leq b_m \end{aligned} \quad \text{ただし } x_1, \dots, x_n \geq 0$$

タイムスタンプ関数の変換規則

immediate (n) \longrightarrow n

next (n) \longrightarrow a m + b

(ただし $n < am + b \leq n - a$)

previous (n) \longrightarrow a m + b

(ただし $n < am + b \leq n - a$)

after (n) \longrightarrow n + c

before (n) \longrightarrow n - c

a, b, cは正の整数

条件式の変換例

統合例の統合要求記述から時刻に関する条件式を抽出

$\text{previous}(\text{next}(u)) \leq \text{Soccer.ITS}$

$\text{Soccer.ITS} < \text{next}(u)$

$\text{previous}(\text{Soccer.ITS}) \leq \text{Soccer.ITS}$

$\text{Soccer.ITS} < \text{after}(\text{previous}(\text{Soccer.ITS}))$

$\text{previous}(\text{next}(\text{Soccer.ITS})) \leq \text{Weather.ITS}$

$\text{Weather.ITS} < \text{after}(\text{next}(\text{Soccer.ITS}))$

$\text{previous}(u) \leq u$

$u < \text{after}(\text{previous}(u))$

変換結果

$$x1 < \text{day} \cdot x2$$

$$\text{day} \cdot x2 \leq x1 + \text{day}$$

$$\text{day} \cdot x2 - \text{day} \leq \text{day} \cdot x3$$

$$\text{day} \cdot x3 < \text{day} \cdot x2$$

$$\text{day} \cdot x3 \leq x4$$

$$x4 \leq \text{day} \cdot x2$$

$$x4 - \text{week} \leq \text{week} \cdot x5 + 3\text{day}$$

$$\text{week} \cdot x5 + 3\text{day} < x4$$

$$\text{week} \cdot x5 + 3\text{day} \leq x4$$

$$x4 < \text{week} \cdot x5 + 4\text{day}$$

$$x4 < \text{day} \cdot x6$$

$$\text{day} \cdot x6 \leq x4 + \text{day}$$

$$\text{day} \cdot x6 - \text{day} \leq \text{day} \cdot x7$$

$$\text{day} \cdot x7 < \text{day} \cdot x6$$

$$\text{day} \cdot x7 \leq x8$$

$$x8 \leq \text{day} \cdot x6$$

$$x8 - \text{day} \leq \text{day} \cdot x9$$

$$\text{day} \cdot x9 < x8$$

$$x8 = \text{day} \cdot x9 + 6\text{hour}$$

$$x1 - \text{week} \leq \text{week} \cdot x10 + 3\text{day}$$

$$\text{week} \cdot x10 + 3\text{day} < x1$$

$$\text{week} \cdot x10 + 3\text{day} \leq x1$$

$$x1 < \text{week} \cdot x10 + 4\text{day}$$

$x1, \dots, x10$ は変数

week, day, hourは正の整数

線形不等式の解の存在判定(1)

$$\begin{array}{r} a_{11} x_1 + \dots + a_{1n} x_n \leq b_1 \\ \vdots \\ a_{m1} x_1 + \dots + a_{mn} x_n \leq b_m \end{array}$$

ただし $x_1, \dots, x_n \geq 0$

この制約を満たす解が存在するか？

→ 線形計画法を用いた判定法

線形不等式の解の存在判定(2)

線形計画問題

$$\text{制約} \left\{ \begin{array}{l} a_{11} x_1 + \dots + a_{1n} x_n \leq b_1 + x_a \\ \vdots \\ a_{m1} x_1 + \dots + a_{mn} x_n \leq b_m + x_a \end{array} \right.$$

$$\text{目的関数} \quad -x_a \quad \text{ただし } x_1, \dots, x_n \geq 0 \quad x_a \geq 0$$

制約の範囲内で目的関数 $-x_a$ の最大値を探す

問題を解き、最大値が0になるならば、そのときの解は元の条件式の解

検証手順

1. 配信要求記述と性質記述から以下の式を導出し、時刻に関する条件式を抜き出す

$$E'(I_1, \dots, I_n, t) = \sigma_{I_k} \text{ITS} \in f^{-1}(t) (E(\sigma_{\text{PROP}1}(I_1), \dots, \sigma_{\text{PROP}n}(I_n)))$$

2. t を $f(u)$ で置き換え、 $\text{PROP}k(u)$ を条件式に追加する

3. 整合性の判定

- 各情報源 li ごとに以下の処理を繰り返す。全ての情報源について結果が真ならば統合要求記述は整合性を満たす
- 元の条件式に $\text{ITS}i > f(u)$ を追加し、線形計画問題を得る
- 問題を解いて、目的関数が0に最大化できれば整合性を満たさない解が存在するので偽、そうでなければ真

4. 充足可能性の判定

- 元の条件式に $f(u) \geq \text{now}$ を加え、線形計画問題を得る
- 問題を解いて、目的関数を0に最大化できれば充足可能性を満たす

まとめ

- 配信型情報源統合環境における統合要求記述の検証手法
 - 配信型情報源の性質の考慮
 - 整合性・充足可能性
 - 実際の検証手法

今後の課題

- プロトタイプシステムを用いた検証手法の評価
- より一般的な情報源の持つ性質への対応