DBSJ Journal, Vol.10, No.1
June 2011

整合性を考慮した注釈伝播

Annotation Propagation with Consistency

青戸 了† 清水 敏之†† 吉川 正俊††

Ryo AOTO Toshiyuki SHIMIZU
Masatoshi YOSHIKAWA

本論文では，注釈の整合性を考慮した注釈伝播手法を提案する．注釈は，一般的に特定のデータ集合に対し，それぞれ特定の意図を持って記述されおり，データ処理によって付与対象のデータが変化することによって，注釈内容が無効となってしまうという問題がある．そのため，データの理解支援という注釈の役割を保証するためには，データの変遷を通して，注釈が示す情報の正確性や妥当性を検証し維持することが必要とされる．本稿では，データ処理によって起こる注釈の品質劣化の問題を解決するため，注釈の意味を表現するための表現モデルを提案する．また，注釈の意味的な定義に従い，データ変遷において生じる注釈の不整合を定式化するとともに，注釈の整合性を維持するための手法を提案する．本論文の提案手法により，注釈の整合性を確保した伝播が可能である．

1. はじめに

近年，ウェブによるデータ流通が一般的に行われるようになった．それに伴い，データ生成過程の複雑化や，データソースの多様化が進んでおり，データ品質の維持や理解支援がより重要な問題となっている．

データ品質や理解支援のための情報を保持する手段として古くからメタデータが利用されている．従来はデータコレクション単位やファイル単位でのメタデータ付与が一般的であったが，データの複雑化や多様化を背景とし，より細かい単位でのメタデータ付与が行わるようになった．細粒度で付加されるメタデータを一般的に注釈（アノテーション）と呼ぶ．注釈は一般的にセル単位やレコード単位もしくは，それらの集合に対して文章を関連付けることで表現される．注釈の記述内容は，データの理解支援を行うためのコメントや，データ品質を表す情報，データ起源や処理過程などの来歴情報などからなる．注釈の代表的な利用例として，curated database[1] が挙げられる．curated database とは，一般的なデータベースとは異なり，人手によるデータ収集によって作成されるデータベースである．注釈は理解支援のためのコメントを付与する目的で用いられることが多く，多彩な起源を持つデータの相互利用のためには不可欠である．また，一般的な商業データや科学データなどにおいては理解支援のためのコメントの他に，データ品質の確保を目的として，起源情報や生成過程に関する情報を含む来歴情報が付与される．上記のように，単一の表に対して複数のメタデータが様々な粒度で生成されるという状況において，それらの品質維持が非常に重要な問題となる．

† 学生会員 京都大学情報学研究科社会情報学専攻修士課程
aoto@db.soc.i.kyoto-u.ac.jp
†† 正会員 京都大学 情報学研究科 社会情報学専攻
{tshimizu, yoshikawa}@i.kyoto-u.ac.jp

FilmTable

title	year	genre	leading_actor1	leading_actor2	director
film1	2000	Mystery	actorA	actorB	director1
film2	2001	Comedy	actorB	actorC	director2
film3	2003	Period film	actorA	actorB	director3
film4	2005	Mystery	actorA	actorB	director4
film5	2008	Mystery	actorA	actorB	director1

図 1 映画データベース

データベース中のデータへの注釈付与に関する研究は現在までに数多く行われている[2]～[4]. 注釈の表現やそれらに対する問合せに関する議論に加え，注釈の伝播に関する問題が存在する．伝播とはデータに付与されている注釈を演算や更新等の処理を通し，それらを出力へ再付与する概念である．起源データに付与されていた情報を，様々な処理を経て生成されるデータに対し正しく継承するという目的のために重要な議論であり，注釈の各表現に対して同様に議論が行われている．従来の注釈伝播に関する議論は，それらがどこから伝播されたのかという整合性（where provenance[5]）のみを保証するものであった．ところが，注釈は一般的に，付与対象としたデータに対し特定の意図をもって生成されるため，演算や更新等によるデータの変遷に伴い，本来持っていた内容に不整合が生じるという問題がある．以下，映画データベースを例に，注釈の品質劣化に関する問題を示す．

［例 1.1］ 図 1 は映画に関する表であり，映画名，公開年，ジャンル，主演 1，主演 2，監督という六つの属性を持つ．この表には映画の内容に関する事実がデータとして格納されており，ユーザがそれぞれ任意の部分のデータにコメントを付与し，それらはユーザ間で共有される．データベースに格納される事実に対し，注釈は一般的に構造化されていない内容や，主観の含まれる内容であることが多い．例として次のような注釈が付与されているものとする．

（1） a_1 ：この二人は共演することが多い．

（2） a_2 ：ミステリー映画においては，同僚役で出演することが多い．

（3） a_3 ：続編の中では一作目と同じ監督の作品が面白かった．

なお，film4 と film5 は film1 の続編であるとする．注釈 a1,a2,a3 はそれぞれ図 1 の影付き部，実線部，破線部に付加されている．この表に対し，主演と監督の関係を抽出するために主演 1，主演 2，監督の三属性を射影するという操作を考える．一つ目の注釈は，actorA と actorB という値が存在すれば意味を持つ内容であるため，不整合とはならない．対して，二つ目の注釈は，同じ値の組の中でも，ジャンルという文脈を仮定された上で生成された注釈であるため，上記のような操作を行った場合，事実としての根拠が失われてしまい，整合性を失う．また，表 FilmTable に対して，film5 の前作に当たるようなレコードが結果に表れないような選択演算が行われたとき，三つ目の注釈は，表内に存在する特定のレコードの存在と，その属性値を仮定して生成されているため不整合となる．

以上のように，仮に伝播が正しく行われた場合でもそれぞれの注釈が仮定している情報が処理によって欠落してしまった場合，注釈の持つ意味について不整合が生じる場合がある．本論文では，問合せによるデータ導出や更新操作などによって生じる注釈の品質劣化の問題を解決するため，注釈の

不整合分析を行う．我々は，分析を行うための議論として，まず意味を考慮した注釈モデルを定義する．文脈と呼ばれる概念を導入することによって，従来研究では取り扱うことのできなかった，注釈の意味を表現する．定義された注釈の意味に従い，演算の適応によって生じる不整合を定義する．さらに，定義された不整合に対し，注釈の整合性を確保するための手法を提案する．提案手法によって，注釈の品質を維持しながら，データ処理を通した伝播が可能となる．

以下構成として，第 2 節において関連研究について述べる．第 3 節において意味を表現するための注釈モデルを定義し，それに対応する注釈の不整合を定式化する．第 4 節において，注釈の整合性を確保するための手法について述べる．第 5 節において，まとめと今後の課題を述べる．

2. 関連研究

これまでにデータベースにおける注釈の表現について様々な研究が行われている． それらは注釈単位当たりの粒度によって分類することができ，セル単位[4]，サブタプル単位[3]，任意の粒度（レコード集合のサブタプル単位）[2]などによる表現がある． 従来の研究は，注釈をデータベース上でいかに表現するか，またそれらをいかに処理を通して伝播させるかについて議論を行っている． 注釈伝播に関する議論では，注釈の位置的な整合性，つまり入力において関連付けられていた注釈は間違いなく出力でも同一のデータに関連付けられることを保証している． しかし，伝播された注釈の中には変遷後のデータ中において意味を成さなかったり，注釈の作成者の意図に反する意味を持つ場合があり，従来の研究ではそれらを取り扱うことはできなかった． よって，本論文では新たに注釈の意味的な整合性を考慮した伝播手法を提案する．

3. 注釈モデル

本節では意味を考慮した伝播を実現するための注釈モデルを定義する．本論文では，任意の組の任意の属性集合に対して関連付けられた文書を注釈と呼び，以下のように表わす．

$$a = (R, T, A, Cont)$$

R,T,Aはそれぞれ，注釈の付加対象となる表，組集合，属性集合を表し，$Cont$は注釈の内容を表わす．注釈は以下の性質を持つものとする．

- **一貫性**

注釈$a = (R, T, A, Cont)$について，関連付けの対象となる組集合Tには以下のような制約が成り立つ．

for all $t \in T, S = \sigma_{A=t.A}(R)$,

$$\bigcup_{s \in S} s = T$$

上式は，注釈の付加範囲が一意に定まることを表している．注釈は必ず表内の値を根拠として関連付けられるため，直感的には，異なる二つの組が，注釈に参照される部分について全く同じ条件を持ちながら，片方の組は注釈が関連付けられるが，他方は関連付けられないというような付加は許されないことを表現している．関連付けの対象となる組集合Tがこの制約を常に満足するためには，Tを以下のように選べばよい．

$$T = \sigma_{Cond}(R)$$

$Cond$は選択条件式と呼ばれ，付加対象である属性集合とその値の組によって構成される論理式である．

次節では，文脈という概念を取り入れ注釈を定義する．

3.1 文脈

文脈とは，直感的には注釈が存在するための背景となる条件である．注釈の付加対象である属性集合において同様の値を持ちながら，特定の組集合のみに注釈が関連付けられている場合，それはそれぞれの組が持つ背景が異なるためである．その違いを判断するには必ず，付加範囲の属性集合以外の部分を根拠とする必要があり，それを判断するための条件が我々の定義する文脈に当たる．また，文脈とは注釈が存在するための根拠であると言い換えることができ，文脈として参照されているデータが注釈の存在を支えている．この根拠の崩れを不整合と呼ぶ．不整合の定義については，次節で詳細を述べる．本論文では，各組が文脈を満足するかどうかという判断を，特定の属性集合の値やレコードの存在を根拠として行う．ここで注意したいのは，文脈が言及しうる範囲として何を用いるのかによって取り扱える不整合の範囲も変化する点である．

本論文では，文脈を文脈条件式と呼ばれる条件式によって表現する．文脈条件式は，以下のように定義される．C_tを組存在条件式，C_lを選択条件式とするとき，$C_t \wedge C_l$は文脈条件式である．Kを表Rのキー属性，yを定数とするとき，$\sigma_{K=y}(R) \neq \phi$は組存在条件節である．組存在条件式は組存在条件節を節として構成される論理式である．選択条件式は属性集合とその値の組によって構成される論理式である．

注釈は文脈Cxを用いて以下のように定義することができる．

$$a = (R, T, A, Cx, Cont)$$

また，文脈を持つ注釈の一貫性は次のような制約を持つことで満たされる．

for all $t \in T, S = \sigma_{A=t.A \wedge Cx}(R)$,

$$\bigcup_{s \in S} s = T$$

注釈の一貫性を満足するため，Tは選択条件式$Cond$と文脈Cxによって，以下のように選ばれる．

$$T = \sigma_{Cond \wedge Cx}(R)$$

よって，以降では，注釈aが関連付けられる組集合Tは，$Cond, Cx$を用いて定められるものとし，

$$a = (R, A, Cond, Cx, Cont)$$

と表現することにする．

最後に，以下のような注釈写像関数と呼ばれる関数μを定義する．表Rに関連付けられている注釈の集合を$\mathcal{A}$とし，注釈$a_i = (R, A_i, Cond_i, Cx_i, Cont_i) \in \mathcal{A}$，任意の組集合$X \subseteq R$に対し，$X \subseteq \sigma_{Cond_i \wedge Cx_i}(R)$のとき，

$$\mu(R, X, A_i, Cx_i) = Cont_i$$

表Rに関連付けられている一つ一つの注釈は，注釈集合から表への写像であると見なすことができる．よって，表Rへ付与されている注釈集合そのものを，上式のような関数を用いて表現することができ，特定の表，組集合，属性集合，文脈を与えることによって，その内容を得ることができる．注釈は組集合を付加範囲として与えられているが，そこに含まれている任意の組の部分集合も同様の内容を持つものとして定義されている．任意の$R, X \subseteq R, A \subseteq att(R), Cx$対して，関連付けられている注釈が存在しない場合には，以下のようになる．

$$\mu(R, X, A, Cx) = \phi$$

［例 3.1］ 例として再び図 1 の映画データベースを考える．注釈a_1, a_2が付加されている属性集合に注目すると，両者は actorA,actorB という値のペアに対して言及されている．とこ

FilmTable

title	year	genre	leading_actor1	leading_actor2	director
film1	2000	Mystery	actorA	actorB	director1
film2	2001	Comedy	actorB	actorC	director2
film3	2003	Period film	actorA	actorB	director3
film4	2005	Mystery	actorA	actorB	director4
film5	2008	Mystery	actorA	actorB	director1

図 2　各注釈の文脈が参照する情報

ろで，各注釈は付加対象の値が同じであるにもかかわらず，異なる組集合に対して関連付けが行われている．これは，各注釈が文脈により異なる内容を持つためである．注釈a_1の内容は，単純に actorA と actorB という値の組み合わせに対して成り立つ，不偏的な性質を述べており，文脈に依存しない内容であるため，$Cx_1 = \phi$と表現される．対して注釈a_2に関しては，特定のジャンルにおける actorA と actorB の組み合わせに対して言及しており，文脈は$Cx_2 = (genre = ``Mystery'')$と表される．注釈$a_3$は，film1 の続編にあたる映画レコード（ここでは film4,film5)が言及対象となっており，なおかつ前作と同じ監督による作品という文脈によって付与が行われている．よって，文脈は$Cx_3 = (\sigma_{title=``film1''}(FilmTable) \neq \phi \wedge director = \sigma_{title=``film1''}(FilmTable).director)$となる．図 2 は，注釈が文脈として用いている情報を図示している．実線で囲われている影付き部分が注釈a_2の文脈の根拠にあたる．同様に，破線で囲われている影付き部分が注釈a_3の文脈の根拠となる．

注釈a_1, a_2, a_3はそれぞれ以下のように表わされる．

$a_1 = (FilmTable, \{leading\_actor1, leading\_actor2\}, (leading\_actor1 = ``actorA'') \wedge (leading\_actor2 = ``actorB''), \phi, Cont_1)$

$a_2 = (FilmTable, \{leading\_actor1, leading\_actor2\}, (leading\_actor1 = ``actorA'') \wedge (leading\_actor2 = ``actorB''), genre = ``Mystery'', Cont_2)$

$a_3 = (FilmTable, \{title\}, (title = ``film4'') \vee (title = ``film5''), ((\sigma_{tilte="film1"}(FilmTable) \neq \phi) \wedge (director = \sigma_{tilte="film1"}(FilmTable).director)), Cont_3)$

3.2 不整合の定義

本節では，前節での注釈の定義から，演算によって生じる注釈の不整合を定義する．注釈は，文脈によって参照されるデータを根拠として整合性を保っていると言える．また，ある組に関連付けられている注釈が整合性を保っているとは，文脈によって参照されているデータがその条件を満足することを言う．

［定義 3.1］(整合性)

注釈をa，付与対象の組集合をT，含まれる組を$t \in T$としたとき，$eval_R(t, Cx) = \text{true}$のとき，注釈$a$は表$R$における組$t$に対して整合性を持つという．ここで，関数$eval_R(t, Cx)$は，文脈$Cx$が表$R$において，組$t$を評価した結果が真の場合にのみ true を返す関数である．$Cx = \phi$であるときは必ず真となる．また，処理によって，評価不能となった属性を含む部分の条件は偽になるものとする．

我々の定義では，表Rへ注釈を生成する際，付与対象である組集合Tを$T = \sigma_{Cond \wedge Cx}(R)$となるように選ぶため，注釈が付与されている組は必ず整合性を持つ．よって，不整合は必ず何らかの処理を実行することによって起こるものであり，それは，文脈によって参照されている情報が表から失われることによって，注釈のもつ文脈が満足されなくなることと定義することができる．

LostData

title	year	genre
film1	2000	Mystery
film2	2001	Comedy
film3	2003	Period film
film4	2005	Mystery
film5	2008	Mystery

π_B(FilmTable)

leading_actor1	leading_actor2	director
actorA	actorB	director1
actorB	actorC	director2
actorA	actorB	director3
actorA	actorB	director4

図 3　射影適応後

LostData

title	year	genre	leading_actor1	leading_actor2	director
film1	2000	Mystery	actorA	actorB	director1
film2	2001	Comedy	actorB	actorC	director2

σ_c(FilmTable)

title	year	genre	leading_actor1	leading_actor2	director
film3	2003	Period film	actorA	actorB	director3
film4	2005	Mystery	actorA	actorB	director4
film5	2008	Mystery	actorA	actorB	director1

図 4　選択適応後

［定義 3.2］(演算によって生じる不整合)

注釈を$a = (R, A, Cond, Cx, Cont)$，$a$の付与対象である組を$t \in \sigma_{Cond \wedge Cx}(R)$とし，表$R$への処理$Op$が行われた時，$eval_{Op(R)}(Op(t), Cx) = \text{false}$ならば，注釈$a$は処理後の表$Op(R)$における組$Op(t)$に対して不整合であるという．

［例 3.2］図 1 の映画データベースに対し，属性$leading_actor_1$,$leading\_actor_2$,$director$(以下Bとする)を射影するという操作($\pi_B(FilmTable)$)を考える．演算を適応した結果は図 3 のようになる．a_1は文脈自由，つまり$Cx_1 = \phi$であるため，全ての$t \in \pi_B(FilmTable)$に対して，$eval_{\pi_B(FilmTable)}(t, Cx_1) = \text{true}$となる．よって，不整合は生じない．注釈$a_2$はジャンルがミステリーという文脈があって初めて意味を持つ内容であり，$Cx_2 = (genre = ``Mystery'')$と表わされていた．しかし，文脈に用いられていた属性である$genre$が射影演算によって失われてしまうため，例えば，一番目の組t_1に注目すると，$eval_{\pi_B(FilmTable)}(t_1, Cx_2) = \text{false}$となり，不整合が生じる．次に，表$FilmTable$に対し，2003 年以降のデータを選択するという処理($\sigma_{year \geq 2003}(FilmTable)$)を考える．演算を適応した結果は図 4 のようになる．注釈a_3は，前作である映画のレコードと，そのレコードが持つ属性を文脈として参照しており，$Cx_3 = (\sigma_{title=``film1''}(FilmTable) \neq \phi \wedge director = \sigma_{title=``film1''}(FilmTable).director)$と表わされた．選択演算によって参照されていた組$t_1$と，それが持つ$director$の属性値が失われるため，$Cx_3$の$(\sigma_{title=``film1''}(FilmTable) \neq \phi$という条件が偽となり，処理後の組$t'_5$に対して，$eval_{\sigma_{year \geq 2003}(FilmTable)}(t'_5, Cx_3) = \text{false}$となり，不整合が生じる．

4. 整合性の確保

前節で注釈の不整合は，演算適応の結果，文脈が満足されなくなることによって生じると定義した．本節では，データの変遷によって生じるそれらの不整合に対し，整合性を確保した注釈伝播を行うための手法を提案する．

整合性確保のための基本的な考え方は次のとおりである．不整合は演算や更新などのデータ処理による情報の欠落によって生じる．そこで，文脈を満たすための根拠であった情

報を，表の代わりに注釈が保持することによって，不整合を防ぐ．つまり，ある文脈によって関連付けられている注釈が存在し，その付加対象である表への操作によって不整合が生じるとき，注釈の保持する内容に，注釈が満たすべき文脈が参照していた情報を加えるという操作を行う．そうすることによって，言及対象となる値の集合が，ある文脈が成り立つ時に限り，特定の内容を持つという，もともとの注釈が持っていた意味を維持することができる．情報を注釈へ保持することに伴い，注釈の付加対象となる範囲も変化する．伝播後に文脈を判断するための情報が欠落してしまう注釈は，欠落してしまう文脈を考えない場合に付きうるすべての範囲を対象として付与される．その中で，注釈に保持された文脈情報を追加の情報として用いることで，整合性が保たれる．以下，再び例 1 の操作を例にとって整合性確保の概要を説明する．

第 3 節で定義したように，注釈の文脈は論理式によって構成されており，葉に“属性＝値”に持ち，接点に and，or を持つ木で表現できる．文脈によって選択される組集合はさらに，木の部分木に対応する組集合に分類され，処理によって満足されなくなる文脈は，部分木の一部に対応する組集合である．処理後も文脈を満たす組集合に関しては，そのまま付与されていた注釈を伝播させればよいが，処理後に文脈を満たさなくなるような組集合や，文脈を仮定すれば付加の対象となるような組集合に対しては，文脈として用いられていた条件を注釈に加えることで，整合性の確保を行う．

整合性を確保するための操作を実現するため，本論文では基本的な六つの関係演算(射影，選択，直積，和，差，属性名変更)に対して拡張を行う．

4.1 文脈の移管

演算拡張の為の準備として以下のような文脈移管関数と呼ばれる，注釈内容が整合性を保つために必要な条件を注釈内容に移管する関数を定義する．

$$\pi_X^A(Cx, Cont) = Cont_{ext}$$

この関数は，入力としてある注釈の文脈Cx，属性集合$X \subseteq att(R)$，注釈内容$Cont$を受け取り，新たな注釈内容$Cont_{ext}$を出力する．ここで$Cont$の定義を以下のように拡張する．

$$Cont_{ext} = (Comment, Cx_{as})$$

注釈内容は，本文となる$Comment$と，内容が成立するための仮定Cx_{as}から構成される．Cx_{as}とは，直感的にはデータそのものから消えてしまった根拠を，内容が成立するための条件として補う情報である．Cx_{as}は注釈生成時には$Cx_{as} = \phi$として与えられる．Cx_{as}は，入力として与えられた属性集合Xに含まれない属性を持つ葉を，文脈木から取り除いた木で表現される条件式である．注釈内容として$Cont_{ext}$を持つ注釈が関連付けられている全ての組は文脈を評価される際，自身が持つ値に加え，Cx_{as}を真にする属性と値の組み合わせを保持するものとして扱われる．つまり，文脈を評価される組は仮想的にCx_{as}に含まれる属性集合を持つものとして拡張される．それによって文脈の評価結果は真となり，整合性の定義に違反することなく注釈の関連付けが行われる．拡張後の組をt_{ext}とし，組tが持つ属性を$\{A_1, A_2, \dots, A_k\}$，Cx_{as}に含まれる属性集合を$\{A_1^{as}, A_2^{as}, \dots, A_l^{as}\}$とする．また，$Cx_{as}$の葉を全て真としたとき，根を真とするために必要なある部分木に含まれる属性集合を$\{A_{m_1}^{as}, A_{m_2}^{as}, \dots, A_{m_n}^{as}\}$，各属性の値を$\{v_{m_1}^{as}, v_{m_2}^{as}, \dots, v_{m_n}^{as}\}$とすると，$t_{ext}$は以下のように定義される．

Algorithm 1 Extension of Project Operator：$\pi_B(R, \mathcal{A})$

Input: $R, \mathcal{A}$
Output: $R', \mathcal{A}'$

```
1: R' ← π_B(R)
2: A' ← ∅
3: for all t' ∈ R' do
4:     for all a = (R, A, Cond, C_x, Cont) ∈ A do
5:         if eval_R'(t', Cond ∧ C_s) = true then
6:             A' ← A' ∪ v(R', A ∩ B, Cond, C_x, Cont)
7:         else if eval_R'(t_ext', Cond ∧ C_s) = true then
8:             A' ← A' ∪ v(R', A ∩ B, Cond, C_x, π_Y^A(C_x, Cont))
               (Y = att(C_x) \ B ∩ att(C_x)
9:         end if
10:    end for
11: end for
```

$$att(t_{ext}) = \{A_1, A_2, \dots, A_k\} \cup \{A_{m_1}^{as}, A_{m_2}^{as}, \dots, A_{m_n}^{as}\}$$

$$t_{ext}.A = \begin{cases} t.A_i & \text{if } A = A_i (1 \le i \le k); \\ v_{m_i}^{as} & \text{else if } A = A_{m_i}^{as} (1 \le i \le n). \end{cases}$$

また，同様にして，表に対して仮定される組集合を注釈へ射影する関数を定義する．

$$\pi_R^t(U, Cont) = Cont_{ext}$$

この関数は，注釈内容$Cont$の仮定Cx_{as}に対し，表Rに組集合Uが持つ属性集合とその値を射影する関数であり，以下のように条件が追加される．

$$Cx_{as} = Cx_{as} \cap (U \subset R)$$

注釈内容として$Cont_{ext}$を持つ注釈が関連付けられている表は，文脈を評価される際，自身が持つ組集合に加え，$\boldsymbol{Cx_{as}}$に含まれる組集合を持つものとして扱われる．射影される組集合Uを用いて，拡張後の表R_{ext}は以下のように定義される．

$$R_{ext} = R \cup U$$

文脈の評価は，まず選択条件式によって関連付けすべき値を持つ組集合が選択され，それらを仮定によって拡張したのちに行われる．また，Cx_{as}は処理の過程で追加されていくものとし，文脈射影によって条件が増加する場合，それらは and 接点で結ばれる．以下，各演算に対する拡張について定義する．

4.2 演算の拡張

本節では，整合性の確保を行うための関係演算の拡張について述べる．拡張後の各関数は，入力として表と，注釈集合の組を受け取り，新たな表と注釈集合を生成し出力する．出力される各注釈は，表名，属性集合，選択条件式，文脈，内容から注釈を構成する関数であるνによって生成される．以下，射影，選択，直積，和，差，属性名変更の拡張について述べる．

・**射影**($\pi_B(R, \mathcal{A})$)： 入力として，表Rと注釈の集合$\mathcal{A}$をとり，表R'と注釈の集合$\mathcal{A}'$を出力する．伝播はアルゴリズム 1 のような手順で行われる．処理後の各組t'について，注釈の持つ文脈が満足されるかを評価し，不整合が生じない場合は，入力の注釈から同様の文脈と内容を持つ注釈を生成する．文脈が満足されないとき，つまり不整合となる場合には，文脈に用いられている属性のうち，出力に射影されなかった属性集合とその値を内容に射影する．その後，組の拡張を行い，再び文脈を評価する．評価結果が真の場合は拡張後の内容を持つ注釈を新たに生成し，出力される注釈の集合へ加える．拡張後も文脈が満たされない場合は，注釈を生成しない．

π_B(FilmTable)

leading_actor1	leading_actor2	director	genre
actorA	actorB	director1	Mystery
actorB	actorC	director2	
actorA	actorB	director3	Mystery
actorA	actorB	director4	Mystery

図 5　射影適応後の注釈

Algorithm 2 Extension of Select Operator : $\sigma_C(R,\mathcal{A})$

Input: $R,\mathcal{A}$
Output: $R',\mathcal{A}'$
1: $R' \leftarrow \sigma_C(R)$
2: $\mathcal{A}' \leftarrow \emptyset$
3: **for all** $t' \in R'$ **do**
4: **for all** $a = (R, A, Cond, C_x, Cont) \in \mathcal{A}$ **do**
5: **if** $eval_{R'}(t', Cond \wedge C_x) = \text{true}$ **then**
6: $\mathcal{A}' \leftarrow \mathcal{A}' \cup v(R', A \cap B, Cond, C_x, Cont)$
7: **else if** $eval_{R'_{ext}}(t', Cond \wedge C_x) = \text{true}$ **then**
8: $\mathcal{A}' \leftarrow \mathcal{A}' \cup v(R', A \cap B, Cond, C_x, \pi^t_{R'}(T_r, Cont))$
$(T_r = \{t_r | t_r \epsilon R, eval_{R-t_r}(t, C_t) = \text{false for } t \epsilon R\})$
9: **end if**
10: **end for**
11: **end for**

［例　4.1］(射影適応後注釈) 表 $FilmTable$ に $\pi_{leading_actor1,leadin_actor2,director}(FilmTable)$という射影演算を適応すると，図 5 のように注釈の伝播が行われる．図 5 の影付き部分が伝播後の注釈a_1，実線で囲われた部分が注釈a_2を表わしている．また，t_1, t_3, t_4の右側に破線で示されている部分は，注釈に移管された条件によって拡張された内容を表わしており，注釈の文脈を評価する際は仮想的に$genre$という属性が存在し，その値は$Mystery$であるものとして扱われる．実際のデータは不整
合であるが，移管された内容によって整合性の定義には違反しない．注釈a_3に関しては，付加対象となっていた属性が結果の表れないため，伝播は行われない．入力の注釈集合$\mathcal{A} = \{a_1, a_2\}$を伝播させた結果を$\mathcal{A}' = \{a'_1, a'_2\}$とすると，
$a'_1 = (\pi_B(FilmTable), \{leading\_actor1, leading\_actor2\}, (leading\_actor1 = ``actorA") \wedge (leading\_actor2 = ``actorB"), \phi, (Comment1, \phi))$
$a'_2 = (\pi_B(FilmTable), \{leading\_actor1, leading\_actor2\}, (leading\_actor1 = ``actorA") \wedge (leading\_actor2 = ``actorB"), genre = ``Mystery", (Comment2, genre = = ``Mystery"))$
となる．それぞれの注釈は表$\pi_B(R)$の$\{t_1, t_3, t_4\}$に関連付けられ，表$\pi_B(R)$における，注釈a'_1の解釈は表Rにおける注釈a_1の解釈と同様である．注釈a'_2の解釈は，“actorA と actorB という値の組み合わせは，それらが属するレコードが$genre$という属性を持ち，なおかつその値が Mystery ならば$Comment2$に記述される性質を持つ”と表現することができる．

・**選択**($\sigma_C(R,\mathcal{A})$)： 入力として，表Rと注釈の集合$\mathcal{A}$をとり，表R'と注釈の集合$\mathcal{A}'$を出力する．伝播はアルゴリズム 2 のような手順で行われる．手順中の組t_rは，文脈Cxによって，存在が参照されている組である．また，C_tは組存在条件式である．処理後の各組t'について，注釈の持つ文脈が満足されるかを評価し，不整合が生じない場合は，入力の注釈から同様の文脈と内容を持つ注釈を生成する．注釈に参照されている組が選択演算によって選択されなかったとき，文脈の評価結果は偽となり不整合が生じる．その場合は，注釈内容，表，

σ_C(FilmTable)

title	year	genre	leading_actor1	leading_actor2	director
film3	2003	Period film	actorA	actorB	director3
film4	2005	Mystery	actorA	actorB	director4
film5	2008	Mystery	actorA	actorB	director1
film1	2000	Mystery	actorA	actorB	director1

図 6　選択適応後の注釈

組に対して拡張を行い，再び文脈の評価行う．評価結果が真の場合は拡張後の内容を持つ注釈を新たに生成し，注釈の集合へ加える．拡張後も文脈が満たされない場合は，注釈を生成しない．

［例　4.2］(選択適応後注釈) 表 $FilmTable$ に $\sigma_{year\geq 2003}(FilmTable)$という選択演算を適応すると，図 6 のように注釈が伝播される．図 6 の影付き部分が伝播後の注釈a'_1，実線で囲われた部分が注釈a'_2，破線で囲われた部分が注釈a'_3を表わしている．また，表下部の破線で表わされた組は，注釈に移管された内容による表の拡張を
表わしている．入力の注釈集合$\mathcal{A} = \{a_1, a_2, a_3\}$を伝播させた結果を$\mathcal{A}' = \{a'_1, a'_2, a'_3\}$とすると，
$a'_1 = (\sigma_{year\geq 2003}(FilmTable), \{leading\_actor1, leading\_actor2\}, (leading\_actor1 = ``actorA") \wedge (leading\_actor2 = ``actorB"), \phi, (Comment1, \phi))$
$a'_2 = (\sigma_{year\geq 2003}(FilmTable), \{leading\_actor1, leading\_actor2\}, (leading\_actor1 = ``actorA") \wedge (leading\_actor2 = ``actorB"), genre = ``Mystery", (Comment2, \phi))$
$a'_3 = (\sigma_{year\geq 2003}(FilmTable), title, (title = ``film4") \wedge (title = ``film5"), (\sigma_{title=``film1"}(FilmTable) \neq \phi \wedge director = \sigma_{title=``film1"}(FilmTable).director), (Comment3, t_1 \in \sigma_{year\geq 2003}(FilmTable)))$
となる．注釈a'_1, a'_2, a'_3はそれぞれ，表$\sigma_{year\leq 2003}(FilmTable)$の$\{t_1, t_2, t_3\}, \{t_2, t_3\}, \{t_3\}$という組集合に関連付けられる．表$\sigma_{year\leq 2003}(FilmTable)$における，注釈$a'_1$と注釈$a'_2$の解釈は表$R$における注釈$a_1, a_2$の解釈と同様である．注釈$a'_3$の解釈は，“film4，film5 という映画は，データベースに存在する film1 という映画と同じ$director$の値を持つとき，$Comment3$に記述される性質を持つ”と表現することができる．

・**直積**($(R,\mathcal{A}_r) \times (S,\mathcal{A}_s)$)： 入力として，表$R,S$と注釈の集合$\mathcal{A}_r, \mathcal{A}_s$をとり，表$R'$と注釈の集合$A'$を出力する．表$R$に関連付けられている注釈集合$\mathcal{A}_r$と，表$S$に関連付けられている注釈集合$\mathcal{A}_s$は，付加範囲と文脈において，ともに独立であるため，それぞれの注釈集合から，単一の注釈集合を生成することによって伝播が可能である．

・**和**($(R,\mathcal{A}_r) \cup (S,\mathcal{A}_s)$)： 入力として，表$R,S$と注釈の集合$\mathcal{A}_r, \mathcal{A}_s$をとり，表$R'$と注釈の集合$\mathcal{A}'$を出力する．$\mathcal{A}_r$と$\mathcal{A}_s$に同一の付加範囲と，文脈を持つ注釈が存在する場合，その内容を統合した注釈を生成する．その他の場合は，各注釈集合に含まれる集合は独立であるため，入力と同様の条件を持つ注釈を生成し，伝播を行う．

・**差**($(R,\mathcal{A}) - S$)： 入力として，表R,Sと注釈の集合$\mathcal{A}$をとり，表R'と注釈の集合$\mathcal{A}'$を出力する．差演算については，注釈の操作を選択演算と同様に定義することができ，伝播はアルゴリズム 3 のような手順で行われる．手順中の組t_rは，文脈Cxによって，存在が参照されている組である．また，C_tは組存在条件式である．

・**属性名変更**($\delta_\theta(R,\mathcal{A})$)： 入力として，表$R$と注釈の集合$\mathcal{A}$をとり，表$R'$と注釈の集合$\mathcal{A}'$を出力する．なお，$\theta$は属性集合を入力として受け取り，属性名の変換を行う関数である．伝播はアルゴリズム 4 のような手順で行われる．属性名変更

Algorithm 3 Extension of Difference Operator : $(R, \mathcal{A}) - S$

Input: $R, S, \mathcal{A}$
Output: $R', \mathcal{A}'$
1: $R' \leftarrow R - S$
2: $\mathcal{A}' \leftarrow \emptyset$
3: **for all** $t' \in R'$ **do**
4: **for all** $a = (R, A, Cond, C_x, Cont) \in \mathcal{A}$ **do**
5: **if** $eval_{R'}(t', Cond \wedge C_x) = \text{true}$ **then**
6: $\mathcal{A}' \leftarrow \mathcal{A}' \cup \upsilon(R', A \cap B, Cond, C_x, Cont)$
7: **else if** $eval_{R'_{ext}}(t', Cond \wedge C_x) = \text{true}$ **then**
8: $\mathcal{A}' \leftarrow \mathcal{A}' \cup \upsilon(R', A \cap B, Cond, C_x, \pi^t_{R'}(T_r, Cont))$
$(T_r = \{t_r | t_r \epsilon R, eval_{R-t_r}(t, C_t) = \text{false for } t \epsilon R\})$
9: **end if**
10: **end for**
11: **end for**

Algorithm 4 Extension of Rename Operator : $\delta_\theta(R, \mathcal{A})$

Input: $R, \mathcal{A}$
Output: $R', \mathcal{A}'$
1: $R' \leftarrow \delta_\theta(R)$
2: $\mathcal{A}' \leftarrow \emptyset$
3: **for all** $a = (R, A, Cond, C_x, Cont) \in \mathcal{A}$ **do**
4: $\mathcal{A}' \leftarrow \mathcal{A}' \cup \upsilon(R', \theta(A), \theta(Cond), \theta(C_x), \theta(Cont))$
5: **end for**

Length

title	length
film2	63
film4	120
film5	115
film6	78

図 7 上映時間

に関しては，入力の注釈が持つ，付加範囲，文脈，内容に含まれる属性名に対し，変更を加え新たな注釈を生成する．θが注釈内容に適応された場合，$Cont$が持つCx_{as}に対して変更が行われる．

［例 4.3］(自然結合)拡張演算を組み合わせた処理として自然結合を考える．結合する表として，図 7 のような表$Length$を考える．表$Length$には図 7 の太線で囲われた部分に注釈a_4が関連付けられており，コメントとして，“非常に長く感じた”という感想が付与されているものとする．これは，ある映画の上映時間について述べられた内容であるため，注釈a_4の文脈であるCx_4は，$Cx_4 = (title = ``film4")$となる．表$FilmTable$と表$Length$との自然結合は以下のように表現される．

$$\sigma_{title=title'}(FilmTable \times \delta_\theta(Length))$$

なお，上式のθは属性$title$の名称を$title'$へ変更する関数とする．図 8 の影付き部分が伝播後の注釈a'_1，実線で囲われた部分が注釈a'_2，破線で囲われた部分が注釈a'_3，太線で囲われた部分が注釈a'_4を表わしている．処理後の表をSとするとし，入力の注釈集合を伝播させた結果を$\mathcal{A}' = \{a'_1, a'_2, a'_3, a'_4\}$とすると，

$a'_1 = (S, \{leading\_actor1, leading\_actor2\}, (leading\_actor1 = ``actorA'') \wedge (leading\_actor2 = ``actorB''), \phi, (Comment1, \phi))$
$a'_2 = (S, \{leading\_actor1, leading\_actor2\}, (leading\_actor1 = ``actorA") \wedge (leading\_actor2 = ``actorB"), genre = ``Mystery'', (Comment2, \phi))$
$a'_3 = (S, title, (title = ``film4") \wedge (title = ``film5"),$

$\sigma_{title=``film1"}(S) \neq \phi \wedge director = \sigma_{title=``film1"}(S).director)$,
$(Comment3, t_1 \in S))$
$a'_4 = (S, length, length = 120, title' = ``film4",$
$(Comment4, \phi))$

となる．伝播はまず属性名変更によって属性$title$の名称と，注釈内で用いられている属性名が変更される．次に，直積によって表と，注釈集合が統合され，最後に，選択演算によって同じ題名を持つ映画レコードが選択される．このとき，選択演算の例と同様に，$t_1 \in FilmTable$が失われるため，その情報は注釈内容に移管される．上記のように，複数の演算による処理が行われた場合も，伝播が正しく行われる．

$\sigma_{title=title'}$ (FilmTable×δ_θ(Length))

title	year	genre	leading_actor1	leading_actor2	director	title'	length
film2	2001	Comedy	actorB	actorC	director2	film2	63
film4	2005	Mystery	actorA	actorB	director4	film4	120
film5	2008	Mystery	actorA	actorB	director1	film5	115
film1	2000	Mystery	actorA	actorB	director1	...	...

図 8 結合後の注釈

5. まとめ

本論文では，データ変遷によって起こる注釈の品質劣化の問題を解決するため，注釈の意味を，文脈という概念を導入することによって表現する注釈モデルを提案した．注釈の不整合を文脈の有効性という側面から定式化し，整合性が失われるような処理に対しては，情報を注釈へ移管するというアプローチによって問題の解決を議論した．また，各関係演算の拡張を行い，整合性を確保した注釈伝播を可能とした．本論文では，文脈は明示的に与えられる場合を想定しているが，付与対象から文脈推定も発展的な議論として存在する．注釈同士の参照を考慮した議論も今後の課題として挙げられる．

[謝辞]

本研究の一部は科研費（22700097）と，文部科学省委託業務研究費国家基幹技術「データ統合・解析システム」の助成を受けたものであり，ここに記して謝意を表します．

[文献]

[1] P. Buneman, J. Cheney, W. C. Tan and S. Vansummeren: “Curated databases”, PODS, pp. 1–12 (2008).
[2] D. Srivastava and Y. Velegrakis: “Intensional associations between data and metadata”, SIGMOD, pp. 401–412 (2007).
[3] F. Geerts, A. Kementsietsidis and D. Milano: “Mondrian: Annotating and querying databases through colors and blocks”, ICDE, p. 82 (2006).
[4] P. Buneman, S. Khanna and W. C. Tan: “On propagation of deletions and annotations through views”, PODS, pp.150–158 (2002).
[5] P. Buneman, S. Khanna and W. C. Tan: “Why and where: A characterization of data provenance”, ICDT, Vol. 1973 of Lecture Notes in Computer Science, pp. 316–330 (2001).

青戸 了 Ryo AOTO
京都大学大学院情報学研究科修士課程在学中．日本データベース学会学生会員．
清水 敏之 Toshiyuki SHIMIZU
京都大学大学院情報学研究科特定助教. 2008 年京都大学情報学研究科博士後期課程修了．情報学博士．ACM，日本データベース学会各会員．
吉川 正俊 Masatoshi YOSHIKAWA
京都大学大学院情報学研究科教授．1985 年京都大学大学院工学研究科博士後期課程修了．工学博士．電子情報通信学会，ACM，IEEE Computer Society 各会員．日本データベース学会理事．