

Contract Expression Language (CEL) – An UN/CEFACT BCF Compliant Technology

January 21, 2004

CONTENTS

INTRODUCTION	3
OVERVIEW OF CEL	5
Functions	5
Data Model	5
Simple Example: Goods for Sale	7
CEL COMPLIANCE TO BCF	9
Introduction	9
Mapping of Data Models	9
Ontology Alignment	10
Modeling BCF Example Contract	12
Modeling BCF Example Business Collaboration	14
Modeling BCF Business Transaction Patterns	15
Interaction with BCF views	17
CONCLUSION	19
For More Information	19
References	19

INTRODUCTION

This document introduces the Contract Expression Language (CEL) [1], an XML-based language to express contractual agreements between business entities, currently under development at the Content Reference Forum (CRF) [2]. Specifically, this document focuses on demonstrating the CEL compliance to the Business Collaboration Framework (BCF), developed by the Techniques and Methodologies Group (TMG) of the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) [3].

The CRF, www.crforum.org, is a consortium of companies, which adopts an open process to develop specifications that will be available as standards. The CRF is made up of hardware and software technology companies, content owners, and service providers. The goal of the CRF is to facilitate the distribution of legitimate content in peered environments with the least friction possible for all stakeholders in the value chain. The CRF intends to do this by enabling the dynamic delivery of the content with the right commercial terms for a given consumer and the proper value chain context.

One of the primary work products being developed at the CRF is the CEL specification and its associated XML schema and reference software. The CEL can be used to express the contractual terms between the participants of electronic content or service distribution value chains to govern the distribution and use of content. The CEL is precise and unambiguous. The syntax and semantics of its expressions are suited for both human and machine interpretations. The CEL is based on the same architecture model and it shares many basic elements with the MPEG21 Rights Expression Language (MPEG REL) [4], which is becoming an International ISO/IEC standard in early 2004.

UN/CEFACT supports activities dedicated to improving the ability of business, trade and administrative organizations from developed, developing and transitional economies to exchange products and relevant services effectively. Its principal focus is on facilitating international transactions through the simplification and harmonization of processes, procedures and information flows, and so on contributing to the growth of global commerce. In support of this vision, UN/CEFACT developed the BCF, www.unbcf.org, to enable business process and information models to be specified in a technology- and implementation-neutral manner that can then be implemented in software using the information exchange syntax and structures of choice.

Because of the common interests in the technologies to describe enforceable e-Business contractual agreements, CRF and UN/CEFACT TMG established a liaison relationship to exchange information and to collaborate on technology development. One of the first collaborative projects is for the technical experts from the two organizations to analyze the compliance of the CEL with the BCF. This document captures the

result of this analysis.

In summary, CEL is a compliant technology for the BCF within the scope of e-Contract expression and interpretation. Specifically, as a BCF project goes down the four-step process (Knowledge Transfer, Create the Business Model, Transform the Business Model, and Implement the Business Model), the CEL is one of the specific technologies that can be used in the last stage to apply the technology-specific production rules to the Business Collaboration Model in its Business Collaboration Schema Specification (BCSS) form.

OVERVIEW OF CEL

The CEL is a language that can be used to express contractual agreements in unambiguous, machine-readable form. It is designed to provide an extensible architectural framework as well as the baseline common and application-specific vocabularies for specifying contracts in many potential applications. The role of contracts written in the CEL is to describe the business agreements (generally contractual) between or among parties participating in the business value chain in a way that, when the contracts are specified, the interests and obligations of the value chain participants can be communicated, interpreted, and executed by machines.

Functions

The primary function of a contract in the CEL is to serve the following purposes:

- Evidence: communicate information conveyed within a contract that can be easily and unambiguously understood;
- Execution: facilitate permissive, obligatory or prohibitory performance within a contract in the appropriate context, integrated with the contracting parties' business processes. This includes determining whether one is allowed to exercise some right, or is required to fulfill some obligation or obey some prohibition; and
- Evaluation: check permissive, obligatory, or prohibitory performance by contracting parties.

The secondary function of a contract in the CEL is to:

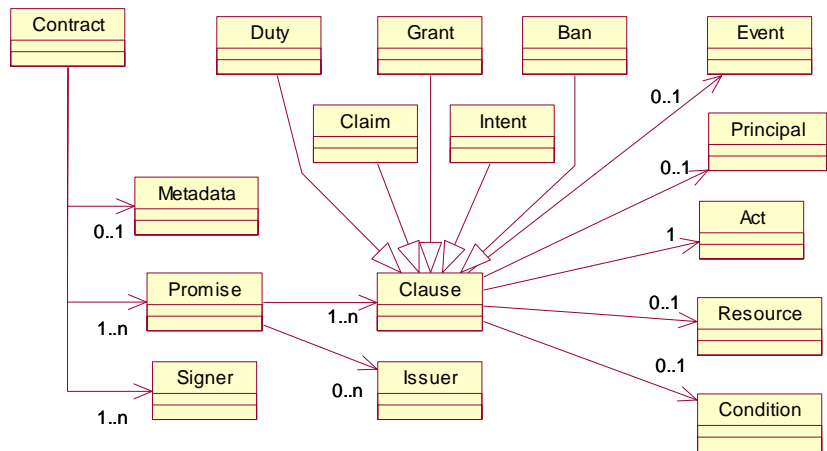
- Facilitate formation of contracts and automate the conventional process of offer, acceptance, agreement, and consideration;
- Facilitate dynamically generated and updated contracts based on, for instance, negotiation and future business models; and
- Facilitate enforcement of contracts with respect to their granted permissions, mandated obligations and stipulated prohibitions.

Data Model

The CEL uses the XML technology to define its syntax and semantics. The extensibility mechanisms provided by the XML Schema [12] enables the separation of the CEL core architectural elements and their relationship in the form of its data model from the details of its extended vocabularies in the forms of a variety of extensions to the core.

The main elements of the CEL data model are: Contract, Promise, Signer, Clause (which can be extended to specific instances like Grant, Duty, Ban, Intent, and Claim), Issuer, Event, Principal, Act, Resource and Condition. The following diagram illustrates these elements and their relationships. In

the diagram, the solid arrows represent a “has a” relationship between the parent element and its respective child elements, and the labels associated with the arrows represent the cardinality of the child elements.



The Contract element is the root element of a CEL contract expression. It contains one or more Promise elements and one or more Signers elements representing parties who pledge to abide by the statements made by the Promise elements in the Contract. Each Promise is further a conglomeration of Clauses and may contain one or more Issuers. Each of the Clauses may contain Event, Principal, Act, Resource, and Condition elements, but all these elements are optional except for Act. The general framework of CEL is based on this EPARC model of a Clause. More specifically, “EPARC” represents the following key components:

1. Event: an optional element representing a triggering event.
2. Principal: an optional element representing an entity or a set of entities that may perform the act specified in the peer element Act.
3. Act: a required element representing an act or a set of acts.
4. Resource: an optional element representing a resource or a set of resources to which the Act applies.
5. Condition: an optional element representing a condition, subject to which the Act may be performed.

Currently CEL has provided five specific types of Clauses that are syntactic and semantic extensions to the Clause: Grant, Duty, Ban, Intent and Claim. All of these extensions share the EPARC model of the Clause. Their semantics are as follows:

1. Grant: whenever the Event is triggered, permit the Principal the right to perform the specified Act over the Resource provided the Condition is met.
2. Duty: whenever the Event is triggered, demand an obligation on

the Principal to perform the Act on the Resource when the Condition is met.

3. Ban: whenever the Event is triggered, impose a prohibition on the Principal to perform the Act on the Resource under the Condition.
4. Intent: whenever the Event is triggered, express the Principal's intent to perform the Act on the Resource under the Condition.
5. Claim: whenever the Event is triggered, assert a fact that the Principal has performed the Act on the Resource when the Condition was met.

It is these specific Clauses that make individual contractual statements to convey permissive, obligatory, prohibitory, intentional, and assertive information, and that, when grouped into Promises made by their Issuers, comprise a CEL Contract expression agreed upon by its Signers.

Simple Example: Goods for Sale

A simple goods-for-sale example is stated as the following:

"Alice and Bob agree that Alice must sell her house to Bob if he pays her \$500,000."

This example uses the Event element "receivePayment", the Principal elements "keyHolder" for Alice and Bob, the Act element "sell" with the constraint "sellTo", and the Resource element "property". The CEL expression below (simplified with omissions of some element namespaces and details) specifies an agreement between Alice and Bob (signed on 11/11/2001) that imposes an obligation on Alice to sell the property to Bob when she receives \$500,000 from Bob.

```
<contract>
  <promise>
    <duty>
      <receivePayment>
        <paymentFlat>
          <rate currencyCode="USD">500000</rate>
        </paymentFlat>
        <payer>
          <keyHolder licensePartId="Bob"/>
        </payer>
        <payee>
          <keyHolder licensePartId="Alice"/>
        </payee>
      </receivePayment>
      <keyHolder licensePartId="Alice"/>
      <sell>
        <sellTo>
          <keyHolder licensePartId="Bob"/>
        </sellTo>
      </sell>
      <property licensePartId="AliceHouse">
        <location>
          <address>SOME ADDRESS</address>
        </location>
      </property>
    </duty>
  </promise>
</contract>
```

```
</property>
</duty>
</promise>
<signer licensePartId="Alice">
  <Signature/>
  <details>
    <timeOfIssue>2001-11-11T11:11:11</timeOfIssue>
  </details>
</signer>
<signer licensePartId="Bob">
  <Signature/>
  <details>
    <timeOfIssue>2001-11-11T11:11:11</timeOfIssue>
  </details>
</signer>
</contract>
```

Introduction

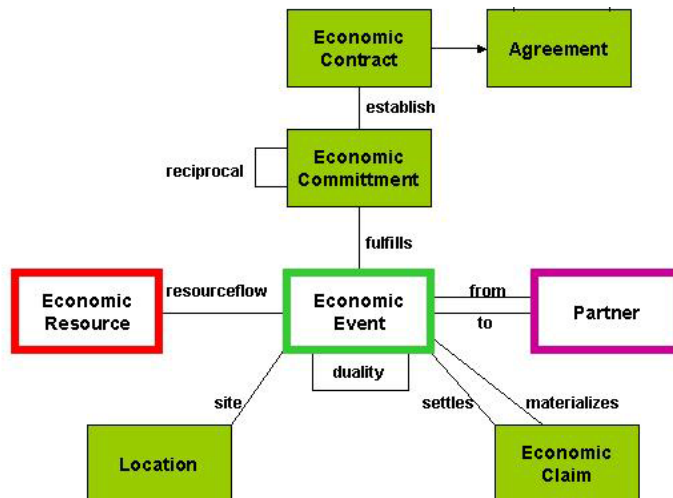
The BCF is a collection of specifications defining electronic business exchange for two or more business partners. The primary goal of the BCF is to capture the business knowledge that enables the development of low cost software components to help the small and medium size companies and emerging economies engage in e-Business practices.

This section presents a number of recommendations based on the gap analysis conducted [10, 11], and provides the following evidence to support that the CEL is a compliant technology to the BCF:

1. mapping the CEL contract data model to the agreement portion of the BCF meta model,
2. aligning the CEL core elements to the BCF transaction ontology,
3. illustrating how the CEL can be used to specify a BCF example contract,
4. demonstrating how the CEL can be used to model a BCF business collaboration,
5. showing how types of CEL expression templates can be designed to specify BCF business transaction patterns, and
6. explaining how CEL extensions and profiles can support different BCF views.

Mapping of Data Models

In the BCF, a commercial trading agreement is modeled, as shown in the following diagram, as part of the business collaboration model according to the UMM Meta Model [5].



This diagram shows that a successful business collaboration involves two dual types of Economic Events, each of which details the Economic

Resources involved in an exchange between two Partners. A Commitment is a promise by a Partner to initiate an Economic Event in the future. Performing the Economic Events fulfills that Commitment. Commitments should always be reciprocated by the other Partner who commits to initiate another type of Economic Event in return. An Economic Contract is a bundle of reciprocal commitments between Partners. A contract is a subtype of the more general object class called Agreement. The two other objects, Claims (for partially completely exchanges) and Locations (identifying places where Economic Events take place), are not essential to the model.

Based on the CEL data model, the following table lists a mapping between the CEL contract elements and the BCF contract elements drawn from [10, 11].

CEL Element		BCF Element		Comments
Contract		Agreement		A CEL Contract contains elements for specifying agreements involving permission, obligation, prohibition, intention and assertion.
Contract		Contract		
Promise				A Promise may specify a Business Collaboration
	Duty		Commitment	A Duty specifies a Commitment
	Event			An Event provides triggering a dependency
	Principal		Partner	
	Act		Event	An Event can be converted into Act
	Resource		Resource	
	Condition			

The duality of the reciprocal events can be specified in CEL with two mutually dependent Duty elements.

Ontology Alignment

The previous table reveals a good ontology alignment between the CEL and the BCF. It involves the BCF concepts of Contract, Business Collaboration, and Commitment.

CEL Contract and BCF Contract

In essence, these Contract elements are identical in semantics. Within the CEL, a Contract is defined to be a set of Promises (Business Collaborations in the BCF as explained below) represented by one or more Clauses of which each is associated with a specific Act that, when performed as a committed business activity, fulfills its obligation. A

Clause requires the elements associated with the Clause to include one required Act (Event in the BCF). Optionally, a Clause can include an Event, a Principal (Partner in the BCF), Resource (Resource in the BCF) and a Condition. In the BCF, the element that associates the Commitment to its related elements is the fulfillment association. The set of obligated, optional or prohibited activities are associated to a Commitment by the annotated Economic Event or Events (as associated by the fulfillment relationship) generated by one or more defined activities. Within the BCF, the declaration of an Event is the mechanism that associates one or more activities to a Commitment. It is the Event that defines the location and the occurrence/non-occurrence of a particular activity. Within the BCF, the two Partners that are associated to a Commitment are required. The BCF also enables for the association of a lifecycle (e.g., phases) to a contract and the aggregation of contracts or agreements within contracts, similar to the inclusion of Promises within the CEL Contract.

Both the BCF and the CEL define the elements of a contract, and both are in essence very similar as they represent common concepts found in contract law. Some of the variance between the two can be attributed in the approach taken to develop a set of semantics suitable for defining a contract. Where the CEL primarily focuses on the contract itself, the BCF focuses on the processes the contract supports. The link between the two is that execution of CEL Contracts results in BCF Business Transactions that form the BCF Business Collaboration.

CEL Promise and BCF Collaboration

The BCF is concerned with Business Collaborations according to Agreements among parties in the forms of Contracts, whereas the CEL focuses on specification of Contracts in such a way that it enables effective interpretation and execution of Contracts. Within a CEL Contract, a number of primitive agreements in the form of Clauses can be grouped together as a Promise. Therefore, CEL Promises can be used to specify a number of related Agreements used for the BCF Business Collaboration.

CEL Obligation and BCF Commitment

The BCF defines a Commitment to be an obligation to perform an action or activity (that is, transfer ownership of a specified quantity of a specified economic resource type) at some future point in time. The CEL element that expresses an obligation is Duty. Hence, the CEL Duty and the BCF Commitment are identical in semantics.

CEL Act and BCF Event

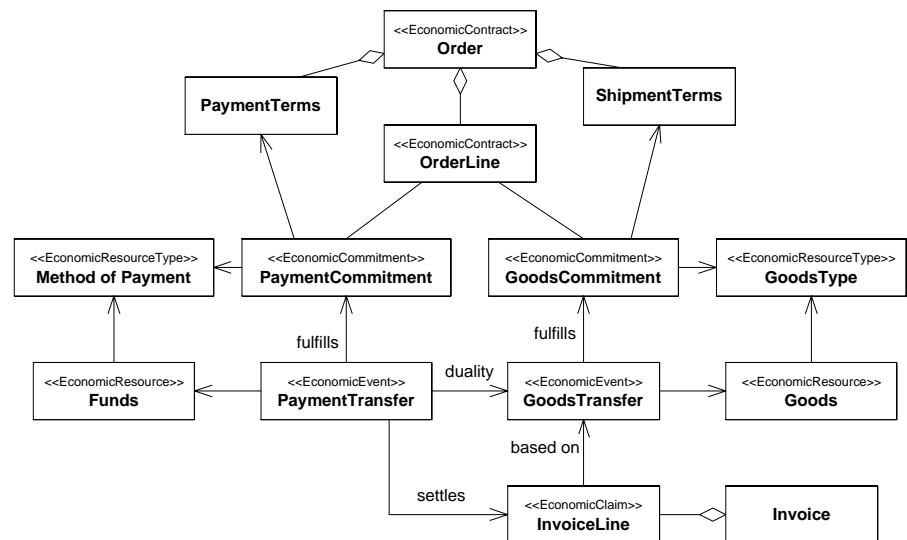
BCF Event specifies a process of changing states that, when it happens, fulfills a Commitment. The CEL Act identifies an act, whose

performance generates an Event. Thus, the CEL emphasizes an Act, while the BCF emphasizes an Event as the performance of an Act. In comparison to programming, specifying Act is like writing a program, while specifying an Event is like describing behavior of a program in terms of processes that are the execution instances of the program.

Within the CEL the Act is the primary element associated with the obligation as the mechanism of fulfillment. Within the BCF, the Business Event that annotates the occurrence of a particular activity is the primary element that affirms fulfillment. Associated with the element Act is a set of business rules/conditions that will necessarily be evaluated to determine when and if the Act has sufficiently completed in order to assert fulfillment. With the BCF it is the occurrence or non-occurrence (existence) of the Business Event that indicates sufficient completion of the activity.

Modeling BCF Example Contract

In [6], an example economic contract is given as follows:



In this diagram, there are two Economic Events, GoodsTransfer and PaymentTransfer, which represent transfers of ownership or control of Economic Resources (Goods and Funds) from one Partner to another (say, Buyer and Seller). The GoodsTransfer Event fulfills the GoodsCommitment, which is one of a pair of reciprocal commitments, and the PaymentTransfer Event fulfills the PaymentComittment, which is the reciprocal of the GoodsCommitment.

Based on the data model mapping, a CEL Contract can be formed as shown below that consists of a Promise that contains two “reciprocal” Duty Clauses. Since the Act of shipment of goods happens after receiving the payment, this Contract specifies the ship-after-pay scenario.

```

<contract>
  <promise>
    <duty>
      <buyer/>
      <pay>
        <methodOfPayment/>
        <paymentTerms/>
      </pay>
      <funds/>
    </duty>
    <duty>
      <paymentTransferredEvent/>
      <seller/>
      <ship>
        <shipmentTerms/>
      </ship>
      <goods/>
    </duty>
  </promise>
  <signer licensePartId="buyer"/>
  <signer licensePartId="seller"/>
</contract>

```

Another Contract may be formed to reflect the ship-before-pay scenario, as follows.

```

<contract>
  <promise>
    <duty>
      <goodsTransferredEvent/>
      <buyer/>
      <pay>
        <methodOfPayment/>
        <paymentTerms/>
      </pay>
      <funds/>
    </duty>
    <duty>
      <seller/>
      <ship>
        <shipmentTerms/>
      </ship>
      <goods/>
    </duty>
  </promise>
  <signer licensePartId="buyer"/>
  <signer licensePartId="seller"/>
</contract>

```

Yet a third Contract may be formed to reflect the simultaneous-ship-and-pay scenario.

```

<contract>
  <promise>
    <duty>
      <buyer/>
      <pay>
        <methodOfPayment/>
        <paymentTerms/>
      </pay>
      <funds/>

```

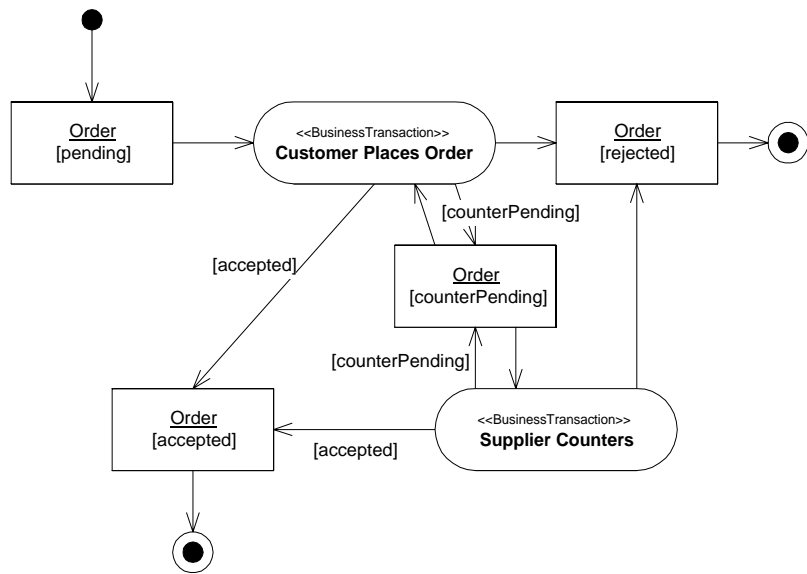
```

</duty>
<duty>
  <seller/>
  <ship>
    <shipmentTerms/>
  </ship>
  <goods/>
</duty>
</promise>
<signer licensePartId="buyer"/>
<signer licensePartId="seller"/>
</contract>

```

Modeling BCF Example Business Collaboration

In [6], an example of BCF business collaboration is given as follows:



Essentially, this diagram shows a collaboration activity between a customer and a supplier. The customer has the right to place an order. The Supplier has the obligation to respond to (counter offer, accept or reject) the Customer's offer.

Thus, a CEL Contract may be formed, as follows:

```

<contract>
  <promise>
    <grant>
      <customer/>
      <place/>
      <order/>
    </grant>
    <duty>
      <orderPlacedEvent/>
      <supplier/>
      <anyOf>
        <accept/>
        <reject/>
      </anyOf>
    </duty>
  </promise>
</contract>

```

```
<counter/>
</anyOf>
<placedOrder/>
</duty>
</promise>
<signer licensePartId="customer"/>
<signer licensePartId="supplier"/>
</contract>
```

Modeling BCF Business Transaction Patterns

To validate the compliance of the CEL to the BCF, showing how to model the BCF business transaction patterns [6, 7] in CEL contract expressions is necessary. A business contract is used mainly to prescribe business transactions that may, will and/or will not occur. Business transactions may be considered as resulting from executing business contracts. Therefore, a business transaction pattern may be prescribed by a type of business contract template. Demonstrating in CEL these types of contract templates whose execution result in the BCF transaction patterns provides a very convincing evidence of the expressiveness of the CEL and its applicability and compliance to the BCF.

The currently predefined BCF business transaction patterns in [5, 6, 7] are: Business Transaction, Query/Response, Request/Response, Request/Confirm, Information Distribution, and Notification. Each of these transaction patterns is modeled in [11] using a CEL expression. For instance, for the "Business Transaction" pattern which models the "offer and acceptance" transaction process that results in a residual obligation between both parties to fulfill the terms of the contract, the following analysis can be made according to the EPARC model of the CEL Clause:

1. There are Principals (for parties or partners) of two types: a **requester** and a **responder**. In general, the requester provides some information to the responder. In many cases, the responder needs to respond with some information.
2. The requesting information that the requester presents to the responder is an **offer**, which, when accepted, will result in a residual obligation between both parties to fulfill the terms of the contract.
3. The responding information (when necessary) that the responder presents to the requester is an **acceptance**, which when approved by the requester, will result in a residual obligation between both parties to fulfill the terms of the contract.
4. The requesting acts that the requester performs are to **offer** the offer and counter-counter-offer, and to **fulfill** the residual contract.
5. The responding acts that the responder performs are to **signal** the reception of the offer, to **signal** the acceptance of the offer (and counter-counter-offer), and to **fulfill** the residual contract.
6. The conditions for performing the acts are:

- a. Signaling the acceptance of the offer (and counter-counter-offer): **within 2 hours** and **in a non-repudiable way**
- b. Signaling the acceptance of the offer: **within 6 hours** and **in a non-repudiable way**
- c. Fulfillment of the residual contract (when accepted): **within 24 hours**

Based on the above analysis, the following type of CEL templates can be used in correspondence to the BCF business transaction pattern. Here, the customer may place an order, and must fulfill the commitments when the order is accepted by the supplier (e.g., transferring payment) within 24 hours. Upon receiving the order from the customer, the supplier must do all the following in sequence: signal the reception within 2 hours and in a non-repudiable way, signal the acceptance within 6 hours and in a non-repudiable way, and fulfill the accepted order within 24 hours.

```

<contract>
  <promise>
    <grant>
      <customer/>
      <place/>
      <order/>
    </grant>
    <duty>
      <acceptanceSignedEvent/>
      <customer/>
      <pay>
        <methodOfPayment/>
        <paymentTerms/>
      </pay>
      <funds/>
      <within24Hours/>
    </duty>
    <issuer licensePartId="supplier"/>
  </promise>
  <promise>
    <duty>
      <orderPlacedEvent/>
      <supplier/>
      <signalNonRepudiable/>
      <receptionOfOrder/>
      <within2Hours/>
    </duty>
    <duty>
      <receptionSignedEvent/>
      <supplier/>
      <signNonRepudiable/>
      <acceptanceOfOrder/>
      <within6Hours/>
    </duty>
    <duty>
      <acceptanceSignedEvent/>
      <customer/>
      <fulfill/>
      <acceptedOrder/>
      <within24Hours/>
    </duty>
    <issuer licensePartId="supplier"/>
  </promise>

```

```
<signer licensePartId="customer"/>
<signer licensePartId="supplier"/>
</contract>
```

These BCF business transaction patterns are used in [7] to “implement” design patterns involving two or more parties (such as service-service, agent-service-service, etc.), and correspondingly the types of CEL templates outlined here can be used to specify CEL templates involving two or more parties in the similar “composed” manner to develop compound CEL expressions.

Interaction with BCF views

By organizing CEL core elements and extended elements (with respect to Event, Principal, Act, Resource, and Condition in particular) into proper extensions and profiles, the CEL can support all the BCF views at different levels from implementation (IFV), to service (BSV), transaction (BTV), requirement (BRV), and domain (BDV).

For instance, an extension to the CEL can be defined to support the BTV for the business requirements within the commercial transaction domain, as follows. It involves extensions to the CEL core elements Event, Principal, Act, and Resource.

onPaymentDueEvent

This triggering Event indicates that the enclosing Clause becomes effective only if the monetary payment due Event occurs. This Event is evaluated to be true only when the payment is due.

agent

This Principal represents a business agent that participates in the business collaboration. The agent may contain a textual child element called agent-name to specify the name of the agent.



transfer

This Act represents the transfer of the Resource by the Principal.

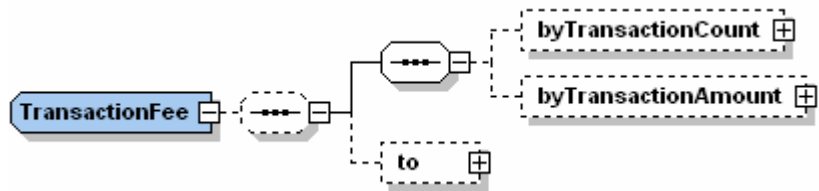
pay

This Act represents an act of monetary payment by the Principal. It may be used in conjunction with the transactionFee Resource.

transactionFee

This Resource element represents a monetary fee. It can be used with the Act pay to express that the Principal pays the monetary sum. The element contains information about the fee amount based on transactions by count or by amount, and the bank account to which

the monetary payment must be made.



The fee amount can be calculated either by using the byTransactionCount along with the rate, or byTransactionAmount, which points to the service that will return the total amount and a percentage element that indicates the percentage of the total amount that is due to the bank account indicated in the to element.

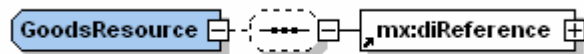
The byTransactionCount child element further contains two child elements, transactionCount and rate. The transactionCount child element identifies the reference service that should be used to retrieve the current transaction count. The rate child element indicates the monetary amount (in the optionally specified currency) per transaction. The total monetary sum represented by the transactionFee Resource in this case is the product of the rate and the count returned by the service specified in transactionCount.

The element byTransactionAmount contains two child elements, total and percentage. The total element identifies the service that can be used to retrieve the total transaction amount that is due, and the percentage element indicates the percentage of the total amount that is due. The total monetary sum represented by the transactionFee Resource in this case is the product of the rate and the total transaction amount returned by the service.

The optional child element of transactionFee specifies the details about the bank account to which the monetary payment is to be made.

goodsResource

This Resource identifies an instance of Goods. When it is present in a Clause, the goodsResource element acts as the Resource. The mandatory child element mx:diReference (an MPEG REL element) identifies a goods item being described by the goodsResource.



With the above extension elements to CEL, one can write CEL contract expressions for business transactions within goods trading applications. For example, one can specify that, when onPaymentDueEvent occurs periodically on the last day of each month, an agent Alice is obligated to pay the (monthly) transactionFee to another agent, Bob's bank account.

CONCLUSION

The CEL is a machine readable language for expressing contractual agreements with unambiguous semantics, broad expressiveness and sound extensibility structure. It is a BCF compliant technology, because its contract data model matches the agreement portion of the BCF meta-model, and its core elements have identical semantics to the BCF transaction ontology. The CEL examples provided in this paper and the CEL-BCF gap analysis documents [10, 11] demonstrate that the CEL can be used to specify BCF contracts, to model BCF business collaborations, to specify BCF business transaction patterns, and to support all of the BCF views.

For More Information

The CEL specification is currently a CRF working draft. The CRF is working toward a CEL candidate specification in 2004, and further alignment with the BCF is one of the key objectives for the candidate specification. For the latest development of CEL and the latest information on the Content Reference Forum, visit the CRF Web site at www.crforum.org.

References

1. CEL: Contract Expression Language, revision 1, May 2, 2003. http://www.crforum.org/candidate/CELWG002_celspec.pdf.
2. Content Reference Forum (CRF), <http://www.crforum.org/>.
3. Business Collaboration Framework (BCF), the Techniques and Methodologies Group (TMG), United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT), www.unbcf.org.
4. MPEG REL, ISO/IEC 21000-5:2003, Information Technology – Multimedia Framework – Part 5: Rights Expression Language.
5. UN/CEFACT Modeling Methodology (UMM) User Guide, CEFACT/TMG/N093, version 20030922.
6. Business Collaboration Framework – High Level Overview, v1.2, August 2003.
7. Business Collaboration Framework – Business Process Patterns, v0.8, August 2003.
8. E-Business Architectural Specification Business Collaboration Framework – Zachman Alignment, v1.0, August 2003.
9. UN/CEFACT, eBusiness Glossary, v0.53, December 2002.
10. CEL and BCF Gap Analysis, September 2003.
11. BCF Gap Analysis for CEL, January 2004.
12. W3C XML Schema, <http://www.w3.org/2001/XMLSchema>.