# NPL: Negotiation Pattern Language

## A Design Pattern Language for Decentralized Coordination and Negotiation Protocols

Adrian Paschke, Christine Kiss

Internet-based Information Systems
Technical University Munich
[paschke, ckiss]@in.tum.de

Samer Al-Hunaty

Loyalty Partner GmbH
Solutions Development
alhunaty@web.de

Currently negotiation covers a huge and unstructured domain of negotiation- and coordination protocols[1] and associated strategies. Researchers and practitioners like system engineers have almost no support in discussing protocol solutions to frequently occurring negotiation problems and in designing and implementing successful protocols and associated negotiation media. Negotiation engineering remains a laborious trial and error process. We propose a novel approach to the description of negotiation and coordination protocols that is based on the idea of design patterns. Design patterns have been used successfully in recent years in the software engineering community in order to share knowledge about the structural and behavioural properties of software and to communicate object-oriented design solutions [1, 2]. In this article we introduce a descriptive, semi-formal design pattern language for negotiation and decentralized coordination protocol solutions. The Negotiation Pattern Language (NPL) uses a combination of narrative human readable descriptions, formal models like Role Activity Diagrams (RAD) and machine-processable XML mark-up. This underpins tool-based cataloguing and publishing of patterns in a machine-readable, semi-formal pattern definition language and supports engineers to understand and apply these patterns. We have collected and described a first set of well-known and successful negotiation and coordination protocols and published them in a web based Design Pattern Library (DPL). The patterns are written from the point of view of a system engineer, but draw on the results of other disciplines such as social choice theory, mechanism design theory or group decision and negotiation theory. This should initiate a community effort, whereas the DPL might be used to gather and share knowledge about protocol designs. We contribute to an interdisciplinary and efficient communication process about design solutions for various negotiation and coordination problems and support system engineers in their design decisions.

## 1. Introduction

In a typical negotiation situation the negotiation parties want to reach mutually beneficial deals through extensive information exchanges. Under most circumstance they are neither strictly competitive nor completely cooperative. They negotiate strategically and mostly insincerely, which means they are trying to exploit and outsmart each other. In agent theory this is called a strategic negotiation with self-interested agents. On the other hand system engineers designing and implementing electronic negotiation media want to accomplish certain goals for the outcome of the negotiations, e.g. global efficiency goals (a social welfare or pareto optimal deal) or stability goals (e.g. a dominat strategy equilibrium [3-5], a Nash equilibrium [4-7] or some other type of equilibrium [4, 5, 8]). One way they can achieve this is to restrict and constrain the negotiation process by an underlying negotiation protocol. [9] A negotiation protocol defines the rules of interaction – the messages (offers and counter offers) which could be exchanged, their allowed sequence and given constraints such as deadline constraints [10] or negotiation budget constraints. Therefore system engineers need a clear understanding of the alternative negotiation protocols and their applicability to solve certain negotiation and coordination problems. They should be able to choose the negotiation protocol which is most suitable for fulfilling the properties and objectives of the intended negotiation media. Currently, negotiation covers a huge and unstructured domain of negotiation protocols reaching from closed and precise protocol rules to partially defined protocols, which give the participants a lot of flexibility in their decision making and information exchange activities. They are implicitly and informally described and are scattered over different literature, reaching from text books, papers and articles to internet sources. Moreover, because negotiation has been researched from a great variety of research disciplines (e.g. anthropology [11], psychology and sociology [12, 13], law [14], economics and management sciences [15, 16], mathematics [17]) and from descriptive, prescriptive and normative perspectives [18], the used terminologies, definitions and concepts differ strongly. As Gulliver noted "… literature is somewhat confused and contradictory about the definition and application of some basic concepts that are used in the study of negotiations." [11] Even the understanding of the term negotiation has no commonly agreed definition. It reaches from restricted

[1] With protocol we do not mean a low level communication protocol, but a high-level negotiation protocol which determines the possible actions the negotiation participant can choose at different points of the interaction.

and precise formal definitions to linguistic or behavioural concepts. For example linguists refer to "dialogue" to clarify a concept as "negotiation of meaning", economists, especially in the game-theoretic research discipline, use the term (bilateral) bargaining synonymously for negotiation [15-17, 19, 20] and researchers in the DAI community equate negotiation to agent coordination [21, 22]. Hence, when different communities talk about certain concepts and use certain terms then there is no guarantee that they mean the same thing. Accordingly, researchers and practitioners need efficient and structured ways of communication about successful protocol solutions. Collecting, describing, cataloguing and publishing such protocols is clearly a community effort which needs a powerful description language and an open infrastructure to gather and share the knowledge. The descriptions must be clear without ambiguity and precise but nevertheless suitable for different communities using different terminologies. They must be easy to use by negotiation engineers in order to facilitate their design decisions. Moreover, the descriptions must be also represented in machine understandable fashion and provide explicit background knowledge to underpin tools for efficient cataloguing, management and retrieval.

In this article we evolve a novel semi-formal (design) pattern language for decentralized coordination and negotiation protocols and present a (web-based) pattern library (online catalogue) with a first collection of well-known protocol descriptions. Design patterns codify successful existing practice by a more or less formalized documentation of the assumptions, structure, dynamics, applicability and consequences of a design decision as well as possible alternatives. They have been used successfully in the software engineering community in the recent years to transfer knowledge about adequate solutions to certain software problem classes and are well suited to establish a body of knowledge for a particular application domain. [1, 2] According to Cline [23] design patterns are usable in several further ways, e.g. as documentation tool to support team work or to facilitate building of robust designs with well understood tradeoffs. There is a growing body of research in the area of formal software design pattern description languages [24-26], but to our knowledge there exist no work applying the methodology of design patterns in the negotiation and coordination domain. With our Negotiation Pattern Language (NPL) we aim to establish a expressive design pattern language to describe, collect and reuse successful protocol solutions in different domains such as voting [27], auctions [28, 29], coalition forming [30-32], bilateral negotiation [33] / bargaining and markets [20], contracting [34], dispute resolution [35], distributed problem solving [36] and other. The patterns are written from the point of view of a system engineer, but draw on the results of other disciplines such as social choice theory [27], game theory [37] and mechanism design theory [5] or group decision [18] and negotiation theory [38]. The language uses a combination of narrative descriptions for human consumption and formal models (so called Role Activity Diagrams (RAD) [39]) embedded within a generic XML mark-up structure to represent protocol patterns in a semi-structured and machine and human-readable fashion. Additional background knowledge is provided utilizing powerful Semantic Web languages (the Web Ontology Language OWL [40]) to represent machine-processable meta data about the protocol definitions and the used terminologies. This underpins an online Design Pattern Library (DPL) which is intended as a tool to catalogue and publish the patterns on the Internet. The DPL supports negotiation designers in their design decision and enables a more thorough and interdisciplinary discussion about successful protocol designs and their use in practice. A native XML database (Tamino [41]) is used as a persistence layer to store and manage large amounts of design patterns written in the NPL, whereas the XML query language XQuery [42] is applied for searching and selecting patterns from the database. Furthermore, an expressive Semantic Web meta layer is lain in between in order to annotate the pattern descriptions with additional background knowledge and to allow the users categorizing, structuring and searching the DPL according to their user-own terminology and conceptual views. The meta data based reasoning mechanisms provided with OWL surpass the traditional keyword based search of common database systems and allow handling inconsistencies between merged domain specific vocabularies. This leads to a highly adaptable, open design pattern library, which provides sophisticated search facilities to the user with a high degree of precision and recall and together with the design pattern approach enables an interdisciplinary and efficient communication process about successful protocol designs.

It is not within the scope of this article to give a comprehensive overview on the vast amount of all negotiation protocols and discuss their merits. Neither will we develop new negotiation techniques, protocols or standards. The presented negotiation pattern language and the design pattern library with a first collection of negotiation patterns should be considered as a contribution to a more thorough and interdisciplinary discussion about frequently used protocol solutions and a more structured and systematic electronic negotiation engineering approach. Our goal is to move beyond the isolated view of a particular discipline or specific negotiation protocol domain and describe well-known and successful protocol solutions in a semi-formal design pattern language in order to initiate an interdisciplinary community effort for building up comprehensive negotiation pattern catalogues as toolboxes for negotiation engineers.

The further article is structured as follows: Section 2 gives a brief introduction of the relevant research streams related to negotiation research. Section 3 discusses related work. In section 4 we introduce the Negotiation Pattern Language (NPL) of decentralized coordination and negotiation protocols. We describe the language structure and the main language constructs. We then evolve a basic categorization of the field of negotiation protocols and give examples of frequently used patterns in each category. This categorization is used as default taxonomy to structure the online Design Pattern Library (DPL). Section 5 describes the DPL architecture and its features. In section 6 we elaborate on the implementation of the NPL and DPL and evaluate our approach in section 7. Finally, in section 8 we give a conclusion of our work.

## 2. Relevant Research Streams

Research on electronic negotiation and associated electronic negotiation media has mainly focused on two issues – the design of protocols and appropriate strategies (of course, apart form research on the negotiation objects itself) [33, 43]. A protocol defines the rules of interaction between the negotiation parties. Typically the process of design is composed of a requirement analysis phase and a synthesis phase. In the analysis phase the needs and desires for a given negotiation situation are investigated and transformed into a more or less formal set of requirements and constraints. Typical questions are: Who participates? What are the negotiation objects in question and what are the goals of the negotiation? What are suitable settings and conditions under which the interaction should take place? Which deals can be made and what sequences of offers and counter-offers are permitted? The extracted requirements are then used in the synthesis phase for a precise specification of the negotiation objects and particular rules [44]. Ströbel and Weinhardt call this the negotiation scenario [45]. In particular, this involves choosing the appropriate negotiation protocol for a particular design problem. In this section we will give a brief overview on the major theoretical streams related to negotiation research which led to different negotiation and coordination protocols and which form a methodological basis to assist a protocol designer in his choice of protocol, which depends on what properties he wants the overall negotiation media to have.

### Game Theory

Game theory [4] represents classes of real life situations as highly abstract models. The involved agents have different goals and preferences, i.e. they are self-interested, and strict rationality is assumed for them. An agent is called individual rational if the payoff an agent gets when participating in a negotiation is not less than the payoff that the agent would get by not participating; otherwise the agent would not participate in that negotiation. Game-theoretic models are divided into non-cooperative models, in which the set of possible actions of individuals are primitive and cooperative models, in which the set of possible joint actions of groups of players are primitive.

### Mechanism Design Theory

Mechanism Design is a special application of game theory which tries to design interaction mechanisms among rational agents which lead to some from of stability (i.e. non-manipulable protocols), such that a self-interested and rational agent is better off behaving in the intended manner of the protocol than in some other. Sometimes this leads to a dominant strategy [3-5], which is independent from the strategies of other agents. However, this is only rarely the case and other stability criteria such as Nash equilibriums [4-7], in which an agent chooses the strategy which is the best response the strategies of the other agents, are needed. The main problem is that there is not always a Nash equilibrium in a game or there are more than one equilibrium.

### Social Choice Theory

In social choice theory (voting) [27] a mechanism selects the outcome solution from the inputs of all agents. The theory distinguishes between settings with truthful voters, where all voters reveal their preferences, i.e. all preferences are known and strategic voters, which insincerely declare their preferences. In the first stetting a social choice rule take as input the preference of all agents and produces an outcome according to the preference order “>”, which in general should be Pareto efficient. Pareto efficiency takes a global perspective and compares the solutions of a mechanism. A solution *a* is Pareto efficient if there is no other solution *b* such that at least one agent is better of in the *b* than in *a* and no agent is worse off in *b* than in *a*. A subset of Pareto efficiency is a social welfare maximizing solution, which is the sum of all agents’ utilities in a solution, i.e. the protocol designer wants to enhance the social good.

### Auction Theory

Auctions are a special setting within mechanism design. Here the auctioneer wants to maximize his own profit, i.e. auctions are usually used in situations where the auctioneer wants to sell an item and get the highest possible payment. Auction theory analyzes protocols and agents' strategies in auctions. Three different value functions can be distinguished: Private value, where the value of an item depends only on the agents' own preferences, common value, where the agents' value of an item depends on the value the other agents give it and correlated value which is a combination of private and common value [28, 29]. Many auction protocols have been developed such as English (first-price open cry), Dutch (descending), Vickrey (second-price sealed-bid) auction with Pareto efficient good allocation mechanisms but also newer protocols with combinatorial bids on bundles, multi-attribute auctions, iterative double auctions or volume discount auctions with supply curves. Here the main problems are winner determination and information feedback.

### Decision Theory, Negotiation / Bargaining Theory and Markets

The subject of descision theory is decision making including decision alternatives, multiple conflicting goals and uncertainty of the negotiation outcome. Agents can make a mutually beneficial agreement, but have a conflict of interest about which one to choose. One of the central tenets of decision theory is rationality and its use in the assessment of decision alternatives and choice. In classical microeconomics often assumptions of monopoly or perfect competition are made and non-cooperative (strategic) game theory is used to select an alternative to reach an equilibrium. However, in real world settings usually these assumptions do not hold and must be relaxed to weakened rationality, non perfect competition and solutions which fulfil certain desirable properties, so called axioms of the negotiation. This is an objective of negotiation (or bargaining) theory [20] [15] which integrates decision analysis and game theory in order to provide formal and meaningful decision support. Two subfields are distinguished: strategic and axiomatic negotiations. In strategic negotiations the negotiation situation is modelled as a sequential game where agents alternate in making offers according to the underlying protocol. The outcome is given by an analysis of which of the players strategies are in equilibrium. In contrast, axiomatic negotiations postulate desired axioms on the solution concept such as invariance, anonymity, Pareto efficiency or independence of irrelevant alternatives.

A special microeconomic market setting [20] is the general equilibrium theory [46]. The theory provides a distributed method for efficiently allocating items among agents based on actual market prices. The goal is to find the best tradeoffs in a multidimensional space of alternatives.

### Distributed Problem Solving (Contracting Theory)

Most traditional negotiation, auction and market mechanisms use a single centralized mediator who is responsible for the communication tasks and dispatching of global prices (or preferences). Distributed contracting theory or problem solving omits this mediator. Instead an agent directly contacts several other agents which leads to a distributed negotiation. A very well-know protocol in this domain is the contract net protocol [34] and its many variants with cooperative agents or self-interested agents. The basic application is (re)allocation of tasks among agents with distributed and nested negotiations (contracting). More recent applications of the contract net protocol are based on marginal cost calculations with individual rationality on a pre contract basis. [47]

### Coalition Formation

The goal of coalition formation [30-32] is to increase the benefits of autonomous agents by forming coalitions that allow them to share resources and cooperate on task execution. Cooperative game-theoretic models such a normal form games or sequential games are applied with self-motivated agents, who join together and the (weak or strong) Nash equilibrium is analyzed to find efficient outcomes for coalition variants.

## 3. Related Work

The theme of (electronic) negotiation and decentralized coordination has received great attention from many academic disciplines and several definitions[2], different negotiation solutions and some prototypes mostly with narrative descriptions and implicitly encoded protocol rules have been proposed. The amount of work is overwhelming and a wide variety of electronic negotiation protocols and associated strategies have been developed especially in the past decades. Their descriptions are given in an informal fashion and are scattered over various literature. Therefore, a major challenge is to identify adequate solutions to recurring design problems and to structure, describe and catalogue these solutions based on an interdisciplinary and precise terminology, in order to provide negotiation designers a

[2] In this article we join the definition from Bichler et.al. [48], who define e-negotiation as an iterative communication and decision-making process between two or more sides represented by two or more agents who cannot achieve their objectives by unilateral actions and who search for a consensus decision.

helpful “toolbox” which assists them in choosing between alternative protocols. Initial works to categorize and structure the area of negotiation protocols and to identify shared concepts and terms has been attempted by several studies examining traditional negotiation solutions. Rosenschein and Zlotkin [33] in their book “Rules of Encounter” give a collection of well-known bilateral negotiation protocols in the context of automated negotiation. Brams and Taylor [35] discuss several procedures for dispute resolution in particular the well-known “Adjusted Winner” protocol. Ferscha and Scheiner [27] make a systematic attempt to provide virtual teams with the power of voting protocols developed in the context of the theory of collective choice. Auction protocols, another wide-spread field of negotiation protocols, are captured for example in Milgrom [49]. These studies in common were focusing under the perspective of a specific research discipline on a particular domain of negotiation protocols such as bilateral negotiations, auctions, voting (see section 2) and treat certain problems and aspects such as applying strategies or reaching certain criteria such as social welfare maximisation, stability, fairness. The protocols in most cases are described implicitly and in an informal fashion. Therefore, more general investigations followed classifying the field of negotiation protocols. Simple taxonomies, which could be found e.g. in Engelbrecht-Wiggans [50] or Friedman [51] where used to organize typical auction designs into hierarchical structures. Lomuscio et. al. [52] and Wurman et. al. [53] presented two further classification approaches which discover similarities concerning the parameters on which a negotiation protocol depends. While the one by Wurman [53] is focused on auctions, Lomuscio [52] treats the automation of negotiations. But a complete and comprehensive classification characterising most negotiation protocols is still missing. However, this precise classification is a necessary condition to develop helpful tools which efficiently support negotiation engineers in their design decisions. The former London classification for electronic negotiation which is now called the Montreal Taxonomy (MT) is one recent approach to analyse and characterise all types of e-negotiations in a comprehensive classification scheme [45]. According to Ströbel and Weinhard [45] the primary intention of the taxonomy “is to stimulate constructive feedback from the electronic negotiations community on the classification scheme in order to achieve a more general consensus”. Therefore, the intention of the Montreal Taxonomy is congruent with our goal. But, while the MT in the first place attempts to collect all relevant negotiation criteria in a kind of word-pair classification scheme and therefore has a lack of formal definition, we try to introduce a more formal pattern language to describe decentralized coordination and negotiation protocols which are usable for human decision makers but also to underpin tool support and machine-automation. Since the word-pairs of the MT classification allow room for interpretation, the selected values are more or less subjective and thus do not have the same meaning for all analysts. Hence, a more formal conceptualization is necessary. We try to address this by the use of design patterns and open, sharable Semantic Web ontologies (terminologies) and XML mark-ups to provide a more formal conceptualization, allow machine-processing in the sense of automated inferences and avoid misunderstandings about used terms. In the next section we will introduce our semi-formal Negotiation Pattern Language (NPL) for negotiation and decentralized coordination protocols, which is based on the concept of design patterns.

## 4. Negotiation Pattern Language - NPL

The field “design patterns” goes back to the early 1980s. Design patterns were used there to solve typical recurring problems in architectural design [54]. In the early 1990s design patterns began to be recognized in software engineering by Helm (1990) and Gamma (1992). In their famous book “Design Patterns – Elements of Reusable software” a.k.a. “The Gang of Four” Gamma, Helm, Jonson and Vlissides [2] described 23 typical design patterns used in software development. Software design patterns are an approach to communicate object-oriented design and facilitate design reuse in software engineering. They have been catalogued and successfully used in particular in the last decade. While the original intention was to support software engineers in their design decisions, there are other use cases as well, e.g. they can be used as a documentation tool, making it easier for a team to absorb new developers or they can be used proactively for building robust designs with well understood tradeoffs. [23] There is a growing body of research in the area of formal software design pattern description languages [24-26]. But design patterns are not just about architecture or software design, they can be useful in many other fields as well. According to a general definition of design patterns “a design pattern addresses a recurring design problem that arises in specific design situations and presents a solution to it” [55]. Hence, design patterns are efficient means to transfer knowledge about successful designs. They codify good existing practices by documenting the assumptions, structure, dynamics and consequences of a design decision in a formalized and uniform format. This uniform structure allows easily reading and comparing the design patterns. We make use of design patterns to develop a pattern language of successful protocol design in the area of negotiation and decentralized coordination protocols, which we call the Negotiation Pattern Language (NPL). To our knowledge this is a “first of a kind” approach and there are no other works which apply the concept of design patterns in the area of negotiation design.

**Name**

**Intent**

Describes the underlying coordination problem and the basic principle of the design solution.

**Also Known As**

Other names of the pattern, if existing.

**Motivation**

Depicts a scenario of a concrete design problem and gives a solution.

**Applicability**

Itemizes the situations or conditions, in which the pattern can be applied.

- Participants characteristics
- Resources
- Other environmental factors

**Structure**

Describes the interaction of the participants, which are participating.

- Protocol rules
  - Interaction rules
  - RAD
  - Allowed negotiation/bidding language
- Allocation rules (if existing) preference elicitation or bid evaluation
- Payment rules (if existing)

**Participants**

Lists the classes and objects, which are described by the design pattern.

**Consequences**

Discusses the advantages and disadvantages of the pattern, and makes suggestions about variations.

- Strategies
- Solution goals (e.g. efficiency, fairness, speed of convergence)
- Computational properties

**Variants**

Brief description of variants, extensions or specializations of the design pattern.

**Known Uses**

Examples of applications, in which the design pattern is used.

**Related Patterns**

Lists patterns, which describe a similar task, and shows the relations.

**Fig. 1: Structure of the Negotiation Pattern Language**

Different design pattern languages have been introduced in the past two decades in particular in the software engineering domain [see [56]]. We adopt the scheme of software design patterns as described in the Gang of Four book [2] and adapt it to the needs of our negotiation domain. Figure 1 gives an overview of the NPL structure and describes the main elements. We define 15 top level elements, which again may have more specialized, subsequent elements such as the possible rules of the message exchange (*protocol rules*) under “*Structure”* which consist of a narrative description of the protocol rules, a formal modelling of the protocol sequence as a Role Activity Diagram RAD [39] and additional information about the supported negotiation or biding languages. Suitable algorithms for preference elicitation, allocation rules, payment rules or bid evaluation (e.g. winner determination) are also described under “*Structure*”. The participating roles and the strategies are described under “*Participants*” and “*Consequences*”. Moreover, in the “*Consequences*” the merits, disadvantages and advantages are discussed and suggestions about possible variations are given. This is one of the most important information, which needs a thorough and interdisciplinary discussion for each individual negotiation pattern, because the choice of a particular protocol will strongly depend on the intended goals and computational properties a protocol designer wants the overall negotiation system to have. As an initial set of evaluation criteria which choose the following:

- **Social Welfare Maximisation:** Social welfare is the sum of all agents’ payoffs or utilities in a given solution, i.e. it measures the global good of the agents. It can be used as a criterion for comparing alternative mechanisms by comparing the resulting sum of the agents’ utilities and finding the maximum outcome.
- **Pareto Efficiency:** A solution is Pareto efficient if, depending on the preferences of all agents, there is no other allocation that increases the utility of one agent without decreasing the utility of another agent.
- **Individual Rationality:** A mechanism is individual rational for all participants, if it is in the interest of an independent agent to participate in the negotiation
- **Stability:** Among self-interested agents, a mechanism should be non-manipulable (i.e. stable) and motivate each agent to behave in the desired manner.
- **Fairness:** in a fair solution no party envies the other party, i.e. all agents have the same opportunities, rights and obligations.
- **Truth revelation:** no agent negotiations insincerely, but is motivated to reveal true preferences and bids truthfully, which leads to non-strategic behaviour.
- **Symmetry:** No agent has complete control of the game, i.e. symmetry guarantees similar “power” to all agents.
- **Win-Win Gains**: In Win-Win solutions a joint gain is possible either through trade-offs based on opposing preferences or though introducing new alternatives during negotiation.
- **Computational Efficiency:** An ideally negotiation mechanism is computational efficient, i.e. has linear or at least polynomial worst case complexity.
- **Communication Efficiency**: Communication among agents should be efficient. For example broadcasting as in contract net might not be ideal with this respect.

- **Distribution of computation**: Distributed computation might be preferable to avoid bottlenecks and single point of failures, as for example in centralized mediator settings.

According to the particular protocol family (auctions, voting, negotiation etc.) different criteria can be chosen to describe the merits of a particular protocol and additional measures might be needed, e.g. monotonicity, simplicity, path independence to assess voting protocols.

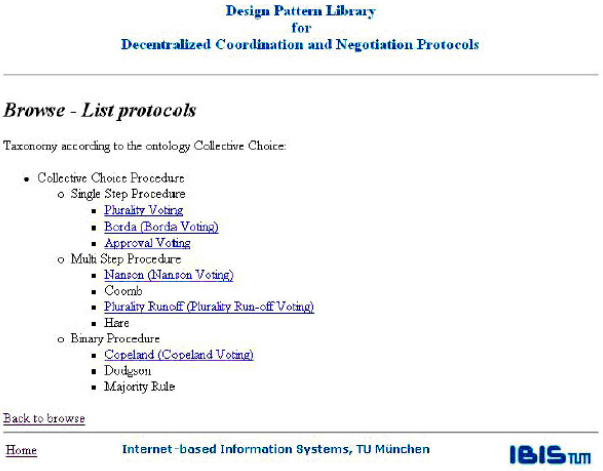
| | |
|---|---|
| Bilateral Negotiations<br>- Monotonic Consession Protocol<br>- Unified Negotiation Protocol etc. | Describes negotiation protocols between two agents. |
| Dispute Resolution<br>- Strict Alternation Protocol<br>- Adjusted Winner Protocol etc. | Describes protocols for the resolution of disputes between two or more disputants. |
| Auctions<br>- English Forward Auction Protocol<br>- Dutch Forward Auction Protocol etc. | Describes well-known auction protocols |
| Voting<br>- Approval Voting Protocol<br>- Nanson Voting Protocol etc. | Describes protocols to hold elections |
| Matching Algorithmus<br>- National Residence Matching Problem | Provides the mechanism for the matching of applicants to offered programs |
| Distributed Problem Solving<br>- Contract Net Protocol etc. | Describes protocols to solve distributed problems |

**Tab. 1: Design Pattern Library Structure**

We gathered a first collection of well-known negotiation and coordination protocols and described them as NPL design patterns (cf. table 1). The patterns are extracted from different literature such as e.g. [27, 33, 35] which treat different aspects – mostly strategic considerations - and belong to different negotiation and coordination domains such as auctions, voting, bilateral negotiations (cf. table 1). We wrote the patterns from the point of view of a negotiation system engineer who seeks to get decision support for his protocol decision problems. However, we draw on the results of other disciplines such as social choice theory, mechanism design theory or group decision and negotiation theory as well in order to give a helpful discussion of the advantages and disadvantages of a particular protocol. *Our goal is to initiate a community effort for cataloguing and publishing successful solutions and stimulate a more thorough discussion about negotiation designs and their use in practice*. This motivates the need for tool support and an open, extendable and efficiently searchable Design Pattern Library (DPL) provided on the Web.

## 5. Design Pattern Library - DPL

The online Design Pattern Library (DPL) [57] is intended to be a helpful toolbox for collecting, cataloguing and publishing negotiation patterns described in the NPL language. It provides an adaptable and efficiently searchable database storage with additional background knowledge and an easy-to-use web-based user-interface. We use the protocol domains defined in table 1 as a default taxonomy to build structured pattern catalogues, which can be "browsed" in a glossary related style. The default taxonomy reflects only one possible view on the complete space of coordination and negotiation protocols. Clearly, many others are possible and different communities might prefer other taxonomies. To address this we make use of Semantic Web technologies, in particular OWL [40] (see implementation – section 5), which facilitate shared terminologies and powerful inference processes to deal with inconsistencies and conflicts between used terminologies. As a result the DLP becomes extensible and adaptable, i.e. users can integrate their own taxonomies via mapping them to the default library structure and use their own terminologies to browse and search the library's pattern catalogues.



Design Pattern Library
for
Decentralized Coordination and Negotiation Protocols

*Browse - List protocols*

Taxonomy according to the ontology Collective Choice:

- Collective Choice Procedure
    - Single Step Procedure
        - Plurality Voting
        - Borda (Borda Voting)
        - Approval Voting
    - Multi Step Procedure
        - Nanson (Nanson Voting)
        - Coomb
        - Plurality Runoff (Plurality Run-off Voting)
        - Hare
    - Binary Procedure
        - Copeland (Copeland Voting)
        - Dodgson
        - Majority Rule

**Fig. 2: Browse View - Collective Choice ontology selected**

The DPL provides three basic functionalities: **Browse**, **Search** and **Modify**. With "Browse" (cf. figure 2) the library can be browsed by a tree structure which corresponds to the default or an user-defined taxonomy. The "Search" consists of three search alternatives: **Simple Search**, **Advanced Search** and **Meta Search**. Simple Search and Advanced Search have a comparable functionality to well-known search engines such as Google with and/or connectives, wildcards, synonyms, spell checking etc. They use the semi-structured XML mark-ups of the NPL to index and query the patterns and traditional keyword search on the narrative description to search for appropriate protocols. Different output formats, e.g. HTML, XML (i.e. NPL), PDF are supported to visualize and represent the patterns on the user interface and make them exchangeable. The "Meta Search" provides highly sophisticated search capabilities which surpass typical keyword search and semi-structured indexing schemes as used in common database systems. The meta search makes use of the additional background knowledge provided by Semantic Web meta data annotations and ontological taxonomies that allows reasoning about the pattern descriptions. Due to the formal semantics of the used ontology language OWL a lot of ambiguity in the used terminologies is removed and it is safer to reason about the pattern definitions. In particular this allows users to define their own vocabularies and use them to formulate search queries, whereas the underlying reasoning engine automatically use conceptual mappings and transitive subclass relations to translate the user-defined concepts to the default meta ontology of the DPL. Moreover, additional meta knowledge can be inferred automatically and the search space of queries will be narrowed. This leads to a higher recall and precision rate than current keyword based search technologies on the Web provide. Finally, the "Modify" function of the DPL is used to insert new design patterns or to modify/delete existing ones. Several web-based forms guide the user through the input process and the data which is inserted into the masks is transformed automatically into the NPL syntax. Alternatively, new design patterns can be directly inserted as NPL written XML documents.

## 6. Implementation

The Negotiation Pattern Language is implemented as a semi-formal XML mark-up language with a combination of machine-processable XML mark-up, human-readable narrative descriptions and formal models (Role Activity Diagrams (RAD) ). Standard Semantic Web vocabularies (RDF based Dublin Core) are used to attach further meta data to the pattern definitions and the pattern documents are catalogued within the taxonomy (t-box) of the central DLP ontology (formalized in OWL) as instances (a-box) of the ontologies' classes. This allows meeting the following design objectives:

- A formal, machine-readable pattern definition language
- An open and extensible representation language facilitating shared terminologies
- A human-readable language with structured narrative and formal pattern descriptions
- Meta data annotations that allow reasoning about pattern descriptions, in particular to provide efficient search functionalities
- A format that fits nicely into the standard Web technologies and the upcoming Semantic Web

The XML Schema of the NPL languages conforms to the previously introduced NPL structure introduced in section 4. As persistence layer to store the collected patterns we use a native XML database – the Tamino Server. Native XML databases are well suited and optimized for semi-structured XML Documents (see e.g. [58] for a discussion of higher-level XML mechanisms, XML query languages and XML databases). The Tamino Server provides useful features such as:

- Sophisticated search functionalities (phonetic search, word stemming, context operations etc.)
- W3C XQuery language support [42]
- Optimized index concepts for semi-structured XML documents

The objectives of the additional Semantic Web layer are:

1. **To share a common understanding of the structure of information among people.**

   If the protocol patterns share and publish the same underlying ontology, information can be easily extracted, aggregated and compared from the different patterns. These aggregated information can be used to answer user queries

2. **To enable reuse of domain knowledge**

   Researchers can use the developed ontology to describe their own domains, for example to build other larger ontologies or to map to other existing ontologies.

3. **To support the search functionality of the DPL**

   Synonyms and part-of-speech information for concepts in the domain will be needed, to allow adequate search functionality. Additionally, if people add new design patterns described in a different semantic language (ontology) a mapping must be provided for equivalent / disjoint concepts and terms.

4. **To dynamically structure the design patterns according to the selected taxonomy/ontology.**

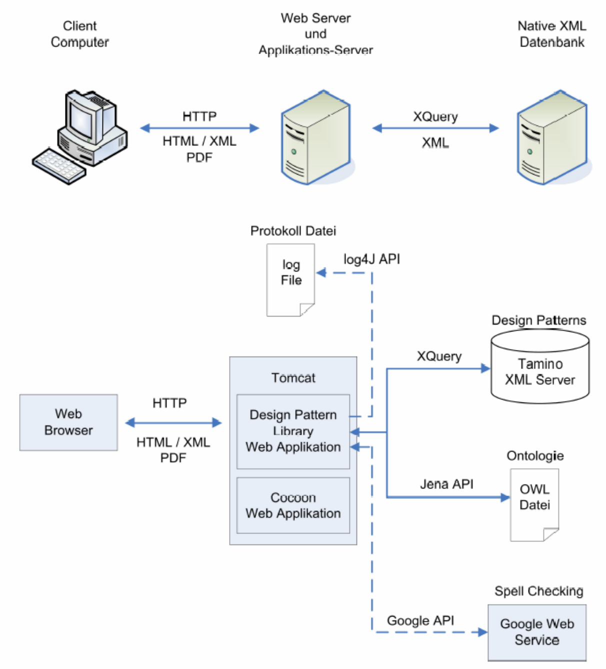


**Fig. 3: Design Pattern Library Architecture**

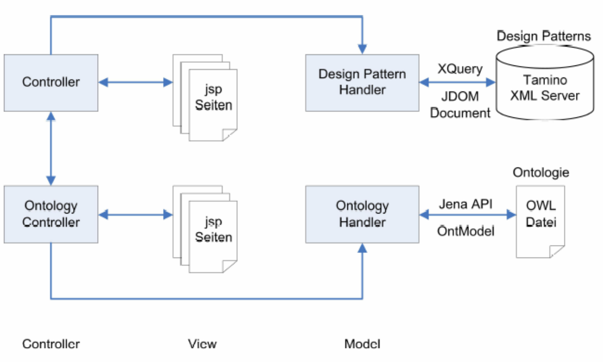


**Fig. 4: Model View Controller Architecture**

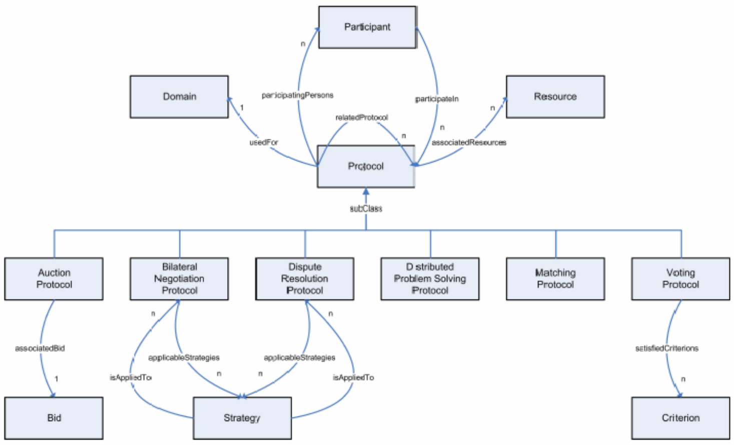


**Fig. 5: Top-Level Default DPL Ontology**

An ontology defines a common vocabulary for researchers who need to share knowledge. It includes machine-interpretable definitions of the concepts in a domain and models their relations. The Semantic Web Ontology language OWL [40] is a promising approach to develop expressive meta data languages with comprehensive cross-linkages and taxonomic sub- and super-class relationships. It enables a compact representation of hierarchical meta data concepts and provides rich inference features for effective and precise query answering. One of the biggest advantages is its openness (open world assumption), i.e. different vocabularies/taxonomies can be explicitly merged or disjoined using constructs such as "sameAs", "disjointWith" and it also offers means to handle arising inconsistencies using non-monotonic defeasible reasoning [59]. Due to the well-defined syntax and the formal semantics which is based on expressive description logics such as *SHIF(D)* and *SHOIN(D)* (a subset of first-order logic) [60] However, it must be noted that the underlying inference processes (e.g. subsumption) in particular for expressive OWL DL and OWL Full sublanguages are computationally complex (e.g. due to equivalence inferencing) and there is always a "tradeoff between the expressiveness of a representational language and its computational feasibility"[61]. We use OWL to represent the default DLP ontology and restrict the expressiveness to OWL DL, so that it is still computationally feasible even for large pattern catalogues. The pattern descriptions (i.e. links to the XML documents within the XML database) are used as individuals (a-box model) within the ontology (t-box model). As inference engine we have selected Jena. Jena is an open-source framework for development of semantic web applications which supports OWL reasoning. Figure 5 shows the first layers of our default DLP ontology with the six protocol domains (see section 4) modelled as classes (subclassOf "Protocol"). For space reasons we cannot discuss the entire default ontology in full detail. We have implemented further ontologies with different protocol taxonomies such as the *Collective Choice* ontology, which subdivides well-known voting protocols into the classes "Single Step Procedures", "Multi Step Procedures" and "Binary Procedures" (cf. figure 2) in order to illustrate the adaptability of the DPL library via merging other ontologies to the default ontology.

## 6. Evaluation

The work described in this article is a novel, first-of-a-kind approach using design patterns to describe negotiation protocols. A qualitative comparison with other approaches is therefore not meaningful. However, design patterns have demonstrated their usefulness in other domains and similar results might be expected in the area of negotiation design. Nevertheless we have attempted to gather some data on the usability and performance of the DLP implementation and the usability of Semantic Web technologies in the presented context. A qualitative benchmarking to conventional search technologies in the Web that are based on key word search or simple name-value-metadata was carried out. The measurement results collected during these assessment cycles are very promising with high levels of precision and recall. Moreover, we have evaluated the system with some initial usability tests, which leads us to add an online help to the DLP in order to support the users, but also pointed out that the user interface requires further development and refinement, whereas the concept of designs patterns is easily usable.

## 7. Conclusion

In this article, we have introduced a novel approach to describing, cataloguing and publishing negotiation protocol design solutions. We have presented a descriptive, semi-formal pattern language, called NPL, to describe negotiation and decentralized coordination protocol solutions together with an online design pattern library DPL and a first collection of frequently used negotiation protocols. The inherent advantage of our approach is that this yields a definition of successful negotiation patterns that is machine processable, but also suitable for a community to share knowledge and discuss new designs and their use in practice taking advantage of the open infrastructure of the Internet. The Web front-end provides comfortable management and search functionalities underpinned by Semantic Web technologies and XML-based query technologies. This should enable researchers and practitioners to communicate effectively about particular protocols and should provide a helpful toolbox for negotiation engineers in their design decisions. Our goal is to initiate a community effort and stimulate a constructive feedback from the electronic negotiation community in order to achieve more general consensus about protocol patterns, used terms and concepts and in the end provide a comprehensive collection of negotiation design patterns.

## Acknowledgements

To the members of the INE SSHRC E-negotiation research program [62] for first discussions about the proposed design pattern approach.

## References

1. Beck, K., *Smalltalk Best Practice Patterns*. 1996: Prentice Hall.
2. Gamma, E., et al., *Design Patterns - Elements of Reusable Object-Oriented Software*. 1995: Addison-Wesley.
3. Mueller, E. and M.A. Satterwaithe, *Strategy-proofness: the existence of dominate strategy mechanism.*, in *Social goals and social organization: Essays in memory of Elisha Pazner*, L. Hurwicz, D. Schneidler, and H. Sonnenschein, Editors. 1985, Cambridge University Press. p. 131-172.
4. Fudenberg, D. and J. Tirole, *Game Theory*. 1991: MIT Press.
5. Kreps, D.M., *A course in Microeconomics Theory*. 1990: Princeton University Press.
6. Maskin, E.S., *The theory of implementation in Nash equilibrium: a survey.*, in *Social goals and social organization: Essays in memory of Elisha Pazner*, L. Hurwicz, D. Schneidler, and H. Sonnenschein, Editors. 1985, Camebridge University Press. p. 173-204.
7. Postlewaite, A., *Implementation via Nash equilibria in economic environments*, in *Social goals and social organization: Essays in memory of Elisha Pazner*, L. Hurwicz, D. Schneidler, and H. Sonnenschein, Editors. 1985, Camebridge University Press.
8. Myerson, R.B., *Bayesian equilibrium and incentive compatibility: an introduction.*, in *Social goals and social organization: Essays in memory of Elisha Pazner*, L. Hurwicz, D. Schneidler, and H. Sonnenschein, Editors. 1985, Camebridge University Press.
9. Sandholm, T.W., *Distributed Rational Decision Making*, in *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, G. Weiss, Editor. 1999, The MIT Press, Cambridge, MA, USA. p. 201--258.
10. Kraus, S., J. Wilkenfeld, and Z. G., *Multiagent negotiation under time constraints.* Artificial Intelligence, 1995. 75(2): p. 297-345.
11. Gulliver, P.H., *Disputes and Negotiations: A Cross-Cultural Prespective.* 1979, Orlando: Academic Press.
12. Druckman, D., *Negotiations: Social Psychological Perspective.* 1977, CA: Sage: Beverly Hills.
13. Pruitt, D.G., *Negotiation Behavior.* 1981, New York: Academic Press.
14. Wetlaufer, G.B., *The Limits of Interactive Bargaining.* Georgtown Law Journal, 1996. 85: p. 1-25.
15. Roth, A.E., ed. *Bargaining: Economic Theories of Bargaining*. Social Science Encyclopedia, ed. A. Kuper. 1995, Routledge: London.
16. Young, O.R., *Strategic Interaction and Bargaining*. Bargaining: Formal Theories of Negotiations. 1975, Urbana, IL.: University of Illinois Press.
17. Harsanyi, J.C., *Rational Behavior and Bargaining Equilibrium in Games and Social Situations.* 1997, Camebridge: Camebridge University Press.
18. Bell, D.E., H. Raiffa, and e. al., *Decision Making: Descriptive, Normative and Perscriptive Interactions*. 1991, Camebridge: Cambridge University Press.
19. Bulow, J. and P. Klemperer, *Auctions vs. Negotiations.* American Economic Review, 1996. 86: p. 180-194.
20. Osborne, M.J. and A. Rubinstein, *Bargaining and Markets*. 1990: Academic Press Inc.
21. Durfee, E.H. and V. Lesser, *Negotiating task decomposition and allocation using global planning*. 1989. p. 229-244.
22. Lander, S. and K. Sycara, *Understanding the role of negotiation in distributed search among heterogenous agents*. 1993. p. 438-444.

23. Cline, M.P., *The Pros and Cons of Adopting and Applying Design Patterns in the Real World*. Communications of the ACM, 1996. 39(10): p. 47-49.
24. Krämer, C. and L. Prechtelt. *Design Recovery by Automated Search for Structural Design Patterns in Object-Oriented Software*. in *Working Conference on Reverse Engineering (WCRE96)*. 1996.
25. Eden, A.H. *LePUS - A Visual Formalism for Object-Oriented Architectures*. in *Conf. on Integrated Design and Process Technology*. 2002. Pasadena, California.
26. Mapelsden, D., J. Hosking, and J. Grundy. *Design Pattern Modelling and Instantiation using DPML*. in *Int. Conference on Tools*. 2002. Australian Computer Society.
27. Ferscha, A. and C. Scheiner. *Collective Choice in Virtual Teams*. in *IEEE 8th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*. 1999. Palo Alto, California.
28. McAfee, R.P., *Auctions and Bidding*. Journal Econ. Lit., 1987. 25: p. 699-738.
29. Milgrom, P. *Auction Theory*. in *5th World Congress: Advances in Economic Theory*. 1987. Cambridge: Cambridge Univ. Press.
30. Sandholm, T.W. and V.R. Lesser, *Coalition formation among bounded rational agents*. Artificial Intelligence - Special issue on Principles of Multi-Agemt Systems, 1997. 94(1-2).
31. Aumann, R., *Acceptable points in general cooperative n-person games*. Contributions to the Theory of Games. 1959: Princeton University Press.
32. Guiasu, S. and M. Malitza, *Coalition and Connection in Games*. 1980: Pergamon Press.
33. Rosenschein, J.S. and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. 1994: The MIT Press, Cambridge, MA, USA.
34. Smith, R.G., *The contract net protocol: High-level communication and control in a distributed problem solver*. IEEE Transactions on Computers, 1980. 29(12): p. 1104-1113.
35. Brams, S.J. and A.D. Taylor, *Fair Division: From Cake-cutting to Dispute Resolution*. 1996: Cambridge University Press.
36. Durfee, E., V.R. Lesser, and D. Corkill, *Cooperative distributed problem solving*. The Handbook of Artificial Intelligence, 1989. 4: p. 83-147.
37. Meyerson, R., *Game Theory*. 1991: Havard University Press.
38. Sebenius, J.K., *Negotiation Analysis: A Characterization and Review*. Management Science, 1992. 38(1): p. 18-38.
39. Ould, M.A., *Business Processes, Modelling and Analysis for Reengineering and Improvement*. 1995: John Wiley & Sons.
40. OWL, *Web Ontology Language (OWL), W3C Recommendation 10th Feb. 2004, http://www.w3.org/2004/OWL*.
41. SoftwareAG, *Tamino Server, http://www.xmlstarterkit.com/*. 2003.
42. W3C, *XQuery*. 2003.
43. Kraus, S., *Strategic Negotiation in Multiagent Environments*. 2001: MIT Press.
44. Wielinga, B.J., A. Akkermans, and T. Schreiber. *A formal analysis of parametric design*. in *Proceedings of the 9th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop Banff*. 1995. Alberta, Canada: SRDG Publications.
45. Ströbel, M. and C. Weinhardt, *The Montreal Taxonomy for Electronic Negotiations*. Group Decisions and Negotiation, 2003.
46. Wellman, M., *A market-oriented programming environment and its applications to distributed multicommodity flow problems*. Artificial Intelligence Research, 1994. 1: p. 1-23.
47. Sandholm, T.W. *An implementation of the contract net protocol based on marginal cost calculations*. in *National Conf. on Artificial Intelligence*. 1993. Washington D.C.
48. Bichler, M.K., G.; Strecker, S., *Towards a Structured Design of Electronic Negotiations*. Group Decisions and Negotiations, 2003.
49. Milgrom, P.R. and R.J. Weber, *A Theory of Auctions and Competitive Bidding*. Econometrica, 1982. 50: p. 1089-1122.
50. Engelbrecht-Wiggangs, R., *Auctions and Bidding Models: A survey*. Manag. Sci, 1980. 26: p. 119-142.
51. Friedman, D., *The Double Auction Market Institutions: A Survey*, in *The Double Auction Market: Institutions, Theories and Evidence*, D. Friedman and J. Rust, Editors. 1993, Addison-Wesley. p. 3-26.
52. Lomuscio, A.R., M. Wooldrige, and N.R. Jennings, *A Classification Scheme for Negotiation in Electronic Commerce*. Agent Mediated Electronic Commerce, ed. F. Dignum and C. Sierra. 2001: Springer.
53. Wurman, P., M. Wellman, and W. Walsh, *A Parametrization of the Auction Design Space*. Games and Economic Behavior, 2001. 35: p. 304-338.
54. Alexander, C., et al., *A Pattern Language*. 1977, New York: Oxford University Press.
55. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. & Stal, M., *A System of Patterns: Pattern-Oriented Software Architecture*. 1996, West Sussex, England: John Wiley & Sons.
56. Hillside.net, *Design Patterns Homepage http://hillside.net/patterns/*. 2003.
57. IBIS, *Internet Based Information Systems, Design Pattern Library, http:\\ibis.in.tum.de*. 2006.
58. Paschke, A. and W. Hümmer, *XML - Höherwertige Mechanismen*, in *Heterogene Informationssysteme*, L. Schlesinger, W. Hümmer, and A. Bauer, Editors. 2002. p. 51-76.
59. Antoniou, G., A. Bikakis, and G. Wagner. *A System for Nonmonotonic Rules on the Web*. 2004. Berlin, Heidelberg: Springer-Verlag.
60. Nardi, D., et al., *The Description Logic Handbook: Theory, Implementation and Applications*, ed. F. Baader, et al. 2002: Cambridge University Press.
61. Levesque, H.J. and R.J. Brachman, *Expressiveness and tractability in knowledge representation and reasoning*. Computational Intelligence journal, 1987. 3: p. 78-93.
62. INE-SSHRC, *E-negotiation research program: http://interneg.concordia.ca/enegotiation/index.html*. 2006.