

Appendix B

Coordination and Organisation

Malone describes coordination as “the process of managing interdependencies between activities” [280]. Seen from this perspective, coordination problems have been at the heart of AI research since the first problems solvers and planners [130, 386, 342]. Pre-occupation with this topic continues to this day with a large body of work in Distributed Artificial Intelligence (DAI) that is directly concerned with methods for coordinating groups of independent “autonomous” processes. For the purposes of this presentation work in this area is divided into two classes:

1. **Coordination:** techniques designed to enable a group of agents to work together in some way to complete a single task or well defined set of tasks (sometimes called a *problem solving episode* [266, 330]).
2. **Organisation:** a temporally stable set of control and information relationships between a group of agents which holds for several unconnected problem solving episodes [158].

Organisational frameworks can therefore be seen as providing a context for individual coordination episodes [158] or scripts/schemas to expedite decision making [167]. It is important to note however that this distinction is very subjective - the notions of “long term” and “multiple task episodes” are highly dependent upon the granularity of decomposition of the system (Organisation is in fact often considered a type of coordination [216]). The decomposition is used throughout this work however to provide some separation between different approaches.

Good entry points for the extensive literature in these areas can be found in works by Grosz [178] (cooperation), Jennings [218] (coordination) and Prietula et. al. [331] (organisation).

More general overviews of DAI and Multiagent Systems research include: [430] and [221]. In [32], Bond and Gasser also provide a useful if dated breakdown of distributed AI research by subject area. Works by Bradshaw [37, 38], Weiss [410], O’Hare and Jennings [310], Huhns and Singh [201] provide useful collections of key papers and concepts in the research field.

B.1 Coordination

The coordination of two or more agents for specific problem solving episodes has been an important area of DAI research since DAI's emergence as a distinct subfield of AI in the late 1980's. Work in the area can be grouped into two themes:

1. **Explicit (Communication-Based) Coordination:** involving agents making explicit references to tasks, roles, assignments of work etc. in their communication with each other. In other words, they actively share coordination information with each other.
2. **Implicit (Communication-Less) Coordination:** in this case, although agents may share common models of coordination occurring there is no direct (explicit) communication relating to coordination activities.

Note that, as with organisation v's coordination, there are grey areas in this distinction since in some cases agents communicate in unconventional ways (e.g. by leaving markers in the environment) or one could argue that agents having internal representations of coordinated activity (without it being communicated) is also explicit. For the purposes of this work however "communication" is restricted to direct message exchange between two or more parties and explicit is only applied to cases which include such communication about coordination. Other distinctions which are often made include selfless v's selfish [415], centralised v's decentralised coordination [158], static v's dynamic coordination [158] and cooperation as opposed to simply being coordinated [178].

B.1.1 Explicit (Communication Based) Coordination

Communication based coordination requires agents to represent and communicate about tasks, task assignments and potentially about motivations for accepting or rejecting task assignments.

Although originally applied to joint intentions approaches, Wooldridge and Jennings' breakdown of collaborative problem solving [434, 431] provides a useful framework for a wide range of coordination approaches involving explicit communication. They break a Cooperative Problem Solving (CPS) process down into four steps (taken from [431]):

1. **Problem recognition:** the CPS process begins when some agent recognises the potential for cooperative action; this recognition may come about because an agent has a goal that it is unable to achieve in isolation, or, more generally, because the agent prefers assistance.
2. **Team formation:** during this stage, the agent that recognised the potential for cooperative action at stage (1) solicits assistance. If this stage is successful, then it will end with a group having a joint commitment to collective action.
3. **Plan formation:** during this stage, the agents attempt to negotiate a joint plan that they believe will achieve the desired goal.

4. **Team action:** during this stage, the newly agreed plan of joint action is executed by the agents, which maintain a close-knit relationship throughout; this relationship is defined by an agreed social convention, which every agent follows.

Most coordination problems include these stages in one form or another even if they turn out to be trivial: plan formation may be simple if there is only one plan, team formation easy if there are only two agents and so forth.

B.1.1.1 Joint Intentions

Joint intentions as a means for modelling coordination underpins a great deal of the work on multi-agent coordination. The approach is based on the notion of *intentions* first proposed by Searle [350] and expanded upon by Bratman [39], Dennett [102] and Cohen and Levesque [79]. An intention is a certain mental state which can be considered as a “commitment to act to bring about a certain state of affairs”. Furthermore, intentions should be (paraphrasing [39]):

- **Realistic:** the agent must believe that the state of affairs it wishes to bring about is achievable (or at least not be aware that it is certainly unachievable).
- **Temporally stable:** intentions should be persistent in some sense (although not completely inflexible).

The former ensures that intentions should be translated into actions at some point and the later suggests that they correspond to medium and long term objectives, which, although they may change (if new information becomes available for example), can be used to guide the agent’s actions.

Joint intentions theory was first proposed by Cohen and Levesque ([267] and [80, 77]) and subsequently taken up by many other authors (see [215] for an analysis for different approaches). Joint intentions extends the idea of a single agent’s intention to a shared mental state amongst two or more agents – all of which then have the same intention. Joint intentions frameworks highlight that (paraphrased from [215]):

- Joint goals are required for joint action;
- All agents in the group must agree that they wish to cooperate to achieve their joint goal.

In [212] and [216] Jennings argues that all coordination frameworks for multi-agent systems can be captured with the notions of commitments and conventions. Commitments are equated with intentions. Conventions with the *social obligations* which come with monitoring validity of commitments and actions which result when commitments need to be changed or modified (such as modifying jointly committed agents). Jennings also introduces the notion of *joint responsibility* [211] which highlights that agents are generally committed to:

1. A joint goal;
2. A recipe (or plan) for achieving that goal.

The effect of the plan becoming invalid is considered to be different from the joint goal becoming invalid. In the former case the joint goal remains and agents are likely to search for another plan, in the later case all joint action is likely to be abandoned.

Joint intentions has therefore become one of the most useful tools in defining descriptions of coordinated activity in agents. Amongst the best known critiques discussing the limits of joint intentions based approaches are:

- **Failure to account for social structure:** Castelfranchi, Conte and colleagues argue that dependence relations amongst agents generate implicit social structures [61, 60] which is fundamental in determining whether agents adopt each other's goals or not [87].
- **Focus on Internal Representations:** in [429] Wooldridge argues that *coherent action* is in fact more important than *coordinated action*. That is, knowing that a system will achieve its goal *G* irrespective of the mental states of the agents.
- **Limited Applicability:** along the same lines, researchers working on emergent coordinated action (such as Franklin, Genesereth, Maes and others) contend that neither explicit mental representation of intentions, goals etc. nor communication about them is necessary for coordination and hence that joint intentions cannot account for all types of coordination.

B.1.1.2 Teamwork

As Tambe points out in [381], it is not necessarily intended that the logical frameworks underlying joint intentions approaches needs to be implemented to build coordinating systems. Rather, the models can be “compiled” and implemented as (for example) coordination rules. The joint intentions approach has been applied in several ways to create functioning coordination mechanisms. The best known are:

1. **STEAM** [384, 383]: based on Cohen and Levesque's Joint intention model, work by Tambe and colleagues on STEAM extends SOAR [254] to allow agents to explicitly model the coordination process. STEAM has also been extended to include “team oriented programming” tools to abstract away the complexity of team programming (TEAMCORE [332]). Target domains including military helicopter scenarios and robotic soccer [238].
2. **GRATE** and **GRATE*** [213]: Jennings' GRATE systems based on the notion of joint responsibility described in the previous section and have been applied to electricity distribution management [215].
3. **COLLAGEN** [338]: Rich and Sidner's interaction planner uses the sharedPlans framework devised by Grosz and colleagues [180, 181, 179] which is in turn based on joint intentions. COLLAGEN supports human users in their interaction with a computer system.
4. **Kinney et. al.** [237]: Kinney and colleagues describe a multi agent planner/scheduler which interleaves pre-planned action plans for coherent

action. Descended from early work by the same group (such as [163] and [335]), the formalisation relies on a joint intentions formalisation with possible worlds semantics.

In each case the important advantage these frameworks bring is to allow the agents to explicitly model the coordination process and react appropriately as the environment changes. This work is intrinsically linked with:

1. **Detecting interactions:** detecting positive and negative interactions between sub-plans carried out by different agents [400].
2. **Monitoring plan and team progress:** as joint actions are carried out it is vital that agents involved monitor progress to ensure that goals and plans remain valid [384, 215]. Papers addressing such monitoring, directly include [227] (using plan recognition by monitoring communication and Knowledge of potential plan structures), [404] (using Markov models) and [380, 382] (sharing agent models to allow agents to simulate each other's states).
3. **Planning and Conflict Resolution:** in agreeing on the plan they will follow to achieve a joint goal, agents will inevitably need to use finer grained coordination and conflict resolution mechanisms to ensure that plans are compatible with each agent's capabilities and objective [431]. Techniques frequently used are protocols such as the contract net [365], negotiation (see Section B.1.1.4, p190) or planning (see Section B.1.1.3, p189).

In addition, work on Multiagent conflict resolution (such as [445] and [67]) and resource allocation [76, 281, 63] is all very relevant to the second point. Work in the AI planning community on interleaving planning with execution (such as [5], [195] and [122]), re-planning [328, 195] and contingency planning [82] are also relevant to the first problem.

B.1.1.3 Planning, Meta-level Communication

The choice of sequences of actions to attain a certain goal has been at the heart of AI work since the birth of the field. Although much of the work done is concerned within the mainstream AI planning community concerns a single planning agent (often acting alone in the world), there is considerable work on planning systems dealing with multiple agents:

1. **Contingency planning:** contingency planning is concerned with developing plans which can handle deviations from the expected execution of a plan. One of the major uses is in environments where the agent executing the plan is not alone and other agents may be acting in the same world – potentially interfering with the plan [82].
2. **Adversarial planning:** is a subbranch of AI planning which deals specifically with situations where hostile agents in the environment may attempt to block achievement of a plan's goals. Work in this area includes ranges from abstractions for Adversarial planning [324, 54] to systems mixing planning and execution [5] and various different planning architectures applied to strategic games [265, 442, 425].

3. **Plan recognition:** plan recognition addresses the problem of identifying plans being executed by other agents and the re-use of this information to inform one's own decisions. Seminal work on this was done by Kautz in [229]. (Note that this work arguably falls into the category of Implicit coordination since there is often no direct communication for coordination purposes - it is repeated here for convenience.)

Work on *distributed planning* includes: Corkill's early work to distribute Sacerdoti's NOAH [342] (see [88]), Foulser and Ephrati [144, 121] looking at the plan merging problem and Rosenschein and Mueller ([339] and [300] respectively) describing plan negotiation. Centralised planning systems specifically designed to plan for multiple agents (*multi-agent planners*) are another area of significant activity. These planners systems break plans down into tasks for individual actors and reason about synchronisation of the plans which must subsequently occur at runtime. Such multi-agent planning approaches include GEMPLAN (see [255, 256]) and systems by Georgeff [163], Cammarata and colleagues [53] and Ephrati and Rosenschein [122].

Plans are also recognised as a powerful tool for explicit communication about coordination coordination. They can be used as an expression proposed future actions which agents can share amongst themselves.

1. **sharedPlans** [179]: works in a similar way to joint intentions based frameworks by combining an agent *intending that something be true* with the axiom that agents with intentions will sooner or later act to make their intentions reality.
2. **Partial Global Planning (PGP)** [117, 97, 95]: PGP is a framework in which agents are able to exchange information about their own planned actions (actions to take, their order, predicted result, time etc.) and those of others. This diffusion of knowledge is then used by the agents in the system to optimise work done by the whole community.
3. **Consensus** [73]: similar to PGP, consensus allows agents to exchange plan information via a blackboard architecture. It was originally applied to cooperating Expert Systems.
4. **Plan Semantics:** efforts are also being made to formulate explicit transferable semantics for plan fragments exchanged between agents [263]. The objective of this is to reduce dependency on local implicit semantics of planning languages used by a particular planning engine.

The approach of using meta-level communication to guide coordination is also taken up in numerous works by Decker, Durfee, Lesser and their colleagues [94, 173] as well as others [315].

B.1.1.4 Negotiation

Negotiation is a process by which agents can "act to resolve inconsistent views and reach agreement" [258]. Negotiation can be applied both in cooperative domains (to resolve resource conflicts between agents working towards the same goal for example [445]) or in competitive environments where each agent intends to maximise its benefit for purely selfish reasons (see [446]). Many negotiation approaches for both of these purposes can be seen as explicit coordination since:

- Agents agree to a set of rules imposed by the negotiation process;
- Agents have the common goal of *reaching agreement*;
- The information exchanged most often involves details of tasks to be done, goods to be exchanged and/or financial or other payment for these goods or services.

Well known techniques for negotiation include:

- **Contract Net Based:** one of the earliest coordination mechanism is the contract net protocol proposed by Smith and Davies [365, 364, 92]. The contract net which is based on a manager agent soliciting proposals from a number of potential contractors before deciding which to allocate the task to. This can be seen as simple for of negotiation and has been extended in many ways by other authors including Tidhar and colleagues [393], Boddy and Dean [30], Fisher and colleagues [137, 136], Osawa and colleagues [315] as well as many others.
- **Game theoretic approaches:** pioneered by Levy, Zlotkin, Rosenschein and colleagues in [445, 446, 447, 340, 339] this work focuses on rational choices agents might make based on evaluation of *utility functions*. A useful illustration of the approach is Levy's application to the pursuit problem in which several agents team together to catch prey and need to resolve contradictions with their own local goals [268]. This theme was extended by authors using other approaches (offering useful comparison) such as reasoning about the social cost of abandoning coordination in works such as [35, 127].
- **Recursive and Multi-Stage Negotiation:** work by Laasri, Conry, Kuwabara and colleagues ([258], [85, 84] and [252] respectively) focuses on multiple rounds of negotiation. Agents are able to counter propose offers if offers they receive are not acceptable and refine plans/solutions.
- **Argumentation Based Negotiation :** one step further than allowing counter proposals is to include explicit reasons for turning down offers when returning counter proposals. This approach was taken in the Persuader systems [378] (although with human participants) and has been taken up by a number of other groups (see [319], [360] and [244]). Castelfranchi argues that argumentation plays an important role in goal adoption [58] and work such as [224] links argumentation to algorithmic problem solving such as Distributed Constraint Satisfaction (using CSP no-good as arguments).
- **Coalition Formation:** whereas most negotiation approaches consider interactions between two agents problems which involve n agents also arise (see [121] for example). Coalition formation [226] deals with n way negotiation or task allocation problems where agents are able to form subgroups. Subgroups of agents can work together complete certain tasks and all solutions do not necessarily include all agents in the environment. Well known work in DAI on coalition formation includes work by Shehory, Kraus [352, 353], Sandholm and Lesser [345]. Work by Keptchel [232]

and Shehory [355] and their colleagues presents strategies for distributed coalition formation which although not necessarily able to find optimal coalitions appears to work effectively.

See [258] for an in depth review of negotiation approaches and [299] for a useful formalisation of the negotiation in terms of language, decision making and process. Applications of negotiation are very diverse include amongst others: telecommunications [85, 252], power load management [438], business process management [307] and transportation scheduling [137, 136].

B.1.1.5 Markets, Auctions and Institutions

Just as with negotiation, the objective of a “market” is to ensure agreement on the price and sale of a good or service (see Section B.1.1.4, p190). As Ygge suggests “local information plus market communication produces global control” [439]. A market system generally provides:

- **Communication infrastructure:** so that agents can connect to the market, receive price information, send bids and learn of outcomes.
- **Rules:** a set of rules governing interaction in the market (e.g. if an agent bids and wins it must be prepared to pay for the good or service bid on).
- **Protocols:** a set of protocols or procedures to be followed for initiating sale, sending/receiving bids and winner calculation.

Markets can therefore be seen as being defined by a set of social laws or organisational rules [145]. The rules of a market being enforced by the institution or environment the agents interact in the context of. Within this context Agents use explicit signals (messages, bids) to indicate their commitments to buying goods or performing tasks - constituting explicit coordination.

Examples of work on market based systems includes:

- **Electricity markets:** Ygge and colleagues describe detailed work on the use of auction mechanism to regulate electricity prices in electricity wholesale markets [438].
- **Building environments:** Clearwater, Huberman and colleagues describe the comparison of PID controllers and market mechanisms to share thermal (heating and cooling) resources at areas of their Xerox Park research facilities [76, 200].
- **Processing power:** Malone [281] and Chavez [63] both describe how market mechanisms can be used to share workstation computing power resources in intranets.
- **Fish markets:** work using market based systems based on the metaphor of a Spanish fish market [317] explores different trading strategies and interaction metaphors [166].
- **Telecommunications:** a large number of researchers have applied market based approaches to telecommunications scenarios, some of the best known include [85, 84] (circuit restoral problems), [250] (routing) [164, 165] (ATM VPC management) and [436, 437] (bandwidth sharing).

- **Transportation planning:** Wellman and others solve transportation scheduling problems using equalibratory markets [411, 412].

A good collection of papers on market based systems can be found in [75].

B.1.1.6 Speech Act Based Coordination

Considering coordination described in terms of intention and commitments (Section B.1.1.1, p 187) has in turn inspired coordination approaches based on *speech acts* [14, 15, 349] and consequently on standard Agent Communication Languages such as KQML [131] and FIPA ACL [133, 134]. These communication languages base interactions between agents on an agreed semantics defined for the communication language (see [253] for KQML and [133] for FIPA ACL). The semantic framework in turn is intended to permit agents to understand the beliefs, desires, intentions and other mental states other agents are trying to communicate to them. Work on coordination based around the idea of speech acts includes:

- **Coordination Primitives:** in [274], Lux and Steiner describe how planning techniques can be integrated with speech act based communication to achieve coordination based on composable coordination primitives [373]. Another approach defining a number of communication primitives can be found in [198, 199] which defines similar speech acts to those used in KQML and FIPA-ACL.
- **Coordination Languages:** the COOL coordination language [21] is designed as a layer modelling sequences of KQML messages with KIF [312] content. Finite state machines are used to keep track of the state of parties in the coordination process. Other examples of coordination languages include [359] and work by Leckie and colleagues (using KQML is work on distributed diagnosis [262] with OPS5 [142] rules and memory states as content).
- **Protocols:** Burmeister [45] presents an approach to coordination based on generic, configurable sequences of speech acts. [67] describes the use of structured, recursive negotiation dialogues for conflict resolution. These theme was then taken up by the Foundation for Intelligent Physical Agents (FIPA) which provides a set of standardised interaction protocols [135] based on FIPA ACL speech acts. These protocols can be used by agents to structure interactions form simple query-response sequences to iterated contract nets and various auction types. Finally protocols have now also been proposed as a potential basis for the semantics of ACLs themselves [325, 362].

B.1.1.7 Coordination Algorithms

In the spirit of the work on coordination protocols, one of the simplest ways to achieve coherent action amongst a number of agents is to apply a well designed (sometimes domain specific) sequence of steps for the agents to follow. Good examples of this approach are:

- **Distributed Constraint Satisfaction (DCSP)** : here a number of agents each own parts of a larger Constraint Satisfaction Problem (CSP). The agents must exchange partial solutions to find a solution to the complete problem which violates none the constraints [440]. Algorithms for solving DCSPs (such as [440, 441]) need to be designed to ensure termination, validity of the solution and (importantly) detection of completion. The resulting sequence of actions is directly analogous to a coordination method. This is neatly demonstrated by work on modelling argumentation based negotiation as a DCSP [224]). Further approaches to DCSP based coordination include [233] and [343]
- **Hierarchical Authority Algorithms**: Durfee [119] describes how a approach similar PGP can be applied to a hierarchy of agents. Agent describe their plans at various levels of abstraction and exchange all high level plans, obtaining more information on those that seem relevant. The approach incorporates a control mechanism based on authority values of each agent (place in the hierarchy) to ensure that the coordination process terminates.

In each of these cases the problem solving strategy (plan, rule sequence or constraint problem) is used to drive the interaction between agents.

B.1.1.8 Coordination Media

As well as the more formal approaches to coordination, there is a body of work which concentrates on the development of coordination software which can be used to support agent interactions. Amongst the best known of these are:

- **Blackboard Systems**: based on the idea of a shared data space in which agents can read and write data blackboard systems where first developed and used for connecting distributed knowledge sources in Expert Systems [405]. More recent work on using blackboards for coordination in multi-agent systems includes [55] and [111].
- **Tuple Space Approaches**: derived from the Linda programming language [57], tuple spaces are similar to a blackboard, acting as shared data spaces in/on which agents can read or write information in the form of vectors of values. Tuple spaces have proved especially useful for mobile agent coordination with several frameworks having been developed, including:
 - TuCSoN [311]: based on distributed tuple centres distributed throughout an Internet environment. Agents are able to interact with tuple spaces and retrieve information deposited there by other agents.
 - LIME [323]: LIME uses the Linda tuple space language in a mobile agent environment – providing its agents with reactive localised intelligence.
 - MARS (Mobile Agent Reactive Spaces) [47, 49, 48]: Mars extends TuCSoN and LIME to include programmable tuple spaces which can be used to enforce local rules in tuple centres. These local rules are then used to add a context dependent element to coordination [49].

- Jada [70, 71]: takes the tuple space concept and implements it in Java (using syntactic extensions to Java) to introduce an object metaphor and multi set rewriting.
- Javaspaces [287]: one of the most popular features of Sun’s Jini Architecture, Javaspaces works in a similar way to its tuple space cousins and Jada by providing multi purpose re-writable object based coordination spaces. These can be used for many applications including (in Jini) matching service offers to users.
- PoliS [69] and Messengers [152]: these systems extend the standard tuple/Java space metaphor further to model logical subspaces or sub-networks. Messengers further extends the Linda-like paradigm to include functions as well as data in the tuple space.

[46] reviews many of these technologies in a mobile agent coordination context, [69] provides a useful review of distributed programming in tuple spaces and [68] discusses the foundations of coordination languages based on pattern matching.

B.1.2 Implicit (Communication-Less) Coordination

While explicit coordination provides very useful models, in some domains it is either not possible or undesirable to provide agents with the mechanisms to communicate for coordination purposes. Reasons for this include:

- **Speed:** although fast coordination approaches (such as [242]) and ways of taking time into account during coordination ([245, 246]) have been proposed, explicit communication about task allocation or actions usually incurs a significant time penalty.
- **Security:** in non-cooperative and open environments it may be undesirable to allow agents to interact with others unless authentication and trust concerns can be addressed. In some cases direct interaction may also be prohibited for fairness reasons (to exclude bidder collusion in auctions for example [344]).
- **Complexity:** explicit coordination often requires considerable reasoning or communication activity on the part of the agent and in many systems (such as ant based systems [347]) this might even outweigh the mechanisms required by agents to achieve their primary tasks.
- **Limited Information:** agents might interact with one another in the environment (e.g. by playing a game of chess or participating in a sealed bid auction) but have no direct communication channel between them.

In conditions such as this, mechanisms which do not require explicit communication about (or in some cases representation of) coordination must be employed for agents to be able to work effectively with one another.

B.1.2.1 Agent Modelling

In some domains, lack of communication about agent's objectives can be compensated for by reasoning about its likely actions based on knowledge about the agent/environment or taking into account observable actions:

- **Recursive Modelling Methods (RMM):** this approach is based on knowing (or guessing) the utility (perceived payoff) of certain actions for agents in the environment [173, 171]. The knowledge of payoffs allows an agent A to construct a tree representing the different views another agent B might take on particular subject (choice). In each of these views A can incorporate what that agent B may consider agent A's view might be (and so on recursively). The tree is therefore used to represent the most profitable (rational) choice A can make given its knowledge of B. RMM is formalised in [185]. Durfee and Vidal suggest ways of making RMM like approaches more tractable [399, 115] by pruning the space of models. Gmytrasiewicz [172] builds on the idea that agents have incomplete models of on another to analyse the utility of sending certain messages (i.e. analysing if there is likely to be a net improvement in the decision another agent may make if it was more aware of another agents goals). In work similar to RMM [376] describes how to model agents for coordination using a finite state machines.
- **Plan Recognition:** in plan recognition, an agent observes the actions of other agents in the environment and attempts to match them to known action plans. This serves to discern the agent's objectives or keep track of progress. Examples of approaches to plan recognition include work by Kautz [229], Tambe and colleagues [380, 385, 382] (based on monitoring communications or access to agents state models which can then be executed in local simulations), Intille [207] and Kaminka [227] (based on monitoring communications).
- **Game Playing:** game playing problems have presented Artificial Intelligence research with many of its greatest challenges and has led to the development of many systems which rely on some kind of model of an opponent. These models range from simple alpha-beta search, game theory ([16, 301] and others) and hypergame models [396] to goal pairing in HTN planning [442], context dependent adversarial HTN planning [425], statistical analysis [146] and plan patterns [324, 418].

Although these techniques are often applied in adversarial (or at least non-cooperative) domains, they could also be used by agents wishing to cooperate but unable to communicate directly.

B.1.2.2 Social Structure

In many examples of Agent systems, agents are able to interact and produce coherent results because they operate in a framework of rules implicit in design or in the design of the system they work in. Agent behaviour is therefore pre-programmed to stay within predefined application boundaries. Some of the best examples of this are:

- **Social Laws** [356, 357]: a large body of work on social laws deals with the use of social norms, relationships or control patterns [159] to govern the actions of agents. The laws may be explicitly enforced by an outside system or simply used as guidelines for developers. There has also been extensive work on the learning or acquisition of social laws [240, 356, 357].
- **Social Power Relations**: Castelfranchi, Conte and colleagues argue that agent interactions are guided by dependence relations which agents may or may not be aware of (e.g. one agent having access to vital resources others need). [61] classifies different types of dependencies and argues that they induce an implicit social structure. [358] uses plan knowledge to infer dependencies (such as external models of goals, actions, resources and plans) and work presented in [87] and [86] discusses how agents adopt goals in the context of social dependencies.
- **Social Structures**: while notions work on social power is concerned with the internal agents perception of dependencies (and subsequent adoption of intentions/goals), Ossowski and García-Serrano [316] argue that the structures induced can also be seen as purely external. This gives rise to a norm constrained environment without particular reference to agent's internal models.

Each of these (very related) approaches can be seen as implicit coordination since agents involved may well not be aware of the restrictions posed on them. However in each case explicit coordination may still be appropriate within the societal context. These cases are discussed further under the heading of organisation (Section B.2, p 199).

B.1.2.3 Emergent Coordination

As Genesereth [161, 162], Maes [276] and Franklin [147] and others point out that coherent action amongst a number of agents can result even if:

- There is no direct communication between the agents;
- There is no a-priori mechanism for assigning or enforcing social rules;
- The agents have their own agendas (goals) which bear no relation to one another. That is, the agents have no intention to coordinate.

In these cases the agents effectively interact via the environment. Each agent's actions change the environment in some way and this in turn stimulates other agents to act in certain ways. [371] remarks that this applies most obviously to purely reactive agents which by definition have no prediction or planning capabilities. It follows therefore that any observable coordination is emergent and cannot be the result of intention. Some of the best known work on applying emergent coordination includes:

- **Tidy Robots**: work by Beckers [23] and Deneubourg [100] shows how teams of robots autonomous are able to collect randomly distributed objects into piles despite having no programmed coordination knowledge or behaviour. The performance perform uniform tasks based on simple behavioural rules and interact with each other indirectly by affecting each others environment.

- **Ant based optimisation:** similar ideas have been applied to solving complex optimisation problems such as the Travelling Salesman Problem [83, 112]. In this work individual ants perform simple exploration in their own local part of the problem space and distributedly add to the quality of the solution.
- **Ant based network traffic routing:** Appleby and Steward [4] as well as Schoonderwoerd and colleagues [347] demonstrate how ant like agents can be applied to telephone call routing problems. Ants randomly explore the network leaving artificial chemical “pheromone trails” as they move. Other ants then follow these trails, slowly reinforcing the fastest, most efficient paths.
- **Wasps:** wasp colonies have also proven a useful metaphor for problem solving. [101] investigates why wasps build such complex nest structure and how can they be used to solve complex problems (this paper also provides a good number of useful references). [72] applies the wasp metaphor to factory configuration and scheduling changes in response to job-mix changes.

In each of the cases one of the primary advantages is the simplicity of the agents involved – creating cheap, robust solutions to complex problems. This theme in some ways a multi-agent reflection of the well know Pengi reactive system [1] which was able to survive in a complex computer game world using relatively simple reactive rules.

Interestingly, several researchers (including [114]) have observed how reactive coordination can lead to emergent specialisations and implicit social structure. This self organisation appears to reflect some of the intuitions on the efficiency of organisations for streamlining decisions [167] and the effect of social dependencies [60].

B.1.3 Meta Coordination

In addition to work concerned directly with particular coordination mechanisms, there is a significant amount of work of managing usage of coordination mechanisms themselves at runtime. This “meta coordination” is reminiscent of work on meta-level reasoning [341, 29], search control ([107, 106, 333]) and realtime planning ([193],[188],[81] and [302]). In these approaches agents (or the agent) reasons about its own computational processes and progress to apply deadlines and constraints to execution – often managing “speed for quality” tradeoffs [188].

In multi-agent systems meta coordination has been applied by various authors to control the coordination process:

- **Choosing between coordination mechanisms:** a number of authors describe work in which agents choose between a number of coordination possibilities: [275], [314], [103].
- **Using Multiple levels of Abstraction:** in [116], Durfee and colleagues describe mechanisms for considering coordination options at various different levels of abstraction. Establishing the correct level of abstraction

for coordination allows detailed coordination when it matters and loose interaction for agents involved in less related activities.

- **Deciding between lone and team action:** several researchers describe mechanisms which allow agents to tradeoff the utility of initiating/participating in a team v's acting alone. Experiments have been performed in grid world predator-prey scenarios ([24] and others). More recent work in this area includes [35] (using a success measure to decide whether coordination is useful) and [196] (trading off agent utility v's "social utility").
- **Dynamic commitment reconsideration:** finally there may be situations where an agent is already committed to a joint (coordinated) action of some kind but due to changes in the environment is no longer convinced of the benefits of completing the task. [127] describes how agents could reason about the costs of breaking of coordinated action. Work such as that on level commitment contracts [346] performs a deeper analysis of the kind of agreements needed between agents to ensure that fairness is observed when and if agents withdraw from coordinated actions.

B.2 Organisation

Fox [145] and Malone [278] were amongst the first to suggest links between human organisation and computational systems. Since then, organisational approaches have become a popular way of supporting coordination and framing control relationships. Current DAI theories on coordination appear to be heading for consensus that at least to some extent "in the organisation lies the power" [194].

The primary distinction between coordination and organisation approaches made in Section B.1, p186 is on the basis of longevity of coordination relationships. A further reason for applying organisational techniques is that it is often useful to analyse behaviour at a *system level* rather than an *agent level*.¹ That is, to model the overall behaviour of a system by modelling the entities involved and the relationships between them. System requirements can then be mapped back into an agent system's implicit or explicit coordination mechanisms. The study of organisational approaches to coordination can be divided into three areas:

- **Characterisations:** notions of organisations, what are they? Do models from social science, management theory and artificial intelligence relate? If so, how do they relate?
- **Formalisations:** the concern of how organisations are modelled, represented, and how agents might know about or use organisational structures.
- **Organisational Design:** dealing with the study of performance of different organisations in particular environments or for particular tasks. Determining which information and control relationships are likely to be most effective.

¹ As noted in the introduction to this section these criteria are both rather slippery however and depend on the perspective of the model used.

Presentation is divided into these three areas. Useful volumes and overviews of DAI work on organisation include works by Carely and Prietula [56, 331].

B.2.1 Characterisations of Organisations

Adam Smith's [366] original theory of the division of labour, characterised organisation as a way of assigning resources and responsibilities to working "units", each with a specific task related to the organisational goal. This definition has been extended and refocused to serve many needs.

- Fox [145] highlights the link between distributed computer systems and human organisations using examples from the Hearsay II speech understanding system [123]. He discusses how "transactional diseconomies" arising from excessive synchronisation force structural changes in organisation. Changing organisational structures are a way of coping with:
 - Complexity: by using abstraction and omission to reduce process interaction (based on Simon's near decomposability [361]).
 - Uncertainty: identifying which organisational structure copes best with adaptation to a particular changing environment – i.e. most effectively supports coordination.
- In [160, 158] Gasser and colleagues define organisations as "settled and unsettled questions about belief and action through which agents view other agents". Settled questions correspond to streamlined "routines" which can be applied in problem solving and unsettled questions are co-ordination problems which would involved additional problems solving, negotiation or the like to resolve. The definition is not structural or externally imposed but deliberately formulated to arise from the agents mental states. The work further characterises organisations as "interlocking webs of commitment" and "patterns of actions" and fundamentally a framework for generating an "expectation of defaults".
- Blanning [27] and Crowston [90] are both more concerned with information distribution and characterise organisations as a collection of information processing entities and an associated information flow. Blanning also goes on to characterise an organisation as a physical symbol system (see [305]) and consider whether it could be considered to be intelligent in its own right.
- Gioia [167] takes the view that organisations streamline decision making by putting in place standard decision making scripts or schemas which limit an agent's action space.
- In a similar vein, work on social laws by Moses, Shoham and colleagues [296] and others [293, 44] argue that explicitly encoded behavioural norms constitute an environment in which behaviour becomes more predictable (everybody driving on the left hand side of the road for example). When these social laws are applied to control relationships (a manager may ask a worker to perform a task and the worker is expected to obey) this defines an organisational (control) structure.

- Werner groups these notions together into the concept of roles and the view of complex systems being inhabited by agents fulfilling pre-determined roles [414]. Roles are related to each other using relations which are in turn annotated by social rules.
- In their work on hierarchical coordination Durfee, Corkill and colleagues ([88, 89] and [119]) see organisation as an allocation of responsibilities for a certain set of tasks. (This work provides a good example of the fine line between organisation and coordination since organisational principles are applied at several levels.)
- Castelfranchi and Jennings view organisations as high level commitments [59] and [215] respectively. [59] clarifies the notion of a social commitment and separates it from collective commitment and sets of internal commitments. He argues that social commitments must be *between* agents. In fact organisations are effectively made up of meta-level commitments: “commitments to commit oneself”. The role of an agent in an organisation is further defined as “the set of commitments of the member to this group” – generating a set of behavioural obligations.

Although there are differences in focus between these characterisations the fundamental ideas of division of information (Blanning, Crowston, Gasser, Smith, Werner) and control (Castelfranchi, Corkill, Durfee, Gasser, Jennings, Shoham, Moses, Werner) permeate all of them. Both aspects are normally acknowledged (Blanning implicitly also goes on to use rules and responsibilities for example). There is also a fundamental aim apparent in the characterisation which is to structure the multi-agent environment to reduce decision complexity, support coordination and reduce system complexity through abstraction.

B.2.2 Formalisation of Organisations

Characterisations of organisations have given rise to a number of formalisms for describing particular organisational instances. These were either developed for their own sake or arose when organisational techniques were invoked to address some other task (such as software engineering). These formalisms can be divided into several categories: rule focused, role focused and classification focused. These are in fact often perspectives on similar underlying models, just presented from a different viewpoint

Rule based approaches describe organisations in terms of rules, laws or conventions which govern interactions in an environment, thereby inducing a social structure. This is an inherently declarative way of defining structure. These approaches derive from the social laws, social power and commitments characterisations of organisations:

- Social laws approaches declare restrictions and permissions for actions and interactions in the environment. These can be defined over social roles [296] to induce effective coordination;
- Notions of social power identify dependencies between agents in the environment. Identifying all (or the most important) dependencies generates an implied organisational structure;

- A network of publicly declared commitments (commitments to commit) also generates an implied organisational structure;
- Blanning also describes a rule based approach for modelling organisational charts in [27]. Rules determine data flow between organisational sub-units in the organisation.

Underlying each of these is the notion of a role as the object of declared rules or laws. In role based approaches the notion of a role is however taken as the primary notion of concern. Role based approaches tackle the problem in a constructivist manner, defining roles, relationships between roles and potential interactions without necessarily determining how or why these exist (the metaphor is procedural rather than declarative.). Approaches include:

- Although not defining roles per-se, in [90] Crowston describes organisations in terms of actors and messages. The descriptions of actors refer to stereotypes (such as “basic programmer”) which appear to be directly equivalent to roles. Data Flow models are then specified for message flow between actors.
- Ferber and colleagues describe the AALAADIN system which models organisations in terms of agents, roles and groups [128, 129]. Roles are filled by agents and groups further are defined in terms of roles, interaction graphs (which roles need to interact with which other roles) and interaction languages (how roles communicate). An organisation is defined as set of groups. [309] describes a design methodology for Multi Agent System design based on AALAADIN.
- Werner’s models [413, 414] define an organisation in terms of a set of roles, each of which describes and abstract agent (relying on a detailed model of agency and communication). Furthermore, roles are seen as composable (potentially using a kind of “role calculus”) and related by structural and modelling relationships such as coupled to, sub-role, specialises, conflicts with, compatible with etc.
- In [231], Kendall describes how software engineering of complex systems can also be described in terms of role models to emphasise the interaction between objects. She defines agent roles as having the following facets: role model, responsibilities, collaborations, external interfaces (to services), relationships (modelling relations such as aggregation, specialisation and so forth), expertise, coordination and negotiation. She also describes the implementation of a “role catalogue” for re-use of role definitions (very similar to the notion of organisational patterns in [444]).
- Parunak [320] continues the analogy with Object Oriented techniques by defining and exploiting extensions to UML to model organisations. The work is based on the AALAADIN [128] model but extends the representation in several ways:
 - To allow aggregation of groups of agents (agencies) to appear as single agents (also agencies). This is inspired by the [138] Holonic manufacturing paradigm and previous work in the area using agencies [308] and compound agents [157];

- Defining roles in terms of dependencies (Integrating work dependency theory by Castelfranchi and others), action templates and interaction in terms of speech act based protocols.
- Drawn from previous work on Partial Global Planning, TAEMS and other projects, So and Durfee [368] propose an organisational model which contains the following elements:
 - The set of tasks and subtasks to be done;
 - The set of agents participating in the organisation
 - An assignment of the tasks and subtasks to the participating agents;
 - A workflow structure which dictates how the tasks and subtasks are to be distributed among agents and how the results and partial results are to be synthesised;
 - (Optionally) A set of resources aside from the agents and a set of constraints on the usage of those resources may apply to agents.
- Agent-oriented modelling techniques often provide system design constructs based on the notion of roles and their relationships. Amongst the best known are:
 - **DESIRE** [40, 41, 42]: modelling in desire is primarily concerned with designing the coordination pathways for a particular task or task type. It can be used to specify task decomposition, information flow and task sequencing.
 - **ADEPT** [308, 220]: divides up the system model into “agencies” each of which is managed by a coordinating agent and contains responsibility for at least one task or a set of sub-agencies in a recursive structure. Service Level Agreements and Service Descriptions are then used to define the interactions amongst agents and agencies.
 - **GAIA** [432, 433]: GAIA specifies systems in terms of interacting roles (an organisation). Roles are defined as sets of:
 - * Responsibilities: defining the functionalities of the role.
 - * Permissions: defining (primarily) which resources the role may access.
 - * Protocols: defining possible interactions with other roles.

Role models are complemented by an interaction model defining a protocol for each type of inter role interaction. In the design phase the organisation model is in effect grounded in an agent model, service model and acquaintance model. In [444], Zambonelli and colleagues suggest that methodologies such as GAIA should be augmented to include organisational rules (attached to relations between roles), organisational structure (explicit representations of the organisation) and organisational patterns (for re-use of common organisational designs).

Finally, classification based approaches are only loosely identifiable as organisation formalisms. Organisational management theory divides up the space

of potential organisational types into several broad classes with different properties. Rather than being concerned with modelling the individual constructs (roles, relations etc.) in and organisation the major concern is to identify the global properties of a system by examining which class it falls into. [145] gives a useful taxonomy of organisational types (by increasing complexity): *person*, *group*, *simple hierarchy*, *multi-divisional hierarchy*, *price system*, *collective organisation* (many orgs sharing long term contracts) and finally, a *general market*.

Another way of dividing up these approaches is into those which advocate the existence of an explicit organisation structure (Zambonelli, Parunak, Ferber and others) and those which rely on an implicit structure arising out of agent models, rules and laws (Shoham, Castelfranchi and others).

B.2.3 Organisational Design

Formalisms for describing organisations open the door to exploring the space of possible organisations, comparing different designs and evaluating them. Many authors have argued that computational models of organisations are a vital support for organisational design, whether for real human organisations (including Carley, Prietula [331], Lin [443], Lant [257] and Gioia [167]) or computational ones (including Fox [145] and Malone [279]).

Presentation of Work in the area is divided up into two main areas: approaches for use in off-line design (Section B.2.3.2, p205) and approaches for on-line design or adaptation of designs (Section B.2.3.3, p206). Before delving into these two areas however, we examine important factors which need to be taken account in design – a subject which is widely covered in the literature (Section B.2.3.1, p204).

B.2.3.1 Critical Factors for Design and Evaluation of Organisations

As in software development, creating organisations takes objectives (tasks to be performed, goals to be achieved) and requirements (quality, cost, performance and the like) through design and evaluation to eventual implementation. It is therefore important to consider the important requirements and influences which must be taken into account in the design process.

The environment in which an organisation is intended to operate (which most authors equate to task and task structure as well as the computational, legislative and market context) is agreed to have a very large impact on the appropriate form for an organisation. More precisely, the two most common factors cited as impacting design are uncertainty and complexity:

- Uncertainty ([260, 153, 154, 145] and [374]): uncertainty is seen as forcing decision making and changing the shape of available information – in turn influencing heavily how decisions should be made and work should be done. Changing environments mean that organisations must react quickly to meet their goals. As organisations expand and diversify, uncertainty in the environment is likely to increase.
- Complexity ([145, 366]): it is complexity that is seen as forcing the division of labour and increasingly decentralised management of resources. As organisations grow in size the amount of data they must process and variety of processes they must execute increases dramatically.

Underlying each of these “top-level” factors are individual environmental factors (“task-environmental factors” in [368]) such as task arrival rate, task complexity, variability, dynamism in demand, market place etc.

Amongst the criteria often used for evaluating organisations are:

- Productivity, profitability, reliability, graph efficiency, graph hierarchy, graph connectivity, graph least-upper-roundedness (from the list in [443]);
- Response time, throughput, system utilisation, communication cost, reliability, availability, solution quality (from [368]).

This shows the range of factors to be considered – from economic to structural and operational. Some of these metrics can be applied off-line (i.e. at the design stage) – such as Lin’s metrics for robustness for example [443]. Other metrics can be evaluated on-line either in simulation of the organisation or during actual operation – such as reaction to particular complex task sequences.

B.2.3.2 Off-line Design

Off-line design of organisations has received a good deal of attention and approaches can be divided into several broad areas: classification based (by example), those based on particular problem solving techniques (AI in particular), those based on testbeds or particular coordination approaches and those based on agent-oriented design methodologies.

- **Classification based:** the type of classes of organisation described by Fox (Section B.2.2, p201) and others by broadly describe the properties of each approach, providing good starting points for standard organisational design approaches. This approach is advocated by Malone [277] and Zambonelli and colleagues (organisational patterns [444]) as well as used in in [443] (for example), to establish a stock set of organisational designs as a basis for evaluations.
- **Automated approaches:** this group includes work by Crowston on using Genetic Algorithms to evolve organisational forms [91] and work by Baligh and colleagues on expert systems for organisational design support [18, 19].
- **Testbeds and simulations:** numerous researchers such as Pollack and colleagues [329], Decker, Lesser and colleagues (with work on TAEMS [98, 96, 330] and DVMT [266]), So and Durfee ([367]) and Cohen and colleagues (PHOENIX [78]) have used simulations to analyse different organisational structures.
- **Agent Oriented Methodologies:** as described in Section B.2.2, p201, many of the agent based design methodologies include an operational model and/or implementation with which designed systems can be simulated (or even directly executed). This is the case for each of ADEPT, AALAADIN and DESIRE.

Finally, work such as that by Glance and Huberman [169, 170] focuses on the effect of varying just one aspect of organisation design (in their case the fluidity – ability of members of the organisation to move between groups and form new ones).

B.2.3.3 On-line Design, Redesign and Adaptation

Contingency theory cited from [260] in [348] and [368] posits that:

1. There is no best way to organise;
2. All ways of organising are not equally effective;
3. The best way to organise depends on the nature of the environment the organisation is to work in.

This suggests that organisations which operate in environments which change over time (i.e. are dynamic) may need to change at execution time to remain effective.

The first use of “organisational re-design” was shown in [89] which used an organisational structure to define responsibilities and interaction patterns to guide problem solving in a distributed sensor network. The organisation relationships were posted on an “organisational blackboard” and could be changed at runtime. Nodes (agents) could accept or reject organisational changes (rejecting if they appeared to have better local information than the higher level organiser. Change was primarily driven by external intervention. Subsequent work in the area has been varied and includes:

- **Gasser, Ishida and colleagues:** [210] describes the principle of providing agents with knowledge of their own organisational situation (organisational knowledge) and the capability to change the state of the organisation using two local, primitive operations “composition” and “decomposition”. [159] goes on to describe how agents can decide of their own accord when to apply such operations. The ideas were applied to distributed production systems which adapted to the effectiveness of the organisation in dealing with certain task arrival patterns. Further work such as [209] and [208] refines the approach to include (for example) the concepts of local and global statistics.
- **Guichard and Ayel:** work in [183] described logical reorganisation of a system very similar to that described by Gasser, Ishida and colleagues. Agents use local primitive operations decomposition and composition to reorganise the logical structure of the system.
- **Corkill, Lesser and colleagues:** [94] demonstrates the advantage of dynamic reallocation of tasks (dynamic organisations) v’s establishing task allocations statically. [120] presents work on evaluating organisations based on performance measures such as response time and reliability. Organisational change is defined in terms of:
 - Role re-allocation: moving tasks between agents and potentially dropping unimportant roles.
 - Local task re-ordering: interleaving local schedules more optimally with the use of local knowledge.

(In these cases the notion of organisation is again based on decomposition for a particular task or task episode).

- **So and Durfee:** So and Durfee [368] define an organisational design model based on 1) an organisational model, 2) a task environment model (defining task and environmental factors) and 2) a performance model. These are coupled with a four step process (see below) to oversee organisation performance and manage organisational change should it be necessary.
- **Coalition formation** [226, 352, 353]: work on coalition formation is very reminiscent of on-line organisation since agents dynamically establish groups which may stay in place for extended periods.

These approaches can be broadly classified into bottom up approaches driven by asynchronous distributed change (Gasser, Ishida and Guichard), top down approaches using generate and test cycles (So and Durfee) and those driven by a particular coordination mechanism (Shehory). Interestingly there appears to little work on organisation design or adaptation in rule based organisational models which could be easily updated by adding rules at runtime.

Building on this work So and Durfee describe a four step cycle for organisational adaptation at execution time (from [368]):

1. **Monitor:** monitor the effectiveness of the organisation in performing its tasks against a known set of evaluation criteria. Compare the results with conditions under which re-organisation should be considered.
2. **Design:** design new organisational structures appropriate to the new situation.
3. **Evaluate:** evaluate the new designs and pick what appears to be the best one.
4. **Implement:** implement the new organisational structure by instantiate components and replacing the old structure.

This meta control loop is reminiscent of the work on meta coordination (Section B.1.3, p198). Although it seems originally intended to apply to a whole organisational structure (i.e. global redesign) it can equally well be applied in a bottom up approach such as Gasser's to describe each agent's local reasoning about organisational change.