AFRL-IF-RS-TR-2003-243
Final Technical Report
October 2003

PRE-GAME-THEORY BASED INFORMATION TECHNOLOGY (GAMBIT) STUDY

Net Exchange

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. N229

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-243 has been reviewed and is approved for publication

APPROVED: /s/

MELVIN J. OSTER
Project Engineer

FOR THE DIRECTOR: /s/

WARREN H. DEBANY, JR.
Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	OCTOBER 2003	Final Jun 02 – Jun 03

4. TITLE AND SUBTITLE
PRE-GAME-THEORY BASED INFORMATION TECHNOLOGY (GAMBIT) STUDY

5. FUNDING NUMBERS
C - F30602-02-C-0078
PE - 61101E
PR - N229
TA - NM
WU - S1

6. AUTHOR(S)
Charles Polk

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Net Exchange
12730 High Bluff Drive, Suite 260
San Diego California 92130

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFGA
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2003-243

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Melvin J. Oster/IFGA/(315) 330-1870/ Melvin.Oster@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.	

13. ABSTRACT **(Maximum 200 Words)**
DARPA sees potential value in a strategic reasoning simulation tool - tentatively named GAMBIT (Game-Theory Based Information Technology). Net Exchange has completed a GAMBIT feasibility study. The study involved characterizing tool requirements, determining the readiness of game theory and IT, demonstrating a historical case in a GAMBIT-like scenario, and recommending a development path forward. The generic GAMBIT scenario has been characterized as Dynamic Hierarchical Gaming (DHG). Game theory is not yet ready to fully support analysis of DHG, though existing partial analysis suggests that a full treatment is practical in the midterm. IT is generally ready to support development of a GAMBIT tool with the critical exception of software agents with sufficient strategic reasoning capability - a good deal of development is required here. Management of R&D among the developers of science instruments for a planetary mission was demonstrated in a distributed software agent environment. The demonstration results from this GAMBIT-like scenario were in accord with observed history. The Diplomacy Test Utility (DTU) has been identified as a path forward. The DTU starts with a known scenario that can be expanded to DHG through a process supported by theory and tested by human play.

14. SUBJECT TERMS	15. NUMBER OF PAGES
GAMBIT, Game Theory, Dynamic Hierarchical Gaming, Strategic Reasoning	85
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

SUMMARY

In June 2002, DARPA funded Net Exchange to conduct a study that would lay the groundwork for a larger GAMBIT program (Game-Theory Based Information Technology). This study effort was called pre-GAMBIT and was completed in June 2003. There were two tracks to the pre-GAMBIT effort: Track #1 characterized the goals of GAMBIT and mapped these into a design and development structure. In Track #2, a GAMBIT-like scenario was demonstrated to illustrate the principles and promise of GAMBIT.

Scenarios of interest to the U.S. security establishment were surveyed and found to fit within a general class of strategic games called Dynamic Hierarchical Gaming (DHG). The current state of game-theoretic modeling was surveyed across the many disciplines that use and advance game theory (e.g., economics, computer science, sociology, biology, and control theory). It was concluded that DHG is beyond rigorous treatment given the current state of modeling. However, DHG amounts to a mingling of coalition games with coordination games. Current game-theoretic analysis has much to say separately about coalition and coordination games. If DHG scenarios could initially be studied decomposed into their coalition and coordination parts, then game theory might be advanced to the point where rigorous modeling of recomposed DHG scenarios could be conducted and tested.

The status of information technology (IT) to support and implement a GAMBIT strategic-reasoning toolset was surveyed. Critical IT components in need of substantial development are software agents that can reason strategically, especially in a DHG scenario, and a communications language the levels the communications field between human and software actors. This latter component is necessary given the perceived requirement that GAMBIT scenarios be of mixed participant types, human and software. For the development of both of these components, a GAMBIT testbed, leading to DHG capability, would be most helpful.

To provide a tangible reference to the intent of GAMBIT as well as the current status of the capabilities required to realize GAMBIT, Net Exchange identified a historical scenario relevant to DHG and demonstrated its simulation under various environments using a distributed software agent architecture. The scenario identified was the management of science instrument R&D for NASA planetary space missions. Cougaar Software supplied the architecture, and in so doing produced a glimpse forward to a GAMBIT software architecture. The various management environments simulated using this pre-GAMBIT system produced data that directly mirrors that from observed history; thus, the demonstration was a success.

Net Exchange concluded its efforts under this contract with the observation that GAMBIT is possible and promising; however, it cannot be attained in one development leap from the current status quo in either game theory or IT. Net Exchange has suggested an interim step – the proposed project that constitutes the Recommendations section of this report involves the use of an established strategic gaming platform, the deceptively simple game of *Diplomacy.*® By

® Diplomacy is a registered trademark of Hasbro Corporation.

adding a bit of formal structure to the on-line implementation of this game, the Diplomacy Test Utility can focus the various strands of extant research while benefiting from the participation of a large and well-trained user base. Incremental enhancement, made robust through an open architecture and verified by repeated human trials, will lead to an instance of a full strategic simulator. Generalization from this instance would result in a GAMBIT toolset.

INTRODUCTION

When interacting with other people, people decide what to do by reasoning strategically.[1] Planners, analysts, and practitioners of U.S. security policy must deal with the key question of strategic reasoning: "What should *We* do in a particular situation given that *They* are present?" A scenario simulation toolset designed within the framework of the formal study of strategic reasoning, game theory, would be a valuable aid, especially if it were accessible as a distributed software application.

A person rarely faces a frontier environment free from concern for the self-interested behavior of other people, and it is not the business of the U.S. security establishment to worry about such rare environments. Reality is played out within a network of interacting self-interests. A reality is one manifestation of this network – the pieces on the board are a projection from the many possible realities that could have resulted given the various strategies each player might have pursued. To plan for reality, and certainly to influence what reality is manifested, you must work from the network down to the board, not from the board.

Military planners, businessmen, chess players, poker players, and football coaches have always understood, in essence, that reality is a subset of strategic reasoning. Today's credible threat space has ballooned and there are neither steady-state properties nor any robust historical precedence on which to plan. The goal of GAMBIT is a strategic reasoning toolset from which numerous scenarios can be scripted and *gamed* by planners and practitioners.

As game theory is the rigorous study of strategic reasoning and as IT has made network-distributed scenario simulation practical, the steps to discerning a GAMBIT toolset are these: (i) Characterize the strategic environments that GAMBIT must address and the nature of the IT required to practically service these environments. (ii) Survey existing developments in the many research fields that employ game theory and assess how these apply to the environments of (i). (iii) Assess existing applied IT capabilities to service the environments in (i). (iv) Highlight an historical example of how disparate fields are brought together and advanced toward a common goal. And, (v) propose a means of advancing from the status quo to the GAMBIT toolset. The balance of this report elaborates on these five steps. But first, an introductory example to game theory may aid those readers not familiar with what is, essentially, a technique of structured thought regarding strategic scenarios.

A Little Game Theory

[1] Here, *strategic* does not refer to a scale of decision or action, but rather to the nature of networked self-interest. When a self-interested actor interacts, or expects to interact, with one or more other self-interested agents, then the actor reasons *strategically*.

Perhaps the most classic example in Game Theory is the Prisoners' Dilemma. The Prisoners' Dilemma is imminently tractable by most anyone, but only because a substantial structure of law, practice, occurrence, and procedure is implicit in the example. All that is fixed and accepted implicitly in the Prisoners' Dilemma must be manipulable in GAMBIT.

The following is a textbook description of the Prisoners' Dilemma:[2]

> The police have arrested two suspects of a crime. However, they lack sufficient evidence to convict either of them unless at least one of them confesses. The police hold the two suspects in separate cells and explain the consequences of their possible actions. If neither confesses, then both will be convicted of a minor offense and sentenced to one month in prison. If both confess, they will be sent to prison for six months. Finally, if only one of them confesses, then that prisoner will be released immediately while the other one will be sentenced to nine months in prison – six months for the crime and a further three months for obstructing the course of justice.

The table below summarizes the scenario in which the two suspects find themselves. The color-coding indicates each suspect's best response to the other's choices. If suspect 2 were to confess, then suspect 1 would be better off confessing. If suspect 2 were not to confess, then suspect 1 would be better off confessing. Suspect 2's strategic analysis is symmetric; thus, both choose to confess.[3]

Table 1: Prisoners' Dilemma

<table>
<tr><td colspan="2" rowspan="2">Prisoners' Dilemma
Numbers represent months in prison (Suspect 1, Suspect 2)</td><td colspan="2">Suspect 2</td></tr>
<tr><td>Confess</td><td>Don't Confess</td></tr>
<tr><td rowspan="2">Suspect 1</td><td>Confess</td><td>-6, -6</td><td>0, -9</td></tr>
<tr><td>Don't Confess</td><td>-9, 0</td><td>-1, -1</td></tr>
</table>

Well, that's it for the classic Prisoners' Dilemma. The fact that the two suspects can be held separately and kept from knowing what the other chooses results in confession being the reasonable choice for each.

Before examining all the structure implicit in the Prisoners' Dilemma, it is helpful to point out one piece of information we know from the outcome of the classic game: the two suspects are not members of the Mafia. If a member of the Mafia confessed and implicated another member, he and those close to him would be killed. In the Prisoners' dilemma, the suspects do not belong to any organizational structure that can enforce a compact of loyalty (a type of contract).

---

[2] Romp, Graham, "Game Theory: Introduction and Applications," Oxford University Press, 1997, page 9.
[3] The outcome Confess/Confess is a Nash Equilibrium – each player's strategy is a best response to the choices of all the other players.

So, what is implicit in the construction of this classic game:

1. The suspects' guilt is not obvious, yet the police know that both committed the crime. This implies a non-trivial investigative structure.

2. The legal system allows plea-bargaining. This implies a certain social-cost compact between the legal system and those protected by the legal system.

3. The police cost/benefit analysis considers letting one criminal loose a better arrangement than spending more investigative resources on the goal of convicting both suspects.

4. The police will not renege on the deal they offer; namely, if only one suspect confesses, then that one will be set free. This implies a legal system monitoring the deal and/or a desire for such deals to be credible in future cases.

5. The legal system is assumed to be able to convict someone with certainty based on a *bought* confession.

6. Neither suspect likes prison, compared to his alternatives.

7. The crime in question does not require any great skill and a highly qualified partner. If the crime represents an act of a highly trained professional and if finding a qualified partner will take many months, then the cost of confessing might be too high.

8. Both suspects know that the other suspect exists and can finger them for the crime. This is not trivially implicit. Imagine a crime organization or terrorist group that has a cellular structure. A member of two separate cells could each be involved in the same conspiracy and not know the identity of the other, or even that another cell is involved. Even if the suspects were transported in the same police car, neither would know if the other were part of their larger organization, a police plant, or an innocent.

The preceding is not an exhaustive list, but it does illustrate the substantial structure that is implicit behind the Prisoners' Dilemma scenario. In particular, there is a tiered police authority structure allied with a prosecution structure and both structures are supposed to abide by a legal code that is supervised by a court structure, all in the supposed service of a *Public* against whom the suspects have committed a crime. As for the suspects, they are petty crooks who are not part of any structure.

So, is the Prisoners' Dilemma some little piece of fluff, an academic play toy that has no broader value? Not at all! Consider the following situation:

> The FBI has informed the local police that a new organized crime syndicate may have begun operating in the municipality. The police have arrested two men accused of damaging the property of a local merchant and threatening the

> merchant's life. The merchant has not given the police any worthwhile information. A security camera in an ATM across the street from the merchant's store caught an image of the two suspects committing the crime. The police have kept this evidence secret, only revealing it to the assistant district attorney who authorized the arrest. The police suspect that the merchant's silence combined with the FBI information indicates that a classic protection scheme is being run. To test this hypothesis, the police separate the two suspects and put them through the Prisoners' Dilemma scenario. If both confess, then they are not part of any organized crime structure. If neither confesses, …

The Prisoners' Dilemma is one possible sub-game within a larger strategic structure. The larger strategic structure is composed of all the implicit elements described above, and more. The generic name we will assign to this full strategic structure is Dynamic Hierarchical Gaming (DHG), but that is jumping ahead. The basis for this generic classification must be established, and that is the point of the next section to this report.

Characterize GAMBIT Scenarios and the IT to service these

Compared with the Cold War, today's security planning is characterized by: (i) Ally/enemy relationships that are more dynamic and greater in number. (ii) A breadth of possible threats that is greater and for which responses need to be planned more quickly. (iii) Tactical margins that are likely to be tighter, necessitating a clearer idea of opponents' intentions. For all these reasons, matters of strategic interaction cannot be relegated to Rules of Thumb or any other experience-based methods that have evolved from the static, limited, narrow, high margin basis of the Cold War. GAMBIT seeks to provide a flexible, easily configured, and broadly applicable means of scenario simulation, which systematically incorporates the strategic interactions within friendlies, among friendlies, and among adversaries.

But modern security concerns go beyond military scenarios. Terrorism pursues the goal of militarism without a national structure or military force; namely, the goal of defeating a nation through conflict. The United States, in particular, and the First World, as a whole, represent a highly networked economy. Panic is short-lived, humans adjust. But a networked economy can be badly disrupted by an attack on a small subset of its nodes and/or links. Net-War is one of the few credible, mortal threats faced by a pre-imminent superpower. All the principal sub-networks (e.g.; electrical, communications, transport, water, financial, petrochemical, agricultural) are vulnerable at their nodes and on their links. Further, sub-net linkages (e.g.; commodities futures markets, gas pipelines to power plants) are also vulnerable. GAMBIT must, therefore, address Net-War scenarios.

GAMBIT, when ready for deployment, will take the form of a software application operating across a distributed network. A GAMBIT host will be able to configure its scenario of interest using a high-order scripting language. GAMBIT actors, structures, and rules will be tailored from generic models to accommodate a wide variety of specific traits. Actors may be human-directed or software agents or both (the case of a human allowing his or her agent to "take over" for a while). The ability to populate a mixed-actor *game* will allow realistically complex scenarios to be run while only having to commit the time of personnel who are key to the goals of the scenario. Further, the ability to operate the scenario over a distributed network means that the humans need not gather at a common location. Thus, GAMBIT will be widely accessible and broadly applicable.

Relevant Scenario Examples

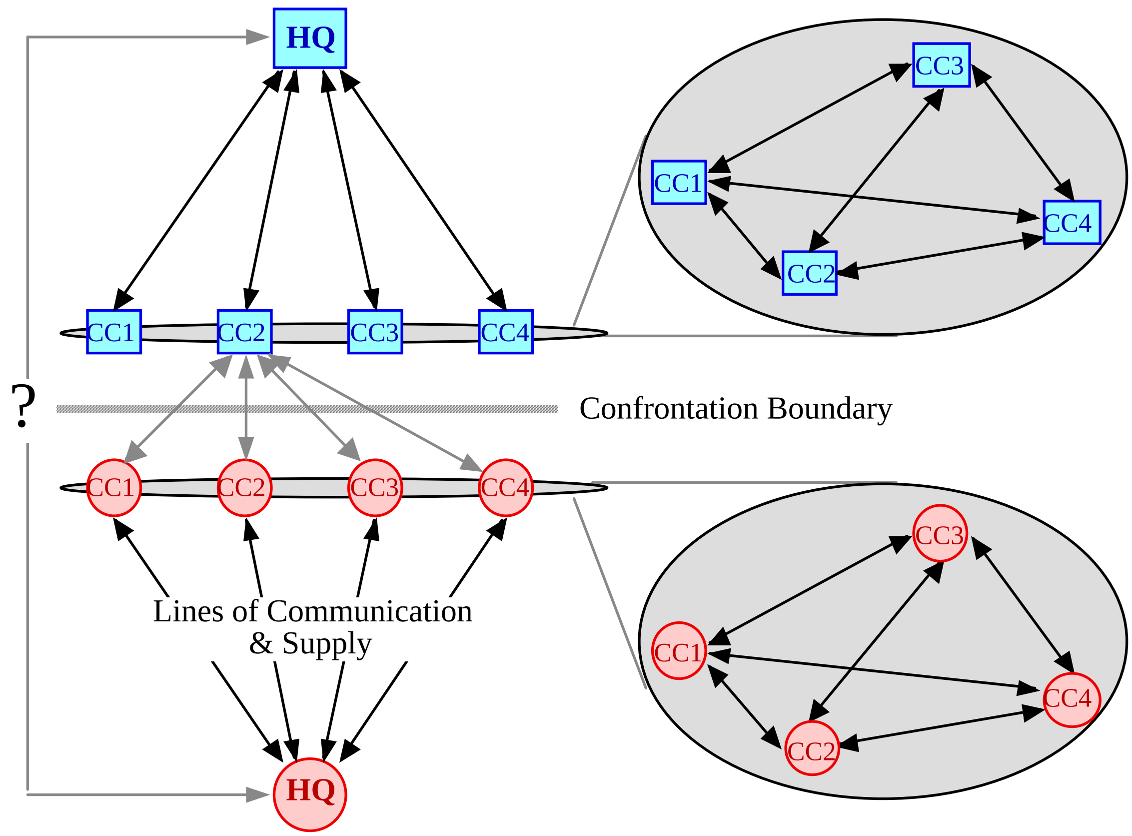
The user of the GAMBIT scenario simulation infrastructure (GSSI) must be able to specify all of the game elements and structures relevant to the scenario. Classes of scenarios will have a common foundation of structure, thus minimal customization will be needed for many types of user once they have specified their class of scenarios. But GSSI must support a range of user scenario specification that is broad enough to handle the simulation needs of the U.S. military and security establishments. A range of likely, relevant scenarios must be considered and their elements modeled in a manner that will promote efficient design and development of GAMBIT. Presented here are three seemingly distinct classes of scenario. As they are described, it becomes evident, however, that there is a high degree of commonality among them. Understanding this commonality is the starting point for GAMBIT design.

Scenario Class #1: Theater Command

Blue and Red forces face each other in a theater of potential or actual combat. Each is organized in a hierarchical command structure. Theater HQ for each allocates supplies to the deployed combat commands (CCs) that face each other across the confrontation boundary. Lines of communication and supply exist between a HQ and its CCs. Communication lines also exist among a side's CCs, with the possibility that these may also reallocate supplies among themselves.

Lines of communication may also exist between opposing forces. The two HQs will likely be in some sort of communication, if not directly, then at a higher level that can be modeled as if they were in direct communication. Communication between opposing CCs is also possible, especially if the definition of communications includes intelligence gathering and combat.

Blue Side Theater Command and Deployments



Red Side Theater Command and Deployments

Figure 1: Classic Theater Command

If this scenario begins prior to combat being initiated, then the actors must decide whether or not to initiate combat and, if attacked, how to respond. Information on how the opposing side is supplied will be key, as will be the supply capabilities and distributions within a side. If this scenario begins after combat is already underway, then the rules of engagement governing the CCs' actions will be quite different.

In this scenario, there are eleven actors, ten of whom are shown. They are a HQ and four CCs per side and The Fog of War (Fog for short). Each HQ is interested in the balance of actions and outcomes across its CCs. Losses in one CC may be completely acceptable if there are gains in others. Each CC, however, may wish to minimize its losses or maximize its combat gains or some mix of both. Fog has no interests, it is Destiny’s agent, a roller of dice that determine the outcome of combat, dice which are loaded by the relative strengths of the combatants.

In this scenario, it is easy to imagine behavior among various subsets of actors that could be characterized as cooperative, competitive, and combative. The CCs of one side may compete with each other for supplies and for meritorious performance; however, collapse of a friendly CC and/or the loss of supply from HQ represent credible threats and provide ample cause for cooperation. Possible combat among opposing CCs is anticipated, but only because this is implicit in the structure of the scenario, and the potential for cooperation among opposing CCs should not be ruled out.[4]

> *Cooperative, competitive, and combative are not fundamental qualities of an actor; rather, they are behaviors exhibited as reasoned responses to the environment in which an actor finds himself.*

A move in the theater command scenario is composed of at least four basic steps: threat assessment, supply allocation, action determination, and outcome resolution. If the scenario is run as a static, or single move game, then these four steps are all that are needed. If the scenario is played as a dynamic, or multiple move game, then these steps are augmented by re-supply (HQ and CC) and maneuver. Opportunities opened up by a dynamic setting include:

- Direct attack on a HQ
- Secondary trading of supplies among CCs
- Move-to-move strategy reassessment
- Altering the number and identity of CCs

An interesting extension to the scenario begins prior to theater deployment of Blue and/or Red. As a side deploys, a HQ must determine its CC structure. The HQ can be considered as possessing a cadre of potential CC commanders, a cadre from which the HQ selects the CC commanders and the resources assigned to each commander. This extension introduces the choice of mission designation, which is a scenario structural component that will affect the decisions of the CC commanders.

An interesting restriction to the scenario is the case where the CCs have no direct communication with each other. This is the designed state of a cellular terrorist network. Also, for point of reference, this is the condition of the accused in the Prisoners' Dilemma – two CCs without direct communication and no command structure facing a law enforcement hierarchy across a confrontation boundary.

Scenario Class #2: The Enemy of my Enemy is my Friend?

As the question mark implies, alliances lie outside the comfortable realm of kin, clan, and country. Italy was with the Central Powers until the shooting started, then went neutral, and later joined the Entente. The Soviet Union was a cautious ally of Nazi Germany and then a still more cautious ally of the Western Allies until the Nazi threat was destroyed. During the Napoleonic Wars, every European power save Britain and Portugal was at one time an ally and at another time an enemy of France. And during the Cold War, France entered and exited the NATO command structure, leaving many to doubt her value as an ally. This is the way major countries behave. Alliance building and maintenance among the likes of unstable regimes, quasi government organizations, warlords, and clans further motivates the simulation and study of The Enemy of My Enemy.

---

[4] Unauthorized truces between French and German divisions in WWI are an example of such cooperation.

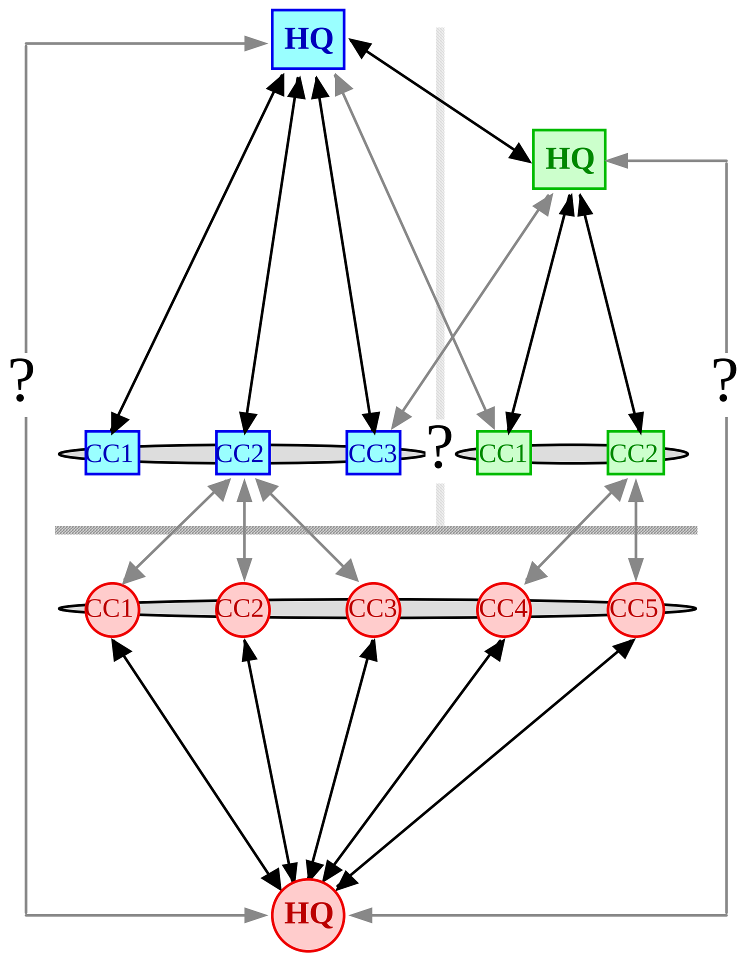


Figure 2: The Enemy of my Enemy is my Friend

In the diagram below, Blue and Green are allied against Red. The alliance could be as simple as a supply route for Blue forces that is in some way facilitated by Green. Alternatively, Blue and Green could share supply, command decision-making, guard each other's flank along a common confrontation boundary with Red, and even command each other's CCs in time of need.[5] In all cases, though, concerns arise regarding perceptions of earnest behavior and the likelihood of a separate peace or even a wholesale redefinition of the line of confrontation (side switching).

This scenario is an extension of Theater Command, but the extension highlights a critical characteristic of the range of actor motivations that GAMBIT must handle. The strategic interests of the various actors are the potential source of instability in cooperative, competitive, and combative relationships. Understanding the interplay of these interests with behavior choice is critical.

Imagine a military force comprised of autonomous or semi-autonomous combat groupings. Such was the military structure of the Roman Republic. On a scale less grand, the lords of medieval

[5] For example, the temporary transfer of command authority over several U.S. Army divisions to British General Montgomery during the Battle of the Bulge.

England were obliged to pay their taxes as money or by providing a commanded formation. In Afghanistan, the Northern Alliance was/is such a structure, and it was critical for the United States to quickly fashion an alliance with it supplied out of Pakistan and Uzbekistan. To be useful in today's world and the world we expect over the next twenty years, GAMBIT must allow for actors to switch among behavior choices – the command hierarchy, confrontation boundary, and behavior mix in a scenario cannot be fixed conditions of a GAMBIT simulation. U.S. command must not only be prepared for forming and maintaining its own alliances, it must also be well-versed in undermining an opposing alliance – Divide and Conquer is the flip side of The Enemy of my Enemy is my Friend.

Scenario Class #3: Net-War

Net-War is about conflict without confrontation boundaries. A confrontation boundary needn't be physical or static, but it does represent a certain structure relative to which opposing actors interact.[6] But structure can be discerned, which gives an overwhelmingly powerful force the opportunity to be overwhelming – thus, Net-War is a credible and reasoned response of the much weaker force. GAMBIT is interested in the French Resistance, the Chindits, and Polish Partisans, as well as Al Qaeda. Therefore, a more general term than terrorist group is necessary. What GAMBIT must accommodate is the strategy and actions of an Asymmetrically-Weak Military Opponent (AWMO).

The following diagram illustrates two Cellular AWMOs engaging an electric power network by attempting to disrupt or destroy nodes and links. There are three types of commercial actors in this network: suppliers (generators), demanders (customers/end users), and conveyors (transmission companies). Each of these actors has an internal organizational hierarchy that could be modeled; however, it is the networked interplay of these actors that is of greatest interest here. Thus, the commercial aspects of Net-War are modeled as *flat*.[7] Opposing the commercial electric power network are one or more cellular AWMOs. Defending the network is a Homeland Security hierarchy that may invest in and deploy three means to protect the network: target hardening (defense), AWMO interdiction (offense), and Node/Link de-emphasis (resiliency). Of these means, only interdiction is illustrated. Hardening and resiliency have to be carried out by the commercial actors and include such structures as the market mechanisms used to coordinate electric power commerce.

Net-War emphasizes actor-to-actor reconnaissance, information sharing, and espionage. These actions are significant in The Enemy of my Enemy; however, they take a more central position in Net-War. GAMBIT must provide a robust environment for information sharing, information stealing, lying, and concealment.

---

[6] There were many non-physical confrontation boundaries in the Cold War; e.g., the Space Race. As for dynamic confrontation boundaries, Sino-Soviet relations were dynamic and this was important to the outcome of the war.

[7] If infiltration into a commercial firm is an important strategy to consider, than a GAMBIT Net-War scenario should include the internal structures of the commercial firms.

Net-War is built up of suppliers, demanders, and services that convey what is bought and sold. All commerce is built up from these three types of actors. Thus, a well-designed GAMBIT will be able to simulate a very broad array of Net-Wars.

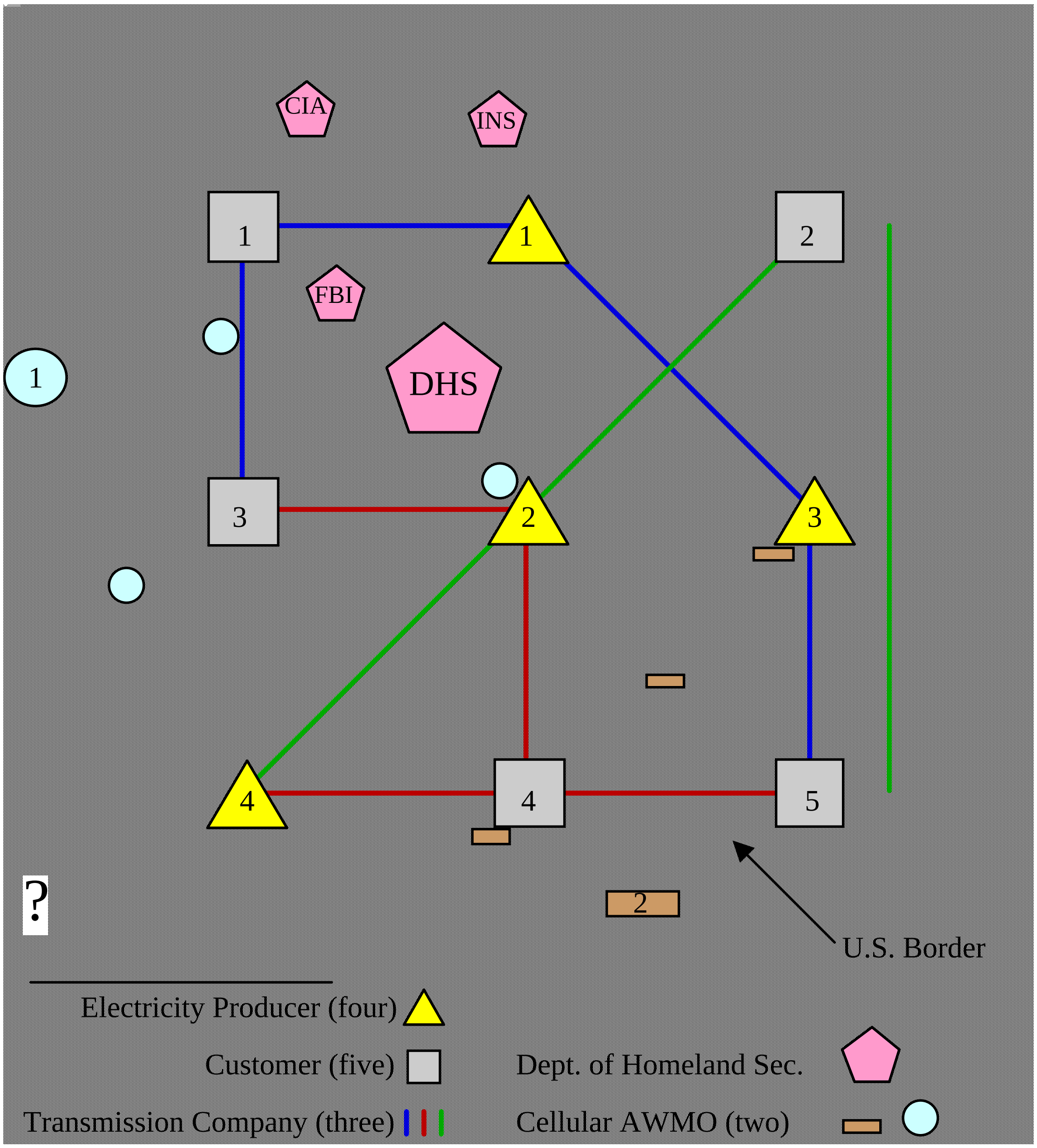


Figure 3: Net War in the Provision of Electicity

Software Architecture

This subsection provides an architectural overview of the possible forms of a GAMBIT implementation. The architecture described is a first pass aimed at creating a design and development structure to indicate feasible functionality and to highlight the imminently achievable as well as what will require additional directed research and development. This subsection presents a collection of high-level views of the GAMBIT infrastructure. The goal of these views is to clearly communicate the scope, organization, and characteristics of the software system. The baseline architecture will communicate the current understanding of:

- The key requirements and desired characteristics of the system
- The structural elements that comprise the system and their associated interfaces
- The logical and physical organization of the system

System Overview

The goal of the GAMBIT infrastructure is to provide a highly configurable, highly scalable software framework for the definition and execution of scenarios that incorporate strategic interactions among Actors. The conceptual model captures the high-level vision for the system in terms of core domain concepts and system features mapped to analysis-level design elements.

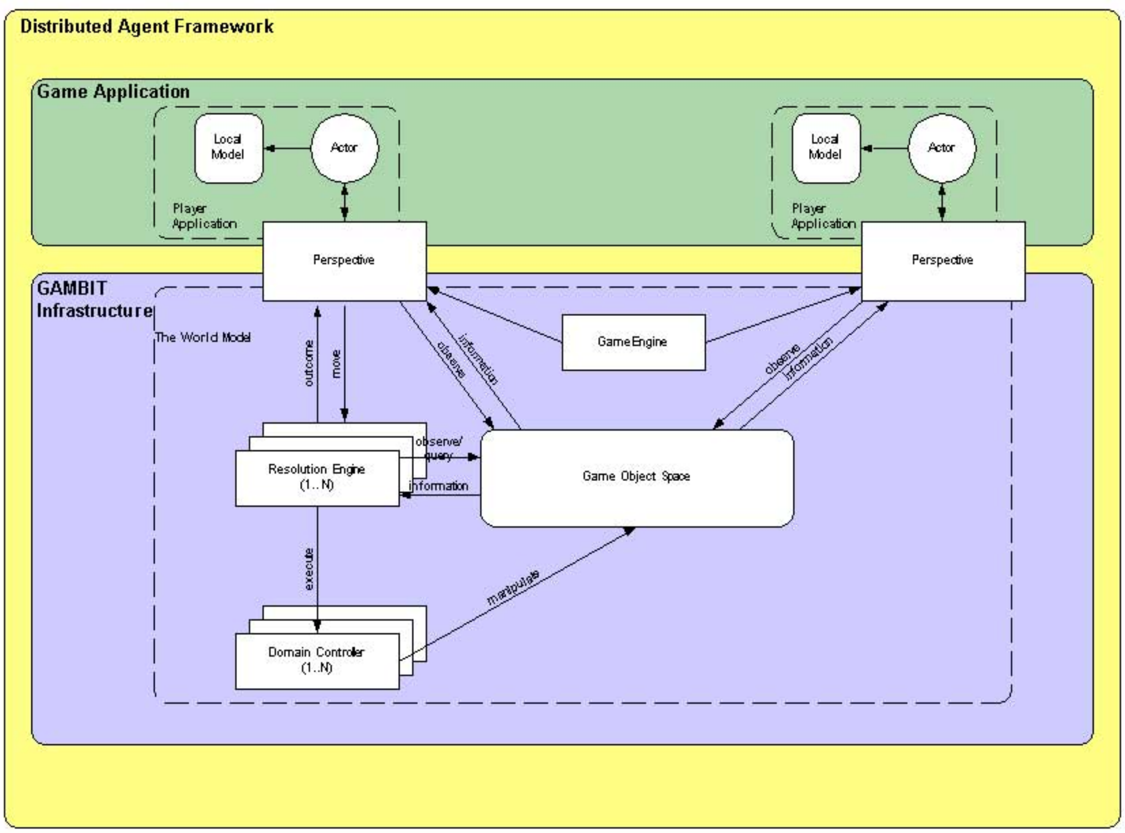


Figure 4: GAMBIT High-Level Conceptual Architecture

Game Object Space

The Game Object Space represents the collection of the entities that exist in the playing field. This includes all physical objects and Actor game tokens, as well as abstract concepts and information relevant to the objects and Actors. All game actions, moves, and informational queries are eventually manifested as manipulations of objects within this space, or result from observation of the state of objects within this space. The Game Object Space will contain little, if any, rules. The rules that govern the access and manipulation of the space will reside in other components within the framework.

Domain Controller

The Domain Controllers will be responsible for handling the manipulation of objects within the Game Object Space. An example of a Domain Controller could be "Nature." These controllers will manipulate objects in the Game Object Space according to the rules set forth in the scenario definition. This may include automatic manipulations, such as weather, as well as the actual manifestation of moves from Actors.

Before an Actor is able to manipulate the Game Object Space, its move will be resolved by the Resolution Engines, which will in turn request the appropriate Domain Controllers to execute the resultant move.

For a given scenario, there may be one or more controllers, each handling different responsibilities of object manipulation. The responsibilities will vary based on domain, as well as level of detail. In this manner, there may be a rich hierarchy of Domain Controllers that delegate different aspects of object manipulation to each other. The actual configuration of Domain Controllers, in terms of number and relationships, will vary based on the scenario type and its rules.

Perspective

Each Actor in a particular game will have a Perspective. Its Perspective is its link to the scenario. All of the Actor's moves and views of the world will be facilitated by its Perspective. The Perspective will contain all of the rules that apply to a particular Actor, which dictate the information they can see, and to some extent, the legal moves they can perform.

The Perspective will be a boundary object that provides services to the Actors, as well as enforces rules on the agent on behalf of the game infrastructure. It will communicate with the Game Engine to control when its Actor is allowed to move or perform other actions. Each of an Actor's moves will be introduced to the scenario, and Resolution Engines, through its Perspective. The Perspective will only allow the Actor to gather information or attempt a move, when it is the Actor's turn, and it will decide if the Actor has sufficient privileges to carry out its requested action. Acceptable informational queries will be drawn from the Game Object Space, and acceptable moves will be submitted to the Resolution Engines.

Game Engine

The Game Engine will be the effective referee of each game. It will contain all of the administrative rules and procedures for the game. It will be in charge of managing turns, enforcing meta-rules, and controlling the flow of the game. The Game Engine will notify each Actor's Perspective when its turn occurs and will enforce that an Actor only takes turns when appropriate. The Game Engine will facilitate the coordination required to enable scenarios in which turns are sequential as well as simultaneous. Additionally, it will enforce time limits where necessary such as: unlimited, fixed, and scenario-constrained. This coordination entails interactions with every Actor's Perspective, as well as the Resolution Engines.

Actor

An Actor is a self-interest driven participant in a scenario. An Actor is linked to the game world through its Perspective. Through the Perspective, an Actor is able to view the world and submit moves.

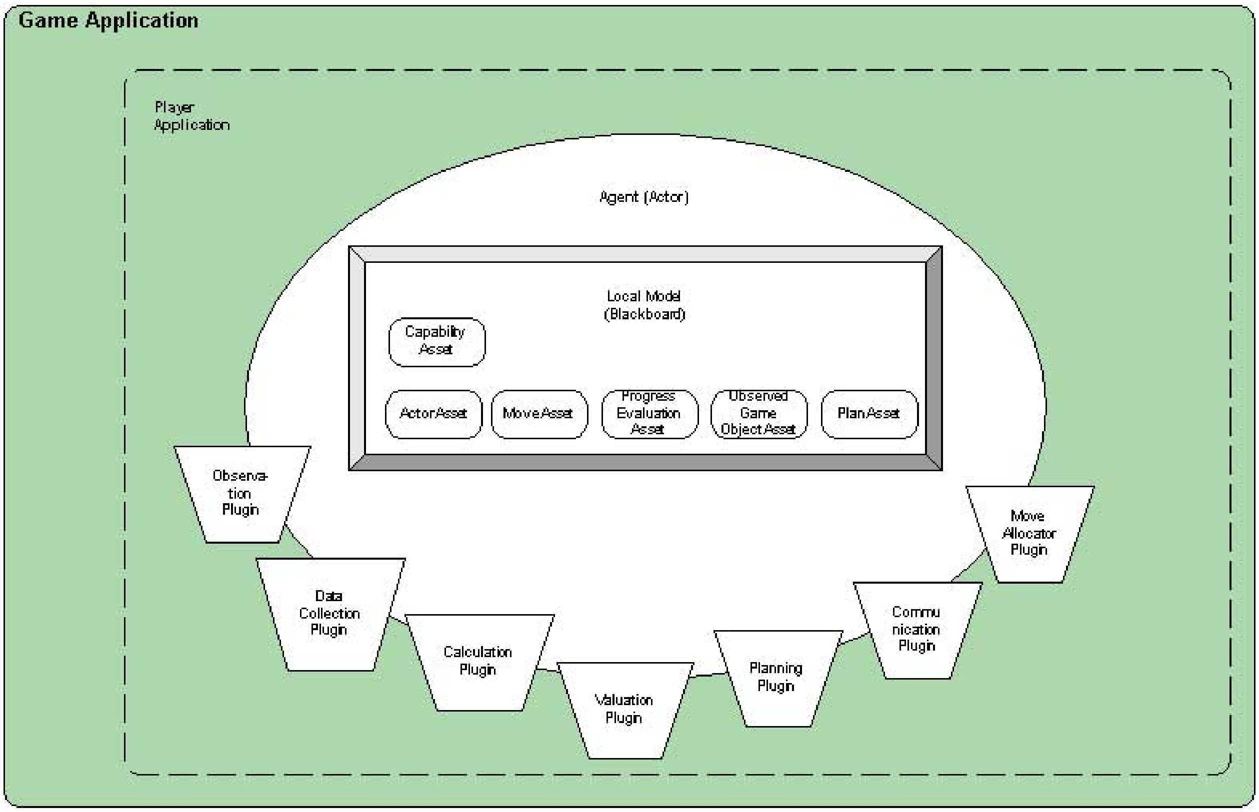


Figure 5: GAMBIT Actor High-Level Architecture

Internally, an Actor is composed of several components that work together to create its emergent strategy. As the Actor must wait for its turn to act and gather information, and only knows what its Perspective allows it to know, it must maintain a Local Model of the real game world. The Actor will use observation and communication components to update and refine its local model, attempting to create a reasonable estimate of the ground truth of the game world, which is present in the Game Object Space. The Actor will have Expertise, which influences the outcome of the Actor's production functions. An Actor's Expertise will evolve as a function of experience, with the Actor applying its knowledge about the success of its previous moves. Using its Expertise and Local Model, as well as other resources, an Actor will create its high level plans and coordinate its moves accordingly. All of an Actor's moves should be in pursuit of Goals against which it continuously evaluates its success. The following diagram depicts the conceptual view of an Actor agent.

Assets

- Actor Asset – contains the Actor's identity and state information, e.g., spatial locality.

- Move Asset – describes the actions that the Actor will submit for a turn. The move may be composed of one or more actions. It is an outcome of planning.
- Progress Evaluation Asset – contains a set of metric values that measure the extent to which a goal or a collection of goals have been met.
- Observed Game Object Asset – is a local representation of an object from the Game Object Space that was received from the Perspective.
- Plan Asset – contains the Actor's set of intended moves as well as future moves.
- Capability Asset – represents an action that an Actor can take. A move is comprised of one or more actions.

Plugins

- Observation Plugin – provides the Actor a mechanism to request information (through its Perspective) about objects in the Game Object Space. Recognizes the positive results of the request, populates the blackboard with observed Game Object Assets.
- Communication Plugin – provides a mechanism for communicating with other Actors. Populates the blackboard with any new relevant information.
- Calculation Plugin – observes game object assets and other relevant information. Performs an aggregation, expansion, or adaptation of the assets such that the Actor may better perform valuation and planning functions based upon the updated information.
- Valuation Plugin – provides the Actor the ability to assess its status in terms of the coins of its realm (e.g., money and liberty). Creates or updates the Progress Evaluation Asset to summarize this assessment.
- Planning Plugin – provides the Actor the ability to perform strategic planning. Considers the state of the Actor's current assets, and updates the Plan Asset with the next move or sequence of moves.
- Move Allocator Plugin – examines the plan asset and submits the next move (if any) to the Perspective.
- Data Collection Plugin – provides the Actor the ability to collect, archive, or report information about the state of Actor.

Resolution Engine

The Resolution Engines are the components in the infrastructure that determine the results of Actors moves. When an Actor attempts to make a move, the move will first be validated by the Actor's Perspective. This involves simple checks, such as verification that it is the Actor's turn, and validation that the move is legal within the course grain rules of the game. If the Perspective accepts the Actor's move, it will submit it to the Resolution Engines. The collection of one or more relevant engines (determined by move context), will work together to resolve the outcome of the move—in terms of manipulation of objects within the Game Object Space.

The Resolution Engines will contain all the fine grain details and rules which determine the extent to which a move is successful, and the corresponding penalties or rewards. Each move submitted to the Resolution Engines by an Actor's Perspective, will initially be received by one of potentially several high-level Resolution Engines. For a particular move, the appropriate high-level Resolution Engine will begin to break the move down into its various sub-moves, if any exist. These sub-moves then go through the same expansion process until they are ultimately decomposed into simple events, which require no further resolution. Once the Resolution Engines determine the result of the requested move, they request that the appropriate Domain

Controllers manifest the move in terms of concrete manipulations of game objects by submitting events to the controllers. The controllers then handle all aspects of realizing the dictated outcome, including any secondary events that are side effects of the primary event, as dictated by the rules of the scenario.

Drivers

This section identifies key software drivers, constraints, and goals affecting the GAMBIT architecture. These architectural drivers provide the framework in which decisions should be made. The key concerns for the architecture are rated in order of relative priority.

Table 2: Prioritization of key GAMBIT

Priority	Driver
High(1)	Configurability
High(2)	Global Availability
High(3)	Extensibility
Med	Scalability

Configurability

A configurable system supports changes to the behavior or structure of a system based upon settable parameters. For the GAMBIT system, this means that the infrastructure should support the definition and execution of a wide variety of scenarios, without the need to write new source code or replace core infrastructure components.

Global Availability

The Gambit architecture must support the execution of scenarios among players that are located in geographically different locations. In order to achieve this goal, the GAMBIT infrastructure should be efficient in it’s use of network bandwidth and should attempt to co-locate tightly-coupled processes.

Extensibility

Extensibility describes the ability of the architecture to acquire new features. The GAMBIT infrastructure must have well-defined extension points that will allow for the seamless addition of new domain and scenario-specific behaviors.

Scalability

Scalability describes a system’s ability to continue to function well as it is increased in size or volume. For GAMBIT, this means that the infrastructure should support scenarios that have only a few players, or scenarios that have a very large number of players (size to be determined) equally well.

Collaboration View

The Collaboration View describes the set of interactions that represent significant, central functionality, plus those that exercise key architectural elements, or that stress or illustrate a specific, delicate point of the architecture. Collectively, these are referred to as the architecturally significant collaborations.

The Collaboration Model (below) describes how objects in the system should interact with each other to get work done. "Configuration" collaborations, as a group, perform the setup and configuration of the game where as the "Mechanics" implement the basic execution events between game components. The "Services" contains a set of collaborations that are available during the game. The collaborations, highlighted in yellow, are further defined in individual collaboration diagrams.

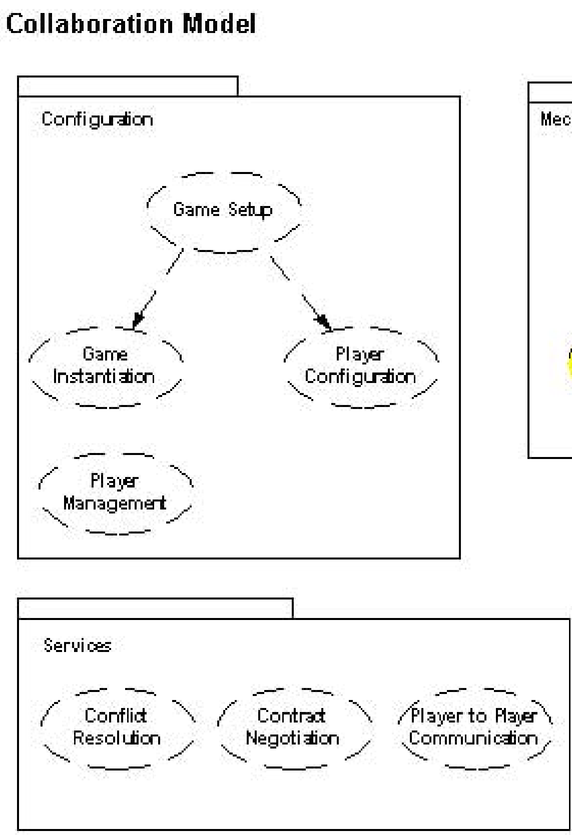


Figure 6: GAMBIT Collaboration UML Class Model

Realized Collaborations

This section shows the behavioral aspect of the software architecture by describing the communication of components to realize specific system behaviors.

Game Execution

The Game Engine initiates the start of the scenario based on input from an Actor or administrative function. It then initiates the Turn Execution of each Actor as appropriate to the rules and type of scenario. It recognizes the end of each Actor's turn as the Resolution Engine returns the result of a requested move. Alternatively, a timeout may occur if the Actor extends beyond its allotted time for the turn.

Turn Execution

Turn Execution Collaboration Diagram depicts the breakdown of interactions between components within the system to execute a player's turn during a game.

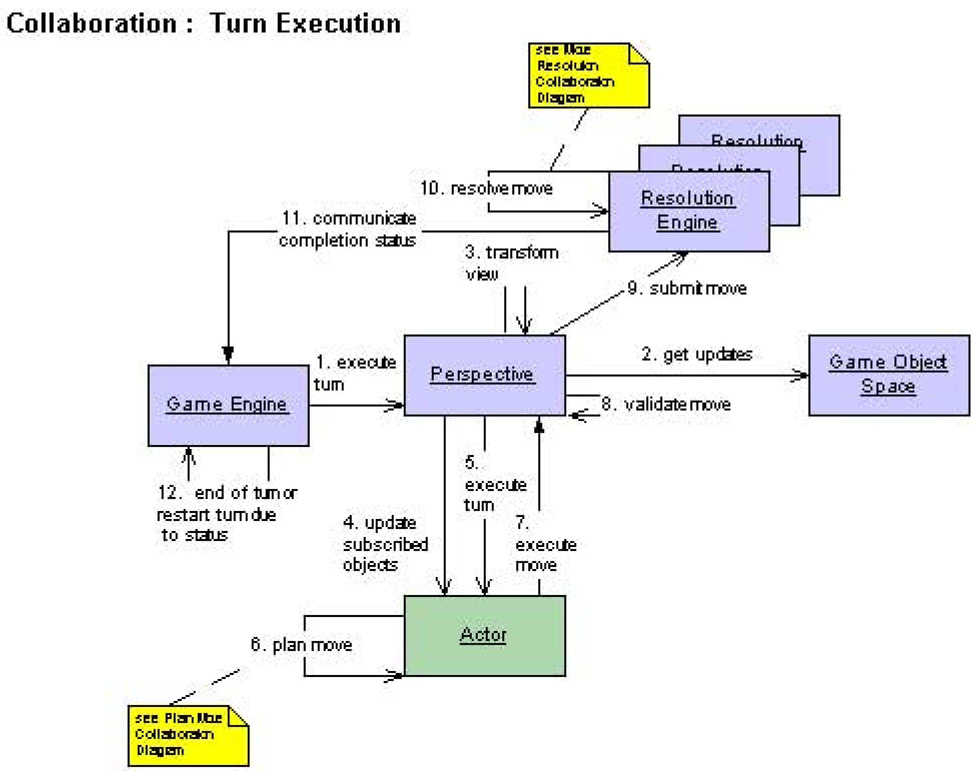


Figure 7: Actor-Perspective Turn Execution UML

Turn Execution Steps:

1. The Game Engine notifies a Perspective that it is that Actor's turn to perform a move.
2. The Perspective retrieves updates of specific Game Object Space entities that are of interest to its Actor.
3. The Perspective creates a view of the observed information that meets the scenario-specific criteria of "fog of war" appropriate to the visibility of the Actor.
4. Using the newly transformed view, Perspective updates the objects that are part of the Actor's subscription.
5. Perspective notifies the Actor that it is time to make a move.
6. Actor plans and creates its next move.
7. Actor submits the move to the Perspective for execution.
8. Perspective validates the Actor's move.

8a. [optional] If the move is invalid, the Perspective notifies the Actor of this and requests a new move (see step 5).

9. Perspective submits the move to the Resolution Engine(s).

10. Resolution Engine resolves the move into sub-moves (if necessary) and passes the move to the appropriate handling-engine.
11. The handling Resolution Engine determines the outcome status of move execution and notifies the Game Engine that the move is complete, along with an indication of the success or failure of the move.
12. Based upon the reported status of the move, the Game Engine determines whether the turn is ended or if the actor should reattempt its turn.

Move Planning

The Move Planning collaboration diagram represents the interaction of components to realize the Move Planning behavior.

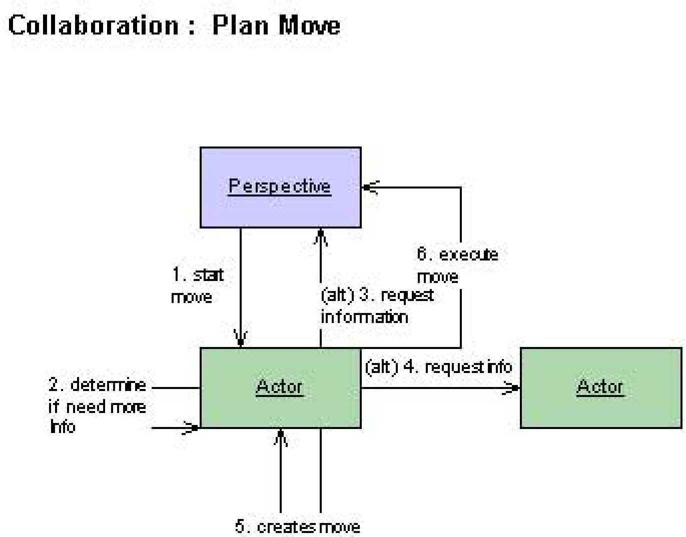


Figure 8: Actor-Actor Plan Move UML Collaboration Model

Plan Move Steps:

1. The Actor receives a “Start Move” notification from its Perspective.
2. Actor determines the need, if any, for further information necessary to determine its next move.
3. [optional] If the actor needs additional Game Object information, it queries the Perspective for that information. (The perspective will return information with the appropriate “fog of war” view applied.)
4. [optional] If the Actor needs additional information from another Actor, it communicates with other actors to acquire that information.
5. The Actor plans and creates its next move.
6. The Actor submits the next move to its Perspective for execution.

Plan Move Activity Diagram

The following activity diagram presents the interactions between Actor Plugins and Assets as depicted in the corresponding descriptions in section 0 of this document. Each step is shown in terms of the actions of the Plugin and resulting updated or created Assets.

Activity Diagram : Plan Move

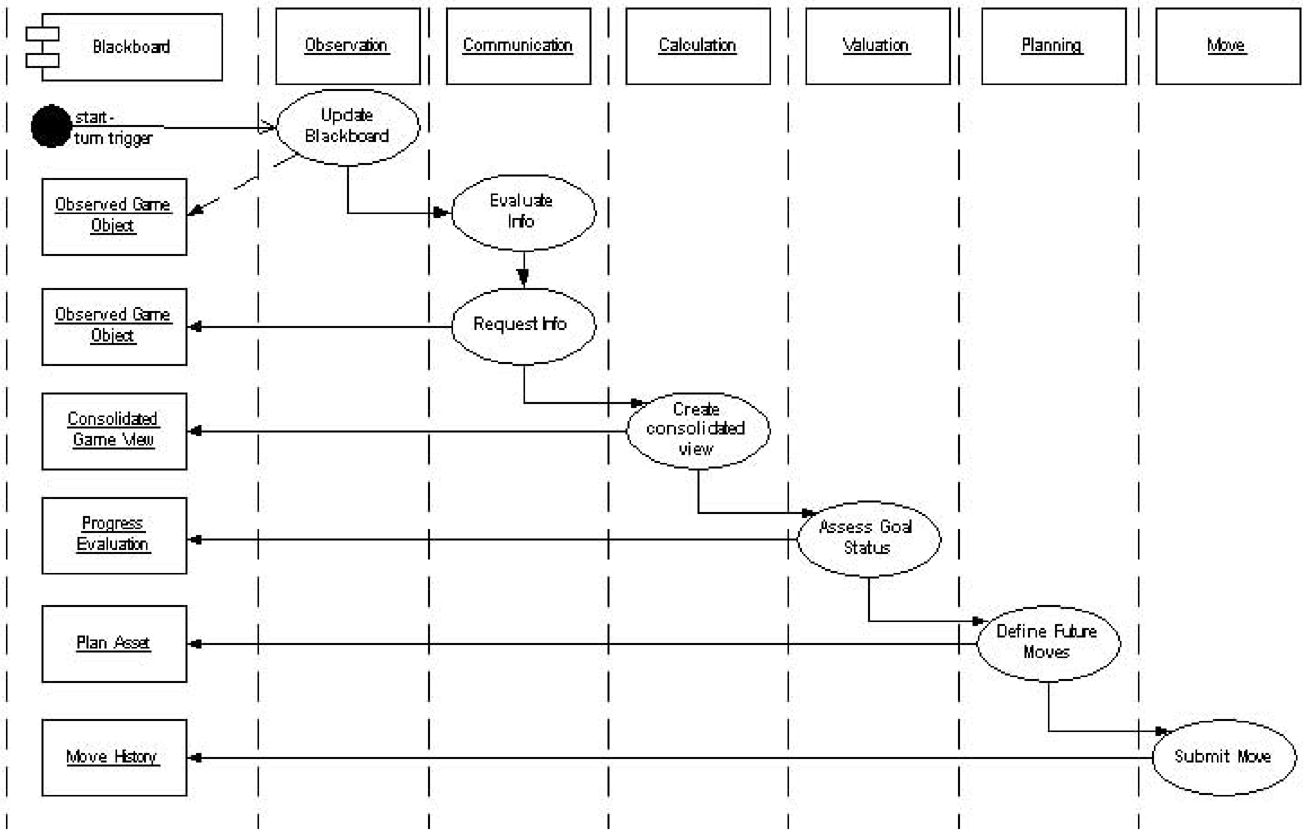


Figure 9: Actor Plan Mode UML Activity Diagram

Move Resolution

The following diagram and its subsequent steps represent the interactions between the components involved in the process of resolving an Actor's requested move.

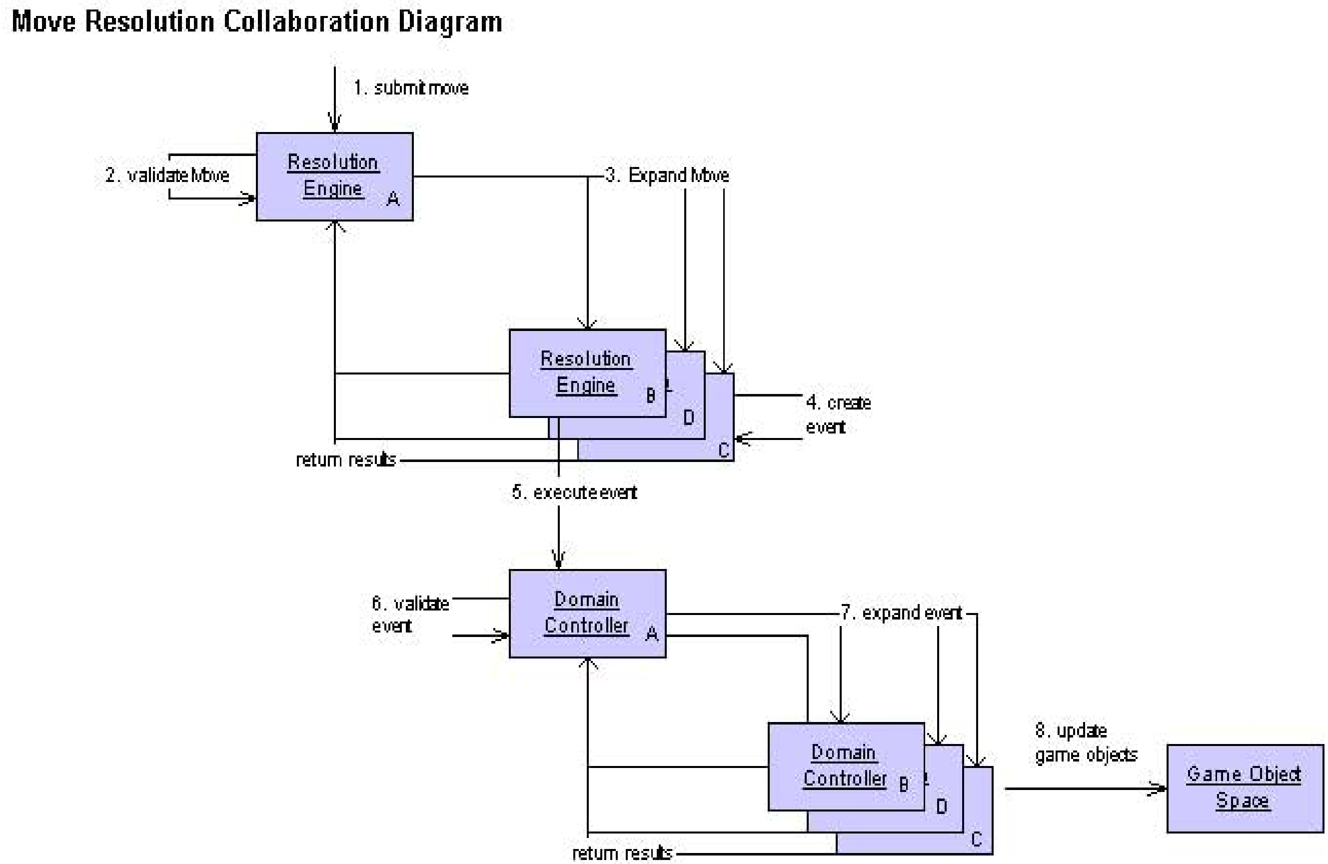


Figure 10: Move Resolution UML Collaboration Diagram

Move Resolution Steps:

1. Perspective submits a new move on behalf of its Actor.
2. A top level Resolution Engine validates the move in terms of the defined rules of the scenario.
3. For each move, the Resolution Engine that handles it may expand the move into additional sub-moves, for processing by other Resolution Engines.
4. When a move has been decomposed to its smallest sub-move (no further sub-moves exist), the Resolution Engine that handles that sub-move will create an event that must occur in the Game Object Space to manifest the result of the sub-move.
5. The Resolution Engine then submits this event to the Domain Controllers.
6. A Domain Controller responsible for handling that particular type of event will validate the event in terms of the constraints of the domain (such as no two physical objects can co-exist in the same spatial location).
7. For each event, a Domain Controller that handles it may expand the even into additional sub-events, for processing by other Domain Controllers.
8. When an event has been decomposed to its smallest sub-event (no further sub-event exist), the Domain Controller that handles that sub-event will update the Game Object Space accordingly (such as moving an objects location).
9.

Assess the Status of Game Theory and IT as they apply to GAMBIT

The information summarized in this section was compiled from a survey of printed research, discussions with various researchers in the relevant fields, and concept/project design work motivated by the research and discussions in the context of the GAMBIT goals.

Quick Review of the GAMBIT Environment

All of the self-interest networks that appear relevant to GAMBIT scenarios fit within one of two classes or a combination of these classes. The first class can be called Coalition Games, in which the formation and stability of one or more coalitions among the actors is the focus of the scenario. The second class can be called Coordination Games, in which the allocation of tasks and resources within an existing and structured group is the focus of the scenario.

Coalition Games are normally associated with terms like competition and collusion; whereas, Coordination Games are normally associated with the term cooperation. But some classic economic examples illustrate that the class label often is a matter of the perspective of interest:

> Different firms are considered to be in competition or, perhaps, in collusion (Coalition Games). The hierarchical organizations that constitute a firm are often thought of as peopled by a *team* of actors in cooperation (Coordination Games). Yet markets working within a hierarchy of law coordinate the actions (and interests) of firms (Coalition Games and Coordination Games), and there can often be substantial dysfunction among the constituents of a firm (Coalition Games not Coordination Games).

Replace “firm” in the examples above with “army” or “nation”, then consider that in this era of asymmetric threats some “armies” may be terrorist networks, and the relevance of Coalition Games and Coordination Games to GAMBIT becomes clear.

There are four qualities of information technology (IT) that would need to be incorporated in a worthwhile GAMBIT toolset: distributed operation, scenario populations that are a mix of human actors and software agents, objective resolutions, and a user composeable architecture. The purpose of a GAMBIT toolset would be to make planners, analysts, and practitioners more adept at strategic reasoning in the real world scenarios they face. Such scenarios are quite varied and often involve large numbers of actors. Scale economies dictate distributed operation (rather than collocation) and the ability to *fill out* a scenario with software agents (rather than requiring potentially hundreds of human actors). The value of testing scenario variations requires that the resolutions determined for a scenario be determined relative to objective rules, rather than subjective judgments. Due to the huge variability in the scenarios of interest to the many different potential toolset customers, scope economies dictate that users be able to compose their own scenarios with the toolset.

The many Branches of Game Theory

Game theory found a home in micro-economics as a means of describing the decision processes and actions of self-interested actors who are engaged in types of commerce where, essentially, there are two few actors for any one actor to hide; e.g., negotiating deals, managing contracted efforts, the formation and maintenance of collusive arrangements, and various types of auctions. Ultimately, all of micro-economics became infused with game theory – the fit was natural and, as the first discipline to fall under the sway of game theory, economics became tightly associated with game theory.

But the first thought experiments that helped to formalize the rigorous treatment of strategic reasoning were not classic economic problems; rather, they were behavioral. The classic game of the Prisoners' Dilemma is a behavioral analysis. The outcome of a Prisoners' Dilemma thought experiment depends on the environment in which the human actors are placed. Such an environment can contain economic and non-economic constructs.

Economics is a subset of human behavior. If game theory proved so applicable to the formal discipline studying one subset of human behavior, then it was only a matter of time before game theory infused political science, sociology, anthropology, and cultural geography. The realization that self-interest is not uniquely human has brought game theory to biology. The fact that mechanical systems interact with other mechanical systems that may be directed by humans with differing intentions has brought game theory to control theory/systems engineering. The fact that computational systems interact with other computational systems that may be directed by humans with differing intentions has brought game theory to computer science, especially where computer science overlaps with economics; e.g., Internet-facilitated commerce.

These disciplines have employed and advanced game theory relative to their own needs and interests. Common threads do not automatically produce a unified theory and certainly do not provide a direct approach to producing a strategic reasoning toolset. However, common threads suggest the potential for common interest in a joint effort that would further the value each discipline finds in game theory. Thus, a focal development leading to a generally useful toolset has the potential for weaving the threads into something quite strong.

What follows in this section is a quick review of game theory in various disciplines. The review takes advantage of the Coalition Games/Coordination Games construct introduced earlier, and is thus subject to the weaknesses of such a broad-brush categorization.

Game Theory in Economics[8]

The use in economics of the class of games/scenarios we are referring to as Coalitions Games is perhaps best exemplified by studies of competition and collusion among firms. The parameters

[8] The literature here is huge and so no attempt will be made to single out any small set of researchers. A reader interested in more detail should refer to any number of graduate-level economics texts.

of these studies deal with the number of firms *in the market*, the *market share* of each, the degree of substitutability or complementarity among the firms' products, the ability of each firm to monitor the actions of each other firm, the ability of a firm to evade regulatory oversight, and whether the scenario is one-shot or long-term. There are substantial parallels between these economic Coalition Games scenarios and the simple behavioral Coalition Games of classic game theory, but all the conditions and terms are, or course, purely economic. The equilibrium solution concept in all these treatments is, generally, some version of best response (Nash).

The use in economics of the class of games/scenarios we are referring to as Coordination Games is best exemplified by mechanism design, also called implementation theory. The intent of mechanism design is to coordinate the individual actions of a group of self-interested actors toward maximizing the attainment of some goal. The classic case is to design a market (or auction) process that matches buyers and sellers such that the greatest group gain in value results. Maximizing group gain is the central purpose of social welfare theory and mechanism design can be thought of as game theory applied to problems of social welfare. Mechanisms are designed such that self-interested actors voluntarily choose strategies that result in the maximization of social welfare, including, implicitly, the choice of accepting the imposition of the mechanism. The equilibrium solution concept is maximization of a social welfare function subject to voluntary and incentive compatible participation – from the actor's point of view, this is best response given the mechanism.

Principal/agent studies are a subset of mechanism design, but with a quality that makes them worth singling out here – P/A studies combine elements of Coalition Games and Coordination Games. In the scenarios considered, the principal is in a position of authority over the agent or agents; e.g., firm/employee, prime/subcontractor. The principal's goal is to maximize its welfare function, which is just its utility (no grand social motive here). An agent's goal is to maximize its utility. Both types of actors pursue their goals within the environment, physical and informational, in which they find themselves; e.g, if deceit and deception can enhance their goal, then an actor uses deceit and deception. When there are multiple agents, it is often the case that they can have the incentive to collude against the interests of the principal – this collusion is a Coalition Game. The scenario between the principal and the agents is a Coordination Game.

Game Theory in Computer Science

Theoretical computer science deals heavily with issues of computational complexity and information processing, often in distributed networks (e.g., multi-processor, LAN, Internet). A classic network model in computer science is a graph of nodes interconnected by arcs. In traditional computer science, the nodes in a network are either compliant or not, and non-compliance results from internal-node malfunction or overloading due to network routing to/through that node. With the advent of the Internet, computer science has begun to deal with concerns that result from the potential for strategic behavior at each node. Algorithmic mechanism design, distributed algorithmic mechanism design, and computational mechanism design are all approaches to deal with computer networks as Coordination Game scenarios. The principle insight behind these approaches is that incentives are as important as computational complexity – incentive compatibility and computational complexity merged into computational compatibility.[9]

The graph-theoretic models of information flow and computation used in computer science have recently been applied to classic Coalition Games scenarios involving many actors, rather than the classic two or three. Traditional game theory would consider strategy spaces in which every actor must consider the potential actions of every other actor. This translates to a graph in which every node has an arc to every other node – the computational complexity of such a structure becomes intractable with realistically small numbers of actors. Computer scientists have begun to examine structuring these graphs so that each node is linked to its neighborhood and then applying learning theory approaches so that a node may update its definition of its neighborhood in dynamic games. This approach offers the prospect of practical compact representations and analyses of multi-actor strategic scenarios, though it is currently limited to certain Coalition Game scenarios.[10]

A separate branch of theoretical computer science, something of a hybrid with classic game theory, has begun to pose the question, "How would classic games, like the Prisoners' Dilemma, be played out on a dispersed information network?" This line of research is quite new and has been labeled Distributed Games.[11] As GAMBIT is meant for the Information Age, this line of thought may well offer valuable input.

Game Theory in the *Living* Sciences

Living, in this context, means that the object of study is an actor that responds to an environment that is very unstructured compared to the rarified models of economics and computer science. The environmental model of a living science is populated by other actors, many of whom are not in the same tribe, society, or even species let alone in the same grouped economy or information network. In this context, living sciences include sociology, biology, and even robotics (as extensions of living actors that must respond to the actions of other actors).[12] These are applications of game theory that are more *real world*.

Living science scenarios are almost wholly Coalition Games or, more to the point, they are wholly not Coordination Games. To be blunt, survival of the fittest has nothing to do with a social welfare function. A social compact, such as an established rule of law, is not a presumption in these scenarios and so there is little implicit structure from which one can assume the imposition of a coordination mechanism. Any group cohesion, such as intra-species loyalty, is a consequence of internal coalition formation not engineered coordination.

---

[9] Representative work includes research by Noam Nisan and Joan Feigenbaum.
[10] Representative work includes research by Michael Kearns and Pierfrancesco La Mura.
[11] See Dov Monderer and Moshe Tennenholtz, 1997.
[12] For an overview of evolutionary game theory, see Herbert Gintis' "Game Theory Evolving"; for game theory applied in sociology, see the work of Colin Camerer, and for examples of game theory in robotic/control systems, see the work of Richard Murray.

Living science game theory treatments are much closer to the behavioral thought experiments of classic game theory than to the mechanism design treatments of economics and computer science. And with respect to classic games, such as the Prisoners' Dilemma and Divide a Dollar, there is evidence that they may provide insight that is not only fresh, but also fundamentally valuable.

In classic game theory, actors use best response and the capability of fully rational reasoning to decide on their moves. This produces some results for classic games that are counter-intuitive and are only made right through the imposition of extreme requirements, like infinite repetition. Living science studies of these same classic games have revealed the significance of learning, pattern behavior, and other reasoning processes. Intuitive results have replaced counter-intuitive ones through realistic and well-defined solution concepts such as bounded rationality, which can account for complex, multi-actor environments and the limited resources that an actor can apply to decision-making. The possible connection to graph-theoretic compact game forms and the merits of expanding these solution concepts to Coordination Game scenarios are apparent.

The Game Theory Status Quo and a Direction Forward

Reality is a mixed structure of Coalition Games and Coordination Games. Many fields of study recognize this: cohesive societies (Coordination Games) engaged in cooperation, competition, or conflict with other societies (Coalition Games); firms (Coordination Games) competing or colluding with other firms (Coalition Games); and families (Coordination Games) jostling for position within a tribe (Coalition Games). And although analysis may focus within Coalition Game or Coordination Game structures, interplay between Coalition Games and Coordination Games is recognized (e.g., treason, mergers, and intermarriage).

With the limited exception of principal/agent theory, there is no game theoretic treatment of mixed and evolving (dynamic) Coalition Game/Coordination Game structures. And such a theory, including appropriate equilibrium solution concepts, is what is needed to underpin and guide the development of a strategic reasoning toolset.

For the purposes of having a name, Dynamic Hierarchical Gaming (DHG) is a decent moniker for a hybrid of Coalition and Coordination Games. As a solution concept, Bounded and Updated Best Response (BUBR) sounds good for the actors and there need not be one for the whole system, only the separate social welfare goals of each sufficiently cohesive group of actors.

From economics, computer science, and the living sciences, the groundwork for DHG exists. What is needed is a process through which all the pieces can be brought together and molded.

What IT is ready to offer

The intuition behind GAMBIT is that information technology can enable a scenario simulator rooted in game theory. The Internet obviously provides the capability to network together a widely distributed group of scenario participants. And the processing power of the common personal computer would seem to offer the capability for vibrant simulations. But whereas high data rate connectivity and graphics cards are the stuff of networked first-person shooter games, these capabilities are not sufficient (and may not even to necessary) for a strategic reasoning toolset. Necessary, and a good-bit less-developed, are (i) software agents, capable of dynamic learning and recursive assessment, (ii) a scenario composition language, (iii) the ability to calculate objective resolutions from a group of moves, and (iv) an IT-based process for controlled group behavioral trials.

Distributed Operations and Processing

Within the context of this deliverable, it is not necessary to review or substantiate what is obvious; namely, current technologies provide the ability to network together group operations involving substantial processing at distributed nodes. Further, it is reasonable to assume that this ability will continue to improve independent and in advance of the needs of any strategic reasoning toolset. As will be clear, processing power and distributed operations are key enablers for software agents and objective resolution determination.

Software Agents

To be practical and useable, a strategic reasoning toolset that supports dynamic hierarchical games (DHG) must be able to incorporate software agents as the actors at any of the player positions in a game/scenario. For example, a DHG with three top-level players (countries) engaged in coalition formation/conflict may require two nested agent groups beneath each top-level (e.g., combat units and logistics). Each agent group would probably contain at least three actors. Thus, in a relatively simple DHG there would be twenty-one actors. Assembling twenty-one people is not a trivial task, and assembling twenty-one people who you want to train in the specific actor roles is harder still. The capability to populate a scenario with a mix of software agents and human actors is fundamental to the toolset being worthwhile.

Recall the GAMBIT goal:

> The goal of GAMBIT is a strategic reasoning toolset from which numerous scenarios can be scripted and *gamed* by planners and practitioners.

It is critical that the human actors who participate in a scenario produced from the envisioned toolset improve their ability to handle real-world situations that involve the strategic reasoning of other humans. But practicality requires that many of the actors in a GAMBIT scenario be software. Thus, the software agents must behave like humans, which in this context means that the software agents must emulate human strategic reasoning.

A software agent that can play the Nash equilibrium strategy in any one of several versions of the Prisoners' Dilemma it not what is needed. Rather, the desired software agent must be able to assess which actors are important for its interests (its neighborhood) and what its choices of action are given the rules and other constraints of its position. Such an agent is more like the actor in Living Science game theoretic treatments than the arch-rational actor in classic

Economics treatments. Such an agent would likely have to allocate limited reasoning resources (imposed by design rather than by any limits in processing power) among learning and recursive assessment of the likely actions of others. In dynamic, multi-turn games, pattern recognition and updating assumptions about its neighborhood would be part of the software agent's behavior.

In this GAMBIT context, a good software agent is not measured by the frequency of its winning, but by the infrequency with which human actors distinguish it as software.[13]

Software agents are being produced by commercial and academic (research) IT. The commercial software agents that implement the self-interested goals of their human masters are mostly bidding agents participating in some sort of structured and well-defined auction process. Other commercial software agents function as non-selfish automata in what amounts to a distributed intelligent system. Academic work in software agents has taken self-interested behavior further into models of biological systems and robotics games, such as a robot actor version of Capture the Flag. Pattern recognition, learning, and limited reasoning resources (bounded rationality) are all coming into the mix. However, all of these agents are scenario specific and none of the scenarios is a DHG.

Scenario Composition Language

DHG represents a huge class of strategic actor arrangements with myriad choices of physical (non-strategic) constraints limiting what actions can be taken. A strategic reasoning simulation toolset that addresses a significant subset of the DHG class must allow a user to compose the DHG instance of interest. Essentially, a user must be given a menu of allowable scenario architectures and enabled features per architecture. This would amount to a high-order, object-oriented language for the toolset. Such languages are routine it modern IT. But given that no instance of a DHG has yet to be thoroughly modeled, let alone implemented as a distributed software application, it is premature to design the scenario composition language. When the root task is better understood, IT should be able to supply the composition language.

Calculating Objective Resolutions

One of the great strengths of game theory is that given a well-defined scenario, if game-theoretic analysis produces a result, that result is objective, repeatable, and testable. Strategic reasoning simulations for the purposes of planning and training should have the quality of objective resolution determination. Traditional war gaming is fraught with subjective resolution judgments. A strategic reasoning toolset that is truly based in game theory should not suffer from subjective resolutions.

Whether player moves are synchronous or asynchronous, they will impact numerous players. Thus, the resolution of moves will be a combinatorial computational problem. The

[13] This should correlate with the frequency of winning, but the emulation of human strategic reasoning is what addresses the GAMBIT goal.

computational complexity of combinatorial problems is well studied. Over the past decade, the commercial practice of implementing combinatorial auctions and markets has taken these studies into practice. IT has provided both the computational power and the algorithmic dexterity to deal practically with increasingly complex combinatorial problems.

The type of combinatorial problems currently serviced commercially are purely economic; e.g., package auctions and interlinked markets. This is a subset of the problems that will arise in a DHG scenario. As with a scenario composition language, it is probably reasonable to assume that once the DHG class of problems is better understood, IT will be able to address the associated combinatorial problems required to calculate objective resolutions. An argument behind this assumption is that understanding the DHG class will require the development of compact game forms that allow for reduced form analysis that will decrease the computational complexity of the problems. This decreased computational complexity will directly facilitate applied solutions of the combinatorial problems needed for objective resolutions.

The IT Status Quo and a Direction Forward

The major applied IT development required involves software agents that can emulate human strategic reasoning at any actor node in a DHG scenario. Progress here, and with all other IT inputs to a strategic reasoning toolset, would be greatly enabled by a focused effort to rigorously establish a DHG case and then generalize from that establishment.

HIGHLIGHT GAMBIT BY DEMONSTRATING AN HISTORICAL SCENARIO

The game The Enemy of my Enemy is my Friend, illustrated below, contains coalition formation (commanders) and hierarchical group coordination (colors). Hierarchical groups can be thought of as coalitions that formed in the past and are stable. Such stability is tested when the lower members of groups can be influenced from beyond the group. Played across multiple turns, this game has the structure of a GAMBIT scenario – Dynamic Hierarchical Gaming (DHG). To provide a meaningful demonstration of what might be possible with a GAMBIT simulation toolset, a scenario was identified that had significant elements of DHG and was robustly historical. Robustly historical means that the scenario has been played out in real life under a variety strategic environments for which data (player motivations, actions, and outcomes) are known. The scenario selected was the management of moral hazard in the group research and development (R&D) task of producing a payload of science instruments for a planetary mission.

Managing group R&D is dynamic in that there are at least two sequential stages, research and development. Managing group R&D is hierarchical in that centrally owned resources are allocated to multiple subordinates so that those subordinates can produce components of a whole, in this case instruments that comprise a payload. And managing group R&D is strategic gaming in that the interests of the subordinates do not completely overlap with the interests of the manager and the uncertainty of research causes a Fog of War that can potentially be exploited by the subordinates to enhance their interests (moral hazard). However, as illustrated below and demonstrated in the work reported, managing group R&D lacks the coalition formation/evolution elements present in a full GAMBIT scenario.

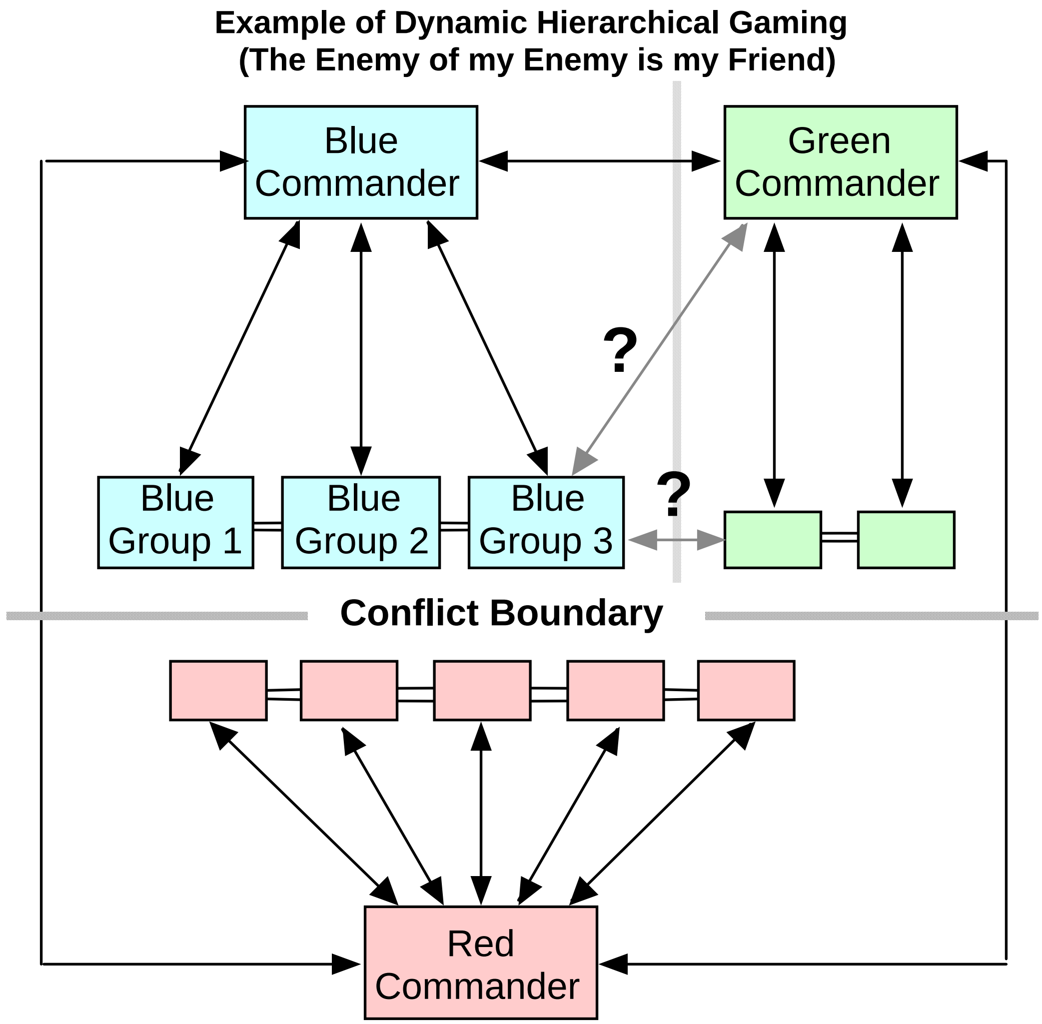


Figure 11: Example of Dynamic Hierarchical Gaming

Managing group R&D is a robustly historical scenario because of the Cassini instance. In traditional R&D management of interplanetary missions by the Jet Propulsion Laboratory (JPL), funds were the loose variable when allocating mass, power, and funding allowances among scientists who were developing instruments. The standard mission overran its science budget considerably. Occasionally, an instrument was removed from development when the loose funds constraint tightened, but, for the most part, scientists were given supplemental allocations when they announced bad research luck and were allowed to quietly benefit from good luck. For Cassini, however, the cancellation of its sister mission due to a cost overrun sent a strong message to JPL: Cassini faced a hard budget limit. A new R&D management approach was needed.

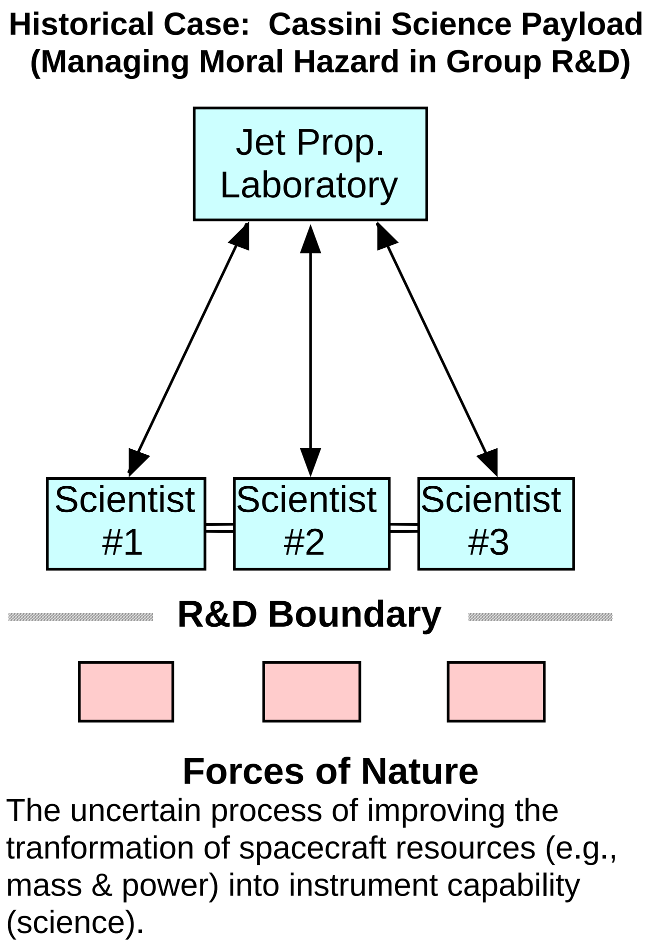


Figure 12: The Cassini Science Payload: Managing Moral Hazard in Group R&D

For Cassini, JPL replaced traditional R&D management with decentralized control. Prior to R&D, the scientists were given title to mass, power, and funding allocations. These allocations summed up to all the available resources – JPL did not hold back any margin. The scientists were given the freedom to trade resources among themselves in the hope that each scientist might use the effects of good luck in one aspect of research to compensate for the effects of bad luck in some other aspect. Cassini launched on time, on budget, and with its full complement of instruments, making Cassini an outlier in the history of large planetary missions.

Model used for the Historical GAMBIT-like Scenario

We modeled the group R&D problem as illustrated below. Congress is assumed to allocate a fixed, predetermined budget in two sequential pieces; B_R for the research phase and B_D for the development phase. JPL has at its disposal a spacecraft with a fixed payload capacity supporting a total instrument mass of M and a total instrument power supply of P. JPL allocates research budgets, b_R, to each of three scientists who then make research investment decisions regarding technologies for mass use, μ, and power use, π. Outcomes of this research are then reported by the scientist to JPL (the reports need not be truthful). Additional funds along with a mass and power allocation are then allocated to each scientist. The instruments are produced, delivered, and the mission is launched.

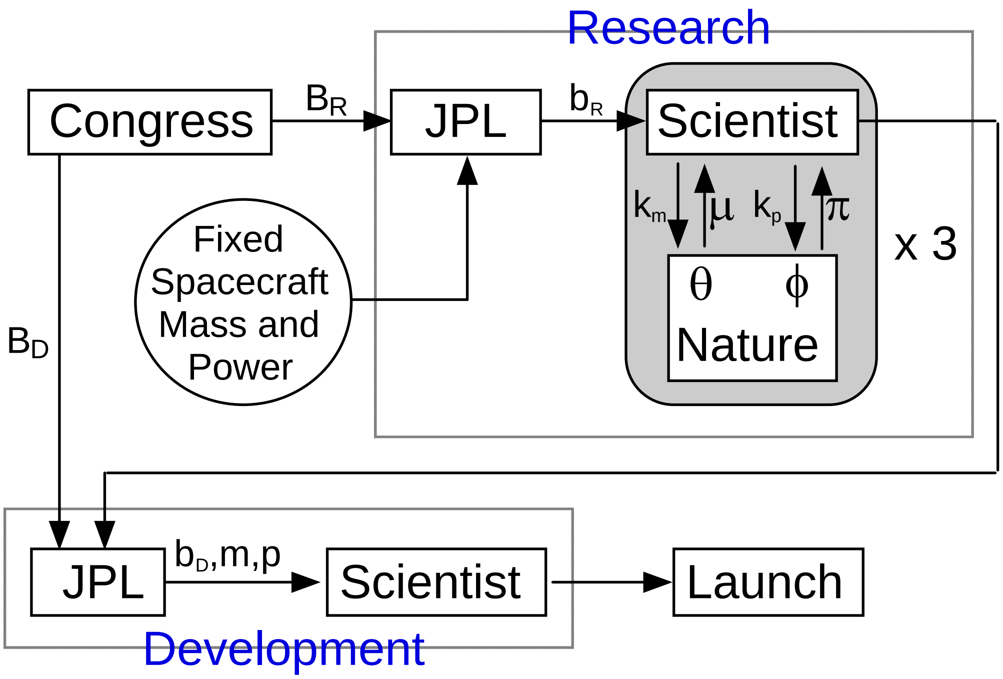


Figure 13: Model of Group R&D on a Planetary Mission

For this model, JPL is assumed to value some balance of science capability from the instruments. Accordingly, for the demonstrations of this scenario, the following functional form was used for JPL's value:

$$V_{JPL} = \min\{S_1S_2, S_1S_3, S_2S_3\} + \beta(\text{Budget} - \text{Expenditure})$$

Each scientist can be assumed to value only the capability of his own instrument plus residual funds, if any. Accordingly, the following functional form was used for each of the three scientist's values:

$$V_{Sc} = f(S, \$) = k_D\left[\mu(k_m,\theta)m + \pi(k_p,\phi)p\right] - \gamma k_D^2 + \varepsilon(\text{residual } \$)$$

The mass- and power-use technologies are the crux of the Research Phase. Money is invested in technology research in the hope of increasing capability above off-the-shelf levels, denoted μ_0 and π_0, both of which are set equal to 1. (This is where Nature will roll die weighted by how much money is invested.) For the demonstration, this process was simplified by assuming that there are only two possible outcomes to any research effort, off-the-shelf and enhanced; e.g., μ_0 and $\mu_0 + \Delta\mu$. The probability of technology states was affected by investment as follows:

$$\mathrm{Prob}[\Delta\mu] = \left[\frac{k_m - \underline{k}_m}{\bar{k}_m - \underline{k}_m}\right]^{\xi} \quad \& \quad \mathrm{Prob}[\Delta\pi] = \left[\frac{k_p - \underline{k}_p}{\bar{k}_p - \underline{k}_p}\right]^{\zeta}$$

$$\xi \text{ and } \zeta \in (0,1]$$

where $\underline{k}_m$ is the minimum investment that must be made to have a chance at enhanced technology, $\bar{k}_m$ makes enhancement a *sure thing*, k_m is the investment decision made, which must fall between $\underline{k}_m$ and $\bar{k}_m$, and ξ and ζ determine the rate at which investment affects the probability of a successful research outcome. For the demonstration, ξ and ζ were both set at 0.5.

Funds allocated for development, k_D, are spent overwhelming on specialized labor. Given the short supply of specialized labor, it is reasonable for there to be sharply decreasing returns to expenditures of k_D above a certain point; thus, the quadratic term in the Scientist valuation function.

Four R&D management processes were modeled to compare best case, historical group R&D management, and Cassini. These four, referred to as Cases, are:

Case #1: Monolithic -- Scientist Robots (*first-best* solution). The problem w/o moral hazard -- the scientists are not self-interested.

Case #2: Agency with Full Information. JPL can costlessly observe Research outcomes. Scientists know that Research outcomes affect Development allocations; therefore, they play a maxmin strategy. This case establishes the upper bound for JPL under traditional management of self-interested agents.

Case #3: Agency with costly monitoring of Research (more realistic than case #2).

Case #4: Agency with property rights and trading Cassini). JPL gives each scientist (b_R, b_D, m, p) before Research. Scientists are allowed to trade resources.

See Appendix A for a more thorough description of the modeling and computational processes used for the demonstration.

Software Architecture for Historical Demonstration

Any eventual GAMBIT toolset will rely on a distributed agent architecture. For the demonstration reported on here, the Cougaar architecture was used. The Cougaar architecture is reported on more fully in the previous section of this report entitled “Characterize GAMBIT Scenarios and the IT to service these”. All of the equipment used in demonstrating this GAMBIT-like scenario was computer hardware owned and operated by either Net Exchange or Cougaar Software.

Scenario Runs Performed

The R&D scenario was run for 378 parameter setting, 126 each for management cases #1, #2, and #4. The only parameters varied during these 378 runs were the overall budget, B_R and B_D, and the technology potentials of the scientists. Seven funding profiles were used for each 126 runs for each management case; these are listed below.

Table 3: Funding Profiles for Cassini R&D Scenario Runs

	Funding Profiles for Scenario Runs						
Research, B_R	2	2	3	3	4	4	5
Development, B_D	3	4	3	4	3	4	5
Total Funding	5	6	6	7	7	8	8

For each of these seven funding profiles, scenario runs were conducted with six different average technology potentials among the three scientists; 1.0, 1.5, 2.0, 2.5, 3.0, and 3.5. The technology potential of a scientist is the mass use technology, μ, or power use technology, π, that he can attain through investing in and succeeding at research. Thus, if the mean of a run is 3.0, the average of the six potentials among the scientists is 3.0.

For each of these six average potential settings, three distributes of scientist technology potentials were run – these were named “mean”, “± 10%”, and “± 20%”. In the mean distribution, all three scientists had the same research potential in both mass and power use technology. In the ± 10% case, one scientist had mean potential in both, a second scientist had 10% greater potential in mass use technology and 10% less potential in power technology, and the third scientist had the 10% applied in the other direction. The ± 20% distribution was defined as with the ± 10% distribution.

The graph on the next page illustrates JPL’s expected value for the 56 runs associated with the funding profile (3, 3) for management cases #1, #2, and #4. Several observations are apparent and amount to a concise analysis of the demonstrated scenario:

1. Variability among the scientists’ technology potential tends to increase JPL’s expected value under all management cases; i.e., for each management case and for any average technology potential among the three scientists (the horizontal axis), JPL’s expected value (the vertical axis) tends to be least for “mean”, greater for “± 10%”, and greatest for “± 20%”. This is to be expected as the variability in potential makes it easier to decide in what to invest.

2. For each distribution of technology potential, management case #1 is the upper bound for JPL expected value. This is as anticipated since case #1 is the model without agency and represents a “first best” solution.

3. It is possible for case #2 to produce lower JPL expected value than case #1, even though there is full information in case #2. The maxmin strategy played by each scientist results in technology investment decisions that are different than the decisions made in case #1. There is nothing in the model used that allows JPL to punish a scientist for making an investment decision that is different than the decision JPL would have made. Thus, full information eliminates the possibility of lying, it does not eliminate all influence from moral hazard.

4. It is possible for case #2 to produce lower JPL expected value than case #4. This is a very strong result. Recall that in case #4 JPL cedes all decision power prior to any research being conducted. The moral hazard that motivates the maxmin strategy of case #2 can be so strong that JPL is better off giving away the resources and all decision authority over their use.

5. Case #3 is not illustrated, but the effect of costly monitoring can be seen by simply lowering the results for case #2. As case #4 does not require any monitoring, even a mild ability on the part of the scientists to conceal the true results of research from JPL would cause the traditional R&D management process (case #3) to be inferior to the decentralized process (case #4).

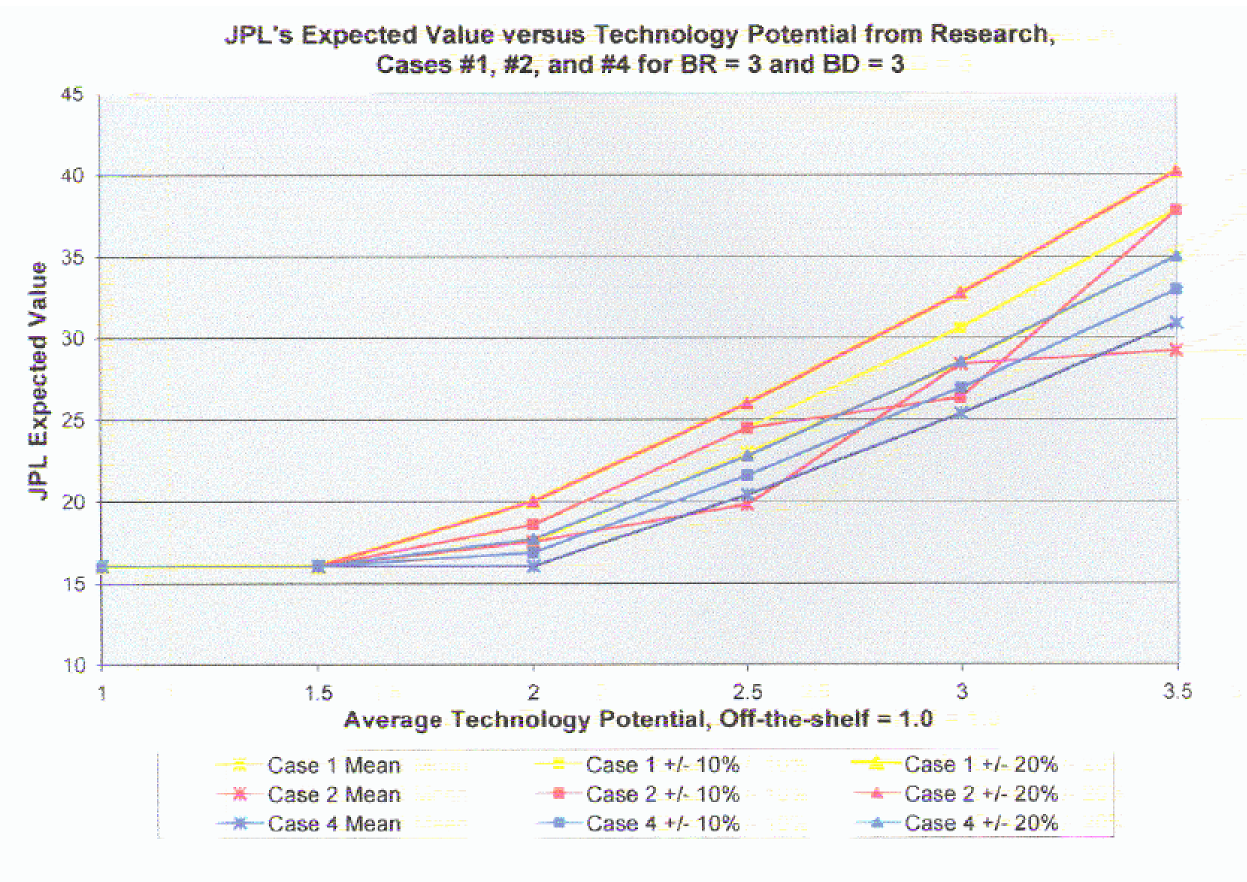


Figure 14: JPL's Expected Value versus Technology Potential from Research

The data for all 378 parameter settings and several additional graphs are included in Appendix B. Further analysis and commentary is included in Appendix C.

The results from the R&D model examined here are completely in accord with the observed performance of the Cassini R&D management process as compared to historic R&D management for JPL planetary missions. Under this pre-GAMBIT contract, the goal of the technology demonstration was to replicate a ground truth instance that had a structure sufficiently analogous to GAMBIT and for which game-theoretic treatment had been shown as relevant and beneficial. This goal has been successfully accomplished.

CONCLUSIONS

A principal goal of this pre-GAMBIT study was to characterize the current uses of game theory and advise DARPA on the readiness of the game theory community and IT technologists to design and product a GAMBIT strategic reasoning toolset. This study concludes that there is potential promise for such a toolset, but that there is a practical need for a focal development task to bring together the insights of disparate practitioners around a tractable task.

The basic situation described so far involves (i) many branches of research using similar thought constructs, and (ii) a formidable array of existing technology that has been applied only partially to the various research endeavors. This is not a new situation. And the solution to this situation is straightforward – a focal development that provides each research community with substantial value and can be built by applying existing technology.

Engage theorists, technologists, and empiricists in the design and operation of a utility from which each can benefit. The combined task engenders focus while the requirement that all garner value results in open use and flexible enhancement policies. This solution has worked well for physical science, why not behavior science? If this is good for particle physics, why not for strategic reasoning? If CERN, why not GAMBIT?

RECOMMENDATIONS: THE DIPLOMACY TEST UTILITY

Net Exchange concluded its efforts under this contract with the observation that GAMBIT is possible and promising; however, it cannot be attained in one development leap from the current status quo in either game theory or IT. Net Exchange recommends an interim step – the use of an established strategic gaming platform, the deceptively simple game of *Diplomacy*. By adding a bit of formal structure to the on-line implementation of this game, the Diplomacy Test Utility (DTU) can focus the various strands of extant research while benefiting from the participation of a large and well-trained user base. Incremental enhancement, made robust through an open architecture and verified by repeated human trials, will lead to an instance of a full strategic simulator. Generalization from this instance would result in a realized GAMBIT toolset.

Introduction to Diplomacy

In the mid-1950s, the board game *Diplomacy* was introduced.[14] It is a game for seven players, each of whom commands one of the major pre-WWI European powers.[15] Each player begins with three pieces.[16] Players gain pieces when they gain valuable territory. When one of them gets eighteen pieces, he or she wins. The rules of movement and conflict resolution are simple and objective. *Diplomacy* can take days to play even though only about 20 moves are required to reach a conclusion. This is a game of negotiation, promises made and promises broken, a game of dynamic coalition formation.

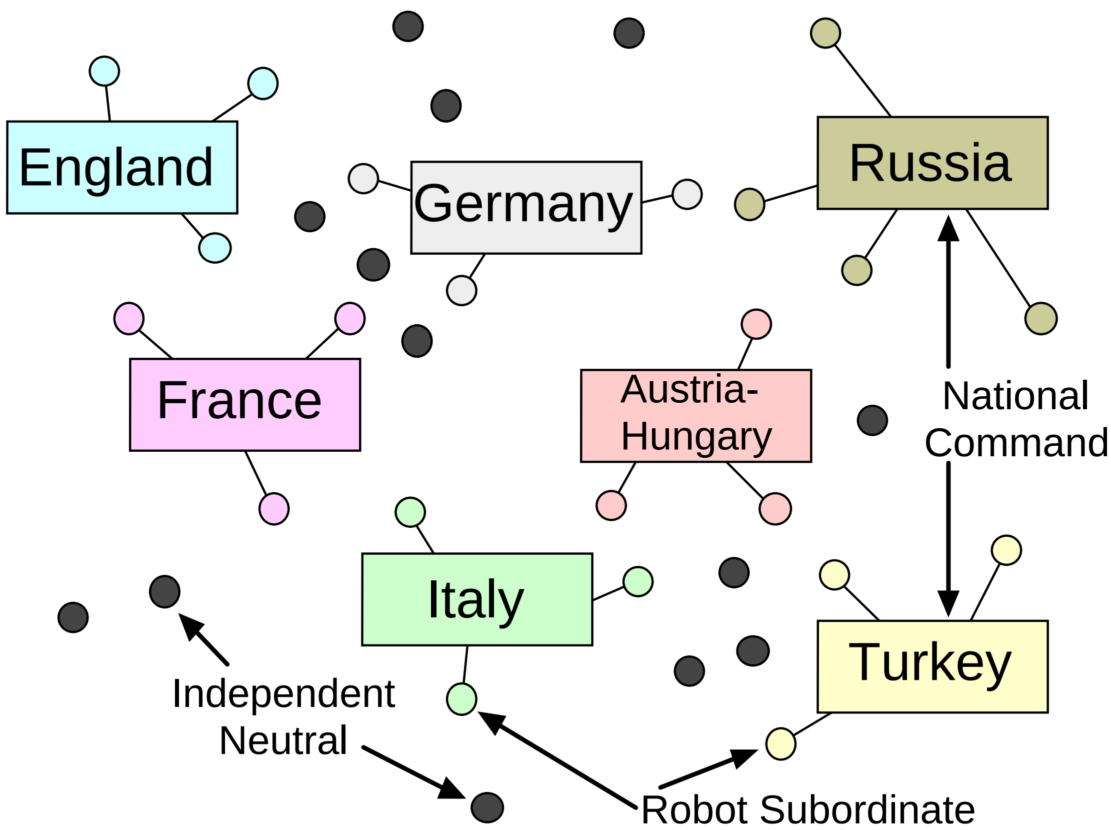


Figure 15: The Distribution of Players and Objectives in Diplomacy

Diplomacy has developed a substantial following with a large literature on strategies for the various players at different points in the game.[17] *Diplomacy* was regularly played by mail – the small number of pieces and objective move resolution rules made *Diplomacy* ideal for this. Players would mail negotiation letters to each other during a prescribed negotiation phase and then each would mail a letter with their actual move for the turn to a game master (often a

[14] The original name was *Realpolitik*, but the euphemism of *Diplomacy* was a better marketing choice.
[15] Austria-Hungary, England, France, Germany, Italy, Russia, and Turkey
[16] Except for Russia, which has four.
[17] See http://www.diplomacy-archive.com/, http://www.diplom.org/ and similar for information on *Diplomacy*, past and current.

magazine published for this purpose). Games would proceed at a two-move-per month pace for approximately a year. *Diplomacy* was a network distributed dynamic Coalition Game long before personal computers and the Internet.

With the advent of personal computers and the Internet, *Diplomacy* has gone on-line and offers the basis on which to build the focal utility to advance theory and practice toward a GAMBIT toolset. A large number of human strategic reasoners, well-versed in the rules and practices of the game and willing to participate for enjoyment, make feasible the repetition and incremental improvement that is necessary for such a utility.

Stage One of the Diplomacy Test Utility

Imagine on-line *Diplomacy* with a mix of humans and software agents whose moves are resolved by an objective resolution engine. If the humans cannot figure out which nations are run by software agents, then that is an indication of a substantial advance in Coalition Game strategic reasoning. An objective resolution engine has already been implemented for the current on-line play. However, a software agent does not exist that is up to the task.

This first stage of the DTU requires four developments by the central DTU project:

1. A structured negotiation language and protocol is necessary to replace the casual human oriented system currently in use. This would still operate asynchronously, but the structure would put humans and software agents on an even plane linguistically. Without leveling language ability, human actors would still be able to discern software agents independent of the strategic reasoning capability of the software agents. And without imposing a structured language protocol, assessing the intent of negotiation would be more difficult, complicating analysis.

2. Operations capability to run numerous *Diplomacy* games simultaneously. DTU Central must attract and train (in the negotiation language) *Diplomacy* players. The skill level of players must be assessed and updated. All games must be archived, including all negotiations and moves. The assignment of human players to nations must be conducted so that the trials are unbiased toward certain game configurations. Human players would have limited interaction with the Utility, but they would be given the incentive to vote on which actors in a game were software agents.

3. An API for software agents and an approval process for accepting/rejecting proposed agents.

4. An API and access policy to the game data archive.

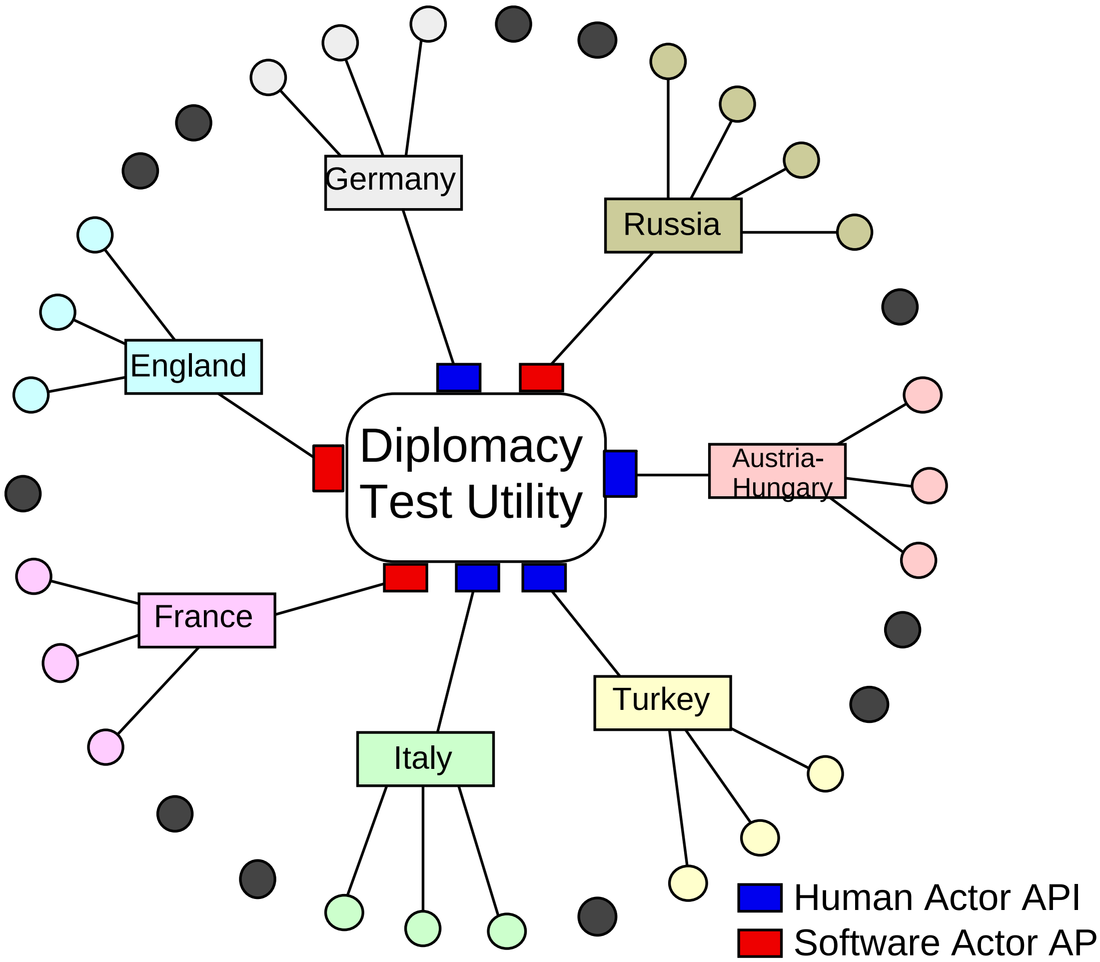


Figure 16: Stage One of the Diplomacy Test Utility

Stage Two of the Diplomacy Test Utility

Nations are not monolithic, but hierarchical and networked organizations. Staying within the general structure of *Diplomacy*, three functions, subordinate to the national leader and perhaps each other, can easily be justified: (i) command of the nation's military units, (ii) production of military supplies, and (iii) distribution of supplies and transport of units (logistics). If these functions are modeled as being performed by groups of self-interested nodes/actors, then under each national leader there is a Coordination Game on which mechanism design can be applied.

At the top level of a Stage Two DTU game, this is still a Coalition Game, but the moves that a national leader can order and the subset of those orders that the leader can expect to have executed depend on the Coordination Games played out below each leader.

The separation of Coalition Game and Coordination Games here is critical – no interaction is allowed between the national Coordination Games. The only international communication is between leaders and there is no international commerce, even to the extent that one nation cannot use the supplies of another. This separation allows for the application of fairly standard game theory to the two separate types of games. Only the leader needs to deal with the interactions between the two types, interactions that are purely physical rather than strategic as no foreign actor can influence a decision beneath a national leader. This separation allows software agent development to be guided by largely existing game theory, while game theorists work on the more complex problems of DHG that will be needed in DTU stage three.

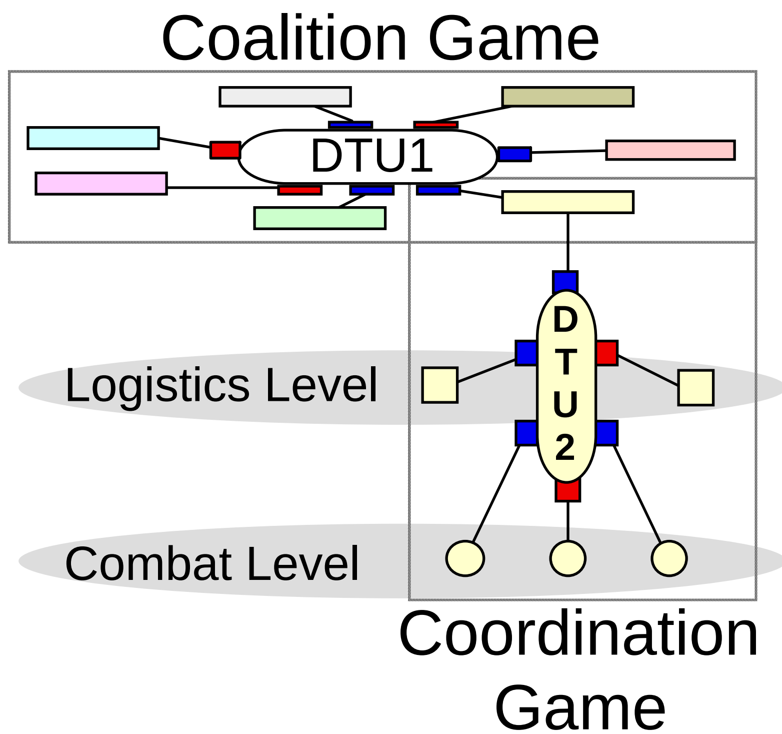


Figure 17: Stage Two of the Diplomacy Test Utility

In DTU stage two, software agents can fill any leader or subordinate function actor position. On-line *Diplomacy* players should be introduced to the new game and given reasons to participate as subordinate actors.[18] The open access policy for software agents would be continued. The one required new development for DTU Central is the process for introducing and using coordination mechanisms within a nation. Various command and control and decentralized (market) means have been designed and some used for commercial scenarios, and it is reasonable to expect many of each to be proposed into DTU stage two.

Stage Three of the Diplomacy Test Utility – DHG

In war, supply gets interdicted, production gets disrupted, and military command gets compromised – international physical transport and communication can be undertaken by and affect subordinate actors. DTU stage three removes the prohibitions that cleanly separated Coalition Games and Coordination Games in stage two. Two new developments need to be

[18] For instance, the common practice in even single player shooter games of being required to gain experience before getting a better gig – "you got to earn your wings".

added to DTU Central: (i) the means of defending and policing against unauthorized international transport and communication and (ii) the means of initiating and conducting international transport and communication (whether authorized or not).

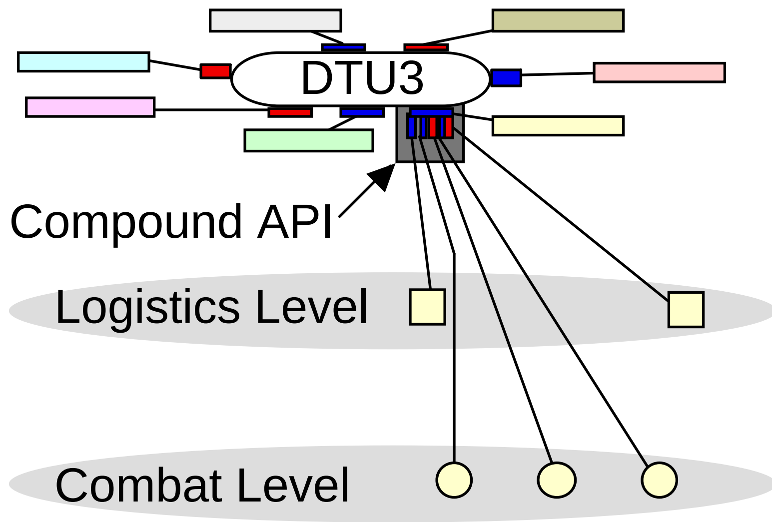


Figure 18: Stage Three of the Diplomacy Text Utility

DTU stage three will be a test bed for dynamic hierarchical games (DHG) within the context of one basic scenario, *Diplomacy*, and benefiting from the economies of having a large reservoir of trained and cheap human subjects. Stage three is the actual utility that will enable the interplay of theory and practice that will result in an understanding of and ability to use DHG. Stages one and two are just stepping stones to this utility.

Outline for Getting Underway

You cannot build CERN without first building the Stanford Linear Accelerator and Fermi Lab. Similarly, DARPA cannot realistically expect to build a universal strategic reasoning utility before even the first instance of a strategic reasoning utility has been built. Once the DTU stage three is built and refined, the requirements for supporting generic strategic scenarios will be much more apparent. Also the merits of a game theory & IT approach will be more defensible; a critical consideration given the degree of funding likely required for a universal strategic reasoning utility.

DTU stage one requires theoretical work for the negotiation language/protocol, software and systems engineering for the APIs and game operations, marketing to both the *Diplomacy* and the research communities, and a licensing arrangement with Hasbro. These are fairly straightforward tasks and a modestly funded effort should be able to produce an operational DTU stage one within a twelve to eighteen-month timeframe.

Given sufficient initial funding, it is highly recommended that certain stage two and three tasks get underway from the start. The command, production, and logistics Coordination Game structure of stage two will require theory development, software engineering, and a good deal of standalone testing. To assure that DTU stage two can begin smoothly once stage one is established and understood, these efforts should start as soon as they can be funded. Regarding DTU stage three, focused theoretical work should begin as soon as possible – this is a hard problem and the better it is understood prior to committing product development resources, the smoother will be development and the better will be the product.

APPENDIX A:
FORMAL MODELING OF R&D MANAGEMENT PROCESSES FOR A PLANETARY MISSION

Decision Models for Cassini Simulation

V. 1.00
Net Exchange
May 21, 2003
Tak Ishikida

Revision history

Version	Date	Author	Description
1.00	May 21, 2003	TI	Original documentation (*CassiniModels00.doc*)

About this document
Describes the decision models to be run in Cassini simulation.

Overview

JPL, three scientists, and nature are the actors in this Cassini simulation. Four decision models are described in this document.

Figure 1 illustrates a general sequence of the various actions.

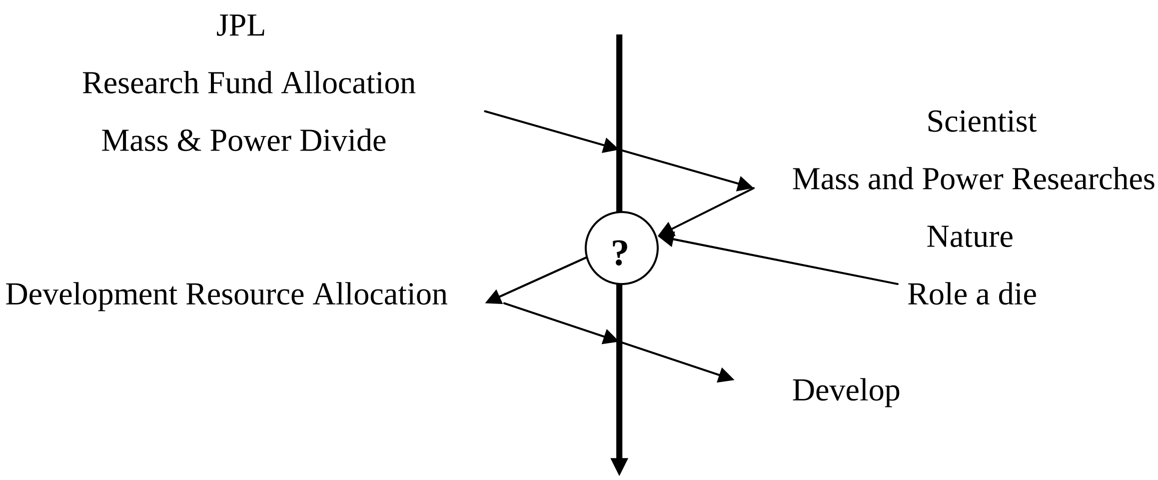


Figure 1. Sequence of Decisions

- The first move is made by JPL; it decides the research fund allocation among the scientists.
- Then a scientist divides the fund between two research areas: mass and power uses. The research effort and nature's draw determine the technological factors (efficiency) of his mass and power uses.
- JPL gets some information about the research outcomes of scientists.
- Then JPL determines the development resource allocation based on the information.
- A scientist determines how much of resource to use.

The main difference among the decision models described here is the amount of information available for JPL symbolized by "?" in Figure 1. The following cases are considered:

Case 1: JPL has complete information and acts as the sole decision maker
Case 2: JPL has complete information on research outcomes
Case 3: JPL can gain partial information by probing scientists
Case 4: JPL has no information and operates exchanges to give an opportunity redistribute development resources.

In all cases, I approach the decision process by a backward induction:

- First figure out the scientist's development fund use.
- Figure out JPL's development resource allocation based on the type of information available to it.
- Figure out scientists research fund division between mass and power researches.
- Finally, figure out JPL's research fund allocation.

Case 1: JPL as a Sole Decision Maker with Complete Information

JPL knows every detail and centrally decides allocation of funds in research phase, and then allocates mass, power, and funds in development phase after observing the results of research efforts by scientists In this model, scientists are the robots that obey JPL's commands.

JPL Decisions After Research Phase Outcomes Are Known

Once the research phase outcomes of each scientist became known, JPL decides the allocation of mass, power, and development funds to the scientists. Since each scientist could have spent research efforts on two areas (one effecting its mass use and the other its power use) and the research results in one of two outcomes (no luck vs. good luck), there are all together 64 possible combinations of research effort outcomes for the three scientists in the simulation. Given a combination of research outcomes and available development funds (note that how this state is reached does not affect the JPL decision on allocation of development resources), JPL finds the allocation that maximizes the value of science and residual funds to JPL.

Allocation quantities of mass, power, and fund take a discrete value. The optimal allocation at each state is found by brute force enumeration. Simulation parameters are described in §6 in detail.

JPL Decision Before Research Phase

The amount of research fund spent on mass and power research affects the probability of having good outcomes: more you spend, better chance of having a good luck (get higher value for a mass and/or power use). Given the funds spent on mass and power researches by each scientist, JPL can compute the probabilities of 64 research outcomes and funds available for the development, and hence the expected value of the values under the research fund distribution. Probabilities of having good luck among scientists are assumed independent. Moreover, the probability of having a good luck for mass use and power use within each scientist are assumed independent. As for development resources, funds allocation for mass and power researches takes discrete value. The program enumerates over the combinations of research funding and finds the maximum expected value.

Case 2: Game with Complete Information

Everyone knows all details as in Case 1. JPL distributes research and development money, but each scientist decides how to use funds; given the research fund, a scientist decides how much of research fund is spent on mass research and how much on power in the research phase, and the scientist decides how much of development fund is actually spent in the development phase. If JPL over-allocates fund, a scientist can keep the money not spent. I have not considered a scientist option of carrying over the unused research fund into the development phase. However, what I did is equivalent of JPL taking back unused fund and adding to the development fund. The unused research fund may or may not be used in the development phase. If used, it may not go back to the scientist from whom JPL took it back. The JPL value under this scheme is at least as large as if a scientist is allowed to carry over unused research fund. Under this scheme scientists have no incentive to leave research fund unused.

Scientist Decision After Research Outcomes Are Known

I assume that each scientist maximizes his value (the value of science and unused fund) given the resources available for his development.

JPL Decision After Research Outcomes Are Known

JPL allocates development resources so that its value is maximized under the scientist behavior described in §3.1. Unlike Case 1, the scientist is assumed to have a positive utility on the unused development fund. Because of that, a scientist may spend less money on the development than in Case 1 when faced with the same research outcomes and resources available. Therefore, the value to JPL can be less in Case 2.

Scientist Decision on How to Divide Research Fund

Fix the amount of research fund allocated. Scientists compete for limited resources in the development phase and since the research outcomes influence the allocation of them, one scientist's decision on how to spend their research fund affects allocation to others. I consider this a game (in the normal form): a strategy of a scientist is the distribution of research fund between mass and power researches, and a payoff of the scientist is the expected value for the scientist under the JPL allocation of development resources described in §3.2. For each combination of strategies from the scientists, we can compute the probabilities of 64 research outcomes and hence we can compute the expected payoff for each scientist under the strategy combination. Given the payoff matrix, I assumed that each scientist plays the max min strategy.[19] An implicit assumption is that the scientists make simultaneous moves (or they can't observe other scientists' moves).

JPL Decision on Research Fund Allocation

JPL computes the response (max min strategy) of each scientist for each allocation of research fund (discrete grid point as in Case 1) and computes the JPL value that comes out of it. JPL selects the research fund allocation that gives the maximum expected value to JPL.

Case 3: Game with Partial Information: Probing

JPL know characteristics of scientists (their value functions, technology factors, etc). But JPL can observe neither how scientists allocate their research funds between power and mass researches nor the outcomes of their researches. JPL, however, has an option of probing scientists to see whether they succeeded or not with some cost. The cost of probing comes out of research/development fund[20]. Although, the probing will increase the efficiency of allocation of development resources, it reduces the available funds for research/development. Thus, there is a tradeoff.[21]

Probing reports scientists' mass and power research outcomes. It tells the effort results in either GoodLuck or NoLuck. However, it is noisy; i.e.,

Prob[Report GoodLuck|GoodLuck] < 1.

Prob[Report NoLuck|NoLuck] < 1.

More money JPL spends on probing, more accurate the report becomes. For simplicity, I assume that the amount of research fund spend by a scientist does not influence the accuracy of probing.[22]

[19] The tie break is arbitrary (the last or first index in enumeration get picked up). That means a (weekly) dominant strategy may not be selected.

[20] In all cases, I assume that unused research fund can be added to the development fund. If we attempt to take money out of the development fund, we can always have an option of leaving the exact amount unused in the research phase as long as there is enough research fund to cover it. So there is no real distinction between whether research fund is spent to probe or development fund.

[21] In Case 4, we compute the expected value without any probing where JPL distributes research fund and announces the development resource distribution at the beginning and lets scientists do their best with the allocation. Unless the expected JPL value with the probing exceed the expected value with no probing, probing is not worth undertaking. For the sake of argument, suppose that probing is noiseless; i.e., probing reveals the true research outcomes. Then the situation is exactly like in Case 2 except the money available for research/development is less by the cost of probing. This gives the upper bound on the expected JPL value with probing since noise in probing lowers the expected JPL value.

[22] Charles tells me that more money (labor) a scientist spends, more likely the probing finds a true outcome.

Scientist Decision After Research Outcomes Are known

Given research outcomes and the allocation of development resources, each scientist maximizes his value.

JPL Decision After Probing Results Are Known

JPL does not directly observe the research outcomes. After probing scientists, JPL updates the probability distribution over research outcomes (64 possibilities) based on the probing results (*a. la.* Bayes Rule to get the posterior). Given the posterior probability distribution and the available development resources, JPL determines the allocation of development resources that maximizes its expected value under the scientist behavior described in §4.1.[23]

Computing Posterior Distribution

I assume independence in research outcome and probing result across different researches and probing efforts. With this, I illustrate how the posterior is computed for a single research effort.

Let p := prob(GoodLuck) be the prior probability of having good luck, f := prob(Report GoodLuck|GoodLuck) be the conditional prob of correctly reporting GoodLuck, and g := prob(Report NoLuck|NoLuck) be the conditional prob of correctly reporting NoLuck. Then the posterior probability of GoodLuck given proving report GoodLuck is

r := prob(GoodLuck|Report GoodLuck) = f*p/[f*p + (1-g)*(1-p)]

Similarly,

s := prob(GoodLuck|Report NoLuck) = (1-f)*p/[(1-f)*p + g*(1-p)]

Note p is the function of research fund spent and f and g are the money spent on monitoring (and possibly the money spent on research as well. Each report results in a distinct posterior distribution.

With six independent researches, there are 64 possible reports and hence 64 possible posteriors for each prior.

Development Resource Distribution

Given a probability distribution on research outcomes and development resources available, JPL can evaluate the expected value under a specific resource allocation assuming scientists behave as described in §4.1:

- For each combination of research outcomes, each scientist responds by maximizing his value given his share of the allocated resources.

[23] Note that the prior distribution of research outcomes depends on how much fund is spent on research (and hence depends on JPL allocation of research fund and a scientist distribution of the allocated fund between mass and power researches) and so does the posterior distribution. If the number of posterior distribution is limited, we may be able to take advantage of the fact but the dependency on research fund allocation makes it unlikely. In Case 2, we dealt with the outcome itself instead of the distribution so it is limited to 64 contingencies (64 trivial distributions). When probability distribution is a (DP) state, it is called an information state in the literature.

- From the scientist's decision on development money use, JPL value is computed for the combination of research outcome.
- Find the JPL values for all the combinations of research outcomes as described in the previous two steps and use the posterior distribution to compute the expected value for the allocation.

By enumerating all possible resource allocations, JPL can find the best allocation under the distribution and the resources available. The difference from Case 2 is that JPL need to compute the best response for more than one combination of research outcomes and compute the expected value.

The (computational) difficulty comes from not so much from having to compute expectation but from the fact that each research fund allocation results in different posterior distribution over the research outcomes.

Scientist Decision on How to Divide Research Fund

Fix an allocation of research fund and money committed on probing. The scientists compete for limited resources in the development phase and since the research outcomes and probing results influence the allocation of them, one scientist's decision on how to spend their research fund affects allocation to others. I consider this a game (in the normal form): a strategy of a scientist is the distribution of research fund between mass and power researches, and a payoff of the scientist is the expected value for the scientist under the JPL allocation described in §4.2:

- For each combination of strategies from the scientists, a scientist can compute the probabilities of 64 research outcomes (the prior distribution).
- Also he can compute the conditional probabilities of correct reporting from the money committed for probing.
- Hence he can compute the probability of reaching to each information state (a posterior distribution, or probing report) considered in §4.2.
- For each posterior distribution, he can follow JPL decision (development resource allocation) described in §4.2 and work out his development fund use (decision described in §4.1) to get his and other scientists expected payoff for the posterior distribution.
- After computing his (and other scientists) expected payoff for each possible posterior distribution (possibly 64 of them), he unconditions (take expectation over possible information state) to get the unconditional expectation of his (and other scientists') payoff under the strategy combination.
- By performing above steps for all possible strategy combinations, he has the expected payoff for each scientist under every strategy combination. This is the payoff matrix.
- Given the payoff matrix, I assumed that each scientist plays the max min strategy. An implicit assumption is that the scientists make a simultaneous move (or can't observe other scientists' moves).

JPL Decision on Research Fund Allocation

Scientist's decision is similar to that in Case 2. JPL computes the response (max min strategy) of each scientist for each combination of research fund and probing commitment (discrete grid point as in Case 1) and computes the expected JPL value that comes out of it. JPL selects the research fund allocation that gives the maximum expected value to JPL.

Case 4: Game with No Information: Resource Exchange

Instead of trying to lean the research outcomes, JPL creates an exchange where scientists can redistribute their allocated resources. Since JPL does not gain any information regarding the outcome of research efforts by the scientists, the probability distribution JPL has on the outcomes is the prior distribution. So we can assume that JPL's decision on development resource allocation is done before the research phase begins.

[Probably something has to be said about the cost of creating and running exchange.]

JPL Decision on Resource Allocation

JPL decides all (both research and development) resource allocation at the beginning. Since the resources are all allocated by the time scientists make research phase decisions, JPL assumes that they will maximize their expected value. The resource allocation is determined in the following manner.

Scientist Decision After Research Outcomes Are Known

Given the research outcomes and available development resources, each scientist maximizes his value.

Scientist Decision on How to Divide Research Fund

For each possible division of research fund between power and mass, he can compute the probability distribution over 4 possible outcomes of his research efforts. For each research outcome, he follows his optimal action and gets the optimal value as described in §5.1.1. Thus he can compute the expected value for each possible division. He chooses the research fund division that gives him the maximum expected value.

JPL Decision

For each combination of research and development resource allocation, JPL computes the response from scientists as described in §5.1.2. By following the scientist action for each allocation, JPL can compute its expected value over 64 contingencies. JPL chooses the resource allocation that maximizes the expected JPL value. This is the best JPL can do without probing in Case 3.

Exchange

Exchange Scheme

An exchange is the place where scientists can swap their resources. A scientist submits various offers for an exchange and if one of them finds a counter offer that matches it, the swap is made.

Table 1 shows an example of exchange offer. In the example, a scientist wants 1 unit of mass and is willing to give away one unit of power and one unit of money. The value is something that indicates the relative preference of this offer among the offers he submits to the exchange.

Table 1. Resource exchange offer

Development Resource	Mass	Power	Money	Value
Quantity	1	-1	-1	5

The exchange collects exchange offers from all scientists and maximizes the aggregate value subject to resource balance—supply for any resource should be short. At most one exchange offer from each scientist will be accepted. If the exchange results in the excess supply of resources, JPL takes them.

<u>Scientist Bidding</u>

A scientist computes the value at the current development resource allocation. He considers exchanges that results in better value and submits them to the exchange. In this simulation, I limited the exchange offers as follows:

- If one technology factor (mass or power) is better or equal to the other, try to gain more of the resource.

The value is set to the difference between the scientist values before and after the exchange if the exchange offer is accepted.

Simulation Parameters

Research fund allocation: {0, 1, 2, 3} for each mass or power research
Mass allocation: {0, 1, 2, 3}
Power allocation: {0, 1, 2, 3}
Development fund allocation: {0, 1, 2, …}
Research outcomes: {NoLuck, GoodLuck} for each mass and power research.

APPENDIX B:
DATA FROM ALL RUNS OF THE GAMBIT-LIKE HISTORICAL SCENARIO

Case 1 Data

Res. Fund	Dev. Fund	SC1 M GL	SC1 M GL	SC3 M GL	SC1 P GL	SC2 P Gl	SC3 P Gl	JPL Expect	SC1 Expect	SC2 Expect	SC3 Expect
2	3	1	1	1	1	1	1	12	4	4	3
2	3	0.9	0.9	1.1	1.1	1	0.9	12	4	4	3
2	3	0.8	0.8	1.2	1.2	1	0.8	12	4	4	3
2	3	1.5	1.5	1.5	1.5	1.5	1.5	12	4	4	3
2	3	1.35	1.35	1.65	1.65	1.5	1.35	12	4	4	3
2	3	1.2	1.2	1.8	1.8	1.5	1.2	12.2785	3.94641	3.24402	4.19043
2	3	2	2	2	2	2	2	13.4641	4.24402	4.24402	3.48803
2	3	1.8	1.8	2.2	2.2	2	1.8	14.0641	4.54162	3.24402	4.78564
2	3	1.6	1.6	2.4	2.4	2	1.6	15.1441	4.03923	3.91068	4.28325
2	3	2.5	2.5	2.5	2.5	2.5	2.5	15.7974	4.1547	4.1547	4.1547
2	3	2.25	2.25	2.75	2.75	2.5	2.25	16.6308	4.44338	3.91068	4.68739
2	3	2	2	3	3	2.5	2	17.4641	4	4.1547	4.97607
2	3	3	3	3	3	3	3	17.4641	4	4.48803	4.1547
2	3	2.7	2.7	3.3	3.3	3	2.7	19.0497	4.51722	4.1547	4.51722
2	3	2.4	2.4	3.6	3.6	3	2.4	20.6354	4.79043	4.1547	4.79043
2	3	3.5	3.5	3.5	3.5	3.5	3.5	20.1068	4.69936	4.69936	4.1547
2	3	3.15	3.15	3.85	3.85	3.5	3.15	21.9568	5.0181	4.1547	5.0181
2	3	2.8	2.8	4.2	4.2	3.5	2.8	23.4162	5.33684	4.1547	5.33684
3	3	1	1	1	1	1	1	16	4	4	4
3	3	0.9	0.9	1.1	1.1	1	0.9	16	4	4	4
3	3	0.8	0.8	1.2	1.2	1	0.8	16	4	4	4
3	3	1.5	1.5	1.5	1.5	1.5	1.5	16	4	4	4
3	3	1.35	1.35	1.65	1.65	1.5	1.35	16	4	4	4
3	3	1.2	1.2	1.8	1.8	1.5	1.2	16	4	4	4
3	3	2	2	2	2	2	2	17.4917	4.05157	4.34715	4.87293
3	3	1.8	1.8	2.2	2.2	2	1.8	18.5481	4.99245	4.29558	4.74789
3	3	1.6	1.6	2.4	2.4	2	1.6	19.9764	4.64199	4.29558	4.98914
3	3	2.5	2.5	2.5	2.5	2.5	2.5	22.9144	4.62892	5.16161	4.98298
3	3	2.25	2.25	2.75	2.75	2.5	2.25	24.4144	5.1135	4.87293	5.46065
3	3	2	2	3	3	2.5	2	25.9144	5.14088	4.97607	5.79743
3	3	3	3	3	3	3	3	28.3509	4.89687	5.93832	5.45028
3	3	2.7	2.7	3.3	3.3	3	2.7	30.5716	5.61722	5.55342	5.86124
3	3	2.4	2.4	3.6	3.6	3	2.4	32.6796	5.99043	5.55342	6.23444
3	3	3.5	3.5	3.5	3.5	3.5	3.5	34.9835	5.62892	6.25093	5.96916
3	3	3.15	3.15	3.85	3.85	3.5	3.15	37.7764	6.40457	5.7698	6.54545
3	3	2.8	2.8	4.2	4.2	3.5	2.8	40.1789	6.83997	5.7698	6.98085
4	3	1	1	1	1	1	1	16.1	4	4	4
4	3	0.9	0.9	1.1	1.1	1	0.9	16.1	4	4	4
4	3	0.8	0.8	1.2	1.2	1	0.8	16.1	4	4	4
4	3	1.5	1.5	1.5	1.5	1.5	1.5	16.9521	5	5	3.48803
4	3	1.35	1.35	1.65	1.65	1.5	1.35	17.8521	5.44641	3.66667	5.35318
4	3	1.2	1.2	1.8	1.8	1.5	1.2	19.3121	4.69282	4.66667	4.62966
4	3	2	2	2	2	2	2	22.9282	5.64273	5.64273	4
4	3	1.8	1.8	2.2	2.2	2	1.8	24.8127	5.19872	4.9245	5.20903
4	3	1.6	1.6	2.4	2.4	2	1.6	27.1101	5.5245	5.11695	5.46815
4	3	2.5	2.5	2.5	2.5	2.5	2.5	29.864	5.78362	5.31631	5.75783

4	3	2.25	2.25	2.75	2.75	2.5	2.25	33.1836	6.30089	5.56033	6.03016
4	3	2	2	3	3	2.5	2	36.8538	6.70812	5.849	6.25783
4	3	3	3	3	3	3	3	38.7922	7.10684	6.09302	6.29558
4	3	2.7	2.7	3.3	3.3	3	2.7	42.6036	7.21991	7.22008	6.04688
4	3	2.4	2.4	3.6	3.6	3	2.4	48.0103	7.15914	7.04145	6.82211
4	3	3.5	3.5	3.5	3.5	3.5	3.5	49.2183	7.59487	7.33013	6.40563
4	3	3.15	3.15	3.85	3.85	3.5	3.15	55.3852	7.49593	7.42635	7.35135
4	3	2.8	2.8	4.2	4.2	3.5	2.8	60.9288	7.96743	7.81125	7.6304
5	3	1	1	1	1	1	1	16.2	4	4	4
5	3	0.9	0.9	1.1	1.1	1	0.9	16.8	4.34641	4	4.34641
5	3	0.8	0.8	1.2	1.2	1	0.8	17.6	4.69282	4	4.69282
5	3	1.5	1.5	1.5	1.5	1.5	1.5	20.0488	5	5	4
5	3	1.35	1.35	1.65	1.65	1.5	1.35	22.3713	5.44641	4.48803	5.93444
5	3	1.2	1.2	1.8	1.8	1.5	1.2	24.7426	5.89282	4.48803	6.38085
5	3	2	2	2	2	2	2	30.4678	6.6188	5.20627	5.9245
5	3	1.8	1.8	2.2	2.2	2	1.8	32.8681	6.27239	6.3094	5.77607
5	3	1.6	1.6	2.4	2.4	2	1.6	36.0856	6.20074	6.41253	6.17404
5	3	2.5	2.5	2.5	2.5	2.5	2.5	39.8564	7.24402	5.82137	7.44017
5	3	2.25	2.25	2.75	2.75	2.5	2.25	44.202	6.55755	7.45028	7.07162
5	3	2	2	3	3	2.5	2	48.9909	6.57735	7.45028	7.7735
5	3	3	3	3	3	3	3	51.0185	8.48803	6.5396	8.5433
5	3	2.7	2.7	3.3	3.3	3	2.7	56.6422	7.53094	8.6943	7.99588
5	3	2.4	2.4	3.6	3.6	3	2.4	62.7751	8.06188	8.6943	8.50003
5	3	3.5	3.5	3.5	3.5	3.5	3.5	62.8619	8.67193	8.02044	7.58426
5	3	3.15	3.15	3.85	3.85	3.5	3.15	71.0049	8.50852	8.66025	7.90268
5	3	2.8	2.8	4.2	4.2	3.5	2.8	78.7232	8.96604	9.3745	8.2102
2	4	1	1	1	1	1	1	16	4	4	4
2	4	0.9	0.9	1.1	1.1	1	0.9	16	4	4	4
2	4	0.8	0.8	1.2	1.2	1	0.8	16	4	4	4
2	4	1.5	1.5	1.5	1.5	1.5	1.5	16	4	4	4
2	4	1.35	1.35	1.65	1.65	1.5	1.35	16	4	4	4
2	4	1.2	1.2	1.8	1.8	1.5	1.2	16	4	4	4
2	4	2	2	2	2	2	2	17.4341	4.48803	4.48803	3.82137
2	4	1.8	1.8	2.2	2.2	2	1.8	18.2341	4.05231	4.48803	4.48803
2	4	1.6	1.6	2.4	2.4	2	1.6	19.0341	4.28325	4.48803	4.48803
2	4	2.5	2.5	2.5	2.5	2.5	2.5	19.4341	4.39872	4.73205	4.48803
2	4	2.25	2.25	2.75	2.75	2.5	2.25	20.6781	6.34209	4.06538	5.36004
2	4	2	2	3	3	2.5	2	22.4102	6.79743	4.06538	5.73205
2	4	3	3	3	3	3	3	22.4102	5.4641	5.79743	4
2	4	2.7	2.7	3.3	3.3	3	2.7	24.9302	5.81051	5.39872	5.07846
2	4	2.4	2.4	3.6	3.6	3	2.4	27.6902	6.15692	5.39872	5.42487
2	4	3.5	3.5	3.5	3.5	3.5	3.5	26.7435	6.04145	6.04145	4.66667
2	4	3.15	3.15	3.85	3.85	3.5	3.15	30.1735	6.4456	5.39872	5.71355
2	4	2.8	2.8	4.2	4.2	3.5	2.8	32.8102	6.84974	5.39872	6.11769
3	4	1	1	1	1	1	1	16.1	4	4	4
3	4	0.9	0.9	1.1	1.1	1	0.9	16.1	4	4	4
3	4	0.8	0.8	1.2	1.2	1	0.8	16.1	4	4	4
3	4	1.5	1.5	1.5	1.5	1.5	1.5	16.9521	5	5	3.48803
3	4	1.35	1.35	1.65	1.65	1.5	1.35	17.8521	5.44641	3.66667	5.35318
3	4	1.2	1.2	1.8	1.8	1.5	1.2	19.3121	4.69282	4.66667	4.62966
3	4	2	2	2	2	2	2	22.9282	5.64273	5.64273	4
3	4	1.8	1.8	2.2	2.2	2	1.8	24.8127	5.19872	4.9245	5.20903
3	4	1.6	1.6	2.4	2.4	2	1.6	27.1101	5.5245	5.11695	5.46815

3	4	2.5	2.5	2.5	2.5	2.5	2.5	29.864	5.78362	5.31631	5.75783
3	4	2.25	2.25	2.75	2.75	2.5	2.25	33.1836	6.30089	5.56033	6.03016
3	4	2	2	3	3	2.5	2	36.8538	6.70812	5.849	6.25783
3	4	3	3	3	3	3	3	38.7922	7.10684	6.09302	6.29558
3	4	2.7	2.7	3.3	3.3	3	2.7	42.6036	7.21991	7.22008	6.04688
3	4	2.4	2.4	3.6	3.6	3	2.4	48.0103	7.15914	7.04145	6.82211
3	4	3.5	3.5	3.5	3.5	3.5	3.5	49.2183	7.59487	7.33013	6.40563
3	4	3.15	3.15	3.85	3.85	3.5	3.15	55.3852	7.49593	7.42635	7.35135
3	4	2.8	2.8	4.2	4.2	3.5	2.8	60.9288	7.96743	7.81125	7.6304
4	4	1	1	1	1	1	1	16.2	4	4	4
4	4	0.9	0.9	1.1	1.1	1	0.9	16.8	4.34641	4	4.34641
4	4	0.8	0.8	1.2	1.2	1	0.8	17.6	4.69282	4	4.69282
4	4	1.5	1.5	1.5	1.5	1.5	1.5	20.0488	5	5	4
4	4	1.35	1.35	1.65	1.65	1.5	1.35	22.3713	5.44641	4.48803	5.93444
4	4	1.2	1.2	1.8	1.8	1.5	1.2	24.7426	5.89282	4.48803	6.38085
4	4	2	2	2	2	2	2	30.4678	6.6188	5.20627	5.9245
4	4	1.8	1.8	2.2	2.2	2	1.8	32.8681	6.27239	6.3094	5.77607
4	4	1.6	1.6	2.4	2.4	2	1.6	36.0856	6.20074	6.41253	6.17404
4	4	2.5	2.5	2.5	2.5	2.5	2.5	39.8564	7.24402	5.82137	7.44017
4	4	2.25	2.25	2.75	2.75	2.5	2.25	44.202	6.55755	7.45028	7.07162
4	4	2	2	3	3	2.5	2	48.9909	6.57735	7.45028	7.7735
4	4	3	3	3	3	3	3	51.0185	8.48803	6.5396	8.5433
4	4	2.7	2.7	3.3	3.3	3	2.7	56.6422	7.53094	8.6943	7.99588
4	4	2.4	2.4	3.6	3.6	3	2.4	62.7751	8.06188	8.6943	8.50003
4	4	3.5	3.5	3.5	3.5	3.5	3.5	62.8619	8.67193	8.02044	7.58426
4	4	3.15	3.15	3.85	3.85	3.5	3.15	71.0049	8.50852	8.66025	7.90268
4	4	2.8	2.8	4.2	4.2	3.5	2.8	78.7232	8.96604	9.3745	8.2102

Case 2 Data

Res. Fund	Dev. Fund	SC1 M GL	SC1 M GL	SC3 M GL	SC1 P GL	SC2 P Gl	SC3 P Gl	JPL Expect	SC1 Expect	SC2 Expect	SC3 Expect
2	3	1	1	1	1	1	1	12	4	4	3
2	3	0.9	0.9	1.1	1.1	1	0.9	12	4	4	3
2	3	0.8	0.8	1.2	1.2	1	0.8	12	4	4	3
2	3	1.5	1.5	1.5	1.5	1.5	1.5	12	4	4	3
2	3	1.35	1.35	1.65	1.65	1.5	1.35	12	4	4	3
2	3	1.2	1.2	1.8	1.8	1.5	1.2	12.2785	3.34641	3	3.34641
2	3	2	2	2	2	2	2	13.4641	3	3.82137	4.1547
2	3	1.8	1.8	2.2	2.2	2	1.8	14.0641	3.80829	3.24402	4.05231
2	3	1.6	1.6	2.4	2.4	2	1.6	15.1441	4.03923	3.24402	4.28325
2	3	2.5	2.5	2.5	2.5	2.5	2.5	14.1308	3.33333	4.39872	4.39872
2	3	2.25	2.25	2.75	2.75	2.5	2.25	16.6308	4.44338	3.57735	4.68739
2	3	2	2	3	3	2.5	2	17.4641	4	3.24402	4.97607
2	3	3	3	3	3	3	3	17.4641	4	3.82137	4.97607
2	3	2.7	2.7	3.3	3.3	3	2.7	16.4641	3.33333	4.97607	4.62966
2	3	2.4	2.4	3.6	3.6	3	2.4	20.6354	4.54641	3.57735	4.79043
2	3	3.5	3.5	3.5	3.5	3.5	3.5	17.7735	3.82137	4.69936	4.69936
2	3	3.15	3.15	3.85	3.85	3.5	3.15	21.9568	5.0181	3.82137	5.0181
2	3	2.8	2.8	4.2	4.2	3.5	2.8	23.4162	5.09282	3.57735	5.33684
3	3	1	1	1	1	1	1	16	4	4	4
3	3	0.9	0.9	1.1	1.1	1	0.9	16	4	4	4
3	3	0.8	0.8	1.2	1.2	1	0.8	16	4	4	4
3	3	1.5	1.5	1.5	1.5	1.5	1.5	16	4	4	4
3	3	1.35	1.35	1.65	1.65	1.5	1.35	16	4	4	4
3	3	1.2	1.2	1.8	1.8	1.5	1.2	16	4	4	4
3	3	2	2	2	2	2	2	17.4917	3.91068	4.1547	4.25783
3	3	1.8	1.8	2.2	2.2	2	1.8	18.5481	4.14162	4.1547	4.34789
3	3	1.6	1.6	2.4	2.4	2	1.6	19.9764	4.18011	3.87293	4.38638
3	3	2.5	2.5	2.5	2.5	2.5	2.5	19.7735	4.48803	4.73205	4.97607
3	3	2.25	2.25	2.75	2.75	2.5	2.25	24.4144	4.58426	4.73205	4.98298
3	3	2	2	3	3	2.5	2	25.9144	4.56353	4.45028	5.0792
3	3	3	3	3	3	3	3	28.3509	4.75598	5.41253	5.55342
3	3	2.7	2.7	3.3	3.3	3	2.7	26.2931	5.04614	5.3094	5.25241
3	3	2.4	2.4	3.6	3.6	3	2.4	32.6796	5.19447	5.3094	5.40074
3	3	3.5	3.5	3.5	3.5	3.5	3.5	29.1621	5.48803	5.62892	5.7698
3	3	3.15	3.15	3.85	3.85	3.5	3.15	37.7764	5.66363	5.62892	5.66363
3	3	2.8	2.8	4.2	4.2	3.5	2.8	40.1789	5.92855	5.52578	6.03168
4	3	1	1	1	1	1	1	16.1	4	4	4
4	3	0.9	0.9	1.1	1.1	1	0.9	16.1	4	4	4
4	3	0.8	0.8	1.2	1.2	1	0.8	16.1	4	4	4
4	3	1.5	1.5	1.5	1.5	1.5	1.5	16.9521	4.66667	3.66667	4.11004
4	3	1.35	1.35	1.65	1.65	1.5	1.35	17.3856	4	4	4.34641
4	3	1.2	1.2	1.8	1.8	1.5	1.2	19.3121	4.69282	3.66667	4.62966
4	3	2	2	2	2	2	2	22.1457	4.87293	4.39872	4.73205
4	3	1.8	1.8	2.2	2.2	2	1.8	24.1282	5.14991	4.42265	4.97128
4	3	1.6	1.6	2.4	2.4	2	1.6	27.1101	4.57883	4.73205	4.57883
4	3	2.5	2.5	2.5	2.5	2.5	2.5	29.864	5.36097	5.11695	5.54651
4	3	2.25	2.25	2.75	2.75	2.5	2.25	30.7342	5.4641	5.0792	5.26474
4	3	2	2	3	3	2.5	2	36.8538	5.36097	4.98298	5.55342
4	3	3	3	3	3	3	3	38.7922	6.5433	5.79743	6.01382

4	3	2.7	2.7	3.3	3.3	3	2.7	39.2786	6.36319	5.95214	5.69356
4	3	2.4	2.4	3.6	3.6	3	2.4	48.0103	6.45251	5.65655	6.1431
4	3	3.5	3.5	3.5	3.5	3.5	3.5	49.2183	7.02443	6.37479	6.40563
4	3	3.15	3.15	3.85	3.85	3.5	3.15	55.3852	6.94745	6.56724	6.32865
4	3	2.8	2.8	4.2	4.2	3.5	2.8	60.9288	7.35159	6.43326	6.73279
5	3	1	1	1	1	1	1	16.2	4	4	4
5	3	0.9	0.9	1.1	1.1	1	0.9	16.8	4.34641	4	4.34641
5	3	0.8	0.8	1.2	1.2	1	0.8	17.6	4.69282	4	4.69282
5	3	1.5	1.5	1.5	1.5	1.5	1.5	20.0488	4	4.66667	5.12201
5	3	1.35	1.35	1.65	1.65	1.5	1.35	18.8488	4	5	4.7122
5	3	1.2	1.2	1.8	1.8	1.5	1.2	24.7426	4.69282	4	4.69282
5	3	2	2	2	2	2	2	27.9043	4	5.64273	6.3094
5	3	1.8	1.8	2.2	2.2	2	1.8	32.8681	5.71014	5.24402	5.32524
5	3	1.6	1.6	2.4	2.4	2	1.6	36.0856	5.62966	5.01382	5.24476
5	3	2.5	2.5	2.5	2.5	2.5	2.5	36.7442	6.40961	5.41445	5.84931
5	3	2.25	2.25	2.75	2.75	2.5	2.25	39.4674	6.15217	5.36107	5.50389
5	3	2	2	3	3	2.5	2	48.9909	6.16852	6.08241	5.98989
5	3	3	3	3	3	3	3	48.8983	6.83652	7.1584	6.40269
5	3	2.7	2.7	3.3	3.3	3	2.7	47.1688	7.08729	6.39872	7.42433
5	3	2.4	2.4	3.6	3.6	3	2.4	62.7751	6.79595	6.39872	7.13299
5	3	3.5	3.5	3.5	3.5	3.5	3.5	62.8619	7.43159	8.03426	6.8597
5	3	3.15	3.15	3.85	3.85	3.5	3.15	71.0049	7.15532	8.1156	6.58343
5	3	2.8	2.8	4.2	4.2	3.5	2.8	78.7232	7.55947	8.05897	6.98757
2	4	1	1	1	1	1	1	16	4	4	4
2	4	0.9	0.9	1.1	1.1	1	0.9	16	4	4	4
2	4	0.8	0.8	1.2	1.2	1	0.8	16	4	4	4
2	4	1.5	1.5	1.5	1.5	1.5	1.5	16	4	4	4
2	4	1.35	1.35	1.65	1.65	1.5	1.35	16	4	4	4
2	4	1.2	1.2	1.8	1.8	1.5	1.2	16	4	4	4
2	4	2	2	2	2	2	2	17.4341	3.82137	4.1547	4.48803
2	4	1.8	1.8	2.2	2.2	2	1.8	18.2341	4.38564	3.82137	4.38564
2	4	1.6	1.6	2.4	2.4	2	1.6	19.0341	3.82137	4.1547	4.28325
2	4	2.5	2.5	2.5	2.5	2.5	2.5	19.4341	3.82137	4.39872	4.73205
2	4	2.25	2.25	2.75	2.75	2.5	2.25	20.5561	4.48803	4.39872	4.68739
2	4	2	2	3	3	2.5	2	22.4102	4.97607	3.82137	4.97607
2	4	3	3	3	3	3	3	22.4102	4.79743	4.73205	5.06538
2	4	2.7	2.7	3.3	3.3	3	2.7	24.9302	5.81051	4.06538	5.07846
2	4	2.4	2.4	3.6	3.6	3	2.4	27.6902	5.66889	3.82137	5.42487
2	4	3.5	3.5	3.5	3.5	3.5	3.5	24.4102	5.4641	5.3094	5.3094
2	4	3.15	3.15	3.85	3.85	3.5	3.15	30.1735	6.4456	4.73205	5.71355
2	4	2.8	2.8	4.2	4.2	3.5	2.8	32.8102	6.84974	4.06538	6.11769
3	4	1	1	1	1	1	1	16.1	4	4	4
3	4	0.9	0.9	1.1	1.1	1	0.9	16.1	4	4	4
3	4	0.8	0.8	1.2	1.2	1	0.8	16.1	4	4	4
3	4	1.5	1.5	1.5	1.5	1.5	1.5	16.9521	4.66667	3.66667	4.11004
3	4	1.35	1.35	1.65	1.65	1.5	1.35	17.3856	4	4	4.34641
3	4	1.2	1.2	1.8	1.8	1.5	1.2	19.3121	4.69282	3.66667	4.62966
3	4	2	2	2	2	2	2	22.1457	4.87293	4.39872	4.73205
3	4	1.8	1.8	2.2	2.2	2	1.8	24.1282	5.14991	4.42265	4.97128
3	4	1.6	1.6	2.4	2.4	2	1.6	27.1101	4.57883	4.73205	4.57883
3	4	2.5	2.5	2.5	2.5	2.5	2.5	29.864	5.36097	5.11695	5.54651
3	4	2.25	2.25	2.75	2.75	2.5	2.25	30.7342	5.4641	5.0792	5.26474
3	4	2	2	3	3	2.5	2	36.8538	5.36097	4.98298	5.55342

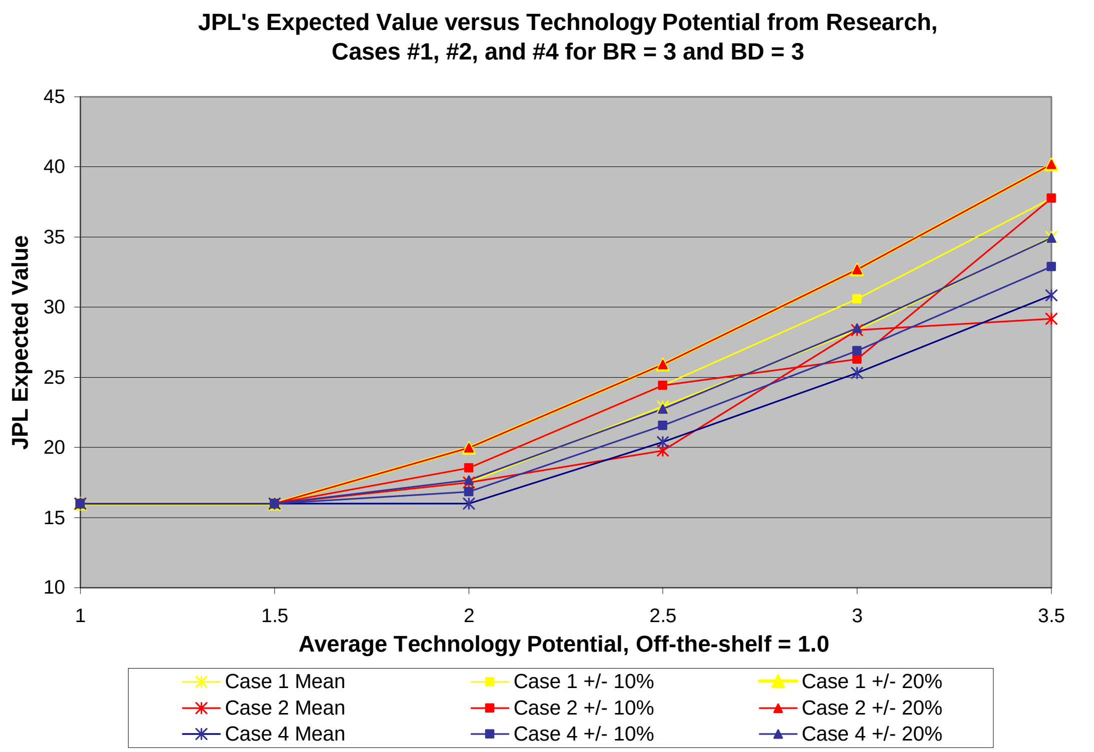
3	4	3	3	3	3	3	3	38.7922	6.5433	5.79743	6.01382
3	4	2.7	2.7	3.3	3.3	3	2.7	39.2786	6.36319	5.95214	5.69356
3	4	2.4	2.4	3.6	3.6	3	2.4	48.0103	6.45251	5.65655	6.1431
3	4	3.5	3.5	3.5	3.5	3.5	3.5	49.2183	7.02443	6.37479	6.40563
3	4	3.15	3.15	3.85	3.85	3.5	3.15	55.3852	6.94745	6.56724	6.32865
3	4	2.8	2.8	4.2	4.2	3.5	2.8	60.9288	7.35159	6.43326	6.73279
4	4	1	1	1	1	1	1	16.2	4	4	4
4	4	0.9	0.9	1.1	1.1	1	0.9	16.8	4.34641	4	4.34641
4	4	0.8	0.8	1.2	1.2	1	0.8	17.6	4.69282	4	4.69282
4	4	1.5	1.5	1.5	1.5	1.5	1.5	20.0488	4	4.66667	5.12201
4	4	1.35	1.35	1.65	1.65	1.5	1.35	18.8488	4	5	4.7122
4	4	1.2	1.2	1.8	1.8	1.5	1.2	24.7426	4.69282	4	4.69282
4	4	2	2	2	2	2	2	27.9043	4	5.64273	6.3094
4	4	1.8	1.8	2.2	2.2	2	1.8	32.8681	5.71014	5.24402	5.32524
4	4	1.6	1.6	2.4	2.4	2	1.6	36.0856	5.62966	5.01382	5.24476
4	4	2.5	2.5	2.5	2.5	2.5	2.5	36.7442	6.40961	5.41445	5.84931
4	4	2.25	2.25	2.75	2.75	2.5	2.25	39.4674	6.15217	5.36107	5.50389
4	4	2	2	3	3	2.5	2	48.9909	6.16852	6.08241	5.98989
4	4	3	3	3	3	3	3	48.8983	6.83652	7.1584	6.40269
4	4	2.7	2.7	3.3	3.3	3	2.7	47.1688	7.08729	6.39872	7.42433
4	4	2.4	2.4	3.6	3.6	3	2.4	62.7751	6.79595	6.39872	7.13299
4	4	3.5	3.5	3.5	3.5	3.5	3.5	62.8619	7.43159	8.03426	6.8597
4	4	3.15	3.15	3.85	3.85	3.5	3.15	71.0049	7.15532	8.1156	6.58343
4	4	2.8	2.8	4.2	4.2	3.5	2.8	78.7232	7.55947	8.05897	6.98757

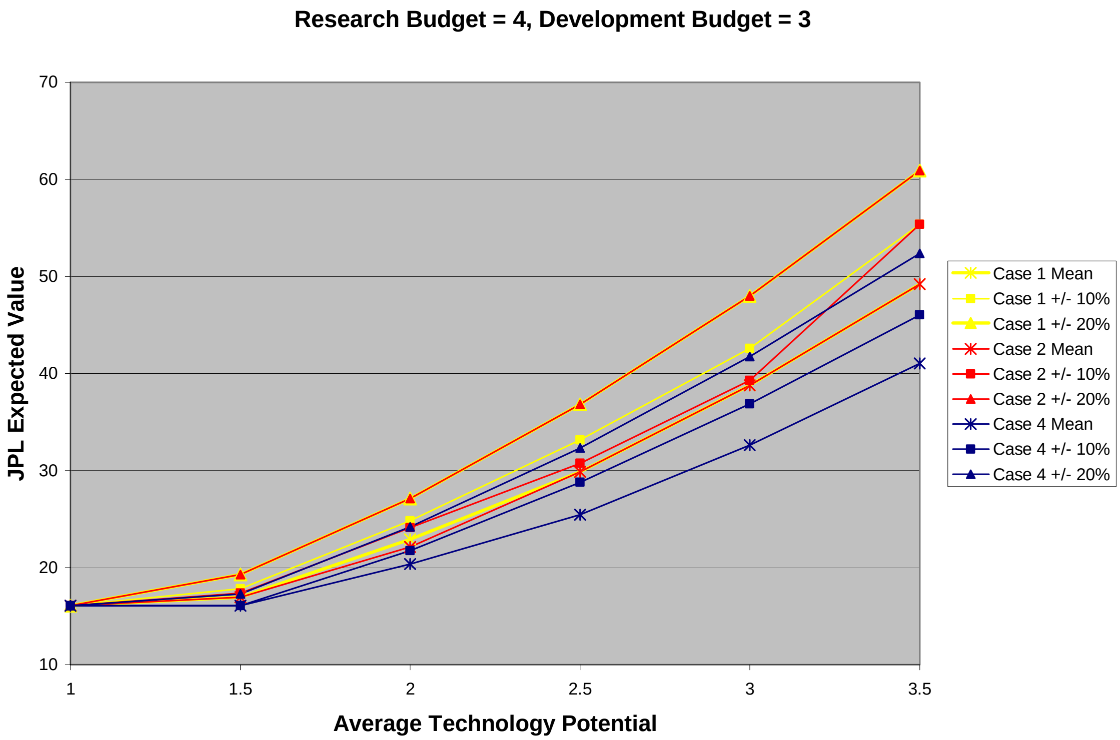
Case 3 Data

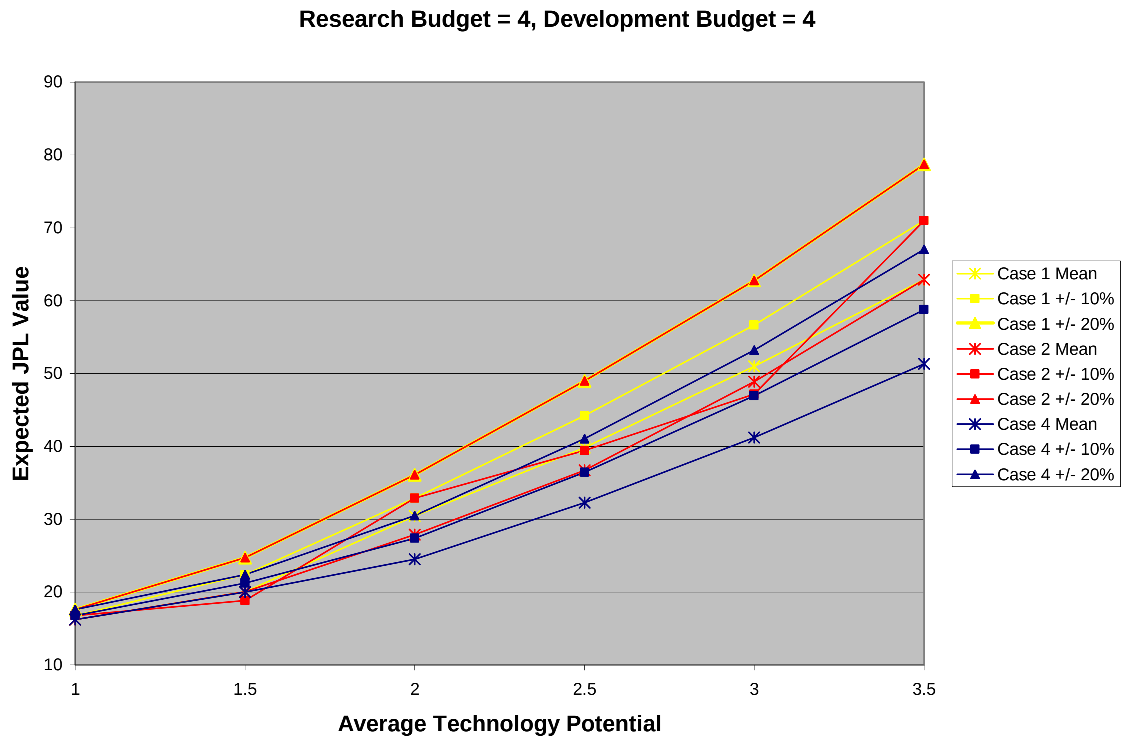
Res. Fund	Dev. Fund	SC1 M GL	SC1 M GL	SC3 M GL	SC1 P GL	SC2 P Gl	SC3 P Gl	JPL Expect	SC1 Expect	SC2 Expect	SC3 Expect
2	3	1	1	1	1	1	1	12	4	4	3
2	3	0.9	0.9	1.1	1.1	1	0.9	12	4	4	3
2	3	0.8	0.8	1.2	1.2	1	0.8	12	4	4	3
2	3	1.5	1.5	1.5	1.5	1.5	1.5	12	4	4	3
2	3	1.35	1.35	1.65	1.65	1.5	1.35	12	4	4	3
2	3	1.2	1.2	1.8	1.8	1.5	1.2	12	4	4	3
2	3	2	2	2	2	2	2	12.6188	3.1547	4	4
2	3	1.8	1.8	2.2	2.2	2	1.8	12.6188	3.38564	4	4
2	3	1.6	1.6	2.4	2.4	2	1.6	13.2	5.42487	3	5.42487
2	3	2.5	2.5	2.5	2.5	2.5	2.5	13.9282	3.73205	3.73205	5
2	3	2.25	2.25	2.75	2.75	2.5	2.25	14.7615	4.02073	5	4.02073
2	3	2	2	3	3	2.5	2	15.5949	4.3094	5	4.3094
2	3	3	3	3	3	3	3	15.5949	4.3094	4.3094	5
2	3	2.7	2.7	3.3	3.3	3	2.7	16.5949	4.65581	5	4.65581
2	3	2.4	2.4	3.6	3.6	3	2.4	17.5949	5.00222	5	5.00222
2	3	3.5	3.5	3.5	3.5	3.5	3.5	17.2615	4.88675	4.88675	5
2	3	3.15	3.15	3.85	3.85	3.5	3.15	18.4282	5.2909	5	5.2909
2	3	2.8	2.8	4.2	4.2	3.5	2.8	19.5949	5.69504	5	5.69504
3	3	1	1	1	1	1	1	16	4	4	4
3	3	0.9	0.9	1.1	1.1	1	0.9	16	4	4	4
3	3	0.8	0.8	1.2	1.2	1	0.8	16	4	4	4
3	3	1.5	1.5	1.5	1.5	1.5	1.5	16	4	4	4
3	3	1.35	1.35	1.65	1.65	1.5	1.35	16	4	4	4
3	3	1.2	1.2	1.8	1.8	1.5	1.2	16	4	4	4
3	3	2	2	2	2	2	2	16	4.73205	4.1547	4.73205
3	3	1.8	1.8	2.2	2.2	2	1.8	16.8309	5.07846	4.1547	5.07846
3	3	1.6	1.6	2.4	2.4	2	1.6	17.6619	5.42487	4.1547	5.42487
3	3	2.5	2.5	2.5	2.5	2.5	2.5	20.366	5.59808	4.73205	5.59808
3	3	2.25	2.25	2.75	2.75	2.5	2.25	21.549	6.03109	4.73205	6.03109
3	3	2	2	3	3	2.5	2	22.7321	6.4641	4.73205	6.4641
3	3	3	3	3	3	3	3	25.3094	6.4641	5.3094	6.4641
3	3	2.7	2.7	3.3	3.3	3	2.7	26.9022	6.98372	5.3094	6.98372
3	3	2.4	2.4	3.6	3.6	3	2.4	28.495	7.50333	5.3094	7.50333
3	3	3.5	3.5	3.5	3.5	3.5	3.5	30.8301	7.33013	5.88675	7.33013
3	3	3.15	3.15	3.85	3.85	3.5	3.15	32.8905	7.93634	5.88675	7.93634
3	3	2.8	2.8	4.2	4.2	3.5	2.8	34.9509	8.54256	5.88675	8.54256
4	3	1	1	1	1	1	1	16.1	4	4	4
4	3	0.9	0.9	1.1	1.1	1	0.9	16.1	4	4	4
4	3	0.8	0.8	1.2	1.2	1	0.8	16.1	4	4	4
4	3	1.5	1.5	1.5	1.5	1.5	1.5	16.1	4	4	4
4	3	1.35	1.35	1.65	1.65	1.5	1.35	16.1	4	4	4
4	3	1.2	1.2	1.8	1.8	1.5	1.2	17.3024	4.38564	5.1547	4.38564
4	3	2	2	2	2	2	2	20.3923	4.52073	4.73205	6
4	3	1.8	1.8	2.2	2.2	2	1.8	21.7643	5.07846	6.3094	5.07846
4	3	1.6	1.6	2.4	2.4	2	1.6	24.2031	5.42487	6.3094	5.42487
4	3	2.5	2.5	2.5	2.5	2.5	2.5	25.4745	7.4641	5.59808	5.59808
4	3	2.25	2.25	2.75	2.75	2.5	2.25	28.8044	6.03109	7.4641	6.03109
4	3	2	2	3	3	2.5	2	32.3509	6.4641	7.4641	6.4641
4	3	3	3	3	3	3	3	32.6427	7.6188	6.4641	6.4641

4	3	2.7	2.7	3.3	3.3	3	2.7	36.8924	6.98372	8.6188	6.98372
4	3	2.4	2.4	3.6	3.6	3	2.4	41.7457	7.50333	8.6188	7.50333
4	3	3.5	3.5	3.5	3.5	3.5	3.5	41.0235	8.7735	7.33013	7.33013
4	3	3.15	3.15	3.85	3.85	3.5	3.15	46.0282	7.93634	9.7735	7.93634
4	3	2.8	2.8	4.2	4.2	3.5	2.8	52.3876	8.54256	9.7735	8.54256
5	3	1	1	1	1	1	1	16.2	4	4	4
5	3	0.9	0.9	1.1	1.1	1	0.9	16.8	4.34641	4	4.34641
5	3	0.8	0.8	1.2	1.2	1	0.8	17.6	4.69282	4	4.69282
5	3	1.5	1.5	1.5	1.5	1.5	1.5	20	5.73205	5.73205	4
5	3	1.35	1.35	1.65	1.65	1.5	1.35	21.2	6.25167	4	6.25167
5	3	1.2	1.2	1.8	1.8	1.5	1.2	22.4	6.77128	4	6.77128
5	3	2	2	2	2	2	2	24.5157	7.4641	7.4641	4.1547
5	3	1.8	1.8	2.2	2.2	2	1.8	27.3668	6.77128	7.4641	5.07846
5	3	1.6	1.6	2.4	2.4	2	1.6	30.5058	7.23316	7.4641	5.42487
5	3	2.5	2.5	2.5	2.5	2.5	2.5	32.2993	9.19615	9.19615	4.73205
5	3	2.25	2.25	2.75	2.75	2.5	2.25	36.4436	8.04145	9.19615	6.03109
5	3	2	2	3	3	2.5	2	41.0313	8.6188	9.19615	6.4641
5	3	3	3	3	3	3	3	41.2376	10.9282	10.9282	5.3094
5	3	2.7	2.7	3.3	3.3	3	2.7	46.9176	9.31162	10.9282	6.98372
5	3	2.4	2.4	3.6	3.6	3	2.4	53.2196	10.0044	10.9282	7.50333
5	3	3.5	3.5	3.5	3.5	3.5	3.5	51.3306	12.6603	12.6603	5.88675
5	3	3.15	3.15	3.85	3.85	3.5	3.15	58.7889	10.5818	12.6603	7.93634
5	3	2.8	2.8	4.2	4.2	3.5	2.8	67.0707	11.3901	12.6603	8.54256
2	4	1	1	1	1	1	1	16	4	4	4
2	4	0.9	0.9	1.1	1.1	1	0.9	16	4	4	4
2	4	0.8	0.8	1.2	1.2	1	0.8	16	4	4	4
2	4	1.5	1.5	1.5	1.5	1.5	1.5	16	4	4	4
2	4	1.35	1.35	1.65	1.65	1.5	1.35	16	4	4	4
2	4	1.2	1.2	1.8	1.8	1.5	1.2	16	4	4	4
2	4	2	2	2	2	2	2	16	4	4	4
2	4	1.8	1.8	2.2	2.2	2	1.8	16.2641	5.07846	4	5.07846
2	4	1.6	1.6	2.4	2.4	2	1.6	17.1138	3.61658	4.73205	6
2	4	2.5	2.5	2.5	2.5	2.5	2.5	17.6603	3.73205	4.73205	6
2	4	2.25	2.25	2.75	2.75	2.5	2.25	19.0263	4.02073	4.73205	6
2	4	2	2	3	3	2.5	2	20.3923	4.3094	4.73205	6
2	4	3	3	3	3	3	3	20.3923	4.3094	5.3094	6
2	4	2.7	2.7	3.3	3.3	3	2.7	21.6766	4.65581	8	4.65581
2	4	2.4	2.4	3.6	3.6	3	2.4	25.0222	5.00222	8	5.00222
2	4	3.5	3.5	3.5	3.5	3.5	3.5	23.8803	4.88675	4.88675	8
2	4	3.15	3.15	3.85	3.85	3.5	3.15	27.9936	5.2909	8	5.2909
2	4	2.8	2.8	4.2	4.2	3.5	2.8	30.9231	5.69504	8	5.69504
3	4	1	1	1	1	1	1	16.1	4	4	4
3	4	0.9	0.9	1.1	1.1	1	0.9	16.1	4	4	4
3	4	0.8	0.8	1.2	1.2	1	0.8	16.1	4	4	4
3	4	1.5	1.5	1.5	1.5	1.5	1.5	16.1	4	4	4
3	4	1.35	1.35	1.65	1.65	1.5	1.35	16.1	4	4	4
3	4	1.2	1.2	1.8	1.8	1.5	1.2	17.3024	4.38564	5.1547	4.38564
3	4	2	2	2	2	2	2	20.3923	4.52073	4.73205	6
3	4	1.8	1.8	2.2	2.2	2	1.8	21.7643	5.07846	6.3094	5.07846
3	4	1.6	1.6	2.4	2.4	2	1.6	24.2031	5.42487	6.3094	5.42487
3	4	2.5	2.5	2.5	2.5	2.5	2.5	25.4745	7.4641	5.59808	5.59808
3	4	2.25	2.25	2.75	2.75	2.5	2.25	28.8044	6.03109	7.4641	6.03109
3	4	2	2	3	3	2.5	2	32.3509	6.4641	7.4641	6.4641

3	4	3	3	3	3	3	3	32.3509	8.6188	6.4641	6.4641
3	4	2.7	2.7	3.3	3.3	3	2.7	36.8924	6.98372	8.6188	6.98372
3	4	2.4	2.4	3.6	3.6	3	2.4	41.7457	7.50333	8.6188	7.50333
3	4	3.5	3.5	3.5	3.5	3.5	3.5	40.1089	7.98483	8.33013	7.33013
3	4	3.15	3.15	3.85	3.85	3.5	3.15	46.0282	7.93634	9.7735	7.93634
3	4	2.8	2.8	4.2	4.2	3.5	2.8	52.3876	8.54256	9.7735	8.54256
4	4	1	1	1	1	1	1	16.2	4	4	4
4	4	0.9	0.9	1.1	1.1	1	0.9	16.8	4.34641	4	4.34641
4	4	0.8	0.8	1.2	1.2	1	0.8	17.6	4.69282	4	4.69282
4	4	1.5	1.5	1.5	1.5	1.5	1.5	20	5.73205	5.73205	4
4	4	1.35	1.35	1.65	1.65	1.5	1.35	21.2	6.25167	4	6.25167
4	4	1.2	1.2	1.8	1.8	1.5	1.2	22.4	6.77128	4	6.77128
4	4	2	2	2	2	2	2	24.5157	7.4641	7.4641	4.1547
4	4	1.8	1.8	2.2	2.2	2	1.8	27.3668	6.77128	7.4641	5.07846
4	4	1.6	1.6	2.4	2.4	2	1.6	30.5058	7.23316	7.4641	5.42487
4	4	2.5	2.5	2.5	2.5	2.5	2.5	32.2993	9.19615	9.19615	4.73205
4	4	2.25	2.25	2.75	2.75	2.5	2.25	36.4436	8.04145	9.19615	6.03109
4	4	2	2	3	3	2.5	2	41.0313	8.6188	9.19615	6.4641
4	4	3	3	3	3	3	3	41.2376	10.9282	10.9282	5.3094
4	4	2.7	2.7	3.3	3.3	3	2.7	46.9176	9.31162	10.9282	6.98372
4	4	2.4	2.4	3.6	3.6	3	2.4	53.2196	10.0044	10.9282	7.50333
4	4	3.5	3.5	3.5	3.5	3.5	3.5	51.3306	12.6603	12.6603	5.88675
4	4	3.15	3.15	3.85	3.85	3.5	3.15	58.7889	10.5818	12.6603	7.93634
4	4	2.8	2.8	4.2	4.2	3.5	2.8	67.0707	11.3901	12.6603	8.54256




Research Budget = 4, Development Budget = 3
JPL Expected Value
70
60
50
40
30
20
10
1
1.5
2
2.5
3
3.5
Average Technology Potential
Case 1 Mean
Case 1 +/- 10%
Case 1 +/- 20%
Case 2 Mean
Case 2 +/- 10%
Case 2 +/- 20%
Case 4 Mean
Case 4 +/- 10%
Case 4 +/- 20%



Research Budget = 4, Development Budget = 4
Expected JPL Value
90
80
70
60
50
40
30
20
10
1
1.5
2
2.5
3
3.5
Average Technology Potential
Case 1 Mean
Case 1 +/- 10%
Case 1 +/- 20%
Case 2 Mean
Case 2 +/- 10%
Case 2 +/- 20%
Case 4 Mean
Case 4 +/- 10%
Case 4 +/- 20%

APPENDIX C:
FINAL PRESENTATION GIVEN AT DARPA ON JUNE 25, 2003

Final Report
& Technology Demonstration
of the pre-GAMBIT effort

Deliverable #6 to DARPA
Under contract #F30602-02-C-0078

25 June 2003

Overview of pre-GAMBIT

DARPA interest in a strategic-reasoning simulation tool

- SR: How self-interested actors decide what to do with other such actors
- Simulation Motive: Reality is a subset of strategic reasoning and today's threats do not fit yesterday's relatively stable experience base.

Game-Theory **B**ased **I**nformation **T**echnology

- Game Theory is the formal study of strategic reasoning.
- Any game-theoretic simulation tool would be implemented with IT.

Goals when funded in June 2002: (all accomplished)

- Assess readiness of game theory for scenarios of interest to DARPA
- Configure a distributed agent IT tool for mixed human/software gaming
- Demonstrate this tool in a known game-theoretic scenario
- Recommend path forward

Conclusion on the Readiness of Game Theory

Scenarios of interest to DARPA

- Staged processes; e.g., logistics in support of operations
- Coalition formation and stability
- Attacks on civil economic systems

General Model: *Games* within & across hierarchies

- Mechanism Design involves strategic reasoning within hierarchies
- *Classic Games* involve top nodes of hierarchies (same level - *flat*)
- SQ: No treatment of mixed system & separate treatments too limited

Needed: Develop one case of *Clash of Organizations*

- Common task to focus efforts of various groups using game theory
- Simulation tool for a specific, representative scenario (*Diplomacy*)
- With one instance well-understood, then generalize to full GAMBIT.

The Readiness of IT

GAMBIT scenario would involve 100 to 1,000 actors

- Impractical to require co-location (in place or even in time)
- Impractical to require all the actors to be humans
- Desirable if human actors could be substituted for by s/w agents

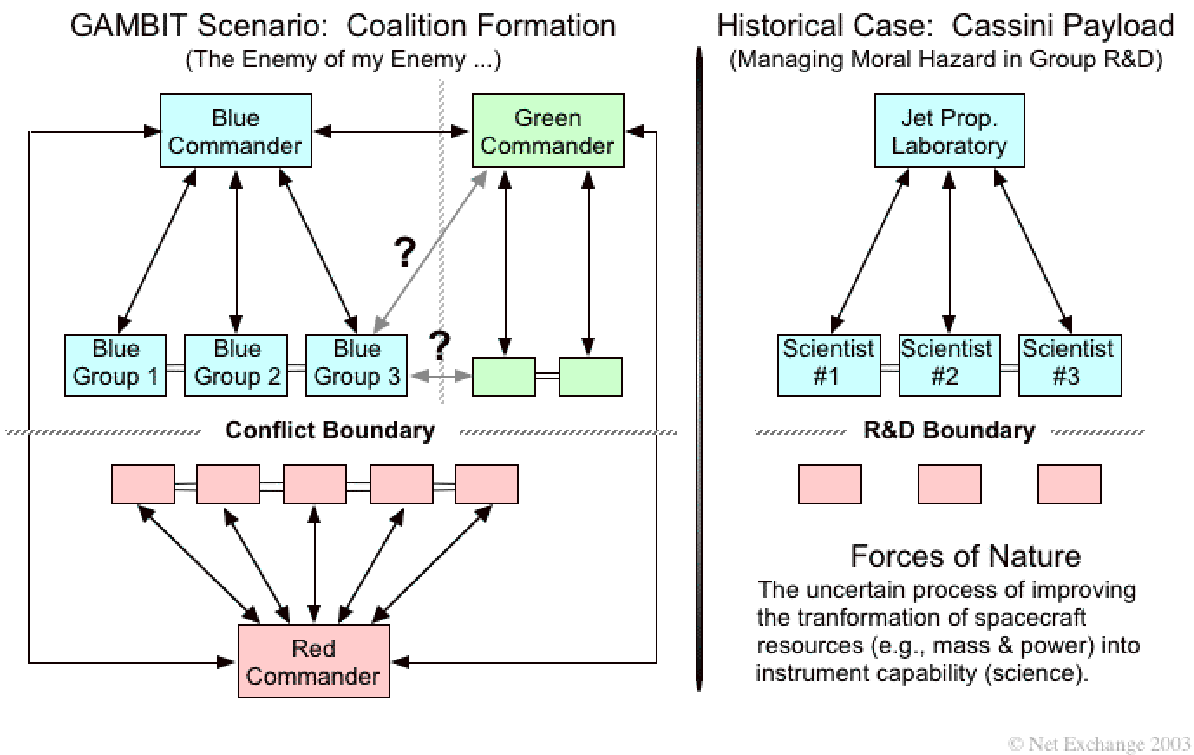
Substantial processing handled by each node

- Mirrors the strategic reasoning that is distributed in reality
- Required for scenarios of this scope to be practical

Needed: Development of strategic software agent

- Able to assess its position in an organization/game
- Pursuit of self-interest within the constraints of its position
- Quality Measure: whether human players can discern it's software

GAMBIT Scenario and analogous Historical Case



Historical Demonstration -- Cassini Instrument R&D

R&D for Cassini Saturn mission faced a hard $ limit

- When trading off mass, power, and funds, funds were always *loose*.
- Congressional cancellation of sister mission due to cost increase signaled that funds would not be a loose variable for Cassini.
- Economists supplied a game-theoretic R&D management approach -- decentralized control over resources (property rights + trading).
- Cassini launched on budget, on time, and with all instruments onboard.

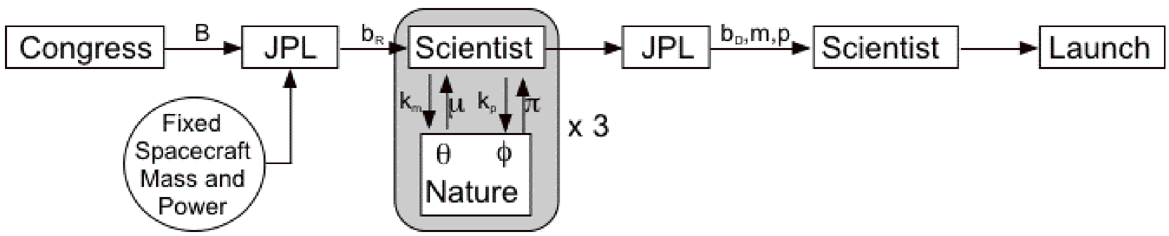
Justification for Cassini as pre-GAMBIT Demonstration

- Hierarchy cooperating on a group task subject to the self-interest of its members *versus* an opponent (Nature doesn't share the group's interest)
- Tractable with existing theory and can be compared to ground truth
- Implementable using existing distributed agent architecture (Cougaar)

Our Model of Cassini

Cassini as a two-phased project

- Research: investments into mass-use and power-use technologies
- Development: mass, power, & funds converted into instruments



JPL wants to maximize the use of mass, power, funds relative to

$$V_{JPL} = \min\{S_1S_2, S_1S_3, S_2S_3\} + \beta(\text{Budget} - \text{Expenditure})$$

Each of three instrument scientists performing R&D want to max

$$V_{Sc} = f(S, \$) = k_D\left[\mu(k_m,\theta)m + \pi(k_p,\phi)p\right] - \gamma k_D^2 + \varepsilon(\text{residual } \$)$$

Four Cases Examined

#1: Monolithic -- Scientist Robots (*first-best* solution)

- The problem w/o moral hazard -- the scientists are not self-interested
- Establishes the benchmark for the best outcome JPL can expect

#2: Agency with Full Information

- JPL can costlessly observe Research outcomes.
- Scientists know that Research outcomes affect Development allocations; therefore, they play a maxmin strategy.

#3: Agency with costly monitoring of Research

#4: Agency with property rights and trading

- JPL endows Scientists with (b_R, b_D, m, p) before Research
- Scientists allowed to trade resources for mutual benefit

Demonstration Results -- Overview

Model implemented w/i modified Cougaar framework

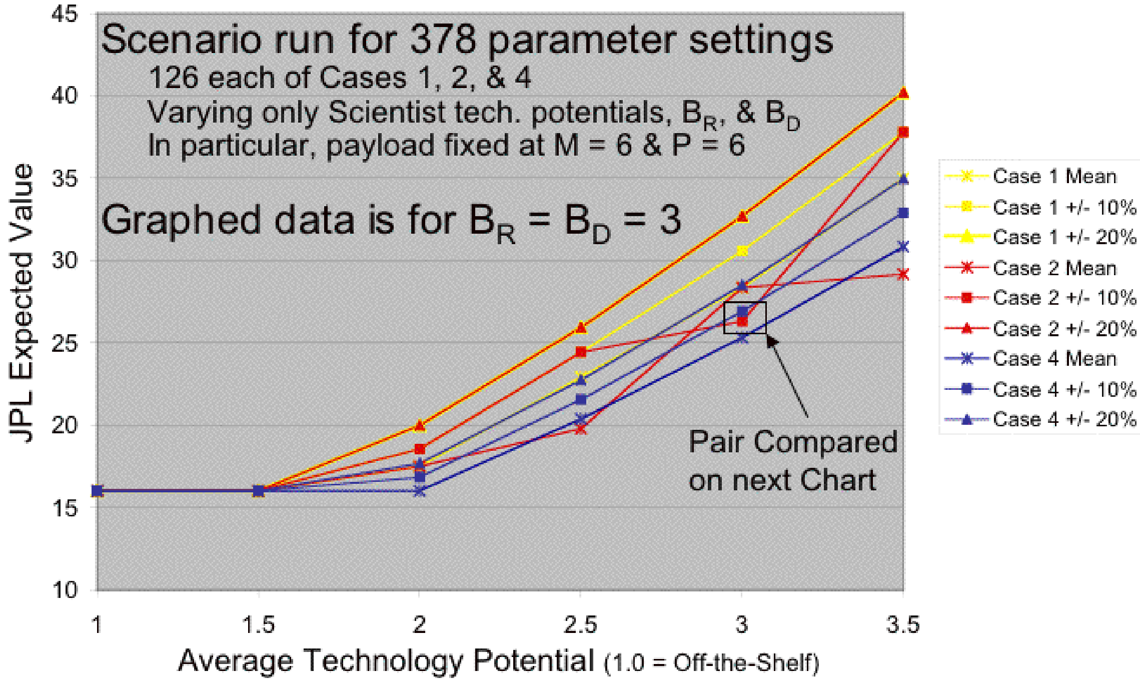
- Actors for JPL, three Scientists, Nature, and an Exchange
- Communication *plumbing* supplied by Cougaar architecture
- *Reasoning* components incorporated via *plugins*

Results mirror historical observations

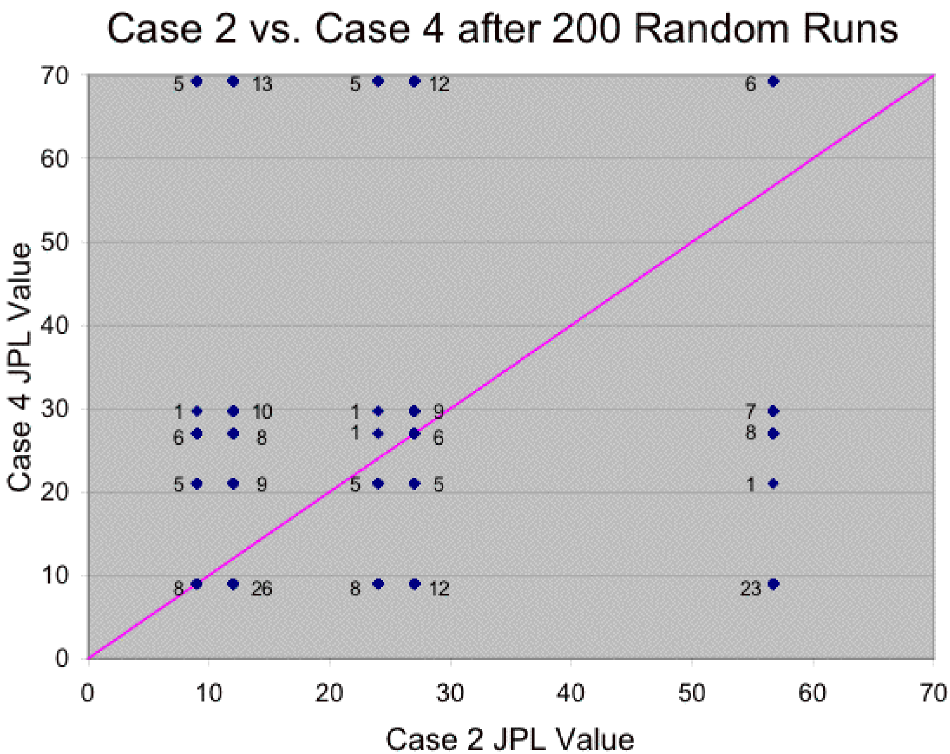
- With tight budgets and substantial research potential there is a substantial moral hazard problem
- Agency with Property Rights can be a superior *Second Best* solution

There is nothing special in the results, which supports the goal of implementing a known example.

Demonstration Results -- Sample Data Graphs



Demonstration Results -- Sample Data Graphs



Commentary on Limitations and Directions Forward

Cassini was run using a classic game theory approach

- All actors were fully rational -- complete dynamic program using all available information and unconstrained by computational resources
- General graph -- all nodes maintained equally aware interconnections

Current approaches to game theory that offer direction

- Structured graphs, directed graphs, graphs with neighborhoods
- Bounded rationality, resource constrained learning, dynamic neighborhood definition/assessment.

Current computer science approaches offering direction

- Theory: Distributed Algorithmic Mechanism Design
- Applied: Distributed Agent Architecture

Next Steps -- The Diplomacy Test Utility

Make use of *Diplomacy*®

- Established strategic game with large, on-line player base
- Well-defined and simple game language
- Large library of player strategy studies and commentaries

A CERN for Strategic Reasoning: People not Particles

- Stage I: Implement the multilateral, multi-period coalition game
- Stage II: Implement a supply, logistics, and command hierarchy below each national node in the coalition game
- Stage III: Allow communication and interaction across hierarchies below the national node level

The utility is produced, qualified entry of s/w agents is supported, & public on-line human play is utilized.

Review of the game *Diplomacy*®

Diplomacy® is a multi-turn coalition formation game

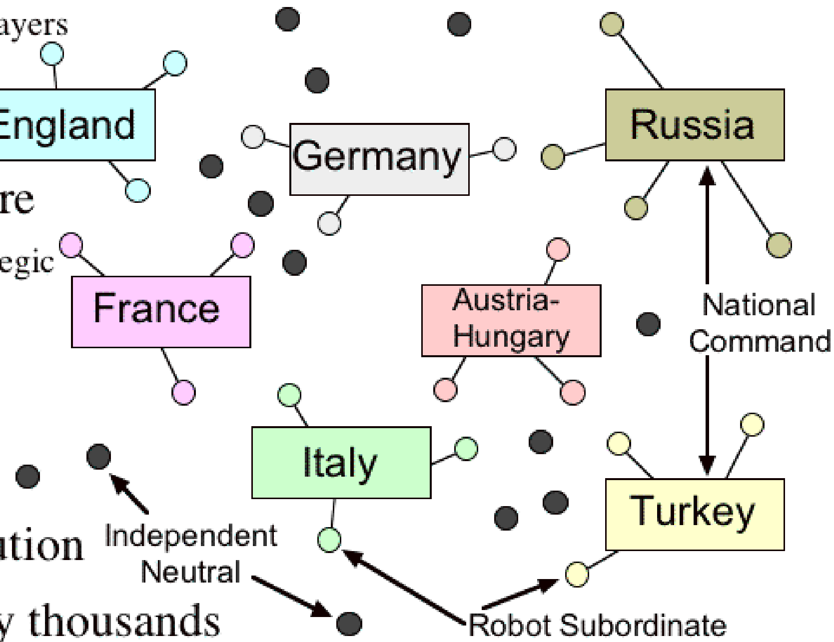
- Europe 1900: 7 Players vie to control 50% of nodes.

Strategic Structure

- Nodes are not strategic
  - Subordinate
  - Neutral
  - Adversary
- Dynamic but *Flat*

Objective Resolution

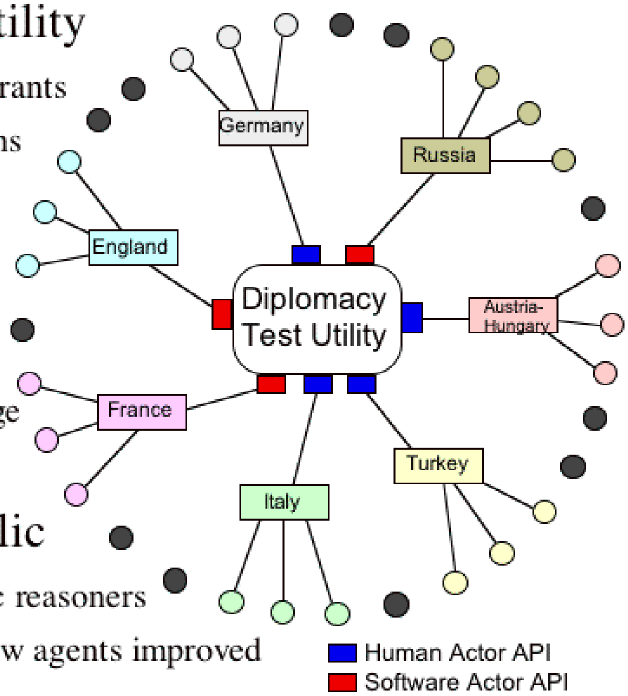
Played on-line by thousands



DTU Stage 1 -- Traditional, Online with Dual API

Basic Diplomacy Test Utility

- Bookkeeping for human registrants
- Cycling of game phases & turns
- Human and software APIs
- S/W actor nomination rules
- Policing allowable moves
- Outcome resolution
- Structured negotiation language
- Archiving & archive access

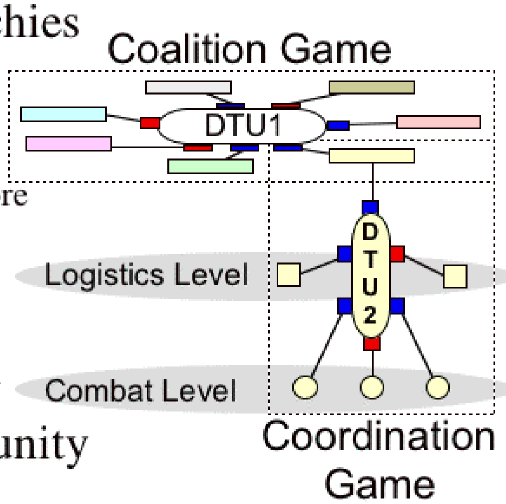


DTU1 is open to the public

- Large, trained pool of strategic reasoners
- New theory tested, strategic s/w agents improved

DTU Stage 2 -- Nations as Isolated Hierarchies

- Introduce an authority hierarchy within each nation
  - Nodes are self-interested actors -- command constrained by incentives
  - Two types of nodes: combat level of the traditional game & logistics
- Simplification: Isolate Hierarchies
  - Coalition Game among national heads
  - Coordination Game w/i each nation
  - DTU1 services coalition game, as before
  - DTU2 services coordination game
  - Approachable with current theory
- Extend human & s/w approach
- Continue to use on-line community



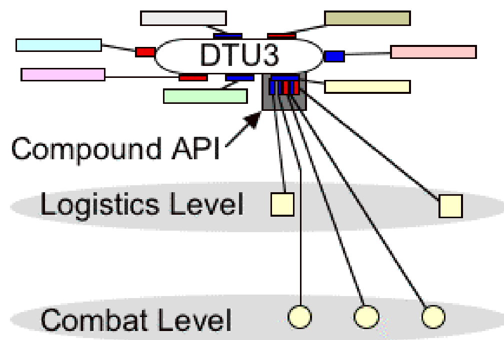
DTU Stage 3 -- Clash of Hierarchies (GAMBIT)

Expose actors within a national hierarchy to actors in other national hierarchies

- Strategies made possible: interdiction of supply, espionage, treason
- Stability would require cost and/or benefit structures that favor established relationships

Requirements for DTU3

- A compound API that maintains national structure yet *exposes* nodes
- Nodes must be able to switch sides
- A deeper negotiation language



An instance of GAMBIT

Stage 3 requires & will result in substantial new theory.