USING GRAPH-THEORETIC MODELS TO ANALYZE AND MANAGE THE COMPLEXITY OF THE DESIGN OF NAVAL SURFACE SHIPS

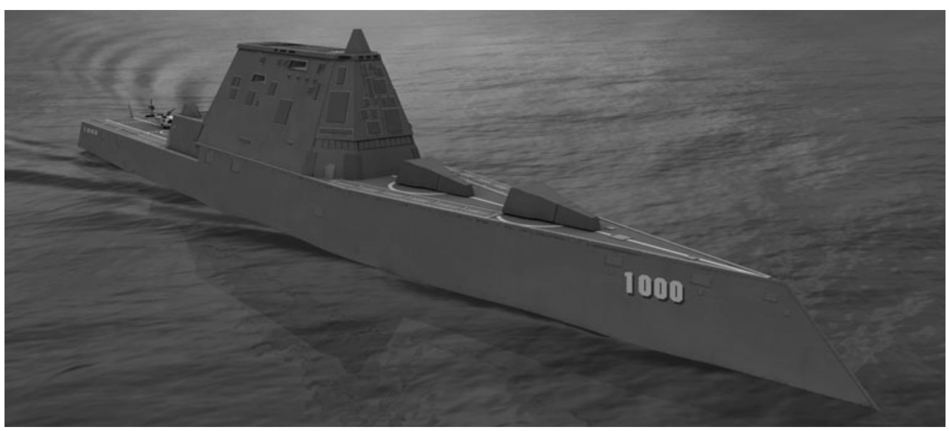
Jay Borthen, E.I.

INTRODUCTION

Naval ships are inherently complex. They must provide all of the necessary means to maintain life at sea for months at a time while simultaneously being able to serve their purpose. Depending on what a ship's purpose actually is, a wide range of systems may or may not be equipped, including weapons, defensive countermeasures, radars, sonars, communication antennas, research equipment, aviation-related equipment, and a multitude of other support systems. All of which are constrained in both space and weight.

As one can quickly conclude, these different systems interact with one another in various ways and in varying degrees. In fact, one of the most difficult aspects of designing any ship is quantifying and tracking the interactions between the numerous systems onboard, both from a functional perspective and from an arrangements point-of-view. Ship designers are often forced to find creative ways to deal with the intricate choreography and planning required to complete the design in an efficient and effective way.

One commonplace approach used in the design of ships today, at least from a topological perspective, is to "divide and conquer". That is, to partition the ship into easier to manage pieces, or *zones*, similar to what is shown in Figure 1, and focus attention on one zone at a time. Unfortunately, however, as history has proven, this method can sometimes lead to costly misalignments in inter-zonal systems such as piping, wireways, ducting, and even primary structure. A supplementary tactic is needed that emphasizes the interdependence of the zones and manages their complexity in a more concrete manner.



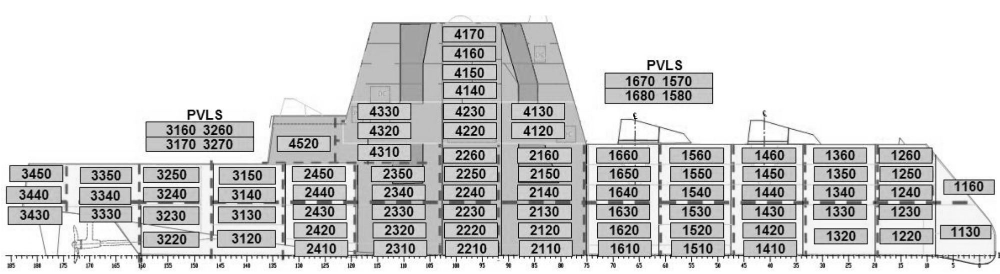


Figure 1: Profile View of a Ship Divided Into Zones

GRAPHING COMPLEXITY

One expert (Billingsley) estimates naval ship complexity in terms of parts-count as being 100 times that of a typical aircraft and 1,000 times that of a typical industrial plant. The complexity of a ship should account for more than just a parts-count though. It should take into consideration the relationships between the systems that those parts construct. Below is a screenshot of a rough system-level arrangement of a surface ship's machinery room which illustrates how compact and dense a ship can physically be. Moving any piece of equipment will almost surely impact other pieces of equipment.

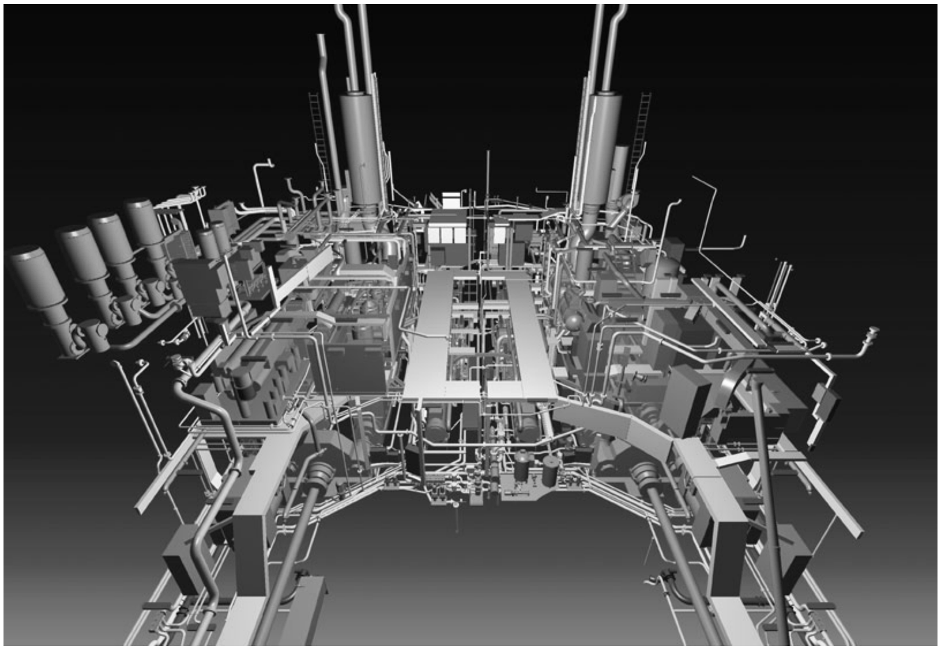


Figure 2: A System-Level Arrangement of a Ship's Machinery Room

Above all, organization is critical and because of this we start by categorizing all the parts and systems in the ship into a discrete set of *design disciplines*, D, such that

D = {Insulation, Electrical Equipment, Foundations, HVAC, Machinery Equipment, Pipe, Primary Structure, Wireways, Outfit & Furnishings}

and we let each element of D be a vertex in a graph, G. The set of edges, E, between the vertices in G will then represent the relationships between each design discipline as shown in Figure 3.

For our discussion we consider only the relationships regarding the physical arrangement of the ship and assume that functional design has been completed and is fixed. It should also be noted that D will vary for each individual ship depending on the type of ship as well as on designer preference.

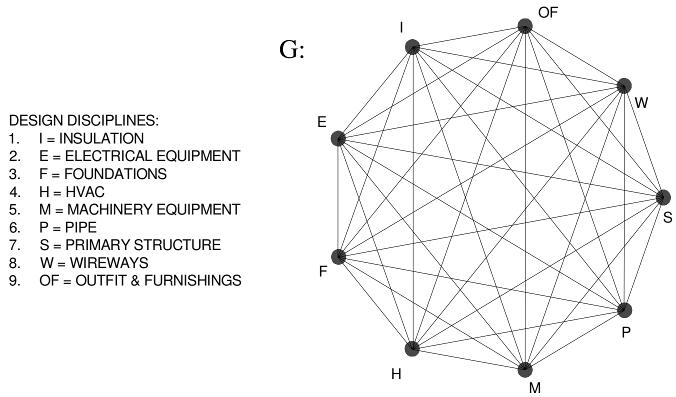


Figure 3: A Simple Complete Graph with Nine Vertices

The cardinality, |D|, of our design discipline set is equal to 9 and because the graph in Figure 3 is a simple complete graph in which one and only one edge is present between vertices, the number of edges, |E|, in the graph G is

$$|E| = {}_{|D|}C_2 = \frac{|D|!}{2!(|D|-2)!} = \frac{|9|!}{2!(|9|-2)!} = \frac{9*8}{2} = 36$$

This can also be shown via the First Theorem of Graph Theory,

$$|E| = \frac{1}{2}\sum_{d\in D} deg(d) = \frac{1}{2}|D|(|D|-1) = \frac{1}{2}(9)(8) = 36$$

Every vertex is connected to every other vertex so the graph is necessarily *regular* where

$$deg(d)_{d\in D} = (|D|-1)$$

If we now replace our set of edges, E, "with a set of ordered pairs of vertices, we obtain a directed graph, or digraph" (Harris et al. pg. 3). An example of this is shown in Figure 4. Each edge has two possible orientations and there are a total of 36 edges. So the number of possible complete digraphs, N, with nine vertices like our graph G_D in Figure 4 will be

$$\text{N} = 2^{36} = 68,719,476,736$$

This is, clearly, too many to reasonably consider. Therefore, assumptions are needed from our design processes in order to decrease this number. By reducing the number of design disciplines, assuming the set of disciplines is not completely connected in every occurrence, and/or fixing the orientation of the ordered-pairs the cardinality of the set of edges, E, will decrease.

In any case, the orientation of each edge in our digraph represents a *domination* relationship between the respective ordered pairs. For example, the directed arc between the vertices H and OF, in Figure 4 indicates that vertex H dominates vertex OF because the directed arc is pointing toward vertex OF. Vertex H is also dominating the vertices F, W, and P and is being dominated by vertices E, I, M, and S. The *king* of a complete digraph, or *tournament*, is the vertex that has the greatest out-degree. The out-degree of the vertex S is the maximum it can possibly be, that is

$$\text{out-deg(S)} = \Delta(\text{G}) = (|D|-1)$$

Where Δ(G) is the maximum degree in the underlying graph of G_D. Therefore, vertex S must be a king. The vertex H is dominating four vertices and therefore has an out-degree of 4. It is being dominated by four other vertices, so it, likewise, has an in-degree of 4. We can conclude that the vertex H is not a king because

$$\text{out-deg(H)} < \text{out-deg(S)}$$

There can be multiple kings in a tournament. If multiple kings are present, however, then the tournament is not *transitive* and therefore must contain a cycle, which, in the context of ship design, amounts to the possibility of design iterations which cause cost increases and schedule delays.

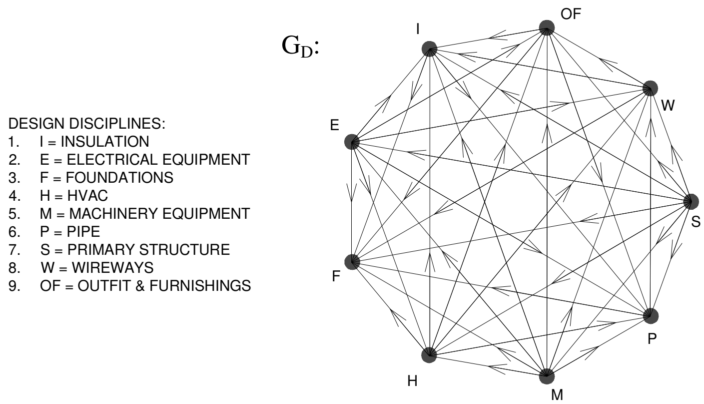


Figure 4: A Complete Digraph

Digraphs are used in many areas where analytics are employed including computer science, sociology, civil engineering, and biology and are frequently found in other fields of mathematics such as game theory, set theory, and combinatorics. Using digraphs in any situation can become tedious and time consuming if the number of vertices and edges are not restricted in some manner. Using the methods outlined above and perhaps even performing a screening experiment with empirical data from previous ship design efforts may be valuable in controlling a data explosion. A screening experiment would help determine exactly what interactions could be neglected. For our purposes we will make some subjective assumptions regarding our set of edges and the number of vertices.

INTEGRATING GRAPHS AND INVARIANTS WITH THE DESIGN PROCESS

The ship's zonal breakdown (Figure 1) is a good place to start our discussion of the integration of graphs with ship design. Obviously, each of the ship's zones interfaces with at least one other zone. This allows us to portray the neighborhood of a zone with a nontrivial *tree*. If we take, for instance, Zone 1620 from the ship profile in Figure 1, its neighborhood can be modeled as in Figure 5.

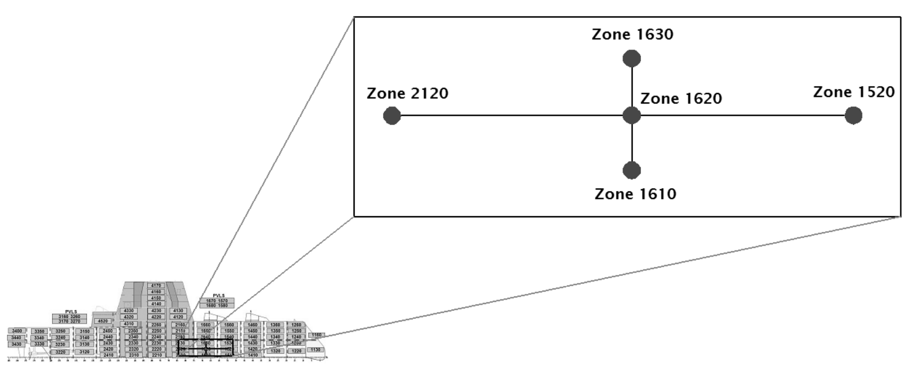


Figure 5: The Neighborhood of Zone 1620

Each zone will contain a subset, D_Z, of elements from our original set of design disciplines, D, such that

$$D_z \subseteq D$$

D_z will depend on the spaces that are in each zone. For example, a zone may contain a galley and mess in which case the machinery discipline can be neglected unless there is a large piece of the propulsion system located in the middle of the dinning area.

Let's take for instance our neighborhood from Figure 5 and say that Zone 1620 contains a galley and mess, Zone 1630 contains the Ship Mission Control, Zone 1520 contains crew staterooms and a lounge, Zone 2120 is a machinery room, and Zone 1610 is a ballast tank. Each zone will have its own set of design disciplines that could be organized as follows.

- D_{1620} = {Insulation, Electrical Equipment, Foundations, HVAC, Pipe, Primary Structure, Wireways, Outfit & Furnishings} = D – {Machinery Equipment}
- D_{1630} = {Insulation, Electrical Equipment, Foundations, HVAC, Pipe, Primary Structure, Wireways, Outfit & Furnishings} = D – {Machinery Equipment}
- D_{1520} = {Insulation, Electrical Equipment, Foundations, HVAC, Pipe, Primary Structure, Wireways, Outfit & Furnishings} = D – {Machinery Equipment}
- D_{2120} = {Insulation, Electrical Equipment, Foundations, HVAC, Machinery Equipment, Pipe, Primary Structure, Wireways, Outfit & Furnishings} = D
- D_{1610} = {Pipe, Primary Structure, Wireways, Outfit & Furnishings} = D – {Insulation, Electrical Equipment, Foundations, HVAC, Machinery Equipment}

We now take our zonal subsets of design disciplines given above and embed them onto their respective vertices in the tree of Figure 5. This gives us the graph shown in Figure 6 and illustrates that Figure 5 is, in actuality, a representation of a set of sets. More specifically, it is a representation of a set of Directed Acyclic Graphs, or DAGs.

A DAG should exist within each zone. Forbidding the presence of directed cycles precludes design iterations from occurring. This constraint accurately reflects the philosophy of later ship design phases and also significantly reduces the number of permutations of our digraph G_D.

Another conclusion that we can draw from the assumption of Directed Acyclic Graphs is the necessity that at least one source and one sink be present within each zone. Hence, there will always be at least one discipline that dominates the zone and this discipline should take precedence over the other disciplines during Detailed Design and arrangements planning. Ultimately, defining the precedent-order of each discipline within each zone would help streamline the design and save time and money.

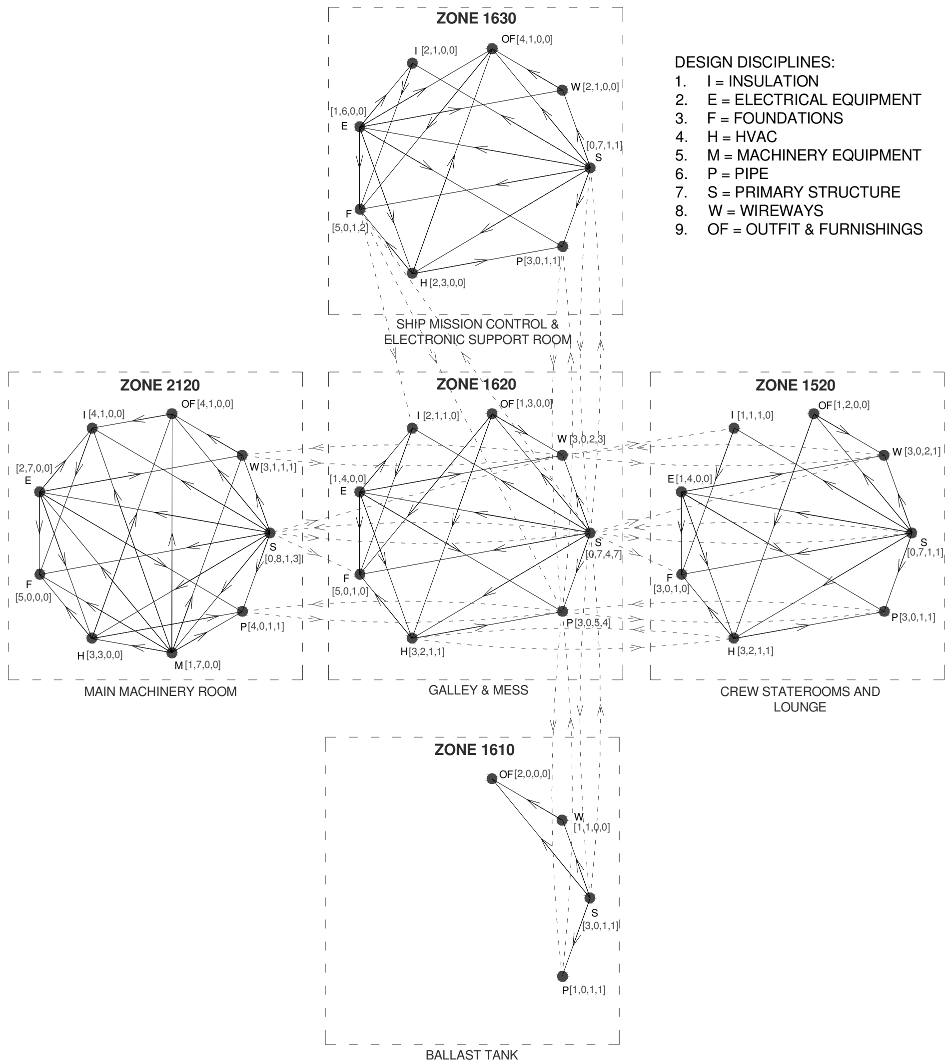


Figure 6: Intra-zonal and Inter-Zonal Complexity

It may also be prudent to employ concepts of *game theory* after establishing zonal DAGs. Setting incentives on the DAGs could potentially be used to sort out compromises in the design. For example, say that there is only enough room in a machinery space for one fan-coil assembly, which helps keep the machinery cool, or one electrical control panel, which is necessary to run the machinery. The open space is ideal for both pieces of equipment, but only one piece of equipment can be chosen to occupy the space. The piece of equipment that is not chosen will need to be relocated, away from its ideal location. Which piece of equipment should be selected?

Similar to the classic problem of the Prisoner's Dilemma, this situation will depend on the consequences of each option, the options being the installation of the fan-coil assembly, the installation of the electrical control panel, or neither. For a further discussion of game theory, the reader is directed to Myerson.

At each vertex in Figure 6, a degree matrix is displayed that quantifies the respective degree invariants. That is,

$$\text{Vertex Degree Matrix} = [a, b, c, d]$$

Where a = Intra-Zonal In-Degree, b = Intra-Zonal Out-Degree, c = Inter-Zonal In-Degree, and d = Inter-Zonal Out-Degree.

INTRA-ZONAL CONNECTIVITY

As the name implies, the concept of intra-zonal interactions concerns the interactions between design disciplines within a specific zone. The intra-zonal degree invariants that are present in our vertex degree matrices above neglect the edges that run between zones. The intra-zonal degree of each vertex in a zone is, however, separated into an in-degree and an out-degree, the sum of which gives the total intra-zonal relationship count for said vertex. The smallest of these sums represents the edge-connectivity, $\lambda(G_Z)$, of the specific zone's DAG. If we find the degree sequence, s_Z, of the underlying graph of Zone 1620, then the edge-connectivity of Zone 1620 will be equal to the minimum quantity in our sequence,

$$S_{1620}\text{: } 7,6,5,5,4,3,3,3$$

$$\lambda(G_{1620}) = 3$$

The higher the intra-zonal edge-connectivity is the more dependent the disciplines in a zone are on one another. An edge-connectivity of zero implies that there is either nothing in the zone or that there are no relationships between the systems within the zone. The maximum possible edge-connectivity given |D| vertices, is equal to (|D| - 1) which is the degree of each vertex in a simple complete graph. In our case, assuming nine design disciplines,

$$\max \lambda(G_Z) = \Delta(G) = (|D| - 1) = (9 - 1) = 8$$

INTER-ZONAL CONNECTIVITY

The inter-zonal degree invariants only take into account the edges that traverse the boundaries of the zones and neglect the edges within zones. Similar to intra-zonal connectivity, the inter-zonal connectivity is described in terms of the number of edges between vertices.

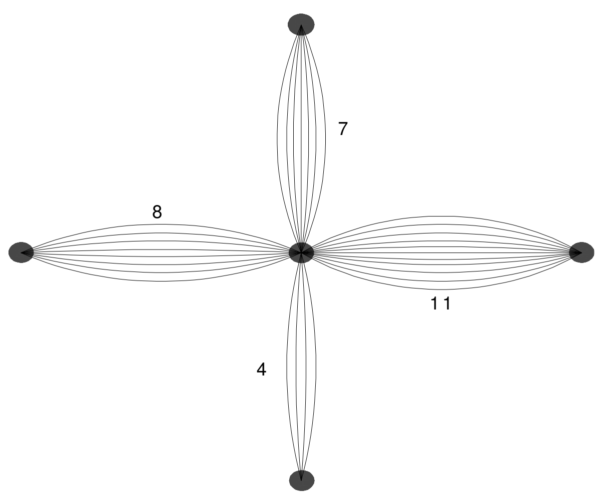


Figure 7: Inter-zonal Connectivity Strength

Every design discipline in our set D will not, and should not, be able to impact objects in adjacent zones. In other words, only certain disciplines within a zone can impact another zone's disciplines. For example, Outfit & Furnishings in one zone will not directly affect the arrangements in an adjacent zone. On the other hand, structural changes in one zone can significantly affect the arrangement of systems in a neighboring zone. Our set of inter-zonal design disciplines, D_I, (i.e. those disciplines that are able to impact neighboring zones) includes primary structure, wireways, pipe, HVAC, and foundations such that

$$D_I = \{\text{Primary Structure, Wireways, Pipe, HVAC, Foundations}\}$$

If we extract the inter-zonal edges from Figure 6 the multigraph in Figure 7 is created. The inter-zonal edge-connectivity provides a reasonable estimate for the relationship strength of a neighborhood. In addition, if we consider the in-degree and out-degree of each zone we can determine what zones have the most influence within the neighborhood.

CONCLUSION

Analyzing the complexity of a ship using graphs and graph-theoretic invariants may be helpful to consider during the transition between Contract Design and Detailed Design in order to help justify cost estimates and scheduling assessments. Graph-theoretic invariants could also help quantify design metrics that have previously been difficult to attain, such as the connectivity of the ship's systems from one zone to another. A further investigation is needed to verify the accuracy of these applications but graph theory has the potential to ultimately reduce risk and help predict problem trends within the design of naval ships.

REFERENCES

Billingsly, Daniel W. *Lessons Learned From Developing and Managing Naval Ship Information from Conception to Disposal*. ASNE Seminar. Washington, DC. 10 December 2009. Keynote address.

Cadmatic. *Nupas-Cadmatic Ship Design Software.* Web. 17 December 2009. <http://www.cadmatic.com/xml/ship_design/ship_design.html>.

Chartrand, G., and L. Lesniak. *Graphs & Digraphs (Fourth Edition).* Washington, DC: Chapman & Hall/CRC, 2005. Print.

Harris, John M., and Jeffry L. Hirst, and Michael J. Mossinghoff. *Combinatorics and Graph Theory*. New York: Springer, 2000. Print.

Myerson, Roger B. *Game Theory: Analysis of Conflict*. Cambridge, MA: Harvard University Press, 1991. Print.

Stanford University. *Stanford Encyclopedia of Philosophy*. Web. 11 December 2009. <http://plato.stanford.edu/entries/game-theory>.