Operationg Room Planning

Agent based system analysis and design

Eric Neher
Course Agent Systems
Blekinge Institute of Technology
Karlskrona, Sweden
eric.neher@students.fhnw.ch

Pascal Büttiker
Course Agent Systems
Blekinge Institute of Technology
Karlskrona, Sweden
pascal.buettiker@students.fhnw.ch

***Abstract*—This paper analyzes the usage of a Multi Agent System (MAS) applied to the Operating Room Planning problem, showing the benefits and shortcomings of such an approach. The Operation Room Planning (ORP) describes an optimization problem which is suited in a dynamic and complex environment. These characteristics match the motivation for Multi Agent Systems. For the design process of the MAS the Gaia methodology is used. Furthermore, this paper points out some ethical problems, which may occur when an autonomous learning machine is used in such a critical environment like a hospital.**

Keywords—Agents; MAS; design; architecutre; learning machines;

I. Introduction

The aim of this report is to present analysis and design of a multi agent system (MAS), which is capable of handling the "Operating Room Planning" problem and giving a proper motivation why to use MAS to solve this problem.

II. Purpose of agent system

The Operating Room Planning (ORP) problem describes the complex dynamic environment of a hospital operating room setting in which patients in general share a common goal: getting well as soon as possible. Survival is one of the strongest instincts of a human which means that a patient, who finds himself trapped in an environment such as the ORP describes, his desires are pretty straightforward and normally don't change. In contrast, his believes and intentions may change. Unfortunately, once a client falls into the custody of a hospital, his believes and intentions become irrelevant.

The problem of the ORP, which has to be solved, is therefore not managing the individual wishes or handle the clients differently because of their distinction in believe. It is about health and death and how latest can be minimalized, which becomes more and more difficult due to increasing surgery demand. Health care is a huge building which requires several administrative instances and since there are numerous clients and just limited resources a clever client management system is inevitable.

Each client has its own health status and his own path of welfare it has to go through. In order to achieve the ORP goal, maximize the success rate of surgeries and minimalize the cost, a very simple but effective method has to be applied: Prioritization. Unfortunately the prioritization alone does not solve the problem hospitals are confronted with. The most difficult part it to manage a plan of the allocation of resources, which allows increasing efficiency, flattening out peak demands and save costs. The ORP is therefore a highly distributed and dynamic optimization problem.

The facts that the patient status can change at any time, as well as the applied medical measurements are everything else than predictable make the environment very dynamic. Further, the pure size of all affected entities, the wide distribution of the problem and finally the interconnected dependencies of all the required resources make this problem very complex. All these characteristics are hard to handle by a central organized system.

In this paper we will discuss a multi agent system approach, which can reduce the overall system complexity by splitting up the problems to different agents and solve the dynamic nature by using autonomous agents capable of handling new situations themselves in order to fulfill the system requirements and produce a plan, which achieves the design objectives.

Departments have personal interests (reduce the own waiting list), common interests (help as many patients as possible, fast as possible) and also require different kind of resources, which they have to share with other departments. These circumstances lead us to declare the departments as agents, which are able to state their desires and communicate (either to cooperate with other departments, or to get resources from resource providers). The purpose of the agents is therefore to interact with each other in a dynamic environment in order to achieve a local as well as global optimum.

III. Agent Architecture

A. General Agent System Architecture

The Operating Room Planning problem contains two major challenges: Creating an intelligent system, which is capable of planning an optimal scheduling and is also able to react in emergency cases. Therefore, such an agent system has to handle the general patient cases, referred as long term planning, where an optimal schedule would be the resulting artifact. Emergency cases, referred as short term planning, must also be handled and are always higher prioritized than the general patient cases. These cases result more in immediate usage of resources rather than planning.

Due to the dynamic nature of patient cases (e.g. a general case can always become an emergency case), the agent system needs a reactive part for handling emergency cases yet the overall problem solving must be handled by a pro-active part for long term planning.

Therefore, our approach would be to design the system as a hybrid agent [1; *Wooldridge, An Introduction to Multi Agent Systems*], which includes both the above-mentioned characteristics: Reactive event handling as well as proactive planning.

This leads to layer architecture for the Agent, where the layers are organized hierarchically; the most basic reactive behaviors are in the bottom layers and the long term planning in the top ones. The primary layers are identified as a reactive layer for handling environmental changes and a planning layer for the long term planning. The reactive layer may affect the planning layer, as the emergency case handling may result in a reprioritization of the overall situation.

B. Agent based Learning

In the next few sections we would like to discuss certain aspects of learning based agent system approaches. In order to make a MAS capable of learning some agents within the system may contain an AI (artificial intelligence). A learning system can be designed in several ways which overall defines its autonomy as well as its potential to change itself.

A simple approach of learning MAS would be a system, which makes its decisions, based upon analysis of environmental history. In spite of its simplicity, this method would provide the programmer the ability of controlling the behavior of the agent very precisely and furthermore predicting the resulting outcomes. However, such approaches are not really dynamic and adaptive by design (if the programmer can exactly predict the behavior of the system we don't consider it very adaptive). Therefore, we will discuss the appliance of genetic programming to the agent system as another possibility of implementing learning functionality.

1) Genetic Programming: More advanced approaches of a learning AI make use of genetic code, also known as evolutionary algorithms. Such a system basically works with an initial set of solutions, more precisely a population, and are able to modify the given set of solutions by either mutate an existing solution or recombine them into new ones. In a further step the new population is tested against a utility method resulting in a unique score for each solution. The scores are compared with each other in order to set up rankings and determine the quality of the code. The whole procedure is preceded as fast as possible and repeated thousands or millions of times. The architectural requirement/challenge of such an approach is the velocity. In order to make use of genetic code in the ORP we would need to be able to simulate the outcome of a given solution in a short amount of time to sample a lot of solutions. Otherwise, the processing would take too long and hence make the system useless. A solution could be a simple entity holding some parameters, which then are changed. In contrast, a solution could also be more complex, representing a part of the agents own code, which can be changed by the agent himself. We will discuss latest a little further.

In order to use genetic programming to modify code, we have to define the "Genes" as pieces of code.

The agent has a set of predefined methods given as well as some control-statements, which are the genes. Those genes can now be assembled and ordered in nearly unlimited ways. The agent modifies the code by reordering, adding and modifying the given statements and runs the new code against its utility simulation method to compare the quality of the new code.

The only way to "control" such a system is to limit the available methods (genes) to the evolutionary algorithm. It should be clear, that it is not possible to prove that the real behavior of the AI as well the development of it follows a strict pattern.

2) Artificial Neural Networks: Artificial neural networks are yet another approach creating an AI, which is capable of learning [2; Bhadeshia H. K. D. H. (1999). "Neural Networks in Materials Science"]. An artificial neural network is basically a simulation of how our brain works. The neural network models a network with nodes (Neurons) connected to each other by different weighted paths, representing some sort of a weighted Graph network. In the process of learning some connections become stronger while others grow weaker. Consequently, the system will continually identify paths with higher quality. A (trained) neural network becomes something, which can no longer be understood by a human. It is merely a black box, which is magically able to reason about a given input.

Key point is the unpredictability of both of the approaches (evolutionary algorithms as well as artificial neural networks). A system, which is able to modify itself to a certain degree, becomes unpredictable. At least, it is not much more predictable than a human.

C. Ethical Issues

The autonomous agents may run into dilemma situations where all decisions may result in loosing one or more lives. We give an example of such a dilemma situation below.

1) The Ethic Schedule Dilemma: Let's assume the following hypothetical situation in a hospital, running an Agent system for scheduling surgeries.

The current waiting list has 4 emergency cases of patients {A, B, C, D} at the top. All four patients need to be operated within tree hours – or they might die. There is only one operating room left. The others are not available for whatever reasons. The estimated operating time for each patient shall be defined as follow:

TABLE I. EXAMPLE SURGERY LIST

Patient	Estimated surgery duration
A	4h
B	1h
C	1h
D	1h

A rational decision is made easily. If we operate patient "A" first, all the other patients are likely to die. If we move "A" to the bottom of the queue, it is very likely to save all but A. So we have one life versus three. The Agent system may therefore decide that patient "A" will die for the sake of a better global optimum. It is a sad situation, but you might agree with the decision of the Agent nonetheless.

Now let's introduce moral factors. We identify our given patients by the following table:

TABLE II. EXAMPLE PATIENTS LIST

Patient	Identity
A	Lucy, 11 years.
B	Marcel, 48 years, Father of Lucy
C	Petra, 89 years
D	Rob, 82 years

Is the decision still that obvious? Is a young life worth more than an old one? Is the wish of one person (a Father is likely to die for his child) less worth than the rational decision?

A human hardly can answer those questions. And whatever the decision is, the responsible "person" must be able to qualify and explain why he decided, including rational and moral points. So how can we expect an Agent to handle such situations? And, if an Agent system answers those questions, which acceptance will they have?

2) Conclusion of Agents capable of learning: *In the above example, you might argue that the situation is very delicate, but there might still be rules at some point to guarantee a given protocol. So, some parameters like live expectancies might influence the decision (which is off course highly questionable). Either way, the problem exists with or without having a learning system.*

However, when a machine has to decide such situation, it is even more delicate. The machine hardly can argue morally and explain its thoughts. If we bring the machine learning approach and such dilemma situations together, we exactly run into this kind of situation.
In the end, the machine decides based upon its learning experience and passed evolution what should happen. The agent's behavior will not be predictable because the system is able to modify itself. It is not a transparent process even when it follows strict rules as software always does. Even if the decision of the agent is 100% rational and correct, it is a decision of a machine, which judges about life. We also have shown before that certain situations are not necessarily solved best using only rational reasoning.

Would a hospital trust the decisions of such an agent system, which is able to learn and modify itself? Would you trust in a computer program that constantly modifies itself? Who is responsible when the autonomous Agent has made a (wrong) decision? This all is coupled to the basic question, how much autonomy an agent should have.

Albeit learning based systems are a very interesting approach when applied in MAS, we see a big issue. The lack of social acceptance of such technology will likely be a deal breaker. A social high sensible environment like a hospital seems to be the wrong place to put such experimental technologies into action. Therefore, we stick to a less powerful, but predictable architecture as described in the initial section of this chapter.

IV. GAIA METHODOLOGY

A. Introduction

Regarding the GAIA Methodology we consider the following steps as most relevant:

- Subdivision of System
- Environmental Model
- Organizational Rules
- Role Model
- Interaction Model

We wanted to make sure that we have selected steps of both analysis and design. Since the artifacts "Agent Model" as well as "Services Model" are very time consuming because of their high degree of detail and due to the fact that the Agent Model pretty much matches our Role Model, we left these two steps out.

However, in the following paragraphs we will discuss the Operation Room Planning problem within the above defined topics of the GAIA Methodology. In addition, we will first describe the requirements of the system. Although the requirement elaboration and elicitation process is not included in GAIA it is essential to have solid knowledge about them.

B. *Vision*

The vision of approaching the Operating Room Planning problem with an agent based system is to optimize surgery scheduling in order to improve surgery efficiency, minimize peak demands and save costs.

C. *Rough Requirements*

- The system needs to be able to do long term planning
- The system needs to be able to do short term planning
- The system has to adjust the planning in case of sudden changes of patients state

D. *Subdivision of System into Sub-Organizations*

We separated our system into the following sub-organizations:

- Patient Manager
- Patient Departments (including Surgery Waiting List System)
- Operation Room Provider
- Surgeon Teams Provider
- Postoperative Provider

1) Patient Manager: We consider patients as entities. The Patient Manager holds all the patients and assigns them to the corresponding departments based on their medical issues. Emergency cases will be assigned to the emergency department.

2) Patient Departments: We consider each department as an individual agent with its own desires, believes and intentions. The desire of a department is to heal its patients. Each patient has its own health status and its own path of welfare it has to go through. The path itself is the product of a patient's health status and several rules within the ORP (see Organizational rules). The patient's may intent to skip steps of his path in order to shorten his healing process. However, the system does not allow such behavior. A patient's path can only be shortened by the system due to negative changes of a patient's health status.

Each department has its own patients based on its field of medical expertise. Each department contains a surgery waiting list system consisting of 2 separate waiting lists; one list for surgeon appointments and another for surgery appointments. Each department is responsible for its own waiting list and therefore defines the prioritization of the patients based on department-related indicators.

3) Surgeon Team Provider: *In order to allow an adequate level of abstractness we do not consider individual surgeons but rather whole surgeon teams. Each surgeon team consists of a set of employees which is defined by the system rules (see Organizational rules). Every team is identified by their field of expertise which defines in which field of surgery the team is specialized. The Surgeon Team provider has an overview of the team's availability.*

4) Operation Room Provider: *The Operating Room Manager holds all information about the individual operating rooms. It knows which operating room is able to handle which kind of surgeries based on the room equipment and has an overview of the room's availability.*

5) Posoperative Provider: *There are postoperative resources within the hospital care such as wards and the ICU (Intensive Care Unit). Each ward as well as the ICU contains a certain available space to offer. The Postoperative Manager is responsible for all these units and is therefore able to provide information about the availability of them.*

E. *Environmental Model*

The environmental model of the ORP consists of an information system containing different types of entities and agents. Patients as entities build the input of the system. The output is an always up-to-date plan of surgery procedures. The core attribute of a patient is its health status which is the base for the patient's periodization within the system. Furthermore, patients get assigned to departments based on their medical issues. The fact that a patient's health status may change at every time makes the environment dynamic.

In order to achieve the system objectives the departments can communicate with each other and are able to bid for resources, i.e. operating rooms, surgery teams and post-operative measurements. We will present more information about the interaction in the section "Agent Interaction".

F. Organizational Rules

- A patient has to follow its assigned path of welfare which means he has to fulfill the following preconditions in order to proceed:

TABLE III. PRECONDITION LIST

Step	Precondition
Appointment waiting list	Referred from primary care
Outpatient appointment	Passed appointment waiting list
Surgery waiting list	Passed outpatient appointment
Surgery	Passed surgery waiting list
Use postoperative resources	Passed surgery
Discharged	Passed postoperative ward or ICU

This rules only has effect as long as a patient's status is not declared as an emergency case

- Each surgeon team consist of a specified allocation of employees
- An emergency case has always a higher priority as a non-emergency case
- The following artifacts must be available in order to proceed a surgery:
  - 1 operating room, which matches the required equipment
  - 1 operating team, which matches the required field of expertise
  - Post-operative measurements (available resources in ward or ICU)

G. Role Model

In the following section we defined the most important roles according to the GAIA concept. [3]

DpApWlHandling (Department Appointment Waiting List Handling)
Description
This role is responsible for managing the appointment waiting list of a department
Permissions
Read healthStatus //health status of incoming patient
Read ApWaitingListStatus //read status of appointment waiting list
Change ApWaitingList //put patient on appointment waiting list
Responsibilities
Liveness: DpApWlHandling = (HandlePatient,InformSurgeonTeamProvider)
Safety: ApWaitingListStatus != empty as long patients are requesting

DpSyWlHandling (Department Surgery Waiting List Handling)
Description
This Role is responsible for managing the surgery waiting list of a department
Permissions
Read healthStatus //health status of patient
Read SyWaitingListStatus //read status of surgery waiting list
Change ApWaitingList //put patient on surgery waiting list
Responsibilities
Liveness: DPPatientHandling = (HandlePatient)
Safety: ApWaitingListStatus != empty as long patients are requesting

DpGoodsRequest
Description
This role handles the bidding for the needed goods
Permissions
Read healthStatus //health status of incoming patient
Read SyWaitingListStatus //read status of surgery waiting list
Request Goods //bid for needed goods
Responsibilities
Liveness: DpGoodsRequest = (AnalyzePatient, RequestGoods)
Safety: RequestGoods as long surgery package is not feasible

DpProposalCall
Description
This role is responsible for sending calls for proposals to other departments
Permissions
Read packageInormation //To find out which goods are missing
Send call for proposals //send task (problem) to solve to other departments
Responsibilities
Liveness: DpProposalCall = (AnalyzePackages, CallingProposals)
Safety: CallingProposals as long surgery package is not feasible

DpAnswerCall
Description
This role is responsible for answering proposal calls from other departments
Permissions
Listen Proposal calls //listen to eventual calls for proposals
Send proposal //send proposal for the received task
Responsibilities
Liveness: DpProposalCall = (ListenForProposalCalls, SendProposal)
Safety: always ListenForProposalCalls

H. *Interaction Model*

In the following section we defined the most important protocols according to the GAIA concept. [3]

InformSurgeonTeamProvider

Purpose	inform the surgeon team provider that appointment has been made
Initiator	DpApWlHandling
Responder	SurgeonTeamProvider
Inputs	Appointment
Outputs	Confirmation

RequestGoods

Purpose	Bid for the needed goods in the corresponding auctions
Initiator	DpGoodsRequest
Responder	operatingRoomProvider, postoperative provider, surgeon teams provider
Inputs	Surgery waiting list, patient's health status
Outputs	Requested goods || reject information

CallingProposals

Purpose	Send calls for proposals in order to solve the given task (problem)
Initiator	DpProposalCall
Responder	DpAnswerCall
Inputs	The task which has to be solved (missing goods)
Outputs	proposal || requested good || nothing

SendProposal

Purpose	Send answer for proposal call
Initiator	DpAnswerCall
Responder	DpProposalCall
Inputs	The tasks (problem) solution: missing goods
Outputs	Confirmation by initiator

V. AGENT INTERACTION

A. *Introduction*

We made ourselves several thoughts about how to approach the interaction handing. We consider both, the CNP and the Auctions approach, as very interesting and both of them have characteristics that fit our architecture as well as some that don't. An auction approach fits due to the fact that we have several individual agents with own desires and resources to fulfill them. The CNP approach fits because we have several instances, which need to solve a task, which they can't on their own. Once we analyzed these characteristics and realized the above-mentioned facts, we concluded it makes sense to make use of both approaches to build our interaction model.

B. *Interaction Model*

The main interaction in our ORP system happens between the different departments and the resources. Each department is self-interested to a certain degree and tries to achieve its own goal, which would be to get all the resources it needs to put its patients into surgery. Therefore, departments are surgery agents, which "protect" their own patients. We consider the resources as goods, i.e. surgeon teams, operating rooms and postoperative measurements such as beds in a postoperative ward or in the ICU. For each type of resources there are one-sided auctions for which departments are allowed to bid. The sellers offer teams, rooms and beds for periods that are not yet reserved. Bidders submit bids for the wanted goods. The price they offer to "pay" is the health status of their next patients represented by a calculated value. Of course the departments do not really pay but the value is needed in order for the seller to determine which department, more precisely which patient will receive the goods. If a bidder wins it will receive confirmation about it. As soon as surgery package is feasible, the surgery appointment for the corresponding patient is fixed. Furthermore, departments are able to interact with other departments in means of solving tasks (CNP). These tasks represent the needed surgeries. For example: If a department was not able to "buy" an operating room for a given time span, it can initiate a task, which would be to make this specific surgery package feasible. Different departments, which may have a room at the given time but still miss another resource, may be willing to help. This is why departments are just self-interested to a certain degree.

The emergency department is the only one that does not bid, since it immediately needs resources rather than planning a surgery. Some resources are always put on hold for emergency cases.

VI. AGENT COMMUNICATION

Consider the following diagram, showing the defined subsystems and Agents, and the interaction paths between them:

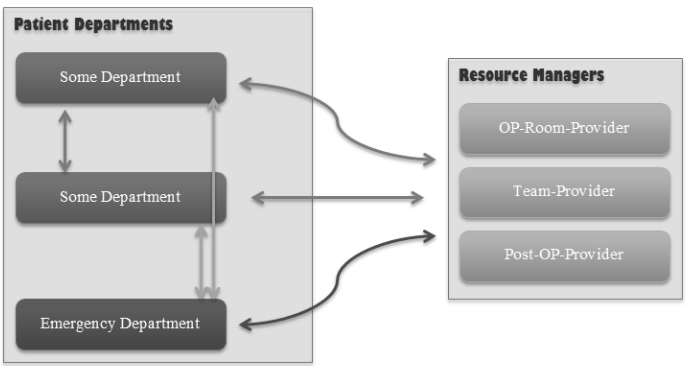

Fig. 1. Communication Model

A. Inter Patient-Department Communication

The patient departments can communicate with each other, using the Contract Net Protocol (CNP). The main purpose of the communication is to organize the joining of resources. This interaction is needed when a single department agent is not able to solve a specific problem. These problems occur if a department does not receive all the resources it needs. For example might an agent require an expert team at a given time slot which is not available anymore. The department can then try to receive help from another department.

Additionally, the emergency department has a communication line with the other departments. As soon as a patient becomes an emergency case, the emergency department is informed. Furthermore, the emergency department can request any allocated resources from another department, as long as those resources are not already in use by a running surgery.

The task oriented situation fits well in the CNP communication pattern. A task of a department represents mainly the problem of lacking resources. The aim of tasks is to complete care packages in order to cure a patient. Therefore, the system requires cooperation between departments via Contract Net Protocol.

B. Resource Auctions

We have identified three resource types, i.e. operating rooms, surgeon teams and postoperative measurements. The resources are uniquely defined by a time slot and a resource type. All resources are time depended and must match with each other in order to build a feasible care package. The resource providers keep track of the current bookings and allocations. They do not allow overlapping bookings. All department agents can join the resource auctions. How departments are able to bid and win in an auction please consider section “Interaction Model”.

1) Just-In-Time-Auctions initiated by Client request: Although auctions are always created and managed by the resource providers, the dynamic matter of the bookings is a problem. The duration of a surgery is always different, so the resource providers cannot just offer an operating room for 2 hours – probably it is required for a longer or shorter time. We assume that for a given operation, we can estimate the overall duration. For this reason, we have invented a new auction system, based upon two phases. First, auction clients can place a request on a resource provider. Then, the resource provider creates appropriate auctions based upon the requests. The department agents (representing clients in auctions) can therefore place requests to the resource providers, of what resources they need and for how long they them need. Based upon this information, the resource providers can create auctions.

2) Emergency Department priority: As soon as the emergency department places a resource request, all affected auctions are immediately closed. An auction is “affected”, when the resource and the time slot overlaps with the request of the Emergency Department. All required resources are allocated to the Emergency Department instantly.

3) Common Ontology: Our solution makes use of auctions and task proposals. Since the possible actions within the system are preset and the agents need to strictly follow certain rules (hot to bid, how to communicate etc.) we figured that the system should use a common ontology.

C. Example interaction FIPA-ACL

A typical request following FIPA-ACL for a resource looks like this:

```
( Sender: Department-XY
  Reciver: Operating-Room-Provider
  MsgId: 234
(Content:
        :command PLACE_REQUEST
        :resource OperatingRoom
        :timeslot  1pm-5pm
) )
```

A response from the resource provider then indicates acceptance or rejection:

```
( Sender: Operating-Room-Provider
  Reciver: Department-XY
  MsgId: 235
  In-reply-to: 234
(Content:
        :command PLACE_REQUEST_ACK
        :auctionID 50231
) )
```

Now, the client has an auction id and is able to start bidding. So he will switch the protocol to an auction based one. The new auction is broadcasted to all agents, which are interested.

VII. Conclusion

We developed the analysis and the design of a Multi Agent System in order to solve the Operating Room Planning problem. It was very interesting and instructive to go through this process. The ORP turned out to be a complex problem with many challenging tasks to manage. We came across some ethical issues, as we were discussing systems capable of learning, which may represent problems in an actual live environment. However, regarding the ORP, we see a lot of potential in MAS. We therefore conclude that MAS would be an effective way of approaching the ORP problem.

ACKNOWLEDGMENT

We would like to thank Marie Persson and Johan Holmgren for their valuable inputs in the MAS module during our course at BTH — which has helped us to get a deep insight into theoretical MAS. Furthermore, we are also very grateful for the guest lecture held by Christian Johansson, who showed us how Multi Agent Systems in industrial areas are currently used: To solve complex routing problems in dynamic environments and how MAS can be applied to new upcoming products.

REFERENCES

[1] Wiley Wooldridge
“An Introduction to Multi Agent Systems”

[2] Bhadeshia H. K. D. H. (1999)
“Neural Networks in Materials Science”
http://www.msm.cam.ac.uk/phase-trans/abstracts/neural.review.pdf
(Accessed March 5th 2013)

[3] Michael Wooldridge, Nicholas R. Jennings, David Kinny
“The Gaia Methodology for Agent-Oriented Analysis and Design”
http://www.cs.ox.ac.uk/people/michael.wooldridge/pubs/jaamas2000b.pdf
(Accessed March 2nd 2013)

[4] Blekinge Institute of Technology
Course Agent Systems
“Case Study ORP”

APPENDIX ONE – CASE STUDY ORP [4]

A. *Introduction*

Most countries today, try to adjust to increasing demand and cost for healthcare services. One of the most expensive areas in healthcare is surgery, which necessitates many expensive resources in terms of staff, equipment, and medical resources. Generally, these resources have to be managed and divided between several departments within the hospital, e.g., orthopaedics, gynaecology and general surgery, in order to meet the total surgery demand.

B. *Operating Room Planning*

The operating room planning includes short term planning and long term planning, i.e., emergency cases and non-emergency cases. Non-emergency cases are described as elective cases and are commonly referred from primary care to a specified department within the hospital care. Before surgery is decided, the patient generally meets the surgeon at the outpatient clinic, i.e., the hospital care. Emergency cases commonly enter the Operating room department passing through the Emergency department as illustrated in Figure 2. However, there are exceptions to this rule; for instance, an elective patient admitted to an inpatient ward can suddenly become an emergency surgery case due to unexpected complications.

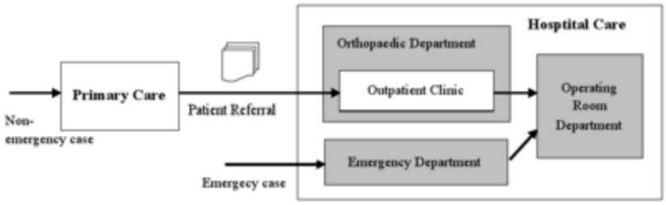

Fig. 2. Illustration of the surgery process

In general, the elective surgery process starts at primary care. The patient is then referred to specialist care for an outpatient appointment. If surgery is decided, the patient is then put on hold for surgery. In reality, the *surgery waiting list* system consists of two waiting lists; one, waiting to meet the surgeon specialist at the outpatient appointment, and one, waiting to be scheduled for surgery after the appointment. Moreover, there is one surgery waiting list system representing each of the operating departments and which are separately managed, i.e., one waiting list system at the Department of Orthopaedics (as depicted in Figure 2), another one at the Department of General Surgery and at the Department of Gynaecology, and so on. Consequently, the allocation of operating room resources affects every surgery waiting list system. In addition, the Operating room department also has to consider a variety of postoperative resources when planning. After surgery, the patients are monitored in a postoperative ward for circulation and respiration, but also for assistance with analgesic before being transferred to the ward or directly discharged. In addition, some patients will need postoperative intensive care and consequently have to be transferred to the Intensive care unit, (ICU) after surgery.