GIS TOOL TO DEMONSTRATE MAURYAN EMPIRE

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Sukesh Babu Mandava
Fall 2011
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Sukesh Babu Mandava:

GIS Tool to Demonstrate Mauryan Empire

Joseph Lewis, Chair
Department of Computer Science

Carl Eckberg
Department of Computer Science

Farid Mahdavi-Izadi
Department of History

7/26/11
Approval Date.
Copyright © 2011
by
Sukesh Babu Mandava
All Rights Reserved
DEDICATION

This book is dedicated to victims of November 26, 2008 Mumbai terror attacks.
ABSTRACT OF THE THESIS

GIS Tool to Demonstrate Mauryan Empire
by
Sukesh Babu Mandava
Master of Science in Computer Science
San Diego State University, 2011

The purpose of this Software is to demonstrate the rise and fall of Maurya Dynasty in a more interactive manner. The application is developed in Java with MOJO (MapObjects Java Objects), provided by ESRI and maps were developed using ARCVIEW software.

This software is a graphical interface tool developed to inspire students to learn and explore facts about Maurya Dynasty. Maps were developed for each and every empire in Maurya Dynasty and prominent locations of each empire are pointed in the respective map. Links to the Internet are provided to make the application more interactive and informative, which could help enhance student’s interest and motivation for learning. Moreover, the students could also customize the application environment to suit their individual interests.

By utilizing modern computer technology to present an important subject in an engaging and interesting fashion, this work could become effective learning software in the classroom not just for students, but also the instructors who can handle bigger class sizes with a greater ease.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT .................................................................................................................. v</td>
</tr>
<tr>
<td>LIST OF TABLES ......................................................................................................... viii</td>
</tr>
<tr>
<td>LIST OF FIGURES .......................................................................................................... ix</td>
</tr>
<tr>
<td>LIST OF ACRONYMS ...................................................................................................... xi</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS .................................................................................................... xii</td>
</tr>
</tbody>
</table>

## CHAPTER

1 **INTRODUCTION** ....................................................................................................... 1  
   1.1 Mauryan Dynasty .................................................................................................. 1  
   1.2 Introduction to GIS ............................................................................................. 10  
   1.3 Advantages of GIS ............................................................................................... 10  
   1.4 Disadvantages of GIS .......................................................................................... 11  
   1.5 Challenges Faced ................................................................................................. 12  
   1.5.1 Collection of Data ........................................................................................... 12  
   1.5.2 Platform Independency .................................................................................... 12  

2 **THE REQUIREMENTS** .............................................................................................. 13  
   2.1 Data Requirements .............................................................................................. 13  
   2.2 Platform Requirements ....................................................................................... 13  
   2.3 Functional Requirements .................................................................................... 14  
   2.4 Optional Requirements ....................................................................................... 14  

3 **TECHNOLOGY** ....................................................................................................... 15  
   3.1 JAVA .................................................................................................................... 15  
   3.1.1 Introduction ...................................................................................................... 15  
   3.1.2 Features ........................................................................................................... 15  
   3.1.3 Usage of Java ................................................................................................... 16  
   3.2 MOJO .................................................................................................................... 17  
   3.2.1 Introduction ...................................................................................................... 17  
   3.2.2 MOJO Package Organization ......................................................................... 17
# LIST OF TABLES

## PAGE

<table>
<thead>
<tr>
<th>Table</th>
<th>Description/Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 6.1</td>
<td>Description of Shape File Header</td>
<td>36</td>
</tr>
<tr>
<td>Table 6.2</td>
<td>Value/Shape Type Mapping</td>
<td>37</td>
</tr>
<tr>
<td>Table 8.1</td>
<td>Custom Toolbars</td>
<td>41</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Initial version of GIS tool.</td>
<td>21</td>
</tr>
<tr>
<td>4.2</td>
<td>GIS tool with functional requirement added.</td>
<td>21</td>
</tr>
<tr>
<td>4.3</td>
<td>GIS tool for Maurya Dynasty.</td>
<td>22</td>
</tr>
<tr>
<td>5.1</td>
<td>Creating a new project.</td>
<td>24</td>
</tr>
<tr>
<td>5.2</td>
<td>NetBeans library manager.</td>
<td>24</td>
</tr>
<tr>
<td>5.3</td>
<td>NetBeans new library.</td>
<td>25</td>
</tr>
<tr>
<td>5.4</td>
<td>NetBeans library manager classpath.</td>
<td>25</td>
</tr>
<tr>
<td>5.5</td>
<td>NetBeans browse JAR folder.</td>
<td>26</td>
</tr>
<tr>
<td>5.6</td>
<td>NetBeans palatte menu.</td>
<td>26</td>
</tr>
<tr>
<td>5.7</td>
<td>Adding library to code</td>
<td>27</td>
</tr>
<tr>
<td>5.8</td>
<td>NetBeans palatte component selection.</td>
<td>27</td>
</tr>
<tr>
<td>5.9</td>
<td>NetBeans palatte component selection.</td>
<td>28</td>
</tr>
<tr>
<td>5.10</td>
<td>Screenshot of NetBeans GUI.</td>
<td>29</td>
</tr>
<tr>
<td>5.11</td>
<td>ArcView tool with country shape layer.</td>
<td>31</td>
</tr>
<tr>
<td>5.12</td>
<td>ArcView tool with Subcontinent countries selected.</td>
<td>31</td>
</tr>
<tr>
<td>5.13</td>
<td>New subcontinent shape file created.</td>
<td>32</td>
</tr>
<tr>
<td>5.14</td>
<td>Selecting editing the polygons option.</td>
<td>32</td>
</tr>
<tr>
<td>5.15</td>
<td>Moving the points to create new shape.</td>
<td>33</td>
</tr>
<tr>
<td>5.16</td>
<td>ChandraGupta shape file overlapped on IndiaMap shape file.</td>
<td>33</td>
</tr>
<tr>
<td>5.17</td>
<td>Ashoka shape file overlapped on IndiaMap shape file.</td>
<td>34</td>
</tr>
<tr>
<td>7.1</td>
<td>MapObjects toolbar.</td>
<td>39</td>
</tr>
<tr>
<td>8.1</td>
<td>Custom toolbar.</td>
<td>41</td>
</tr>
<tr>
<td>8.2</td>
<td>Co-Ordinates display.</td>
<td>45</td>
</tr>
<tr>
<td>8.3</td>
<td>Contact us.</td>
<td>46</td>
</tr>
<tr>
<td>9.1</td>
<td>Application overview.</td>
<td>48</td>
</tr>
<tr>
<td>9.2</td>
<td>Maps that can be clicked.</td>
<td>48</td>
</tr>
<tr>
<td>9.3</td>
<td>Chandragupta enabled.</td>
<td>49</td>
</tr>
</tbody>
</table>
Figure 9.4. ChandraGuptha and Bindusara map’s enabled.................................49
Figure 9.5. XY icon enabled........................................................................50
Figure 9.6. XY selection window. ................................................................50
Figure 9.7. POI in Maurya Dynasty.................................................................51
Figure 9.8. Use of Arrow Tool........................................................................51
Figure 9.9. Hotlink tool..................................................................................52
Figure 9.10. Hotlink webpage.........................................................................52
Figure 9.11. Wikipedia hotlink.................................................................53
Figure 9.12. Print option. ..............................................................................53
Figure 9.13. Help icon. ...............................................................................54
Figure 9.14. Identity tool. ............................................................................54
Figure 9.15. Tool tip of zoom to full extent.....................................................55
Figure 9.16. Tool tip of pan. ......................................................................55
## LIST OF ACRONYMS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSP</td>
<td>Java Server Pages</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>JDK</td>
<td>Java Development Kit</td>
</tr>
<tr>
<td>JAR</td>
<td>Java Archive</td>
</tr>
<tr>
<td>BC</td>
<td>Before Christ</td>
</tr>
<tr>
<td>AD</td>
<td>Anno Domini</td>
</tr>
<tr>
<td>CE</td>
<td>Common Era</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographic Information System</td>
</tr>
<tr>
<td>ESRI</td>
<td>Environmental Systems Research Institute</td>
</tr>
<tr>
<td>MOJO</td>
<td>Map Object Java Object</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>JRE</td>
<td>Java Runtime Environment</td>
</tr>
<tr>
<td>SDK</td>
<td>Source Development Kit</td>
</tr>
<tr>
<td>TOC</td>
<td>Table Of Content</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

I like to express deepest gratitude to Dr. Joseph Lewis, Dr. Carl Eckberg, Farid Mahdavi, who are abundantly helpful to complete the thesis. This thesis would not have been possible until Sandeep Banala and Srinath Surapaneni has assisted me. It’s an honor for me to complete my Master in San Diego State University, thanks for providing me infrastructure support.

I am grateful to my roommates for their editorial Support. Deepest gratitude is also due to Pavan Kesana, Anusha Mandava, and Soni Mikkilineni for their help and inspiration. It is a pleasure to thank my friends in India who made my US journey possible.

I am indebted to my family Anil Mandava, Lakshmi Mandava, Rakesh Mandava, Dr. Bhavana Mandava, Dr. Aaradhana Katragadda and God for there continues support in all forms.
CHAPTER 1

INTRODUCTION

1.1 MAURYAN DYNASTY

Mauryan dynasty, one of the most important in rich Indian history, made its mark in the cultural development of India, it led to great development in culture, arts and economy of the country. Indian relations with other countries especially with the western countries improved greatly. Major part of the Indian subcontinent was united and a secure central administration was formed under this rule, which led to the formation of basis social structure and its caste systems. The Mauryan Empire was indeed the first attempt in India to secure administrative centralization on an extended scale. All the big institutions of ancient Indian society and state came into existence. Another important trend was the development in religious and philosophical ways, in which Buddhism was developed, which now transformed to one of the three great religions.

Mauryan’s ruled for a long time in history, where important changes in society, economy and politics happened. During sixth and fourth centuries BC, the first large states formed which led to intense struggle between them for power and resources. Sources say that there were many wars and clashes between people from different tribes, confederacies and monarchies. Evidence from different sources suggested that an important emperor of this time Chandragupta Maurya, who fought much for power. But few scholars disagreed on the sequence of events in this struggle for power over the land. Chandragupta was from Kshatriya cast of Magadha, who studied at great Taxila all his early life, learning different sciences. He met his tutor Kautilya Chanakya, with whom he made plans to capture the throne. After his education, he gathered an army from different regions along with Chanakya Kautilya and very soon he headed a huge army. He led a powerful anti-Nanda, the local dynasty, coalition and so he favored alliance between Alexander and Porus. Chandragupta made a mistake of ignoring the border provinces. Instead of capturing town-by-town he invaded the center directly and faced a bad defeat. So because of his unsuccessful actions, he looked for an alliance with the next powerful ruler in northwest India. Porus and Chandragupta decided to divide the empire
after the victory over Nandas. Soon the Greek forces under Alexander moved west and left the country completely as the army wanted to return home. The occupied territory was divided among the satraps including the Indian rulers. This situation in Punjab favored Chandragupta and his forces started gaining the required strength. After the death of king Porus he overtook both the Porus and Nandas kingdoms. He even took over Punjab and planned a major strike against the Nandas. The two most important periods in Chandragupta’s life were when he fought against Greeks and other the capture of throne. The western sources say that he was assisted by non-monarchic groups of northwest India, few Indian sources say that Chandragupta was assisted by some of non-monarchic units of Northwest India. These non-monarchic forces in Punjab were successful in strongly resisting Alexander’s forces during his invasion. Chandragupta has taken advantage of current circumstances in Punjab and took advantage of them and as the situation was favoring he was confident to get their support against foreign invaders for independence. Chandragupta liberated the Punjab and promised the people safety and prosperity; he assured a safe rearguard. His biggest goal was to capture Magadha throne, for which he advanced to Pataliputra, where he wanted to establish Maurya Dynasty. His previous victory over Greeks and the liberation of the Punjab from the foreign invaders had undoubtedly helped him to get closer to his dream of establishing Mauryan Dynasty. On the way to Magadha throne, Chandragupta has signed a treaty with Seleucus, where Chandragupta receives some regions of Ariana, in return Chandragupta sent Seleucus war elephants. This treaty with Seleucus has resulted in a great alliance between them, Seleucus even sent an ambassador to Chandragupta’s court. Peace with Chandragupta enabled the Seleucus to embark on a long march against Antigonus, whose forces were routed at Ipsus (301 BC). The Mauryan’s war-elephants under general Seleucus played an important role in his victory. Even after the end of treaty Chandragupta maintained good relations with.

Chandragupta was considered as one of the greatest of all Indian kings in the Indian history and his name was associated with the legends who made great Indian history. After Chandragupta’s time, his son Bindusara succeeded him; he was known as “the slayer of his foes”. The situation at the court of Bindusara was extremely complicated; there was a continuous struggle for power among the Mauryan princes and ministers who wanted to gain influence from the King. The Mauryan ruler Bindusara maintained and continued the diplomatic relations with Seleucid Empire. Bindusara followed his father Chandragupta’s
policy of further developing relations with other western countries. Bindusara captured 16 towns and extended Mauryan’s rule from the Eastern Ocean to the western ocean (conquered regions in southern India). The Mauryan dynasty reached its highest level under their third ruler, the well known and greatest of all times Asokavardhan Maurya, son of Bindusara. Asoka is considered one of the most outstanding statesmen of ancient India. Mauryan Empire under Asoka ruled the sub-continent from Kashmir and the Himalayas in the northwest to the Bay of Bengal in the east. During his rule he formed diplomatic relations with the Hellenistic states in West and other countries in the East. To suppress a rebellion against King’s officials in Taxila, Bindusara sent Asoka to West India. He further helped by capturing Khasa, another region against the Mauryan Empire. The major political event occurred in Asoka regime was the war against Kalinga, a powerful state on the eastern coast of Bay of Bengal, which is the biggest war he fought. He led thousands of men in war, which led to almost hundred and fifty thousand prisoners and nearly hundred thousands were killed. The result over the war against Kalinga changed Asoka to embrace Ahimsa, non-violence and approach to conquest. Indeed this directed Asoka towards Buddhism and after the war he abandoned the policy of creating a unified Indian state. Asoka did not discard his predecessor’s foreign policy; he still maintained foreign policy. The methods of its implementation were modified in line with the new political situation. In Kalinga it seemed difficult for Asoka to establish a stable rule, because the people of Kalinga were unhappy with the new ruler to whom they lost their freedom. Only few states in south India were freed of his rule during Mauryan regime.

The executive body in the kingdom included the King, the King was assisted by number of ministers, who were selected based on their proven ability and character. The number of ministers appointed in the court changed often and met in court for discussing public problems and public businesses. If there was a difference between the views of them they voted for the decision. King was the head of the ministers; he could consult anyone by his own and was present in general meetings. The rule was based on a book written by Kautilya, called Arthasastra. The central machinery of administration envisaged in the Arthasastra may well be accepted as a representation of its condition towards the close of Chandragupta reign, and Asoka introduced some changes into the system. The revenue was collected all over the kingdom and was brought under central administration. Collector general (Samaharta) was in charge for collecting the revenue or Gopa; the Gopa had charge of
five to ten villages in which he supervised the maintenance of boundaries. Empire was divided into different provinces. A governor, who was chosen by the king, ruled each province. These governors mostly belong to the family of royals. The situation with villagers was always the same in all of the Indian history; they did not completely own their lands, though they enjoyed a good freedom in their lands under this central administration.

The judiciary system in Mauryan time consisted of two sets of courts and it also consisted of village tribunals, which were headed by elder people and headmen. The King was the highest of all, but he could not take the responsibility for maintaining the total judicial administration by him, as it was a huge empire. The empire had a standing army, which was always kept ready for all of its needs, internal and external. Army was divided into groups headed by superintendents. These superintendents worked alone or under the commander-in-chief of the Mauryan army. Commander-in-chief was responsible for the army on the whole and he along with the King made inspections of the army from time to time.

The Mauryan achievement laid in the ability to weld the diverse parts of the subcontinent into a single political unit and to maintain an imperial system for many years. The economy in the Mauryan time was flourishing and was in a stable state which was the result of the central administration of one unified political state for many years. Most of the revenue came from land revenue collected from all over the empire. Some part of the income was from trade, which was less when compared to the land revenue. The development and expansion in the agriculture and the administration system led to a strong imperial system and gave steady income from land revenue. This change is confirmed by the theories of Kautilya. According to Kautilya, all the uncultivated land should be used for settlement as villages for people who had no place to live and for those deported because of war. It is likely that about one hundred and fifty thousand persons deported from Kalinga by Asoka after the successful campaign settled in this manner. In the account Megasthenes it was said that there were no slaves in the rule, but on contrast other Indian sources talk of the different categories of slaves. Slaves were not used for large-scale production of any sort, though they were present along with hired labor working in the lands, mines and few other places, along with domestic slavery. In the case of revenue and taxes of the kingdom, sources gave different opinions that the land was totally under state control and others say it was both private and individual ownerships. Taxes imposed were different, one for the land being cultivated and other for the produce of the land.
In case of irrigation system this state was maintained for limited time in limited areas, private people who owned and cultivated lands also owned them. Different types of income from the tax imposed on internal and external trade was increasing substantially in the region. Roads were built connecting all the places in the empire. All of these along with efficient administration resulted in creating one single and strong economy, which included all the regions of the empire.

Education in the Mauryan Dynasty improved drastically with time. By the end of the 6th Century B.C, the Vedic rituals and sacrifices gradually increased and the priests (Brahmins) started being benefited, as the other castes were never allowed to study or perform any of the rituals. During those days, education was confined only to the Brahmins and Non-Brahmins were never allowed to do Uppanayana, is a traditionally ceremony performed to mark the point at which boys began their formal education. During Mauryan regime two big religions Buddhism and Jainism gained importance. Buddhism was more prominent during Asoka’s time. Asoka appointed Mahamatras or supervisors who were given the responsibility of spreading cultural awareness and providing education for the people. In all the important divisions of his empire, the emperor directly appointed these Mahamatras. Buddhist monasteries or Vihara’s were the centers of education. The first mention of a Vihara happens to be in Banavasi. An inscription in the Brahmin script dating back to the third century C.E. indicates that, a Chutu princess called Shivaskanda Nagashree donated a vihara in this location. Hiuen Tsang, the Chinese pilgrim who traveled to south India in about 640 C.E., wrote about Buddhism. Mauryan period lead to a revolution in education, which can be seen in the form of the book, Arthasastra - a Sanskrit classic on the principles of political-economic organization written by Chanakya who was the principal advisor of the first Mauryan Emperor. The time of the Mauryan was refered to as Golden Age of India in sciences, mathematics, astronomy, religion and Indian philosophy. The peace and prosperity gained under the leadership of Asoka enabled the pursuit of scientific and artistic endeavours of that time. Few concepts in numeric system were invented in India during the reign of the dynasty. Mauryan India also enjoyed an era of social harmony, religious transformation, and expansion of sciences and knowledge. Legend states that Chandragupta's success was due in large measure to his adviser Kautilya, the Brahma author of the Arthasastra (Science of Material Gain), a textbook that outlined governmental administration and political strategy.
The study of a number of records of the Mauryan Empire showed that Brahmanism was the mainstream religion during the Empire's reign that had its support mainly from the Kings, the Nobles and the rich Brahmanic householders. The era also saw the inception of two other religions Jainism during the rule of Chandragupta Maurya and Buddhism during the rule of Asoka. Hinduism had always been the only predominant religion before the advent of Mauryan dynasty. Early years of Mauryan dynasty saw the loss of fame of Hinduism that it used to have previously. This was due to the attempted reforms of Vedic religions by secret groups in the kingdom. However until the reign of Asoka, Buddhism and Jainism have not taken center stage in the religious reformation in the kingdom. Many aspects of Hinduism were questioned and opposed by certain groups for its complex and expensive ritual process. Nevertheless, reformation in religion was very limited in the early years of the dynasty. Until the rule of Asoka, Hinduism had been the dominant religion of the Mauryan kingdom. Fueled with the opposition from some groups within the kingdom and Asoka's initiative in spreading Buddhism lead to the downfall of Hinduism during the reign of Asoka post Kalinga war. In spite of this, Asoka entertained Hindu Brahman priests and ministers in his kingdom. However, this downfall was short lived and Hinduism retained its glory after the death of Asoka.

A group called the Sramana started the ascetic movement, which started in the Nanda-Maurya period. In Buddhist texts these people Sramana, are referred to as wanderers who specialized in ethics, philosophy, nature lone and mysticism. There were other movements called Ajivika and Nirgrantha that were believed to take place during the Mauryan period. These religious movements played an important role in spreading Buddhism. The greatest event in the history of Buddhism in this period was the conversion of Asoka. Asoka's patronage has made the religion spread to distant lands far beyond the Mauryan Empire. Asoka, after the Battle of Kalinga has renounced violence and started to embrace Buddhism, which lead to establishment of Buddhism all over the country. Asoka’s interest in spreading peace and harmony all over the country has made him to embrace Buddhism as it was based on the principles of ahimsa and forgiveness. He established Buddhist monasteries and encouraged Buddhist monks in his empire to endure. Buddhism was preached and well understood among the people. Many edicts corroborated that Asoka himself has taken the lead in spreading the religion in the form of Dharma. He has ordered his officials to spread the
message of Dharma on rocks, pillars and monuments all over the empire. Asoka entrusted his officers with this mission within his own empire and the deputed missionaries to the foreign countries for the same cause. According to the Ceylon chronicles, the credit for the initiative of this movement is attributed to Tissa Mogaliputta, who was the president of the Third Buddhist council. The Buddhist propaganda during the reign of Asoka was lead by a series of missionary activities, which were masterminded by Asoka himself. The very first mission activity did not have large impact outside the empire but has made great impact within the empire with immense success. This has lead to many other missionaries, which finally paved the way to adornment of Buddhism. Buddhism saw decline in the empire after the death of Asoka. It was also considered a factor that caused some indifference within the empire, which lead to the downfall of the Mauryan Empire. After the decline of Mauryan Empire for various political and economic reasons, people started converting back to Hinduism and this saw the downfall of Buddhism.

Jainism came to light during the reign of Chandragupta Maurya but the impact on Hinduism was very little. Jainism was embraced by people mainly because it had no expensive rituals, didn't have the caste system as in Hinduism and easy to understand with no prior knowledge of Sanskrit. However it failed to spread like Buddhism because some of its Ahimsa principles were too hard to follow for the common men and no major missionary activities took place to preach it all over the empire. Apart from religious movements like Buddhism and Jainism there were also theistic movements that took place during the Mauryan period. Cult of Vasudeva and Arjuna can be considered as theistic movements, which later on became religious. Some epigraphic records of the second century confirmed that the cult of Vasudeva was being followed not only in the country but also in foreign kingdoms. Another cult that started during the same time was the cult of Samskarana, the elder brother of Vasudeva. No clear documentation is mentioned as to when this movement originated but certain inscriptions assert the fact that Samskarana doesn't have as much as following as the cult Vasudeva. Although there are many reforms that took place during the Mauryan only Buddhism has seen major success but eventually declined after the death of Asoka. Hinduism saw slight decline during Asoka but regained its glory after the decline of the Empire. On the Whole the Maurya played a vital role in bringing religious reforms in Indian history, proving that all religions could co-exist together with peace and prosperity.
Over the years of glorious rule, Asoka emasculated lot of wealth. However during the final years of administration, with his generosity towards Buddhist community, he depleted state treasury. At this time Sampadi, who was his heir to throne was appointed to look after the state affairs. Sampadi was warned by the royal ministers about the Emperor’s generosity and advised that he should be avoided from state affairs. Annals about Asoka’s final years corroborated that Ministers and Sampadi seized Asoka’s power and dismissed him from the affairs of the state. All of theses events transformed the political situation in the Mauryan Kingdom into extreme complications. Provinces that were under the Mauryan Kingdom were in pursuit of independence and started revolution against the central authority. As mentioned earlier, the kingdom was also reeling under financial difficulties. This period also saw protest against Buddhism from all corners of the empire. People from the Brahmanic circles, who had strong influence on the spiritual life of the society, were waiting for such an opportunity to regain their glory in the empire. Most influential people of the kingdom at that time Pushyamitra, founder of the Sunga dynasty, Sampadi the Crown prince and Queen Tisyaraksita, wife of Asoka were all against the pro-Buddhist policy and contributed to the goals of the Brahmanic policy. Attention was repeatedly been drawn to the fact that there were no inscriptions dating from the very last year of Asoka’s rule. The above data explains this fact. In the years when the emperor had virtually no power, his decrees were probably no longer engraved on stone.

During the Mauryan period the kingdom saw rebellious movements in certain provinces. In the Taxila province this happened twice. Asoka’s father, Bindusara sent him to Taxila on one such situation and in another case Crown prince Kunala was asked to take care of such rebellion movement in both the cases the townspeople of Taxila reassured the King that they were against the state official but not the king. Such rebellion movements always took place but had little impact on the smooth governance of the Empire. However during the last days of Asoka, Kunjarakarna who belonged to royal family carried out rebellion movement in Taxila. Sources indicate that Crown prince Kunala was blinded by the orders of Queen Tisyaraksita as though issued on the behalf of the king. It is tempting to treat the information of the sources about Kunjarakarna’s considerable power as evidence of the beginning of decline of Mauryan Empire, which became particularly obvious in the last years of Asoka’s rule. During this rule many such rebellion movements have taken place all over the kingdom.
in various provinces. During this period, due to the increase in the number of such movements and lack of treasury, the central authority was unable to ensure strict control over the entire territory and this lead to autonomous regions. It is likely that the division took place after the death of Asoka. A little over half century from his death the Mauryan dynasty came to an end with seizure of the Throne by Pushyamitra, during this period the Empire saw frequent change of its ruler, which suggest the dynasty’s decline.

In 206BC it was recorded that Seleucid’s King, Antiochus, the great, who had a successful campaign in the west territories of Armenia and Partia had concluded war with a peace treaty with the Bactrian King. Antiochus has crossed the Hindukush, for the renewal of the treaty between the Mauryan’s and the Seleucids. He not only received war elephants and landmasses as part of treaty, but also his army continued to deep marching into the territories of Mauryas. By the end of 206 BC Mauryan’s have lost some of their western provinces. Pushyamitra, commander in chief of the army, started a rebellion movement, in which he lead his army against the last of the Mauryan rulers, Brhadratha and seized the throne. This was in the year 185BC and marked the end of the Mauryan dynasty.

The decline of the Mauryan Empire, one of the greatest ancient empires of the east as seen is due to a number of domestic and external reasons. The Mauryan Empire consisted of several provinces with different tribes and people who spoke a multitude of languages. Such Heterogeneous Empire was unified by means of well-administrated system, strong army and the policy of Dharma. Thus it can be seen that weakening of such strong system has caused the collapse of the empire. Many Scholars have opinionated that Asoka’s policies have resulted in disorganization of the administration of the country, which subsequently resulted in the Downfall of the Empire. His pro-Buddhist ideology received strong opposition from the Brahmanic circles who had great influence both politically and economically. This very strong opposition also resulted in decline of Mauryan Empire. As mentioned earlier the policy of Dharma played a major role in unifying the vast heterogeneous Mauryan Empire. Successors of Asoka failed to maintain this policy, which attributed to their weakness in administration of the state and proved fatal to the unity of the empire causing it to disintegrate. In the days of its prosperity an Empire did enhance the glory of the race and it’s achievements in political, economical, educational, religious spheres. The Mauryan Empire will be remembered in
ancient history of the east for its vision in creating a united sub-continent and succeeding in it as well.

1.2 INTRODUCTION TO GIS

GIS abbreviates Geographic Information Science. GIS involves the display and analysis of location based data, maps being the most common example, census data being the most common example of a large database with spatial components. ESRI, among others, sells products to the GIS community.

Study through visualization is the best key for remembering and understanding better. The software is always developed keeping in mind the preferences of the target-user and in what capacities can the new software replace the existing system used by the target user. MapObjects is lightweight GIS software that can be used to create and customize GIS applications. This report discusses the different functionalities of MapObjects and how this is used to make the project. In this thesis, the emphasis is given to develop shape for Mauryan Empire.

In backend the system distributes the data into different sources (shape files, dbf files) so that the data is well organized in a specific context (shape files for geographical features, dbf file for data) and the front end, User Interface (UI), displays the results. The data that is linked with the project in the form of web pages is easy to edit and modify. When a word document is saved in the form of a web page then that document can be linked via hotlink tool, described in the GIS Tools section (Chapter 8), and can be mapped on the Map of Dynasties at any location via co-ordinates (longitude and latitude). The application of labels using Legend Editor explained in GIS Tools section makes the locations on the Map easy to understand.

The document is organized into ten chapters where Chapter 3 talks about the technology, chapter 5 talks about the tools used to design the code, chapter 6 explains the different shape we use, chapter 7 and chapter 8 explains the tool designed in the GIS tool.

1.3 ADVANTAGES OF GIS

A picture can say thousand words. GIS is a special purpose digital database in which a common spatial coordinate system is the primary means of reference. GIS allows us to map the location of objects. Countries, Counties, cities, water lines, roads fire hydrants and natural
resources are all examples of data that can be inventoried and displayed using GIS. Terrain models can be generated to aid with 3-D visualization. Densities and quantities of a specific item in a given area can be calculated and displayed, as can population changes over time. Specific street addresses and coordinate data (i.e., longitudes and latitudes) can be accurately added on a map using geocoding methods. GIS is related to other database applications, which is an additional advantage. All the information in GIS is related to spatial reference. Future conditions of resources can be predicted based on current and historical data. GIS provides a very effective means for graphically conveying complex information. Layouts created with a GIS are extremely useful when included in reports and presentations. GIS can help user organize and centralize data. GIS database can link all of the organization’s digital data together based on a location, such as address. GIS adds advantage like better accessing to better quality and time-relevant data will help us to make better decisions. Spatially enabled websites can provide an excellent tool for education and public information. Websites can be developed with GIS software for either the Internet or a company intranet, which can help us or an organization or department effectively convey information to members of a private group, or to the public at large. Maps can be created dynamically and served over the web. Field Collected GPS Data. The great appeal of the GIS stems from their ability to integrate great quantities of information about the environment and to provide a powerful repertoire of analytical tools to explore this data. GIS are now used extensively in government, business, and research areas for a wide range of application including environmental analysis, planning, locational analysis, and understanding the history. GIS has an added advantage to add different layer of map easily, and it is a very powerful tool for marketing and environmental studies and many other applications.

1.4 DISADVANTAGES OF GIS

Every coin has two sides, GIS has few drawbacks as well. The first issue is the substantial time required to analyze and compile the necessary data. The system is quiet expensive, especially the ESRI software. The good news is we see lot of progress in open GIS. Another common pitfall in GIS application development is capturing more data than required by the application, for which we have to gather different data from different maps;
where there is an often discrepancy from one map to another. Another limitation to GIS is as earth is round and the geographic error can be increased as we get into larger scale.

1.5 Challenges Faced

The software development cycle includes analysis of requirements, design, implementation and testing of the code (software) before it is delivered to the end user. Some of the challenges are worth mentioning in the development of this thesis.

1.5.1 Collection of Data

Data plays a very important role in completion of any project, in my thesis I faced difficulties in collecting 2400 years old data, as there are no proper documents. Shape files are very important in completion of my thesis; I had faced difficulties in obtaining the subcontinent shape files. But thanks for ArcView, which helped me in creating my own shape files.

1.5.2 Platform Independency

After the collection of the data the next challenge was, how will this data run on different platforms? If the end user has Mac OS and project is made in accordance with the windows environment then the project will be of no use for the end user. Hence, there was a need to make the project to run on different operating systems. The project is now platform independent and thus, it can run on Mac OS, Windows, UNIX or Linux. All the browsers supported like Firefox, opera, conqueror, epiphany, Mozilla, Netscape and Internet explorer. So the web pages can be viewed in any browser that the computer or laptop has without any issue in the display. The web pages are created using Microsoft Word and are easy to change and modify in future use.
CHAPTER 2

THE REQUIREMENTS

Prof. Farid Mahdavi, History Department, San Diego State University, has an idea to implement the new technologies in his teaching. To make San Diego State University History students understand better we had to develop a GIS tool to demonstrate Indian History of Mauryan Dynasty. Mauryan’s had ruled the Asian subcontinent from 3BC to 1BC. The thesis is developed under the guidance of Dr Joseph Lewis, Computer Science Department, San Diego State University.

The requirements are gathered and classified into Data requirements, Platform requirements, Student centric requirements and Functional requirements.

2.1 DATA REQUIREMENTS

Data plays a major role in completing any project, in this section we refer to data that is represented in the GIS tool.

Farid Mahdavi, Department of History, San Diego State University has helped me throughout the thesis to research the data which is required in completion of the thesis. The main focus of the work is to develop different sets of maps, which are ruled by different rulers of Mauryan Dynasty. We had to collect the information related to the ruler of the dynasty.

I have collected all the required data and their contribution during their rule in Mauryan Dynasty. Different maps are created for each popular ruler of the dynasty, which displays the land they had conquered.

2.2 PLATFORM REQUIREMENTS

We have to develop a tool that can be used by a wide variety of students in San Diego State University. The idea behind the GIS tool is to make it platform independent, so we can accomplish the requirement. Java was used as a development language with Map Objects Java Objects edition as a GIS solution. The entire setup was framed in windows XP, which has its own advantages. The user does not require knowing much about to IDE’s to execute the tools.
They need to have minimum knowledge on using the command prompt. These requirements have been discussed in detail in chapter 3.

2.3 FUNCTIONAL REQUIREMENTS

In this section we will refer the functional requirements of the GIS tool, which needs to be added. We had added these requirements to help the end user so they can use the tool more efficiently. The following tools are to be developed:

a. Print Option, to print the map as required.
b. Add data layers, to add different sets of data files.
c. Help icons, which pops and describes the functionality of tools.
d. View maps of rulers as required.
e. Navigate through web pages to understand better.

2.4 OPTIONAL REQUIREMENTS

This section describes about the optional requirements that refer to look and feel of the GIS tool, which describes how the overall look and feel should be.

- Users should add the data files as required and should be able to navigate.
- Need to have tool tip text for every tool.
- Images should be added in the web pages.
- Should use bigger font.
CHAPTER 3

TECHNOLOGY

3.1 JAVA

In section 3.1 we introduce JAVA and we discuss the features and usage of JAVA.

3.1.1 Introduction

Java is a high-level programming language developed at Sun Microsystems (now Oracle Corporation), released in the year 1995. Java is developed with "WRITE ONCE RUN EVERYWHERE PHILOSOPHY". The language is an object oriented and enforces the user to use object-oriented style while programming. The language at its advent is aimed to be portable across all platforms, which is a major disadvantage of languages like C and C++ at that time.

3.1.2 Features

Some of the Features of the language are listed below:

- **Object Oriented**: Java language is object oriented. It strictly enforces object language paradigm unlike C++ where the style of programming is left to the choice of the programmer. Being object oriented the language offers features like inheritance, polymorphism, data abstraction and data encapsulation. They play an important role in development of the software adhering to software development principles. The other advantage of object-oriented programming is the flexibility of the code, and design patterns which when employed can provide clean and pragmatic solutions to complex problems. Design patterns must always be seen from the context of the language and there are well-documented design patterns for Java language.

- **Portability**: This is another disadvantage of languages like C++ and C where applications developed on one machine cannot be imported onto other machines. This drawback is overcome in Java by introduction of virtual machine. Since the Java code is run on this virtual machine it doesn't matter on which platform the virtual machine is running, this idea helped in maintaining the Java language portable. The disadvantage of such approach is that there should be java runtime installed on the machines where java should be run. This is eventually not a concern since many platforms come with java runtime installed.

- **High performance**: Java language though interpreted ensures high performance as the interpretation is done not on the source code but on the byte code to which the source code is compiled. So the compiler optimization that needs to be done is carried in the
phase when the source code in text is compiled to byte code. This byte code is then interpreted by java virtual machine. This architecture strikes perfect balance between portability and performance. Though Java cannot surpass C++ in performance it can be as fast as C++ with some good coding practices.

- **Automatic Memory Management:** This is a real boon to programmers where they need not worry about memory allocation and de-allocation unlike in C and C++ where it is the responsibility of the programmer to allocate and free up memory. The major disadvantage of memory management is it hampers programmer productivity, as the programmer instead of the algorithm programmer is busy trying to figure out memory allocation and de-allocation. The scenario where the programmer is responsible for memory management can lead to memory leaks, which have historic reputation for of being very hard to debug. Java implements the automatic memory management as garbage collector which is a low priority thread that runs in the background checking for objects in the heap that are no longer required for the program and freeing them and adding the memory to the existing memory pool. The disadvantage of garbage collector is this runs as a low priority thread which make the freeing up of memory not as fast as it can be. Though method exits to change its priority such methods are cumbersome and not much effective.

- **Concurrent programming:** Java offers language support for concurrent programming in the form of threads. Which can be thought of as lightweight processes. Threads not only are memory efficient but also take up less time than processes to startup. This feature is effective to take use of the multiprocessor architecture prevalent these days. The disadvantages of thread is synchronizing access to shared memory, such disadvantages are also addressed in later versions of java language that have the inbuilt synchronized class.

- **Library support:** Java has standard libraries for all kinds of programming like webbing development, GUI development, networking programming. This standardization is very important in order to build large enterprise applications. Java has a proven track record in the enterprise community for the very same reason.

### 3.1.3 Usage of Java

Java is released in two versions as an open source with general public licence license and as java enterprise edition with enterprise licensing. We have seen equal acceptance of the language in the open source community as well as the enterprise community. Many applications prove the wide acceptance of java language and it's popularity as a programming language. Many universities teach Java as primary programming language and many research projects have been incorporated in java making the language favorite language in the academic world too.
3.2 MOJO

In this section 3.2 we introduce MOJO and we explain how to setup and run mojo.

3.2.1 Introduction

MOJO abbreviated, as "Map Objects with Java" is a product of ESRI, a California based company headquarters in Redlands, California. It is primarily comprised of three major components, a Java application, an example of component architecture, and a GIS development tool.

MOJO is developed in pure java hence it is portable like java and can run on any machine with JRE installed on it. MOJO is based upon component architecture that makes it flexible in incorporating into other applications. The application program can make use of the interfaces and other functionalities of the code, once it imports the component files. There are many advantages using component code, these include flexibility to change the behavior of the code, and integration of many other components. The concept of component architecture is based on modular programming, where a developer can reuse other software developed and he can integrate into his existing application without any configuration and installation overhead.

3.2.2 MOJO Package Organization

The MOJO package installation has root directory ESRI under which all the folders and files containing the package area installed. The hierarchical organization of the package is shown below:

```
ESRI
|--
|---- MOJO
|   |-- Documentation
|   |   |-- MapObjects_WebHelp
|   |   |-- GettingStarted
|   |   |-- MOJ_UML_ClassDiagram
|   |   |-- ProgrammersReference
|   |   |-- jre
```
The Documentation contains the MapObjects web help for the class’s and the object references. Getting started, webpage has detailed documentation on how to use map objects with JBuilder. JRE is essential in order to run java. Lib folder contains all the class libraries for MOJO. Sample folder contains tutorial and sample applications, for a quick start over. Scripts folder contains bat files for compiling and running the application. These scripts are batch scripts that run on Windows NT environment, they help in automation of environment variable configuration like setting PATH CLASS PATH and other environment variables before compiling and executing the application.

3.2.3 Setting Up MOJO

The minimum version of JAVA SDK required for MOJO is 1.4. This applies to both MOJO versions.

- Download MOJO and copy the moj Compile and moj Run files into the folder ESRI\MOJO\Examples.
- If JAVA SDK is Downloaded and installed in folder like C:\jdk1.5.0 modify the moj Compile file to set JDKHOME=C:\jdk1.5.0
- In the same moj Compile file set the MOJ20 or MOJ10 (depending on which version you are using) to the following:
  set MOJ10=C:\ESRI\MOJ10 or set MOJ20=C:\ESRI\MOJ20
- Repeat the same modifications for moj Run as well.
3.2.4 Running MOJO

Now we are ready to run the sample applications:

- Open the command prompt and navigate to ESRI\MOJ20\examples.
- Compile the QuickStart.java file:
  moj_compile QuickStart.java
- Run the application:
  moj_run QuickStart
CHAPTER 4

PROTOTYPE

A prototype is an early sample or model built to test a concept. In software, prototype and alpha are often used interchangeable in realm of software development. It is referred to an early stage of development stage. It is common for prototype to refer to an initial version that demonstrates the very basic of functionality. Ones the alpha grade software is designed with basic functionalities we integrate additional feature into the software, which would be called as beta version. Beta software can be used for testing the entire software and to adjust the program to respond correctly during testing. Often the end users may not provide a complete set of application objectives, detailed input, processing, or output requirement in the initial design stages. Once the beta version test is completed new requests will be added to improve user interface and few additional feature can also be added.

Functional prototype is next version of software, here we incorporate all the requested features into the software and regressive testing is performed on the software before releasing the final product.

1. Initial version. (See Figure 4.1).
2. Prototype with functional requirements. (See Figure 4.2).
3. Final GIS Tool with all requirements. (See Figure 4.3).
Figure 4.1. Initial version of GIS tool.

Figure 4.2. GIS tool with functional requirement added.
Figure 4.3. GIS tool for Maurya Dynasty.
CHAPTER 5

DEVELOPMENT TOOLS

5.1 DEVELOPING THE CODE USING NETBEANS IDE AND MAPOBJECTS

The easiest way to develop the code is by using the GUI; the traditional notepad style used to write the java programs is hard to debug the compiler errors. In this chapter we will discuss the configuring steps for the Net Beans IDE. This provides an ability to integrate with ESRI MapObject Java Edition and it is a very powerful GUI builder. We can prototype and design swings GUI by dragging and position GUI. Below are the steps to configure NetBeans with MOJO.

1. Create a new project.
   a. Start NetBeans and choose File > New Project (see Figure 5.1).
   b. Name the project and browse for an appropriate directory to save project.

2. Create an engine library.
   a. Select Tools->Libraries (see Figure 5.2).
   b. Click on new library button (see Figure 5.3).
   c. Select the newly created Library, select the Add JAR/Folder (see Figure 5.4).
   d. Navigate to the folder where the MOJO jars are copied (see Figure 5.5).

3. Create a Java application.
   a. Adding the Swing/AWT components (see Figure 5.6).
   b. Select the ESRI MOJO 2.0.x library (see Figure 5.7).
   c. Select all the components to be added in palette (see Figure 5.8).
   d. Select the beans folder (see Figure 5.9).
   e. Screenshot of NetBeans GUI after configuring the MOJO (see Figure 5.10).

We should be able to see all the design objects in the palette window on the right hand side, now these objects can be selected and dropped to build the application.
Figure 5.1. Creating a new project.

Figure 5.2. NetBeans library manager.
Figure 5.3. NetBeans new library.

Figure 5.4. NetBeans library manager classpath.
Figure 5.5. NetBeans browse JAR folder.

Figure 5.6. NetBeans palette menu.
Figure 5.7. Adding library to code.

Figure 5.8. NetBeans palette component selection.
Figure 5.9. NetBeans palette component selection.
Figure 5.10. Screenshot of NetBeans GUI.
5.2 Creating the Shape File Using ArcView

ArcView is part of ArcGis suite; ArcGis consist of a group of Geographic Information System (GIS) software products produced by ESRI. ArcView allows one to view spatial data, create layered maps and perform the basic spatial analysis. It can view and edit GIS data help in a flat file database.

In the chapter we create the shape file of different rulers who ruled the Maurya Dynasty. Creation of shape file is done as part of data requirement, below are the steps included in creation of required shape file:

1. We need to have a world shape file.
   Master shape file can be created from the world map shape file, Figure 5.11 shows the ArcView tool that has world shape layer. We need to select the subcontinent courtiers and built a new shape file (see Figure 5.11).

2. We need to edit the file and create a new Asia subcontinent shape file (see Figure 5.12).
   We need to export the selected layer as a new shape file, this new shape file will be saved as IndiaMap shape file (see Figure 5.13).

3. Edit the polygons to create the new emperor shape file.
   We need to click Editor - > Start Editing (see Figure 5.14).
   Move the points across the map to create required shape (see Figure 5.15).
   We need to move all the required point in the image found in Figure 5.15 and create the required shape file; ones all the points in the polygon are moved. We need to export the shape file (see Figure 5.16).
   By following the same process we can other required shape file (see Figure 5.17).
Figure 5.11. ArcView tool with country shape layer.

Figure 5.12. ArcView tool with subcontinent countries selected.
Figure 5.13. New subcontinent shape file created.

Figure 5.14. Selecting editing the polygons option.
Figure 5.15. Moving the points to create new shape.

Figure 5.16. ChandraGupta shape file overlapped on IndiaMap shape file.
Figure 5.17. Ashoka shape file overlapped on IndiaMap shape file.
CHAPTER 6

SHAPE FILES

A shape file stores non-topological geometry and attribute information for the spatial features in a data set. The geometry for a feature is stored as a shape comprising a set of vector coordinates. Shape files can support point, line, and area features. Area features are represented as closed loop, double-digitized polygons. Shape files handle single features that overlap or those are noncontiguous. They also typically require less disk space and are easier to read and write. Shape files are more advantageous over other data sources when considered faster drawing speed and editing ability. Computer programs can be created to read or write shape files using the technical specification. An ESRI shape file consists of a main file, an index file, and a dBase table.

The shape files comprise of three mandatory and several optional files. Three mandatory files are as follows:

- Shape file (.shp) - the file that stores the feature geometry.
- Index file (.shx) - the file that stores the index of the feature geometry.
- dBase file (.dbf) - the file that stores attributes of the features in database.

6.1 SHAPE FILE

The Shape file (.shp) contains the primary geographic reference data in the shape file. The file consists of a single fixed length header followed by one or more variable length records. Each variable length record is made up of a fixed length record header followed by variable length record contents.

The main header is 100 bytes (in length) and contains about 17 fields (9 fields each 4 bytes + 8 field each eight bytes). The representation is shown in Table 6.1 [1] and 6.2 [1].

A variable length record header is 8 bytes in length and mainly comprises of two fields, which contain data for record number and content length. There is a 1:1 correspondence between the shape type in the shape file header and its contents in the variable length record.
### Table 6.1. Description of Shape File Header

<table>
<thead>
<tr>
<th>Position</th>
<th>Field</th>
<th>Value</th>
<th>Type</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte 0</td>
<td>File Code</td>
<td>Code 9994</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 4</td>
<td>Unused</td>
<td>0</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 8</td>
<td>Unused</td>
<td>0</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 12</td>
<td>Unused</td>
<td>0</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 16</td>
<td>Unused</td>
<td>0</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 20</td>
<td>Unused</td>
<td>0</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 24</td>
<td>File Length</td>
<td>File Length</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 28</td>
<td>Version</td>
<td>1000</td>
<td>Integer</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 32</td>
<td>Shape Type</td>
<td>Shape Type</td>
<td>Integer</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 36</td>
<td>Bounding Box</td>
<td>Xmin</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 44</td>
<td>Bounding Box</td>
<td>Ymin</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 52</td>
<td>Bounding Box</td>
<td>Xmax</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 60</td>
<td>Bounding Box</td>
<td>Ymax</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 68*</td>
<td>Bounding Box</td>
<td>Zmin</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 76*</td>
<td>Bounding Box</td>
<td>Zmax</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 84*</td>
<td>Bounding Box</td>
<td>Mmin</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 92*</td>
<td>Bounding Box</td>
<td>Mmax</td>
<td>Double</td>
<td>Little</td>
</tr>
</tbody>
</table>

Table 6.2. Value/Shape Type Mapping

<table>
<thead>
<tr>
<th>Value</th>
<th>Shape Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Null Shape</td>
</tr>
<tr>
<td>1</td>
<td>Point</td>
</tr>
<tr>
<td>3</td>
<td>PolyLine</td>
</tr>
<tr>
<td>5</td>
<td>Polygon</td>
</tr>
<tr>
<td>8</td>
<td>MultiPoint</td>
</tr>
<tr>
<td>11</td>
<td>PointZ</td>
</tr>
<tr>
<td>13</td>
<td>PolyLineZ</td>
</tr>
<tr>
<td>15</td>
<td>PolygonZ</td>
</tr>
<tr>
<td>18</td>
<td>MultiPointZ</td>
</tr>
<tr>
<td>21</td>
<td>PointM</td>
</tr>
<tr>
<td>23</td>
<td>PolyLineM</td>
</tr>
<tr>
<td>25</td>
<td>PolygonM</td>
</tr>
<tr>
<td>28</td>
<td>MultiPointM</td>
</tr>
<tr>
<td>31</td>
<td>MultiPatch</td>
</tr>
</tbody>
</table>


6.2 INDEX FILE

The index file (.shx) contains the offset of the corresponding main file record from the beginning of the main file. The index file header is identical to shape file header. The file length stored in the header is total length of file, which is in 16 bit words. The content length stored in the index record is same as the value stored in the shape file record header.

6.3 DBASE FILE

A dBase file (.dbf) consists of a header record and data records. The header record defines the structure of dbf file and contains any other information related to the table. The
header record starts at file position zero. Data records follow the header, in the consecutive bytes, and contain the actual text of the fields. Any set of fields can be present in the table.

The Three requirements are as follows:

- The file name must have the same prefix as the shape and index file. Its suffix must be .dbf.
- The table must contain one record per shape feature.
- The record order must be the same as the order of shape features in the shape (*.shp) file.

6.4 Optional File

The optional file is as follows:

- .prj: Projection format, a plain text file describing the projection using well-known text format.
- .sbn and .sbx: a spatial index of the features.
- .fbn and .fbx: a read-only spatial index file.
- .ain and .aih: a theme’s attribute file.
- ixs: a geocoding index for read-write shape files.
- mxs: a geocoding index for read-write shape files.
- atx: an attribute index for the .dbf file.
- .shp.xml: geospatial metadata in XML format.
- .cpg: a code page file for .dbf.
CHAPTER 7

MAPOBJECTS TOOLBARS

MapObjects Java Edition contains multiple classes whose methods can be used in workspace to provide the functionality. Toolbars are options which are included as part of functional requirement of the GIS tool. This provides commonly used functions such as zoom to full extent, zoom in, zoom out, pan, identity and etc. In this chapter we discuss how these tool can be integrated into GIS tool [2].

The Figure 7.1 shows the screenshot tools developed in GIS tool.

![Figure 7.1. MapObjects toolbar.](image)

7.1 ZOOMPANTOOLBARS

COM.ESRI.MO2.UI.TB.ZOOMPANTOOLBAR

The ZoomPanToolBar uses functions provided by ZoomPanToolBarActions class, this class extends BaseToolBar. This class allows an end-user to work on the map with different options. The actions supported by the Toolbar include:

- **ZoomIn** – provides a tool for clicking or dragging a rectangle on the map in order to zoom in.
- **ZoomOut** – provides a tool for clicking or dragging a rectangle on the map in order to zoom out.
- **ZoomToSelected** – zooms the map to the extent of all selected features in the selected layer.
- **ZoomToFullExtent** – zooms the map to the extent of all layers within the map.
- **GoPrevious** – Zooms to the previous extent stored in the extent history.
- **GoNext** – Zooms to the next extent stored in the extent history.
- **Pan** – Provides a tool for dragging the map to a new location without altering the zoom level.
- **PanOneDirection** – pans the map in one of four directions, north, south, east, or west.
- **Identify** – performs an Identify on the specified "selected" layers.

### 7.2 Selection Toolbars

**COM.ESRI.MO2.UI.TB.SELECTIONTOOLBAR**

This Toolbar provides functions that perform future selection bases upon Attribute or query. This class extends the BaseToolBar class. The actions supported by this ToolBar include:

- **Identify** – provides a tool for clicking on features in the map, and displaying attributes about those features.
- **Find** – opens a dialog for locating features whose attributes contain an end-user provided string.
- **Search** – Opens a dialog for locating features based on a predefined “stored query.”
- **Query** – Opens a dialog for locating features based on a query that an end – user constructs.
- **Select** – Provides a tool for selecting features by rubber banding a shape in the map.
- **Buffer** – Opens a dialog for constructing a buffer polygon around the currently selected features.
- **Attributes** – Displays attributes of the currently selected features.

### 7.3 Layer Toolbars

**COM.ESRI.MO2.UI.TB.LAYERTOOLBAR**

This Tool Provides Functionality to the user to add or delete the selected layer in table of contents. This class extends the BaseToolBar class. The actions supported by this tool bar are:

- **Add Layer** – Adds a new layer to the Table of Contents.
- **Delete layer** – Removes a layer from Table of Contents.
CHAPTER 8

CUSTOM TOOLBAR

In this chapter we will discuss about the custom toolbar, which have been as part of functional requirement. In the above chapter we discuss MapObjects. Custom toolbars are created to make easy interface to user and to allow additional functionalities to enhance the usability of GIS tool. In order to create the custom toolbars we need to use MOJO methods, important methods are explained briefly. Figure 8.1 shows the custom toolbar snapshot [2]. Figure 8.1 shows the screenshot of custom tool bar.

![Custom toolbar](image)

Figure 8.1. Custom toolbar.

The tools, which are added in the first set of toolbar, are custom tools, and this chapter clearly explains the functionality of each the tool (see Table 8.1).

<table>
<thead>
<tr>
<th>Number</th>
<th>Tool Name</th>
<th>Class Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Print</td>
<td>com.esri.mo2.ui.bean.Print</td>
</tr>
<tr>
<td>2</td>
<td>Add Layer</td>
<td>com.esri.mo2.ui.tb.LayerToolBar</td>
</tr>
<tr>
<td>3</td>
<td>Pointer</td>
<td>Arrow.java</td>
</tr>
<tr>
<td>4</td>
<td>Distance</td>
<td>DistanceTool.java</td>
</tr>
<tr>
<td>5</td>
<td>Help</td>
<td>Help.java</td>
</tr>
<tr>
<td>6</td>
<td>XY</td>
<td>AddXYtheme.java</td>
</tr>
<tr>
<td>7</td>
<td>Hot Link</td>
<td>HotLink.java</td>
</tr>
</tbody>
</table>

8.1 PRINT

Print tool add printing functionality to GIS tool, where a user can select a map and print it. When a user clicks on the print icon, a print dialog box opens where he can select the printing the options. The following code has been implemented in order to add this tool:
8.2 Add Layer

Add Layer tool provides user an important functionality where user’s can add a new layer (map) to the existing Table Of Contents, in the TOC users have functionalities where they can switch between different layers. The following code has been implemented in order to add this tool:

```java
AddLayerDialog addLayer = new AddLayerDialog();
addLayer.setMap(map);
addLayer.setVisible(true);
map.redraw();
```

AddLayerDialog class has been developed in order to display a dialog box to request the path for shapefile. The new shapefile is added to map as a layer and we redraw the map. AddLayerDialog class extends the JDialog.

8.3 Pointer

In GIS tool, when a user selects any tool, the selected tool will be in active state until other tool has been selected. User might want to have functionality where they can use the GIS tool with no tool selected. The Pointer tool sets the state to normal state which means no tools are selected. User needs to click the pointer icon in order to trigger this functionality. The following code has been implemented in order to add this tool:

```java
map.setSelectedTool(arrow);
```

We can select listener to false in order to accomplish pointer tool, setSelectedTool method is used to set the currently selected tool,

8.4 Distance Tool

When the users want to calculate the distance between two points on the map, Distance Tool can be used. Distance between the two points selected is displayed on bottom status bar; the following code has been implemented in order to add this tool:

```java
com.esri.mo2.ui.bean.Print mapPrint = new com.esri.mo2.ui.bean.Print();
mapPrint.setMap(map);
mapPrint.doPrint();
```
DistanceTool distanceTool = new DistanceTool();
map.setSelectedTool(distanceTool);

DistanceTool has been developed to enable distance measurement tool, in this class we store the points clicked and we calculate the distance between the points, and we display on the status bar. DistanceTool class extends DragTool class.

8.5 Help Tool

HelpTool is a user-friendly tool, which talks about other tools; this is very useful for the end user who has a minimal knowledge of GIS tool initially. If user faces any problem in determining the functionality of any tool they can use this tool to learn about them. Users need to click on this tool to enable the HelpTool, later click on other tool. GIS tool will display a text area, which explain functionality of other tool and how to use it. The following code has been implemented in order to add this tool:

```java
ActionListener toolName= new ActionListener () {
    public void actionPerformed(ActionEvent ae) {
        if(helpOn)
            helpWindow = new HelpWindow("Description of tool");
    }
}

helpOn variable in above code will be true when HelpTool has been selected. HelpWindow is the class that has been implemented in the code. In this class we display the content passed in arguments in a text area. This class extends the JFrame and implements the ActionListener.

8.6 XY Tool

XY Tool is used to add location as a layer in table of contents; points will be projected on the map. This tool works in the same style as AddLayer Tool discussed in chapter under 8.2 section. In AddLayer Tool we will be adding the shape file as a layer, under XY Tool we have a functionality to add csv (comma separated files), this file will have latitude and longitude. When we click on XY Tool it will open a window where we have to show the path
of csv file. Ones file is chosen users can see the projection of points on the map. The following code has been implemented in order to add this tool:

```java
AddXYTheme addXYTheme = new AddXYTheme();
addXYTheme.setMap(map);
addXYTheme.setVisible(false);
map.redraw();
```

AddXYTheme class has been developed in order to display a dialog box to request the path for csv file. Data (longitude and latitude) in the csv is collected and projected on the map as a layer and we redraw the map. AddXYTheme class extends the JDialog.

### 8.7 Hot Link

HotLink Tool is very helpful for the end user of the application; this tool will provide the functionality to display the webpages of the particular location, which are projected on the map. When a user clicks the HotLink Icon (globe icon) hot link option is activated. Now users have a privilege to learn about points that are projected. Users can click on any projected points in order to open a webpage, this webpages talks about the history of the selected point. Pages are displayed as the normal HTML pages; these pages can be modified to add additional information to point. Each webpage has hotlink that will redirect them to Wikipedia pages. The following code has been implemented in order to add this tool:

```java
webJavaButton.addActionListener( new ActionListener()

    public void actionPerformed(ActionEvent ae) {
        web = true; }
    });
```

```java
if (web)
    Process pc = Runtime.getRuntime().exec("cmd.exe /c start "
   +url +"";
```

We have a mouse listener enabled to compare the latitude and longitude clicked is equal to any of the points in csv (points projected). If the case matches we store the URL path into url variable. And we run the getRuntime method in order to open a webpage with URL path.
8.8 ADDITIONAL TOOLS

In addition to the MOJO tools and Customized tools there are few additional tools that can help the end user few of them are Help Menu, Co-Ordinates Display and Contact Us Icon.

8.8.1 Help Menu

Help Menu is the last menu item in the menu display; this menu has three menu items:

- General Help – General help will display a window which talks about the GIS tool, and initial steps to do done in order to display the map.
- Labeling Help – Help you to understand how can we label.
- HotLink Help – This explains who can we use the HotLink option. And benefits of them.

8.8.2 Co-Ordinate Display

In the GIS tool at bottom of status bar we display the co-ordinates of the pointer, this would help the end-user to gather co-ordinates of a location

The Figure 8.2 shows the example of co-ordinate system.

Figure 8.2. Co-Ordinates display.
8.8.3 Contact Us

Contact Us help window has the contact information about the GIS tool, this Contact US menu is located under file menu.

The Figure 8.3 shows the screenshot of Contact Us menu.

Figure 8.3. Contact us.
CHAPTER 9

DEMONSTRATION

The following sections contain the screenshots explaining the very important features in the GIS tool. Figures 9.1 to 9.16 demonstrate the GIS tool:

1. Application main page (see Figure 9.1).
2. Enable other maps. We need to check the Table of Contents in the legend (see Figures 9.2 to 9.4).
3. Adding points of interest.
   a. Please click XY icon to open a window, where we have to select the path (see Figure 9.5).
   b. Please navigate to data folder, please load Mauryan_dynasty.csv (see Figure 9.6).
   c. Below points appear (see Figure 9.7).
4. Arrow tool. This tool will uncheck all other active tool functionalities (see Figure 9.8).
5. Hotlinks.
   a. Enabling the HotLink action, (click on the Globe) to enable the hotlink button, to get the webpage displayed please click on any points (see Figure 9.9).
   b. Click on the any point, to open a web page (see Figure 9.10).
   c. Example webpage, (we can use hotlink to open wikipedia) (see Figure 9.11).
6. Print option. We can print maps by using the print option (see Figure 9.12).
7. Help icon (see Figure 9.13).
8. Identity tool (see Figure 9.14).
Figure 9.1. Application overview.

Figure 9.2. Maps that can be clicked.
Figure 9.3. Chandragupta enabled.

Figure 9.4. ChandraGupta and Bindusara map’s enabled.
Figure 9.5. XY icon enabled.

Figure 9.6. XY selection window.
Figure 9.7. POI in Maurya Dynasty.

Figure 9.8. Use of Arrow Tool.
Figure 9.9. Hotlink tool.

Figure 9.10. Hotlink webpage.
Figure 9.11. Wikipedia hotlink.

Figure 9.12. Print option.
Figure 9.13. Help icon.

Figure 9.14. Identity tool.
Figure 9.15. Tool tip of zoom to full extent.

Figure 9.16. Tool tip of pan.
CHAPTER 10

FUTURE ENHANCEMENTS

With the number of GIS tool increasing in Department of Sciences and Department of History at San Diego State University, a common tool should be developed where the end users have an advantage to combine two different GIS tool and view one on another. Tool should be changed as a web interface where the end user can execute the GIS tool without downloading it.
BIBLIOGRAPHY

WORKS CITED


WORKS CONSULTED

S. Banala, GIS software to demonstrate Indian history (Delhi Sultanate), Master’s thesis, San Diego State University, San Diego, CA, 2011.


